

LMS V3.4B

Unterprogramm-Schnittstelle

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 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

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

1	Einleitung	7
1.1	Kurzbeschreibung des Produkts	7
1.2	Zielgruppe des Handbuchs	7
1.3	Konzept des Handbuchs	8
1.4	Verwendete Metasprache	10
2	LMS als Unterprogramm	11
3	Aufrufvorbereitungen	15
3.1	Aufrufvorbereitungen und Rückkehrinformationen	15
3.2	Funktion und Aufbau der Parameterstrukturen	17
	CB Control Block	18
	EA Element Attributes	21
	ED Element Description	23
	EI Element Information	26
	EM Element Mask	29
	ER Element Record	33
	FD File Description	34
	LA Library Attributes	34
	LD Library Description	37
	LI Library Information	38
	PA Protection Attributes	41
	RD Record Description	45
	TA Type Attributes	46
	TD Type Description	48
	TI Type Information	48
	TID TOC-Identifikation	50
	Zusammenhang: Funktion - Parameterstruktur	51

3.3	Übersicht über die Subcodes	53
4	Unterprogrammfunktionen	55
4.1	Übersicht über die Funktionen	55
4.2	Beschreibung der Funktionen	57
	ADD Aufnehmen einer Datei als Element	58
	CLOSE Schließen eines Elementes	62
	CLOSLIB Schließen einer Bibliothek	64
	COPY Kopieren eines Elementes	66
	COPYLIB Kopieren einer Bibliothek	70
	COPYSTR Kopieren eines Delta-Baumes	72
	DEL Löschen eines Elementes	74
	END Beenden eines Unterprogrammzugriffs	76
	GET Lesen eines Satzes	77
	GSYSELEM Lesen einer Elementangabe aus einer S-Variablen	80
	INIT Eröffnen eines Unterprogrammzugriffs	82
	LOCK Sperren eines Elementes	84
	LST Ausgeben eines Elementes nach SYSLST	86
	MODEA Ändern der Elementattribute	89
	MODEP Ändern des Elementschutzes	91
	MODLA Ändern der Bibliotheksattribute	93
	MODTA Ändern der Typattribute	95
	OPENGET Eröffnen eines Elementes zum Lesen	97
	OPENPUT Eröffnen eines Elementes zum Schreiben	100
	OPENUPD Eröffnen eines Elementes zum Lesen und Schreiben	104
	PROVIDE Reservieren und kopieren eines Elementes	107
	PUT Schreiben eines Satzes	111
	REN Umbenennen eines Elementes	112
	REORGLIB Bibliothek reorganisieren	114
	RETURN Rückgabe eines Elementes	116
	SEL Ausgeben eines Elementes in eine Datei	119
	SHOWLA Anzeigen der Bibliotheksattribute	121
	SHOWTA Anzeigen der Typattribute	123
	TOC Fortsetzen von TOCPRIM oder TOCSEC	125
	TOCPRIM Suchen eines Elementes im primären Inhaltsverzeichnis	127
	TOCSEC Suchen eines Elementes im sekundären Inhaltsverzeichnis	131
	UNLOCK Aufheben einer Elementensperre	135

4.3	Programmierhilfen	137
4.3.1	Symbolische Namen	137
4.3.2	Aufbau des Sekundärsatzes (Satzart 163)	141
4.3.3	Aufbau des Attributsatzes (Satzart 164)	142
5	COBOL-Schnittstelle	145
<hr/>		
5.1	Der Verbindungsmodul LMSUP1	145
5.2	Generierung der Parameterstrukturen für COBOL	146
	LMSCOBCB	147
	LMSCOBEA	148
	LMSCOBED	148
	LMSCOBEI	149
	LMSCOBEM	151
	LMSCOBFD	153
	LMSCOBLA	153
	LMSCOBLD	154
	LMSCOBLI	155
	LMSCOBPA	157
	LMSCOBRD	158
	LMSCOBTA	159
	LMSCOBTD	160
	LMSCOBTI	160
5.3	Programmierhilfen	162
	Symbolische Namen LMSCOBEQ	162
	Aufbau eines Satzes der Satzart 163	165
	Aufbau eines Satzes der Satzart 164	166
5.4	Beispiel	168
6	C-Schnittstelle	177
<hr/>		
6.1	Der Verbindungsmodul LMSUP1	177
6.2	Include-Elemente für den C/C++-Compiler	177
	Include-Element LMS.H	178
	Include-Element LMSREC.H	195
6.3	Beispiel	200

7	Assembler-Schnittstelle	209
7.1	Der Verbindungsmodul LMSUP1	209
7.2	Generierung der Parameterstrukturen für Assembler	210
	LMSASSCB	211
	LMSASSEA	213
	LMSASSED	214
	LMSASSEI	215
	LMSASSEM	218
	LMSASSFD	221
	LMSASSLA	222
	LMSASSLD	224
	LMSASSLI	225
	LMSASSPA	227
	LMSASSRD	229
	LMSASSTA	230
	LMSASSTD	232
	LMSASSTI	233
7.3	Programmierhilfen	236
	Symbolische Namen LMSASSEQ	236
	Aufbau eines Satzes der Satzart 163	239
	Aufbau eines Satzes der Satzart 164	240
7.4	Beispiel	242
	Literatur	253
	Stichwörter	255

1 Einleitung

1.1 Kurzbeschreibung des Produkts

Das Bibliotheksverwaltungssystem LMS (Library Maintenance System) kann 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. Dabei wird LMS dynamisch nachgeladen. Diese Unterprogrammchnittstelle kann auch im Extended-System-Bereich (XS) verwendet werden.

1.2 Zielgruppe des Handbuchs

Dieses Handbuch richtet sich an LMS-Anwender und Programmierer, die die vielfältigen Möglichkeiten des LMS in eigenen Programmen nutzen wollen.

Zum Aufruf des LMS als Unterprogramm sind neben der Kenntnis der wichtigsten BS2000-Kommandos und dem Vertrautsein mit LMS vor allem Assembler- oder C- oder COBOL-Kenntnisse unbedingte Voraussetzung.

1.3 Konzept des Handbuchs

Dieses Handbuch beschreibt ausschließlich die Unterprogramm-Schnittstelle für LMS.

Aufbau des Handbuchs

im Einzelnen finden Sie im Kapitel

- **LMS als Unterprogramm**

Übersicht über Funktionen und Anwendungsmöglichkeiten.

- **Aufrufvorbereitungen**

Beschreibung der Aufrufvorbereitungen, Returncodes und der Aufbau der Parameterstrukturen.

- **Unterprogrammfunktionen**

Beschreibung aller Funktionen in alphabetischer Reihenfolge mit Aufruf, Parameterliste und Returncodes.

- **COBOL-Schnittstelle**

Beschreibung der Parameterstrukturen für COBOL und Beispiel zu einem COBOL-Hauptprogramm.

- **C-Schnittstelle**

Beschreibung der Parameterstrukturen für C und Beispiel für ein C-Hauptprogramm.

- **ASSEMBLER-Schnittstelle**

Beschreibung der Parameterstrukturen für ASSEMBLER und Beispiel für ein ASSEMBLER-Hauptprogramm.

Die genaue Beschreibung des LMS, einschließlich der LMS-Anweisungen und -Meldungen, ist dem Benutzerhandbuch [1] zu entnehmen.

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>.

1.4 Verwendete Metasprache

Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

2 LMS als Unterprogramm

Über die Unterprogramm-Schnittstelle stehen dem Benutzer komfortable Möglichkeiten zur Bearbeitung von Programmbibliotheken und deren Elemente zur Verfügung. Die nachstehenden LMS-Funktionen werden direkt aus einem Benutzerprogramm (COBOL, C, Assembler) aufgerufen. Die Kontrolle verbleibt im Benutzerprogramm.

Eröffnen bzw. Beenden des Unterprogramm-Zugriffs

Mit der Funktion INIT wird ein Unterprogramm-Zugriff eröffnet und mit END wieder beendet. Bei jedem Aufruf von INIT wird eine Unterprogramm-Zugriffsidentifikation angelegt. So können parallel ablaufende Unterprogramm-Zugriffe identifiziert werden.

Funktionen für die Unterprogramm-Schnittstelle

Folgende Funktionen können über die Unterprogramm-Schnittstelle aufgerufen werden:

- Mit ADD können Dateien als Elemente in eine Bibliothek aufgenommen werden.
- Mit CLOSLIB kann eine Bibliothek geschlossen werden.
- Mit COPY können Elemente kopiert werden.
- Mit COPYLIB können Bibliotheken logisch kopiert werden.
- Mit COPYSTR können Delta-Elemente strukturerhaltend kopiert werden.
- Mit DEL können Elemente gelöscht werden.
- Mit GSYSELEM wird der Inhalt einer S-Variablen gelesen, als Angabe eines Elementes (Bibliothek, Elementname, Version und Typ) interpretiert und in eine für die Unterprogramm-Schnittstelle nutzbare Form umgesetzt.
- Mit LOCK können Elemente gegen Veränderung gesperrt werden.
- Mit LST können Elemente nach SYSLST ausgegeben werden.
- Mit MODEA können Elementattribute eingestellt werden.
- Mit MODEP kann der Elementschutz eingestellt werden.
- Mit MODLA können Bibliotheksattribute eingestellt werden.
- Mit MODTA können die Typattribute eingestellt werden.
- Mit PROVIDE kann ein Element „ausgeliehen“ werden.
- Mit REN können Voll-Elemente umbenannt werden.
- Mit REORGLIB kann der für eine Bibliothek erforderliche Plattenspeicherplatzbedarf vermindert werden.
- Mit RETURN kann ein „ausgeliehenes“ Element zurückgegeben werden.
- Mit SEL können Elemente in eine Datei ausgegeben werden.

- Mit SHOWLA können Bibliotheksattribute ausgegeben werden.
- Mit SHOWTA können die Typattribute angezeigt werden.
- Mit UNLOCK können Elemente von einer Sperre wieder befreit werden.

Funktionen für das Lesen oder Schreiben von Elementen

Soll ein Element gelesen oder geschrieben werden, muss es mit einer der drei OPEN-Funktionen eröffnet werden.

- Mit OPENGET können Elemente zum Lesen eröffnet werden.
- Mit OPENPUT können Elemente zum Schreiben eröffnet werden.
- Mit OPENUPD können Elemente zum Lesen und Schreiben eröffnet werden.

Ein gleichzeitiger Aufruf von OPENGET, OPENPUT und OPENUPD auf dasselbe Element ist nicht möglich.

Bei allen OPEN-Funktionen wird eine Zugriffspfad-Identifikation festgelegt. So können mehrere Elemente zur gleichen Zeit eröffnet werden. Diese Zugriffspfad-Identifikation muss bei nachfolgenden GET- oder PUT-Aufrufen immer angegeben werden.

- Mit GET kann ein Satz gelesen werden.
- Mit PUT kann ein Satz geschrieben werden.
- Mit CLOSE muss das Element nach der Bearbeitung explizit geschlossen werden.

Funktionen für das Suchen nach Elementen

Um Informationen über Elementeninträge zu erhalten, gibt es drei TOC-Funktionen.

- Mit TOCPRIM kann ein Element im primären Inhaltsverzeichnis gesucht werden.
- Mit TOCSEC kann ein Element im sekundären Inhaltsverzeichnis gesucht werden.
- Mit TOC kann nach weiteren Elementen gesucht werden.

TOCPRIM und TOCSEC dienen dazu, die Suchkriterien für ein Element festzulegen und die Elementeninträge über das erste Element, das diesen Kriterien entspricht, auszugeben. Gleichzeitig muss bei beiden Funktionen eine TOC-Identifikation festgelegt werden. Soll mit den angegebenen Suchkriterien nach weiteren Elementeninträgen gesucht werden, muss die TOC-Funktion aufgerufen werden. Sie dient zur Fortsetzung von TOCPRIM und TOCSEC. Um zu kennzeichnen, welche Funktion fortgesetzt werden soll, muss beim Aufruf von TOC die TOC-Identifikation angegeben werden.

Funktionen für den Elementschutz

Der Elementschutz kann mit folgenden Funktionen eingestellt werden:

- Mit MODEP wird der Elementschutz für ein bestimmtes Element eingestellt.
- Mit MODTA wird der initiale Elementschutz für alle neuen Elemente eines bestimmten Elementtyps eingestellt.
- Mit MODLA wird der initiale Elementschutz für alle neuen Elemente einer Bibliothek eingestellt.

Mit den folgenden Funktionen kann der Elementschutz angezeigt werden:

- Mit den TOC-Funktionen wird der Elementschutz für ein bestimmtes Element ausgegeben.
- Mit SHOWTA werden die Typattribute eines bestimmten Elementtyps ausgegeben.
- Mit SHOWLA werden die Bibliotheksattribute einer Bibliothek ausgegeben.

Funktionen für den Versionsautomatismus

Für den Versionsautomatismus gibt es folgende zwei Funktionen:

- MODTA zum Einstellen der Konvention.
- SHOWTA zum Anzeigen der gültigen Konvention.

Als Konvention steht STD-SEQUENCE, MULTI-SEQUENCE oder STD-TREE zur Verfügung.

Funktionen für die Unterstützung erweiterter Zeichensätze in LMS

Für die Unterstützung erweiterter Zeichensätze in LMS gibt es die folgenden Funktionen (siehe auch [2]):

- MODEA zum Einstellen und Verändern eines Zeichensatznamens.
- Die TOC-Funktionen zum Anzeigen des einem Element zugeordneten Zeichensatznamens.

Funktionen zur Verwaltung der zulässigen Speicherungsform

Versionen von Daten können in Voll- oder in Delta-Technik gespeichert werden. Zur Verwaltung der für eine Bibliothek (einen Typ) zulässigen Speicherungsform dienen die Funktionen:

- MODLA zum Einstellen der zulässigen Speicherungsform für Elemente einer Bibliothek.
- MODTA zum Einstellen der zulässigen Speicherungsform für Elemente eines Typs.
- SHOWLA zum Erfragen der zulässigen Speicherungsform für Elemente einer Bibliothek.
- SHOWTA zum Erfragen der zulässigen Speicherungsform für Elemente eines Typs.

Die Speicherungsform eines Elements wird beim Erzeugen eines Elements gewählt.

Funktionen zur Unterstützung eines Ausleihverfahrens

- MODLA zum Einstellen des Write Control für die Bibliothek und zum Festlegen des initialen Ausleihrechts für alle neuen Elemente der Bibliothek.
- MODTA zum Einstellen des Write Control für einen Typ und zum Festlegen des initialen Ausleihrechts für alle neuen Elemente des Typs.
- MODEP zum Einstellen des Ausleihrechts für ein Element.
- MODEA zum Einstellen des Elementzustandes (FREE oder INHOLD).
- PROVIDE zum Reservieren eines Elementes einer Quellbibliothek und anschließendem Kopieren in eine Ausgabebibliothek.
- RETURN zur Rückgabe eines Elementes einer Quellbibliothek in eine Ausgabebibliothek, sofern die zur Zielversion angegebene Basis in der Ausgabebibliothek vom Benutzer reserviert ist. Löschen des Elementes in der Quellbibliothek und Rückgabe der Reservierungen in der Ausgabebibliothek.
- SHOWLA zum Erfragen des Write Control und des initialen Ausleihrechts einer Bibliothek.
- SHOWTA zum Erfragen des Write Control und des initialen Ausleihrechts eines Typs.
- Die TOC-Funktionen zum Erfragen des Elementzustandes (FREE oder INHOLD) und des Ausleihrechts.

Funktionen zur Unterstützung der make-Funktionalität

- MODEA zum aktualisieren des Änderungsdatums.

3 Aufrufvorbereitungen

Zur Verknüpfung des Hauptprogramms mit LMS muss der Rufer den Modul LMSUP1 aus der Bibliothek SYSLNK.LMS.034 fest hinzubinden.

Der Installationsort dieser Bibliothek ist über IMON frei wählbar.

Mit Hilfe der Built-In-Funktion INSTALLATION-PATH kann der Installationsort von SYSLNK.LMS.034 ermittelt werden:

```
/SET-VARIABLE LIBRARY-NAME =INSTALLATION-PATH-  
                ( LOGICAL-ID = 'SYSLNK'-  
                  ,INSTALLATION-UNIT = 'LMS'-  
                  ,VERSION = '3.4'-  
                  ,DEFAULT-PATH-NAME = '$.SYSLNK.LMS.034')
```

Hauptprogramme, die LMS als Unterprogramm aufrufen wollen, können sowohl im 24- als auch im 31-Bit-Adressierungsmodus ablaufen.

3.1 Aufrufvorbereitungen und Rückkehrinformationen

Aufrufparameter versorgen

Vor dem Aufruf einer Funktion muss der Rufer bestimmte Felder der Parameterstrukturen versorgen. Welche Felder versorgt werden müssen, hängt von der aufgerufenen Funktion ab. Bei der Beschreibung jeder Funktion sind diese Felder in der Tabelle der Aufrufparameter aufgeführt. Es ist darauf zu achten, dass die Reihenfolge der Felder, wie sie in dieser Tabelle aufgeführt werden, eingehalten wird.

Rückkehrparameter

Nach dem Aufruf einer Funktion werden in bestimmte Felder der Parameterstrukturen Rückkehrparameter (z.B. Returncodes, MSG-Schlüssel) geschrieben. In welche Felder geschrieben wird, hängt von der aufgerufenen Funktion ab. Bei der Beschreibung jeder Funktion sind diese Felder in der Tabelle der Rückkehrparameter aufgeführt.

Returncodes

Der Returncode ist ein Zeichen lang und informiert über den Erfolg/Misserfolg der aufgerufenen Funktion.

Detailliertere Informationen erhält man aus den Meldungsschlüsseln zu LMS, DMS und PLAM. Diese Meldungsnummern sind sedezimal verschlüsselt und müssen dementsprechend bewertet werden. Die DMS-Codes werden bereits sedezimal ausgegeben und können direkt ausgewertet werden, während die LMS- und PLAM-Codes für die Auswertung (durch /HELP) noch von sedezimal in dezimal umgerechnet werden müssen.

Folgende Returncodes werden an der Unterprogramm-Schnittstelle zurückgegeben:

Returncode	Bedeutung
X'00'	Die Funktion wurde erfolgreich ausgeführt.
X'04'	Der Satz-Puffer wurde bei GET zu kurz angegeben (maximale Satzlänge 32 KByte bzw. 256 KByte bei Format-B Sätzen).
X'08'	Element- oder TOC-Ende wurde erreicht.
X'0C'	Die Funktion wurde wegen eines Benutzerfehlers abgebrochen.
X'14'	Die Aufrufparameter sind fehlerhaft oder unvollständig.
X'18'	Die Aufrufreihenfolge ist unzulässig.
X'1C'	Interner LMS-Fehler.

3.2 Funktion und Aufbau der Parameterstrukturen

Über die Parameterstrukturen wird die gesamte Datenübergabe vom Hauptprogramm zum LMS bzw. vom LMS an das Hauptprogramm abgewickelt. Die Parameterstrukturen müssen im Hauptprogramm definiert werden. Dazu stehen

- für COBOL entsprechende COPY-Elemente (siehe [Seite 145ff](#)),
- für C ein entsprechendes INCLUDE-Element (siehe [Seite 177](#)) und
- für Assembler entsprechende Makros (siehe [Seite 209ff](#))

zur Verfügung.

Mit den gleichen COPY-Elementen bzw. INCLUDE-Elementen bzw. Makros (Form L) können die Parameterstrukturen initialisiert, d.h. sinnvoll vorbesetzt werden. Die Parameterstrukturen der Unterprogramm-Schnittstelle sind:

- der Kontrollblock CB,
- die Beschreibungen ED, FD, LD, RD und TD,
- die TOC-Parameter EI, EM und TID,
- der Satzpuffer ER,
- die Bibliotheksattribute LA, LI und PA,
- die Typattribute TA und TI,
- die Elementattribute EA.

Felder der Parameterstrukturen

Numerische Felder: 2 oder 4 Byte

Indikatoren: 1 Byte. Es gibt symbolische Werte zur Bearbeitung dieser Felder.

Stringfelder: Alle eingegebenen Strings müssen linksbündig in die entsprechenden Felder geschrieben werden. Die eingegebenen Strings werden bis zum ersten Leerzeichen (Blank), maximal jedoch in Feldlänge ausgewertet.

CB Control Block

Der CB identifiziert die Schnittstellenversion (SCBVERSION), den angesprochenen Zugriff (ACC) und die gewünschte Funktion (FUNCTION). Zusätzlich kann noch ein Subcode (SUBCODE) angegeben werden. Bei allen Funktionen werden ein Returncode (RETURNCODE) und Fehlerschlüssel für LMS, DMS und PLAM (LMS-MSG, DMS-MSG, PLAM-MSG) zurückgegeben. Nach dem INIT-Aufruf steht im Feld LMSVERSION die Versionsnummer der LMS-Installation. Die Verarbeitungsoperanden werden beim INIT-Aufruf auf Standardwerte gesetzt.

Der Kontrollblock CB wird bei allen Funktionen benötigt. Er muss immer der erste Parameter der Parameterliste sein.

Wird der Kontrollblock CB mit einer neuen Version übersetzt, müssen alle anderen Programmteile, die LMSUP-Strukturen benutzen, ebenfalls übersetzt werden.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung [nach INIT]
SCBVERSION	Schnittstellenversion	2	X'F0F4'
FUNCTION	Funktionscode	1	1 x X'01'
SUBCODE	Subcode	1	1 x X'40'
ACC	UP-Zugriffsidentifikation	4	4 x X'FF'
RETURNCODE	Returncode-Klasse	1	1 x X'00'
	reserviert	1	1 x X'00'
LMS-MSG	LMS-Meldungsschlüssel	2	2 x X'00'
DMS-MSG	DMS-Meldungsschlüssel	2	2 x X'00'
PLAM-MSG	PLAM-Meldungsschlüssel	2	2 x X'00'
LMSVERSION	LMS-Version	12	12 x X'40'
	reserviert	4	4 x X'00'
*****	LMS-Parameter *****		
DESTROY	physikalisches Überschreiben Y: Daten mit X'00' überschreiben N: Daten nicht überschreiben	1	1 x X'40' [N]

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung [nach INIT]
FCB	Dateiattribute S: wie gespeichert bzw. Standardwert I: ISAM Q: SAM C: wie im Katalog beschrieben	1	1 x X'40' [S]
RKEY	ISAM-Schlüssel und Dateiattribute Y: ISAM-Schlüssel und Dateiattribute speichern N: ISAM-Schlüssel und Dateiattribute nicht speichern	1	1 x X'40' [N]
OVERWRITE	logisches Überschreiben Y: Zielelement/Datei wird überschrieben (YES) N: Zielelement/Datei wird nicht überschrieben (NO) O: Zielelement/Datei muss existieren (ONLY) E: Zielelement/Datei wird erweitert (EXT) A: Alle Zielversionen zum angegebenen Namen werden überschrieben (NAME) (nur bei COPY, sonst Fehler)	1	1 x X'40' [N]
COLUMN-P	Spaltenzahl pro Zeile: 132/80	2	H'0' [132]
LINE-P	Zeilenzahl pro Seite LINE-P= 60; $21 \leq \text{LINE-P} \leq 255$ 0: keine Überschrift ausgeben	2	H'0' [60]
PROT-IND	PROTECTION INDICATOR S: Zielelemente bzw. Dateien erhalten Standard-schutz (STD) M: Zielelemente bzw. Dateien erhalten Schutz der Quelle (Element bzw. Datei) (SAME)	1	1 x X'40' [S]

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung [nach INIT]
ATTR-IND	ATTRIBUTES INDICATOR (COPY-LIB) S: die Zielbibliothek wird mit Standardschutz erzeugt (STD) M: die Attribute der Bibliotheksdatei (Quelle) werden übernommen (SAME)	1	1 x X'40' [N]
INFO	zu bearbeitender Teilbereich eines Elements S: Standardbereich bearbeiten (STD) X'01': eigentlichen Text (Satzart 1) bearbeiten (TXT) X'02': Kommentar/Dokumentation (Satzart 2) bearbeiten (COM)	1	1 x X'40' [S]
LD-RETURN	das Feld NAME der LD als Rückkehrparameter für vollständigen DMS-Dateinamen der Bibliothek nutzen: N: Name wird nur geschrieben, falls Bibliothek durch Dateikettungsamen gegeben Y: vollständiger Bibliotheksname mit CATID und USERID nach ACS-Umsetzung, bzw. Trefferbibliothek bei Bibliothekslisten	1	1 x X'40' [N]
	reserviert	4	4 x X'40'

EA Element Attributes

Mit EA werden die Elementattribute eingestellt und das Änderungsdatum aktualisiert.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
USER-DATE	vom Benutzer vergebbares Datum	14	14 x X'40'
USER-TIME	vom Benutzer vergebbare Uhrzeit	8	8 x X'40'
CCS-NAME	Coded-Character-Set-Name	8	8 x X'40'
HOLD-STATE	Hold Flag: Zustand, der dem Element zugeordnet wird ' ': UNCHANGE '-': FREE 'H': INHOLD reserviert	1	1 x X'40'
		8	8 x X'40'
MOD-DATE-IND	MODIFICATION DATE INDICATOR steuert das Aktualisieren des Änderungsdatums 'O': OLD 'S': NEW (SYSTEM DATE) reserviert	1	1 x X'40'
		56	56 x X'40'

Die Beschreibung EA wird bei der Funktion MODEA benötigt.

Hinweise für die Eingabe

- **USER-DATE**
Die ersten 10 Zeichen des Datums müssen immer die Form YYYY-MM-DD haben. Das Datum wird jedoch nur syntaktisch auf die Form zzzz-zz-zz überprüft. Die restlichen 4 Zeichen werden nicht überprüft, sie sollten aber die Form dddB haben, wobei ddd den laufenden Tag und B Leerzeichen bezeichnen.
- **USER-TIME**
Die Uhrzeit wird nicht syntaktisch überprüft. Sie sollte die Form HH:MM:SS haben.
- **CCS-NAME**
Die eingegebenen Werte werden nicht auf Systemzulässigkeit geprüft. Soll einem Element kein Zeichensatz zugeordnet werden, so muss das Schlüsselwort '*NONE' mit abschließendem Leerzeichen eingetragen werden. Werden nur Leerzeichen angegeben, bleibt der CCS-Name des jeweiligen Elementes unverändert bestehen.

- HOLD-STATE
Zustand, der dem Element zugeordnet wird. Zu setzende Werte:
 - FREE Nur der HOLDER und der Bibliothekseigentümer dürfen den Zustand auf FREE ändern.
 - INHOLD Der neue Zustand des Elementes ist INHOLD. Die USERID desjenigen, der den Zustandswechsel von FREE auf INHOLD änderte, wird als HOLDER eingetragen. War der Zustand bereits INHOLD, wird der UP-Aufruf abgewiesen. Nur wer Ausleihrecht für das Element hat, kann HOLDER werden.
- MOD-DATE-IND
Steuert, ob das Änderungsdatum (EI: MODIFI-TIME und MODIFI-DATE) aktualisiert oder unverändert bleibt. Mögliche Werte:
 - NEW Im Element werden die aktuelle Uhrzeit und das aktuelle Datum als Zeitpunkt der letzten Änderung eingetragen.
 - OLD Der Zeitpunkt der letzten Änderung bleibt unverändert.

ED Element Description

Die ED identifiziert ein Element. Bei folgenden Funktionen gibt es Einschränkungen:

- OPENGET
 - TYP Die Elementtypen C und L sind nur erlaubt, falls beim OPENGET der Subcode EXTRA gesetzt wird.
- OPENPUT, OPENUPD
 - TYP Die Elementtypen C und L sind nicht erlaubt, da Elemente dieser Typen nicht verarbeitbare Sätze enthalten.
- COPY, COPYSTR, DEL, OPENGET, PROVIDE, REN, RETURN, SEL
 - Die Elementbezeichnung des Eingabeelementes wird nicht auf LMS-Konventionen überprüft (Ausnahme: der Elementtyp bei COPY, PROVIDE, REN, RETURN).
- ADD, COPY, COPYSTR, LOCK, OPENPUT, OPENUPD, PROVIDE, REN, RETURN, UNLOCK

Für die Elementbezeichnung des Ausgabeelementes gelten folgende Regeln:

TYP Der Elementtyp muss ein gültiger LMS-Standardtyp (S, M, P, D, J, X, R, C, L, H, F, U) oder ein davon abgeleiteter Typ sein (siehe [1]). Für Delta-Elemente sind die Standardtypen (R, C, L, H, F, U) und davon abgeleitete Typen nicht zugelassen.

NAME

Der Name muss den Konventionen entsprechen (siehe [1]).

VERSION

Die Version muss den Konventionen entsprechen (siehe [1]). @ als Versionsangabe ist nicht erlaubt. Will der Benutzer die höchstmögliche Version erzeugen, muss er explizit X'FF' gefolgt von mindestens einem Leerzeichen angeben. Die Zeichenfolge Vn. (n=0-9) wird nicht in V0n. umgesetzt.

- ADD, COPY, OPENPUT, OPENUPD, PROVIDE, REN, RETURN

USER-DATE

Die ersten 10 Zeichen des Datums müssen immer die Form YYYY-MM-DD haben. Das Datum wird jedoch nur syntaktisch auf zzzz-zz-zz überprüft. Die restlichen 4 Zeichen werden nicht überprüft. Sie sollten die Form dddB haben, ddd bezeichnet den laufenden Tag des Jahres, B bedeutet Leerzeichen.

USER-TIME

Die Uhrzeit wird syntaktisch nicht überprüft. Sie sollte die Form HH:MM:SS haben.

- ADD, COPY, OPENPUT, PROVIDE

STORE-FORM

Für die Speicherungsform muss einer der Werte STD, DELTA oder FULL angegeben werden.

- ADD, COPY, OPENPUT, PROVIDE, REN, RETURN

Für die Elementbezeichnung des Ausgabeelementes gelten folgende Regeln:

Das Versionsfeld der ED für das Ausgabeelement erhält aufgrund des Versionsautomatismus abhängig vom Subcode folgenden Inhalt:

Subcode	Inhalt
INCP	Präfix ¹
HIGP	Präfix ¹
INCB	Basis

¹ Das Präfix darf auch aus der leeren Version (Leerzeichen) bestehen, jedoch maximal 23 Zeichen lang sein.

Ist der Subcode ungleich einem Leerzeichen (Blank), wird bei Delta-Elementen die Basis-ED (ED2 bzw. ED3 bei COPY) ignoriert.

Hat die Ziel-ED einen Elementtyp, für den die Konvention STD-TREE gilt, werden abgekürzte Versionen (z.B. 2.3) in das interne Format (002.003) umgewandelt.

Bei Verwendung des Versionsautomatismus werden die genannten Felder ebenfalls dazu genutzt, die geschriebene Zielversion zurückzuschreiben. Dabei werden dort spezifizierte Präfix- und Basisangaben überschrieben.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TYP	Elementtyp	8	8 x X'40'
NAME	Elementname	64	64 x X'40'
VERSION	Elementversion	24	24 x X'40'
STORE-FORM	Speicherungsform des Elementes 'S': STD: (FULL oder DELTA) 'D': DELTA: Deltaspeicherung 'V': FULL: Vollspeicherung	1	C'V'
USER-DATE	vom Benutzer vergebbares Datum	14	14 x X'40'
USER-TIME	vom Benutzer vergebbare Uhrzeit	8	8 x X'40'

Die Beschreibung ED wird bei folgenden Funktionen benötigt:

ADD, COPY, COPYSTR, DEL, GSYSELEM, LOCK, LST, MODEA, MODEP, OPENGET, OPENPUT, OPENUPD, PROVIDE, REN, RETURN, SEL, UNLOCK.

Bei den Funktionen ADD, COPY, DEL, LOCK, LST, MODEA, MODEP, OPENGET, OPENUPD, OPENPUT, PROVIDE, REN, RETURN, SEL und UNLOCK darf bei der Angabe des Quell- und Zielelementes in der ED im Feld EDVERS die Zeichenfolge *HIGH, gefolgt von einem Leerzeichen, enthalten sein. Der Anwender kann damit die höchste Version zum angegebenen Elementnamen spezifizieren, ohne diese vorher mit der TOC-Funktion ermitteln zu müssen. LMS ermittelt intern die höchste Version und liefert diese im Feld EDVERS zurück, d.h. die Zeichenfolge *HIGH wird von LMS überschrieben. In einer Schleife muss der Anwender das Feld EDVERS immer wieder mit *HIGH neu versorgen.

EI Element Information

Der EI liefert die Elementinformation bei den TOC-Funktionen. Er ist ein reiner Rückkehrparameter. Ohne Erweiterungen ist EI identisch mit dem Format von ED. Man kann sich mit TOC den Bibliotheksinhalt auflisten lassen und die gelieferte Elementinformation sofort für die Bearbeitung des Elementes verwenden. Beim Aufruf von TOCPRIM wird die Erweiterung 1 und 3, beim Aufruf von TOCSEC werden die Erweiterungen 2 und 3 versorgt, wenn der Subcode=LONG gesetzt ist. Die Felder USER-DATE und USER-TIME werden nur bei Subcode=LONG versorgt. CSECT-Namen mit mehr als 32 Zeichen werden zur Ausgabe im Feld SEC-NAME auf 32 Zeichen abgeschnitten.

Aktuell nicht ausgaberelevante Felder sind mit ihren Initialisierungswerten versorgt.

Die Erweiterung 3 erhält bis auf die Kennwörter die Felder von PA, den CCS-Namen, den Elementzustand (FREE/INHOLD), die USERID des Holders und die Felder ACCESS-DATE, ACCESS-TIME und ELEMENT-SIZE. In dem PA sind auch die möglichen Angaben für die einzelnen Anzeigefelder der Schutzattribute beschrieben (siehe [Seite 41](#)).

Durch die Erweiterung 3 ist die Anzeige von Schutzattributen möglich. Kennwörter werden nicht angezeigt. Es wird jedoch ausgegeben, ob ein Kennwort vergeben ist.

Zudem kann zu jedem Element der gespeicherte Zeichensatz-Name ausgegeben werden. Dabei bedeutet die Ausgabe `*NONE` die Ausgabe von „kein Code“.

Ist der Elementzustand INHOLD, wird im Feld HOLDER die USERID des Holders ausgegeben (Holder ist derjenige, der den Elementzustand von FREE auf INHOLD änderte), sonst Leerzeichen. Im Feld ELEMENT-SIZE wird die Anzahl der PAM-Seiten (2K-Einheit) für die Speicherplatzbelegung des Elementes angezeigt.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TYP	Elementtyp	8	8 x X'40'
NAME	Elementname	64	64 x X'40'
VERSION	Elementversion	24	24 x X'40'
STORE-FORM	Speicherungsform des Elementes `D`: DELTA: Delta-Speicherung `V`: FULL: Voll-Speicherung	1	1 x X'40'
USER-DATE	vom Benutzer vergebenes Datum	14	14 x X'40'
USER-TIME	vom Benutzer vergebene Uhrzeit	8	8 x X'40'
	***** Erweiterung 1 ***		
CREATION-DATE	Datum der Erzeugung des Elementes	14	14 x X'40'
CREATION-TIME	Uhrzeit der Erzeugung des Elementes	8	8 x X'40'
MODIFI-DATE	Datum der letzten Änderung des Elementes	14	14 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
MODIFI-TIME	Uhrzeit der letzten Änderung des Elementes	8	8 x X'40'
	***** Erweiterung 2 ***		
SEC-NAME	Referenzname	32	32 x X'40'
SEC-ATTRIBUTE	Referenzattribut	8	8 x X'40'
	reserviert	5	5 x X'40'
	***** Erweiterung 3 ***		
P-TIND-READ	Anzeigefeld für Leserecht	1	1 x X'40'
P-READ-OWN	Anzeigefeld für Leserecht (Eigentümer)	1	1 x X'40'
P-READ-GRP	Anzeigefeld für Leserecht (Gruppe)	1	1 x X'40'
P-READ-OTH	Anzeigefeld für Leserecht (andere)	1	1 x X'40'
P-READ-PIND	Anzeigefeld für Leserecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-WRIT	Anzeigefeld für Schreibrecht	1	1 x X'40'
P-WRIT-OWN	Anzeigefeld für Schreibrecht (Eigentümer)	1	1 x X'40'
P-WRIT-OTH	Anzeigefeld für Schreibrecht (andere)	1	1 x X'40'
P-WRIT-PIND	Anzeigefeld für Schreibrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-EXEC	Anzeigefeld für Ausführungsrecht	1	1 x X'40'
P-EXEC-OWN	Anzeigefeld für Ausführungsrecht (Eigentümer)	1	1 x X'40'
P-EXEC-GRP	Anzeigefeld für Ausführungsrecht (Gruppe)	1	1 x X'40'
P-EXEC-OTH	Anzeigefeld für Ausführungsrecht (andere)	1	1 x X'40'
P-EXEC-PIND	Anzeigefeld für Ausführungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-READ	Anzeigefeld für Leseguard	18	18 x X'40'
P-GUARD-WRIT	Anzeigefeld für Schreibguard	18	18 x X'40'
P-GUARD-EXEC	Anzeigefeld für Ausführungsguard	18	18 x X'40'
CCS-NAME	Coded-Character-Set-Name	8	8 x X'40'
P-TIND-HOLD	Anzeigefeld für Ausleihrecht	1	1 x X'40'
P-HOLD-OWN	Anzeigefeld für Ausleihrecht (Eigentümer)	1	1 x X'40'
P-HOLD-GRP	Anzeigefeld für Ausleihrecht (Gruppe)	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-HOLD-OTH	Anzeigefeld für Ausleihrecht (andere)	1	1 x X'40'
P-HOLD-PIND	Anzeigefeld für Ausleih-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'40'
P-GUARD-HOLD	Anzeigefeld für Ausleihguard	18	18 x X'40'
HOLD-STATE	Hold Flag: Zustand, der dem Element zugeordnet ist '-': FREE 'H': INHOLD	1	1 x X'40'
HOLDER	USERID des Holders	8	8 x X'40'
ACCESS-DATE	Datum des letzten Zugriffs für das Element	14	14 x X'40'
ACCESS-TIME	Uhrzeit des letzten Zugriffs für das Element	8	8 x X'40'
	reserviert	1	1 x X'40'
ELEMENT-SIZE	Anzeigefeld für Elementgröße	4	4 x X'00'
	reserviert	40	40 x X'40'

Der TOC-Parameter EI wird bei folgenden Funktionen benötigt:

TOC, TOCPRIM, TOCSEC.

EM Element Mask

Der EM steuert die Elementauswahl bei den TOC-Funktionen. Er ist ein reiner Aufrufparameter.

Die Erweiterung 3 enthält bis auf die Kennwörter die Felder von PA. In der PA sind die möglichen Angaben für die einzelnen Felder der Schutzattribute beschrieben. Zusätzlich bietet die Erweiterung 3 die Auswahl über die CCS-Namen, den Elementzustand (FREE / INHOLD), die USERID des Holders, die Felder ACCESS-DATE, ACCESS-TIME und die Felder E-SIZE-MIN und E-SIZE-MAX. Damit ist die Auswahl über bestimmte Schutzattribute, nicht aber über bestimmte Kennwörter möglich. Durch die Angabe von Leerzeichen (Blank) bzw. von X'00000000' bei E-SIZE-MIN und X'FFFFFFFF' bei E-SIZE-MAX können auch beliebige Werte zur Auswahl zugelassen werden.

Bei Kennwörtern ist die Auswahl gemäß der Indikatoren (Existenz eines Kennwortes) möglich.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TYP	Elementtyp	20	20 x X'40'
NAME	Elementname	132	132 x X'40'
VERSION	Elementversion	52	52 x X'40'
STORE-FORM	Speicherungsform des Elementes	6	6 x X'40'
USER-DATE	vom Benutzer vergebenes Datum	32	32 x X'40'
USER-TIME	vom Benutzer vergebene Uhrzeit	20	20 x X'40'
	***** Erweiterung 1 ***		
CREATION-DATE	Datum der Erzeugung des Elementes	32	32 x X'40'
CREATION-TIME	Uhrzeit der Erzeugung des Elementes	20	20 x X'40'
MODIFI-DATE	Datum der letzten Änderung des Elementes	32	32 x X'40'
MODIFI-TIME	Uhrzeit der letzten Änderung des Elementes	20	20 x X'40'
	***** Erweiterung 2 ***		
SEC-NAME	Referenzname	68	68 x X'40'
SEC-ATTRIBUTE	Referenzattribut	20	20 x X'40'
	reserviert	14	14 x X'00'
	***** Erweiterung 3 ***		
P-TIND-READ	Maske für Leserecht	1	1 x X'40'
P-READ-OWN	Maske für Leserecht (Eigentümer)	1	1 x X'40'
P-READ-GRP	Maske für Leserecht (Gruppe)	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-READ-OTH	Maske für Leserecht (andere)	1	1 x X'40'
P-READ-PIND	Maske für Leserecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-WRIT	Maske für Schreibrecht	1	1 x X'40'
P-WRIT-OWN	Maske für Schreibrecht (Eigentümer)	1	1 x X'40'
P-WRIT-GRP	Maske für Schreibrecht (Gruppe)	1	1 x X'40'
P-WRIT-OTH	Maske für Schreibrecht (andere)	1	1 x X'40'
P-WRIT-PIND	Maske für Schreibrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-EXEC	Maske für Ausführungsrecht	1	1 x X'40'
P-EXEC-OWN	Maske für Ausführungsrecht (Eigentümer)	1	1 x X'40'
P-EXEC-GRP	Maske für Ausführungsrecht (Gruppe)	1	1 x X'40'
P-EXEC-OTH	Maske für Ausführungsrecht (andere)	1	1 x X'40'
P-EXEC-PIND	Maske für Ausführungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-READ	Maske für Leseguard	40	40 x X'40'
P-GUARD-WRIT	Maske für Schreibguard	40	40 x X'40'
P-GUARD-EXEC	Maske für Ausführungsguard	40	40 x X'40'
CCS-NAME	Coded-Character-Set-Name	20	20 x X'40'
P-TIND-HOLD	Maske für Ausleihrecht	1	1 x X'40'
P-HOLD-OWN	Maske für Ausleihrecht (Eigentümer)	1	1 x X'40'
P-HOLD-GRP	Maske für Ausleihrecht (Gruppe)	1	1 x X'40'
P-HOLD-OTH	Maske für Ausleihrecht (andere)	1	1 x X'40'
P-HOLD-PIND	Maske für Ausleihkennwort Indikator	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-HOLD	Maske für Ausleihguard	40	40 x X'40'
HOLD-STATE	Hold Flag: Elementzustand ' ': ANY '-': FREE 'H': INHOLD	1	1 x X'40'
HOLDER	USERID des Holders	20	20 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
ACCESS-DATE	Datum des letzten Zugriffs für das Element	32	32 x X'40'
ACCESS-TIME	Uhrzeit des letzten Zugriffs für das Element	20	20 x X'40'
	reserviert	3	3 x X'40'
E-SIZE-MIN	Untergrenze der Elementgrößenauswahl (PAM-Seiten, 2K-Einheit)	4	X'00000000'
E-SIZE-MAX	Obergrenze der Elementgrößenauswahl (PAM-Seiten, 2K-Einheit)	4	X'FFFFFFFF'
	reserviert	64	64 x X'40'

Der TOC-Parameter EM wird bei folgenden Funktionen benötigt:

TOCPRI, TOCSEC.

Zulässige Muster für alle Stringfelder:

Muster	Bedeutung
*	Ersetzt eine beliebige (auch leere) Zeichenfolge.
/	Ersetzt genau ein beliebiges Zeichen.
<s1:s2>	Ersetzt eine Zeichenfolge, für die gilt: <ul style="list-style-type: none"> – Mindestlänge = min(L's1,L's2) – Maximallänge = max(L's1,L's2) – sie liegt in der alphabetischen Sortierung zwischen "s1" und "s2" (einschließlich) – "s1" und/oder "s2" dürfen auch leer sein.
<s1:s2,...>	Listenform der Art "s1:s2" <ul style="list-style-type: none"> – Für jede Bereichsangabe gelten die obigen Regeln. – Die Musterliste ersetzt alle Zeichenfolgen, auf die eine der Bereichsangaben zutrifft (ODER-Verknüpfung). – Die Längenmerkmale gelten paarweise, d.h. jeweils für eine Bereichsangabe. <p>Mengenklammern ('<' bzw. '>') müssen paarweise vorhanden sein. In Mengen dürfen die Zeichen '*', '/', '<', und '>' nicht vorkommen.</p>
-s	Ersetzt alle Zeichenfolgen, die dem Muster nicht entsprechen. Das Minuszeichen darf nur am Beginn der Musterzeichenfolge stehen.

Hinweis für die Eingabe:

- Bei einer Bereichsangabe für USER-, CREATION- und MODIFICATION-TIME sind nur Stundenangaben der folgenden Form erlaubt: <HH:HH>*

Bei Angabe des USER-, CREATION- und MODIFICATION-DATE sind nur Datumsangaben der folgenden Form erlaubt: YYYY-MM-DD*

Ein Datumsbereich muss in der folgenden Form angegeben werden:

<YYYY-MM-DD:YYYY-MM-DD>*

Die Angaben sollen stets mit dem Stern '*' abgeschlossen werden, da der Stern für das Julianische Datum steht.

LMS sucht sowohl das angegebene Datum als auch das Julianische, das automatisch bei der Elementerzeugung angelegt wird.

Wird der Stern weggelassen, setzt LMS anstelle des Julianischen Datums vier Leerzeichen. Dies führt dann bei der Auswertung der Elementmaske EM auf Fehler.

- Um Elemente zu suchen, denen „kein Code“ zugeordnet ist, muss im Feld CCS-Name das Schlüsselwort '*NONE ' mit abschließendem Leerzeichen (Blank) eingetragen werden. Werden Leerzeichen eingegeben, so werden Elemente mit beliebigem CCS-Namen ausgewählt.
- CSECT-Namen mit mehr als 32 Zeichen werden vor dem Mustervergleich auf 32 Zeichen abgeschnitten. Die Angabe längerer Muster im Feld SEC-Name ist dafür bedeutungslos.

ER Element Record

Mit ER werden Elementsätze zwischen Rufer und LMS in beiden Richtungen übergeben. Elemente bestehen aus Sätzen variabler Länge, d.h. dem Datenbereich ist immer ein 4 Byte langer Satzkopf vorangestellt.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
	Satzkopf	2	Satzlänge max. 32K (incl. Satzkopf)
		1	reserviert
		1	Satzart
	Satzbereich (siehe GET)	variabel	aktueller Satz

Beim Lesen eines Satzes wird der Satzkopf in ER von LMS versorgt. Der Wert der Satzlänge errechnet sich aus dem Minimum der Satzlänge des gelesenen Satzes und der in RD vom Benutzer angegebenen Pufferlänge. Es können die Satzarten 1-159, 163 und 164 gelesen werden, wobei die Satzarten 1-159 vom Benutzer frei wählbar sind.

Beim Schreiben eines Satzes (PUT) sind die ersten 4 Byte in ER für den Satzkopf reserviert. Der Satzkopf in ER wird bei PUT jedoch nicht ausgewertet.

Der Satzpuffer ER wird bei den Funktionen GET und PUT benötigt.

Ausnahme

Format-B Sätze (Satzart 160)

Format-B Sätze haben keinen Satzkopf. Ihre Länge beträgt ein Vielfaches von 2K, maximal 256 K. Der Satzpuffer (ER) muss dementsprechend groß gewählt werden (siehe auch OPENGET und GET).

FD File Description

Der FD identifiziert eine Datei. Falls ein Dateikettungsname (LINK) vorhanden ist, wird die dazugehörige Datei bearbeitet. Ein zusätzlich bei FD angegebener Dateiname (NAME) wird ignoriert. LMS trägt in das Feld NAME den zu LINK gehörenden Dateinamen ein. Ist kein Dateikettungsname vorhanden, wird die durch NAME bezeichnete Datei bearbeitet.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PASSWORD	Kennwort lt. PASSWORD-Kommando	4	4 x X'00'
LINK	Dateikettungsname	8	8 x X'40'
NAME	Dateiname	54	54 x X'40'

Die Beschreibung FD wird bei den Funktionen ADD und SEL benötigt.

LA Library Attributes

Mit LA kann ein Verwaltungsrecht (Administrationsrecht) für eine Bibliothek vergeben und die für Elemente zulässige Speicherungsform eingestellt werden. Ein Ausleihverfahren kann bibliotheksweit initialisiert werden.

Bei WRITE-CTRL=DEACT gibt es keine zusätzlichen Prüfungen beim Erzeugen oder Überschreiben von Versionen.

Bei WRITE-CTRL=ACTIV darf nur dann eine Version 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 wird 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 HOLD-STATE, HOLDER und alle Rechte auf die neue Version übernommen, sofern durch die aktuelle Anweisung nicht das Eintragen anderer Werte verlangt wird.

Mit ACCESS-DATE=KEEP kann in der Bibliothek für Elemente das Datum des letzten Zugriffs geführt werden.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-TIND-ADMI	Eingabefeld für Verwaltungsrecht N: kein besonderer Schutz Y: Standardschutz einstellen G: Schutz durch GUARD einstellen Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-OWN	Eingabefeld für Verwaltungsrecht (Eigentümer) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-GRP	Eingabefeld für Verwaltungsrecht (Gruppe) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-OTH	Eingabefeld für Verwaltungsrecht (alle anderen) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-PIND	Eingabefeld für Verwaltungsrecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-PSWD	Verwaltungsrecht-Kennwort	4	4 x X'00'
P-GUARD-ADMI	Verwaltungsguard	18	18 x X'40'
STORE-FORM	Speicherungsform für Bibliothek `N`: NONE: (Wirkung wie STD) ` `: UNCHANGED `S`: STD: (FULL oder DELTA) `V`: FULL: Element `D`: DELTA: Element	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
WRITE-CTRL	Write-Control für Bibliothek 'N': NONE: (Wirkung wie 'D') ' ': UNCHANGED 'A': Write-Control für Bibliothek aktivieren 'D': kein Write-Control (deaktivieren)	1	1 x X'40'
ACCESS-DATE	Zugriffsdatum führen ' ': UNCHANGED 'N': kein Zugriffsdatum führen (NONE) 'K': Zugriffsdatum führen (KEEP)	1	1 x X'40'
	reserviert	34	34 x X'40'

Die Beschreibung LA wird bei der Funktion MODLA benötigt.

LD Library Description

Der LD identifiziert eine Bibliothek. Falls ein Dateikettungsname (LINK) vorhanden ist, wird die dazugehörige Bibliothek bearbeitet. Ein zusätzlich bei LD angegebener Bibliotheksname (NAME) wird ignoriert. LMS trägt in das Feld NAME den zu LINK gehörenden Bibliotheksnamen ein. Ist kein Dateikettungsname vorhanden, wird die durch NAME bezeichnete Bibliothek bearbeitet. Die Länge des Bibliotheksnamens wird durch das erste Leerzeichen bestimmt. Ist aber im Feld MAX-NAME-LEN ein kleinerer Wert eingetragen, wird dieser Wert als Länge des Bibliotheksnamens angenommen. Maximal kann der Bibliotheksname die im Feld MAX-NAME-LEN eingetragene Länge (1-54) haben. Es wird empfohlen, MAX-NAME-LEN=54 zu setzen.

Das Feld NAME wird auch als Rückkehrparameter genutzt, abhängig vom Feld LD-RETURN des CB:

- LD-RETURN=NO Ist die Bibliothek durch einen Dateikettungsname gegeben, so trägt LMS in das Feld NAME den zu LINK gehörenden Bibliotheksnamen ein. Andernfalls bleibt das Feld NAME unverändert.
- LD-RETURN=YES LMS trägt in das Feld NAME den vollständigen DMS-Dateinamen der Bibliothek mit CATID und USERID nach ACS-Umsetzung ein bzw. die Trefferbibliothek bei Bibliothekslisten.

Beachten Sie dabei, dass bei LD-RETURN=YES das Feld NAME der LD vor jedem UP-Aufruf mit dem Bibliotheksnamen versorgt werden muss, wenn die gleiche Bibliothek angesprochen werden soll.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PASSWORD	Kennwort lt. PASSWORD-Kommando	4	4 x X'00'
LINK	Dateikettungsname	8	8 x X'40'
	reserviert	8	8 x X'00'
MAX-NAME-LEN	Maximale Länge des Bibliotheksnamens	2	Längenfeld
NAME	Bibliotheksname	54	54 x X'40'

Die Beschreibung LD wird bei folgenden Funktionen benötigt:

ADD, CLOSLIB, COPY, COPYLIB, COPYSTR, DEL, GSYSELEM, LOCK, LST, MODEA, MODEP, MODLA, MODTA, OPENGET, OPENPUT, OPENUPD, PROVIDE, REN, REORGLIB, RETURN, SEL, SHOWLA, SHOWTA, TOCPRI, TOCSEC, UNLOCK.

LI Library Information

Der LI beschreibt das Verwaltungsrecht und die Initialwerte für den Elementschutz. Kennwörter werden nicht angezeigt. Es wird jedoch ausgegeben, ob ein Kennwort vergeben ist. Die Belegung der Anzeigefelder siehe [Seite 41](#).

Zusätzlich werden die Bibliotheksgröße, die freien 2K-Seiten, das Bibliotheksformat und der UPAM-Schutz ausgegeben. Die zulässige Speicherungsform für die Bibliothekselemente, die Indikatoren für das Ausleihverfahren und das Führen des Zugriffsdatum für Elemente werden angezeigt.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-TIND-ADMI	Anzeigefeld für Verwaltungsrecht	1	1 x X'40'
P-ADMI-OWN	Anzeigefeld für Verwaltungsrecht (Eigentümer)	1	1 x X'40'
P-ADMI-GRP	Anzeigefeld für Verwaltungsrecht (Gruppe)	1	1 x X'40'
P-ADMI-OTH	Anzeigefeld für Verwaltungsrecht (alle anderen)	1	1 x X'40'
P-ADMI-PIND	Anzeigefeld für Verwaltungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-ADMI	Anzeigefeld für Verwaltungsguard	18	18 x X'40'
STORE-FORM	Speicherungsform für Bibliothek 'N': NONE: (Wirkung wie STD) 'S': STD: (FULL oder DELTA) 'V': FULL: Element 'D': DELTA: Element	1	1 x X'40'
WRITE-CTRL	Write-Control für Bibliothek 'N': NONE: (Wirkung wie 'D') 'A': Write-Control für Bibliothek aktiviert 'D': kein Write-Control (deaktiviert)	1	1 x X'40'
ACCESS-DATE	Zugriffsdatum führen 'N': kein Zugriffsdatum führen (NONE) 'K': Zugriffsdatum führen (KEEP)		1 x X'40'
	reserviert	24	24 x X'40'
LIB-FORM	Bibliotheksformat '2': NK2-Format '4': NK4-Format	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
UPAM-PROT	Anzeigefeld für UPAM-Schutz `Y` : Bibliothek ist UPAM geschützt `N` : Bibliothek ist nicht UPAM geschützt	1	1 x X'40'
FILE-SIZE	Bibliotheksgröße in 2K-Seiten	4	4 x X'00'
FREE-SIZE	Anzahl freier 2K-Seiten	4	4 x X'00'
P-TIND-READ	Anzeigefeld für Leserecht	1	1 x X'40'
P-READ-OWN	Anzeigefeld für Leserecht (Eigentümer)	1	1 x X'40'
P-READ-GRP	Anzeigefeld für Leserecht (Gruppe)	1	1 x X'40'
P-READ-OTH	Anzeigefeld für Leserecht (andere)	1	1 x X'40'
P-READ-PIND	Anzeigefeld für Leserecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-WRIT	Anzeigefeld für Schreibrecht	1	1 x X'40'
P-WRIT-OWN	Anzeigefeld für Schreibrecht (Eigentümer)	1	1 x X'40'
P-WRIT-GRP	Anzeigefeld für Schreibrecht (Gruppe)	1	1 x X'40'
P-WRIT-OTH	Anzeigefeld für Schreibrecht (andere)	1	1 x X'40'
P-WRIT-PIND	Anzeigefeld für Schreibrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-EXEC	Anzeigefeld für Ausführungsrecht	1	1 x X'40'
P-EXEC-OWN	Anzeigefeld für Ausführungsrecht (Eigentümer)	1	1 x X'40'
P-EXEC-GRP	Anzeigefeld für Ausführungsrecht (Gruppe)	1	1 x X'40'
P-EXEC-OTH	Anzeigefeld für Ausführungsrecht (andere)	1	1 x X'40'
P-EXEC-PIND	Anzeigefeld für Ausführungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-READ	Anzeigefeld für Leseguard	18	18 x X'40'
P-GUARD-WRIT	Anzeigefeld für Schreibguard	18	18 x X'40'
P-GUARD-EXEC	Anzeigefeld für Ausführungsguard	18	18 x X'40'
P-TIND-HOLD	Anzeigefeld für Ausleihrecht	1	1 x X'40'
P-HOLD-OWN	Anzeigefeld für Ausleihrecht (Eigentümer)	1	1 x X'40'
P-HOLD-GRP	Anzeigefeld für Ausleihrecht (Gruppe)	1	1 x X'40'
P-HOLD-OTH	Anzeigefeld für Ausleihrecht (andere)	1	1 x X'40'
P-HOLD-PIND	Anzeigefeld für Ausleih-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-GUARD-HOLD	Anzeigefeld für Ausleihguard	18	18 x X'40'
	reserviert	68	68 x X'40'

Die Beschreibung LI wird bei der Funktion SHOWLA benötigt.

PA Protection Attributes

Der PA legt den Elementschutz und den Kreis der Ausleihberechtigten bzw. die Voreinstellungen für diese Attribute fest.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-TIND-READ	Eingabefeld für Leserecht N: kein besonderer Schutz Y: es soll der Standardschutz vergeben werden G: es soll ein Schutz durch GUARD vergeben werden Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-READ-OWN	Eingabefeld für Leserecht (Eigentümer) N: kein Leserecht Y: Leserecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-READ-GRP	Eingabefeld für Leserecht (Gruppe) N: kein Leserecht Y: Leserecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-READ-OTH	Eingabefeld für Leserecht (andere) N: kein Leserecht Y: Leserecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-READ-PIND	Eingabefeld für Leserecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-READ-PSWD	Leserecht-Kennwort	4	4 x X'00'
P-TIND-WRIT	Eingabefeld für Schreibrecht N: kein besonderer Schutz Y: es ist Standardschutz vergeben G: es soll ein Schutz durch GUARD vergeben werden Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-WRIT-OWN	Eingabefeld für Schreibrecht (Eigentümer) N: kein Schreibrecht Y: Schreibrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-WRIT-GRP	Eingabefeld für Schreibrecht (Gruppe) N: kein Schreibrecht Y: Schreibrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-WRIT-OTH	Eingabefeld für Schreibrecht (andere) N: kein Schreibrecht Y: Schreibrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-WRIT-PIND	Eingabefeld für Schreibrecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-WRIT-PSWD	Schreibrecht-Kennwort	4	4 x X'00'
P-TIND-EXEC	Eingabefeld für Ausführungsrecht N: kein besonderer Schutz Y: es ist Standardschutz vergeben G: es soll ein Schutz durch GUARD vergeben werden Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-EXEC-OWN	Eingabefeld für Ausführungsrecht (Eigentümer) N: kein Ausführungsrecht Y: Ausführungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-EXEC-GRP	Eingabefeld für Ausführungsrecht (Gruppe) N: kein Ausführungsrecht Y: Ausführungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-EXEC-OTH	Eingabefeld für Ausführungsrecht (andere) N: kein Ausführungsrecht Y: Ausführungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-EXEC-PIND	Eingabefeld für Ausführungsrecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-EXEC-PSWD	Ausführungsrecht-Kennwort	4	4 x X'00'
P-GUARD-READ	Eingabefeld für Leseguard	18	18 x X'40'
P-GUARD-WRIT	Eingabefeld für Schreibguard	18	18 x X'40'
P-GUARD-EXEC	Eingabefeld für Ausführungsguard	18	18 x X'40'
P-TIND-HOLD	Eingabefeld für Ausleihrecht N: kein besonderer Schutz Y: es ist ein besonderer Schutz vergeben G: Ausleihrecht durch GUARD geregelt Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-HOLD-OWN	Eingabefeld für Ausleihrecht (Eigentümer) N: kein Ausleihrecht Y: Ausleihrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-HOLD-GRP	Eingabefeld für Ausleihrecht (Gruppe) N: kein Ausleihrecht Y: Ausleihrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-HOLD-OTH	Eingabefeld für Ausleihrecht (andere) N: kein Ausleihrecht Y: Ausleihrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-HOLD-PIND	Eingabefeld für Ausleihrecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-HOLD-PSWD	Ausleihrecht-Kennwort	4	4 x X'00'
P-GUARD-HOLD	Eingabefeld für Ausleihguard	18	18 x X'40'
	reserviert	84	84 x X'40'

Der PA wird bei folgenden Funktionen benötigt: MODEP, MODLA, MODTA.

RD Record Description

Der RD legt die Wiedergewinnungsadresse, die Satzart und die Satzlänge fest.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
REC-ACC-ID	Zugriffspfad-Identifikation	4	4 x X'FF'
BUFFER-LEN	Pufferlänge von ER	4	4 x X'00'
RECORD-LEN	Satzlänge	4	4 x X'00'
	reserviert	3	3 x X'00'
RECORD-TYP	Satzart	1	1 x X'01'
RECORD-NR	Satznummer	4	4 x X'00'
	reserviert	8	8 x X'00'

Die Beschreibung RD wird bei folgenden Funktionen benötigt:

CLOSE, GET, OPENGET, OPENPUT, OPENUPD, PUT.

TA Type Attributes

Der TA legt den Elementtyp und die geltende Konvention für diesen Typ fest. Für einen Benutzertyp kann der übergeordnete Typ eingestellt werden. Mit TA kann ein Verwaltungsrecht (Administrationsrecht) für einen Typ vergeben und die für Elemente zulässige Speicherungsform eingestellt werden. Ein Ausleihverfahren kann typweit initialisiert werden (siehe [Seite 34](#)).

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
CONVENTION	Konvention für den Typ N: keine Konvention S: Konvention STD-SEQUENCE M: Konvention MULTI-SEQUENCE T: Konvention STD-TREE	1	1 x X'40'
	reserviert	3	3 x X'40'
V-EXAMPLE	Versionsbeispiel für die Konventionen STD-SEQUENCE und MULTI-SEQUENCE	24	24 x X'40'
P-TIND-ADMI	Eingabefeld für Verwaltungsrecht N: kein besonderer Schutz Y: Standardschutz einstellen G: Schutz durch GUARD einstellen Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-OWN	Eingabefeld für Verwaltungsrecht (Eigentümer) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-GRP	Eingabefeld für Verwaltungsrecht (Gruppe) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-ADMI-OTH	Eingabefeld für Verwaltungsrecht (andere) N: kein Verwaltungsrecht Y: Verwaltungsrecht Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-ADMI-PIND	Eingabefeld für Verwaltungsrecht-Kennwort N: kein Kennwort vergeben Y: Kennwort vergeben Blank: die aktuelle Einstellung wird nicht verändert	1	1 x X'40'
P-GUARD-ADMI	Eingabefeld für Verwaltungsguard	18	18 x X'40'
STORE-FORM	Speicherungsform für Typ 'N': NONE: Einstellung der Bibliothek gilt ' ': UNCHANGED 'S': STD: (FULL oder DELTA) 'V': FULL: Element 'D': DELTA: Element	1	1 x X'40'
WRITE-CTRL	Write-Control für Typ 'N': NONE: Write-Control der Bibliothek gilt ' ': UNCHANGED 'A': Write-Control für Typk aktivieren 'D': kein Write-Control (deaktivieren)	1	1 x X'40'
SUPER-TYPE	Eingabefeld für übergeordneten Typ <alphanumeric 1..8> '*NONE': es gibt keinen übergeordneten Typ Blank: die aktuelle Einstellung wird nicht verändert	8	8 x X'40'
	reserviert	47	47 x X'40'

Die Beschreibung TA wird bei der Funktion MODTA benötigt.

TD Type Description

Der TD legt den Elementtyp fest.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TYP	Elementtyp	8	8 x X'40'
	reserviert	8	8 x X'40'

Die Beschreibung TD wird bei folgenden Funktionen benötigt: MODTA, SHOWTA.

TI Type Information

Der TI liefert den Elementtyp, den übergeordneten Typ, den Basis-Typ, die geltende Konvention, das Verwaltungsrecht und die Voreinstellungen für die Schutzattribute. Die genaue Belegung der Schutzattribute ist auf [Seite 41](#) zu finden. Kennworte werden nicht ausgegeben. Es wird jedoch angezeigt, ob ein Kennwort angegeben ist. Die zulässige Speicherungsform für den Typ und ein Indikator für das Ausleihverfahren werden angezeigt.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TYP	Elementtyp	8	8 x X'40'
	reserviert	8	8 x X'40'
CONVENTION	Konvention für den Typ N: keine Konvention S: Konvention STD-SEQUENCE M: Konvention MULTI-SEQUENCE T: Konvention STD-TREE	1	1 x X'40'
	reserviert	3	8 x X'40'
V-EXAMPLE	Versionsbeispiel für die Konventionen STD-SEQUENCE und MULTI-SEQUENCE	24	24 x X'40'
P-TIND-ADMI	Anzeigefeld für Verwaltungsrecht	1	1 x X'40'
P-ADMI-OWN	Anzeigefeld für Verwaltungsrecht (Eigentümer)	1	1 x X'40'
P-ADMI-GRP	Anzeigefeld für Verwaltungsrecht (Gruppe)	1	1 x X'40'
P-ADMI-OTH	Anzeigefeld für Verwaltungsrecht (andere)	1	1 x X'40'
P-ADMI-PIND	Anzeigefeld für Verwaltungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-GUARD-ADMI	Anzeigefeld für Verwaltungsguard	18	18 x X'40'
STORE-FORM	Speicherungsform für Typ 'N': NONE: Einstellung der Bibliothek gilt 'S': STD: (FULL oder DELTA) 'V': FULL: Element 'D': DELTA: Element	1	1 x X'40'
WRITE-CTRL	Write-Control für Typ 'N': NONE: Write-Control der Bibliothek gilt 'A': Write-Control für Typk aktiviert 'D': kein Write-Control (deaktiviert)		1 x X'40'
SUPER-TYPE	Anzeigefeld für übergeordneten Typ <alphanum-name 1..8> **NONE': es gibt keinen übergeordneten Typ	8	8 x X'40'
BASIS-TYPE	Anzeigefeld für Basis-Typ <alphanum-name 1..8> **NONE': es gibt keinen Basis-Typ	8	8 x X'40'
	reserviert	39	39 x X'40'
	***** Schutzattributet ****		
P-TIND-READ	Anzeigefeld für Leserecht	1	1 x X'40'
P-READ-OWN	Anzeigefeld für Leserecht (Eigentümer)	1	1 x X'40'
P-READ-GRP	Anzeigefeld für Leserecht (Gruppe)	1	1 x X'40'
P-READ-OTH	Anzeigefeld für Leserecht (andere)	1	1 x X'40'
P-READ-PIND	Anzeigefeld für Leserecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-WRIT	Anzeigefeld für Schreibrecht	1	1 x X'40'
P-WRIT-OWN	Anzeigefeld für Schreibrecht (Eigentümer)	1	1 x X'40'
P-WRIT-GRP	Anzeigefeld für Schreibrecht (Gruppe)	1	1 x X'40'
P-WRIT-OTH	Anzeigefeld für Schreibrecht (andere)	1	1 x X'40'
P-WRIT-PIND	Anzeigefeld für Schreibrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-TIND-EXEC	Anzeigefeld für Ausführungsrecht	1	1 x X'40'
P-EXEC-OWN	Anzeigefeld für Ausführungsrecht (Eigentümer)	1	1 x X'40'
P-EXEC-GRP	Anzeigefeld für Ausführungsrecht (Gruppe)	1	1 x X'40'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
P-EXEC-OTH	Anzeigefeld für Ausführungsrecht (andere)	1	1 x X'40'
P-EXEC-PIND	Anzeigefeld für Ausführungsrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-READ	Anzeigefeld für Leseguard	18	18 x X'40'
P-GUARD-WRIT	Anzeigefeld für Schreibguard	18	18 x X'40'
P-GUARD-EXEC	Anzeigefeld für Ausführungsguard	18	18 x X'40'
P-TIND-HOLD	Anzeigefeld für Ausleihrecht	1	1 x X'40'
P-HOLD-OWN	Anzeigefeld für Ausleihrecht (Eigentümer)	1	1 x X'40'
P-HOLD-GRP	Anzeigefeld für Ausleihrecht (Gruppe)	1	1 x X'40'
P-HOLD-OTH	Anzeigefeld für Ausleihrecht (andere)	1	1 x X'40'
P-HOLD-PIND	Anzeigefeld für Ausleihrecht-Kennwort	1	1 x X'40'
	reserviert	4	4 x X'00'
P-GUARD-HOLD	Anzeigefeld für Ausleihguard	18	18 x X'40'
	reserviert	52	52 x X'40'

Die Beschreibung TI wird bei Funktion SHOWTA benötigt.

TID TOC-Identifikation

Der TID identifiziert über die TOC-Identifikation die gewünschte TOC-Funktion. Dadurch kann verzahnt in mehreren TOC's gleichzeitig gesucht werden. Für TID sind die Werte 1-10 zugelassen. Der Wert muss in das Feld binär und rechtsbündig eingetragen werden.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
TID	TOC-Identifikation	4	aktuelle TOC-ID

Der TOC-Parameter TID wird bei folgenden Funktionen benötigt:

TOC, TOCPRIM, TOCSEC.

Zusammenhang: Funktion - Parameterstruktur

Die folgende Tabelle zeigt den Zusammenhang, welche Funktion welche Parameterstrukturen benötigt. Die Ziffern geben die Reihenfolge der Parameter für die jeweilige Funktion an. Optionale Parameter stehen in Klammern.

Funktion (F-Code)	Parameterstruktur															
	CB	EA	ED	EI	EM	ER	FD	LA	LD	LI	PA	RD	TA	TD	TI	TID
ADD (X'08')	1		4(5)				2		3							
CLOSE (X'13')	1											2				
CLOSLIB(X'1C')	1								2							
COPY (X'0A')	1		3,5(6)						2,4							
COPYLIB (X'1B')	1								2,3							
COPYSTR (X'0B')	1		3,5						2,4							
DEL (X'07')	1		3						2							
END (X'02')	1															
GET (X'11')	1					3						2				
GSYSELEM (X'1F')	1		3						2							
INIT (X'01')	1															
LOCK (X'0C')	1		3						2							
LST (X'14')	1		3						2							
MODEA (X'1A')	1	4	3						2							
MODEP (X'15')	1		3						2		4					
MODLA (X'16')	1							3	2		4					
MODTA (X'18')	1								2		5		4	3		
OPENGET (X'0E')	1		4						3			2				
OPENPUT (X'0F')	1		4(5)						3			2				
OPENUPD (X'10')	1		4						3			2				
PROVIDE (X'1D')	1		3,5(6)						2,4							
PUT (X'12')	1					3						2				
REN (X'06')	1		3,4						2							
REORGLIB (X'20')	1								2							
RETURN (X'1E')	1		3,5,6						2,4							
SEL (X'09')	1		3				4		2							

Funktion (F-Code)	Parameterstruktur															
	CB	EA	ED	EI	EM	ER	FD	LA	LD	LI	PA	RD	TA	TD	TI	TID
SHOWLA (X'17')	1								2	3						
SHOWTA (X'19')	1								2					3	4	
TOC (X'05')	1			3												2
TOCPRI (X'03')	1			3	5				4							2
TOCSEC (X'04')	1			3	5				4							2
UNLOCK (X'0D')	1		3						2							

3.3 Übersicht über die Subcodes

Die Subcodes können nur bei bestimmten Unterprogrammfunktionen angegeben werden. Sie werden im Feld SUBCODE des Kontrollblocks CB angegeben. Wird kein Subcode angegeben, so muss das Feld SUBCODE im Kontrollblock CB auf Leerzeichen (X'40' / UNUSE) gesetzt werden.

Die Subcodes bestimmen:

- das Ausgabeformat von Elementen,
- die Art des Lesen bzw. Schreibens von Sätzen und Elementen,
- die Steuerung des Versionsautomatismus.

Die folgenden Funktionen besitzen einen Subcode:

TOCPRIM, TOCSEC, GET, CLOSE, LST, ADD, COPY, OPENGET, OPENPUT, PROVIDE, REN, RETURN und COPYLIB.

Die nachfolgende Tabelle beinhaltet alle Subcodes, deren Bedeutung und die Zuordnung zu den jeweiligen Unterprogrammfunktionen.

Subcode	Bedeutung	Funktion
SHORT	Ausgabeformat:	
	Die Elementbezeichnung und die Speicherungsform werden ausgegeben.	TOCPRIM
	Elementbezeichnung, Sekundärname, -attribut und Speicherungsform werden ausgegeben.	TOCSEC
LONG	erweitertes Ausgabeformat:	
	Zusätzlich zur Elementbezeichnung und Speicherungsform werden Benutzer-, Erzeugungs- und Änderungsdatum und -uhrzeit sowie CCS-Name, Schutzattribute, Ausleihrechte, Elementzustand, Holder und Zugriffsstempel mit ausgegeben.	TOCPRIM
	Zusätzlich zur Elementbezeichnung und Speicherungsform werden Benutzer-, Erzeugungs- und Änderungsdatum und -uhrzeit, Sekundärname und -attribut sowie CCS-Name, Schutzattribute, Ausleihrechte, Elementzustand, Holder und Zugriffsstempel mit ausgegeben.	TOCSEC

Subcode	Bedeutung	Funktion
DIR	Direktes Lesen eines Satzes: Ausgeben des Satzes, der durch RECORD-TYPE und RECORD-NR im Kontrollblock RD gekennzeichnet ist.	GET
SEQ	Sequentielles Lesen eines Satzes: Es wird als nächstes der Satz ausgegeben, der bezogen auf den letzten GET-Aufruf dieselbe Zugriffspfad-Identifikation hat.	
WRITE	Ausgabeelement zurückschreiben: Der zuletzt geschriebene Elementzustand wird in die Bibliothek zurückgeschrieben.	CLOSE
RESET	Ausgabeelement verwerfen: Der zuletzt geschriebene Elementzustand wird verworfen.	
SYM	Darstellungsart eines Satzes: Die Sätze werden abhängig vom Elementtyp dargestellt.	LST
HEX	Darstellungsart eines Satzes: Der Satz wird zeichenweise und sedizimal übereinander ausgegeben.	
INCP	Versionsautomatismus: Erhöhung mit Präfix	ADD COPY OPENPUT PROVIDE REN RETURN
INCB	Versionsautomatismus: Erhöhung mit Basis	
HIGP	Versionsautomatismus: Das Element mit der höchsten Version innerhalb des Präfix wird überschrieben	
UNUSE	Alle Versionen werden explizit angegeben.	
EXTRA	Ein Element wird unabhängig von der Existenz von Format-B Sätzen zum Lesen geöffnet.	OPENGET
UNUSE	für zukünftige Erweiterungen	alle bisher nicht genannten Funktionen

4 Unterprogrammfunktionen

Das nachfolgende Kapitel enthält zuerst eine Übersicht aller Unterprogrammfunktionen und ihrer Bedeutung. Danach folgt die Beschreibung der Funktionen in alphabetischer Reihenfolge.

4.1 Übersicht über die Funktionen

Funktion	(F-code)	Bedeutung
ADD	(X'08')	Aufnehmen einer Datei als Element
CLOSE	(X'13')	Schließen eines Elementes
CLOSLIB	(X'1C')	Schließen einer Bibliothek
COPY	(X'0A')	Kopieren eines Elementes
COPYLIB	(X'1B')	Kopieren einer Bibliothek
COPYSTR	(X'0B')	Kopieren eines Delta-Baums
DEL	(X'07')	Löschen eines Elementes
END	(X'02')	Beenden eines Unterprogrammzugriffs
GET	(X'11')	Lesen eines Satzes
GSYSELEM	(X'1F')	Lesen einer Elementangabe aus einer S-Variablen
INIT	(X'01')	Eröffnen eines Unterprogrammzugriffs
LOCK	(X'0C')	Sperren eines Elementes
LST	(X'14')	Ausgeben eines Elementes nach SYSLST
MODEA	(X'1A')	Ändern der Elementattribute
MODEP	(X'15')	Ändern des Elementschutzes
MODLA	(X'16')	Ändern der Bibliotheksattribute
MODTA	(X'18')	Ändern der Typattribute
OPENGET	(X'0E')	Eröffnen eines Elementes zum Lesen
OPENPUT	(X'0F')	Eröffnen eines Elementes zum Schreiben
OPENUPD	(X'10')	Eröffnen eines Elementes zum Lesen und Schreiben
PROVIDE	(X'1D')	„Ausleihen“ eines Elementes einer Quellbibliothek

Funktion	(F-code)	Bedeutung
PUT	(X'12')	Schreiben eines Satzes
REN	(X'06')	Umbenennen eines Elementes
REORGLIB	(X'20')	Bibliothek reorganisieren
RETURN	(X'1E')	„Rückgabe“ eines Elementes einer Quellbibliothek
SEL	(X'09')	Ausgeben eines Elementes in eine Datei
SHOWLA	(X'17')	Anzeigen der Bibliotheksattribute
SHOWTA	(X'19')	Anzeigen der Typattribute
TOC	(X'05')	Fortsetzen von TOCPRIM oder TOCSEC
TOCPRIM	(X'03')	Suchen eines Elementes im primären Inhaltsverzeichnis
TOCSEC	(X'04')	Suchen eines Elementes im sekundären Inhaltsverzeichnis
UNLOCK	(X'0D')	Aufheben einer Elementsperr

4.2 Beschreibung der Funktionen

Die Beschreibung der LMS UP-Funktionen ist nach folgenden Punkten gegliedert:

- kurze Beschreibung der Funktionalität
- Besonderheiten (z.B. Subcodes)
- notwendige Parameterstrukturen: Aufruf- und Rückkehrparameter

Hinweis:

Bei den Parameterstrukturen ist die im Handbuch aufgeführte Reihenfolge unbedingt einzuhalten.

ADD Aufnehmen einer Datei als Element

Mit ADD kann eine Datei als Element in eine Bibliothek aufgenommen werden. Mit FD wird die Datei, mit ED1 das Element und mit LD die Bibliothek, in die es aufgenommen werden soll, angegeben.

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB
Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)).

Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED1 ein Datum bzw. eine Uhrzeit eintragen. Soll durch LMS das aktuelle Datum und die aktuelle Uhrzeit eingetragen werden, müssen die Felder mit Leerzeichen besetzt sein.

Beim Zielelement wird das Katalogattribut CCS als Elementattribut eingetragen.

Die Parameter OVERWRITE, RKEY, DESTROY und PROT-IND werden ausgewertet.

Wird OVERWRITE=EXTEND angegeben, muss gelten:

- Im Element dürfen keine ISAM-Schlüssel vorhanden sein.
- Enthält das Element Dateiattribute (Satzart 164), müssen diese mit den Attributen der Datei übereinstimmen.
- Enthält das Element keine Dateiattribute, darf die Datei nicht RECORD-FORMAT=FIXED haben.
- Der CCS-Name der Datei muss mit dem des Elementes übereinstimmen.

OVERWRITE=NAME ist nicht zulässig.

Das Feld ED1.STORE-FORM bestimmt die Speicherungsform für das anzulegende Element. Die Speicherungsform darf den Einstellungen der Typattribute bzw. Bibliotheksattribute nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherungsform haben. Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-Baumes sind.

STORE-FORM=FULL

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

STORE-FORM=STD

Das Element wird gemäß der für den Elementscope geforderten Speicherungsform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

Ist die geforderte Speicherungsform DELTA, wird bei Subcode UNUSE die Basis als Standardbasis bestimmt. Bei allen anderen Subcodes wird die Basis durch die Angabe in ED1.VERSION bestimmt.

STORE-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. Bei Subcode UNUSE muss zusätzlich die Beschreibung ED2 angegeben werden, die das Basiselement bezeichnet. Bei Verwendung des Versionsautomatismus wird die ED2 ignoriert, die Basisversion muss dann in der ED1 angegeben werden.

Behandlung von Delta-Elementen - ohne Versionsautomatismus:

- Soll das Element als Delta-Element aufgenommen werden, muss gelten:

ED1.STORE-FORM=DELTA, ED1.TYP=ED2.TYP, ED1.NAME=ED2.NAME

- Soll das Element als erstes Element eines Delta-Baums aufgenommen werden, darf ein Element dieses Typs und Namens noch nicht existieren und es muss zusätzlich gelten:

ED1.VERSION=ED2.VERSION

- Soll das Element als weiteres Element eines Delta-Baums aufgenommen werden, muss mit ED2 das bereits existierende Basiselement angegeben werden und es muss gelten:

ED1.VERSION \neq ED2.VERSION

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Schnittstellenversion
	FUNCTION	Funktionscode X'08'
	SUBCODE	Versionsautomatismus (UNUSE oder INCP oder INCB oder HIGP)
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	RKEY	ISAM-Schlüssel und Dateiattribute
	OVERWRITE	logisches Überschreiben
	PROT-IND	Elementschutz
	LD-RETURN	vollständiger DMS-Dateiname in LD

Parameter- struktur	Feld	Bedeutung
FD	PASSWORD LINK NAME	Datei-Beschreibung Kennwort lt. PASSWORD-Kommando Dateikettungsname Dateiname
LD	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamens Bibliotheksname
ED1	TYP NAME VERSION STORE-FORM USER-DATE USER-TIME	Element-Beschreibung (Ziel) Elementtyp Elementname Elementversion Speicherungsform vom Benutzer vergebenes Datum vom Benutzer vergebene Uhrzeit
ED2	TYP NAME VERSION	Element-Beschreibung (Basis) falls ED1.STORE-FORM = DELTA und Subcode = Blank Elementtyp Elementname Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
FD	NAME	Datei-Beschreibung Dateiname (falls Datei durch Dateikettungsnamen gegeben)
	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (Zielelement) Elementversion (falls Aufruf mit *HIGH oder bei Versions- automatismus)

CLOSE Schließen eines Elementes

Mit CLOSE kann ein Element, das mit OPENGET, OPENPUT oder OPENUPD eröffnet wurde, explizit geschlossen werden. Beim Schließen werden die benötigten Ressourcen freigegeben. Das Feld REC-ACC-ID des RD wird auf 4 x X'FF' gesetzt.

Bei CLOSE sind zwei Subcodeangaben möglich. Sie werden nur nach einem OPENPUT oder OPENUPD ausgewertet.

SUBCODE=WRITE

Der zuletzt geschriebene Elementzustand wird in der Bibliothek als gültig erklärt. Das Element, das bisher in der Bibliothek unter diesem Namen vorhanden war, wird überschrieben. Auf dieses Element kann nicht mehr zugegriffen werden.

SUBCODE=RESET

Der zuletzt geschriebene Elementzustand wird verworfen. Das Element, das bisher in der Bibliothek unter diesem Namen vorhanden war, bleibt erhalten.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'13'
	ACC	Subcode (WRITE oder RESET)
RD		UP-Zugriffsidentifikation
	REC-ACC-ID	Satzbeschreibung Zugriffspfad-Identifikation

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
RD		Satzbeschreibung
	REC-ACC-ID	Zugriffspfad-Identifikation

CLOSLIB Schließen einer Bibliothek

Mit CLOSLIB kann eine Bibliothek explizit geschlossen werden. Die Bibliothek wird nur geschlossen, falls keine Elemente der Bibliothek mehr geöffnet sind. Sind Elemente offen oder ist ein Element der Bibliothek gegen Veränderung gesperrt (LOCK), wird CLOSLIB mit Returncode X'18' (Aufrufreihenfolge unzulässig) abgelehnt.

Bibliotheken werden durch eine der folgenden Funktionen eröffnet:

ADD, COPY, COPYSTR, DEL, LOCK, LST, MODEA, MODEP, MODLA, MODTA, OPENGET, OPENPUT, OPENUPD, PROVIDE, REN, RETURN, SEL, SHOWLA, SHOWTA, TOCPRIM, TOCSEC.

Die eröffneten Bibliotheken bleiben aus Performancegründen offen. Sie werden i.a. implizit bei END oder bei Ressourcemangel geschlossen. Ein UP-Benutzer weiß im Allgemeinen nicht, ob eine Bibliothek noch geöffnet ist. Daher ist CLOSLIB auch für geschlossene Bibliotheken möglich und liefert den Returncode OK (X'00').

Bei Schließen von Bibliothekslisten gibt es keine Trefferbibliothek und es kann daher kein DMS-Dateiname bestimmt werden. Die Eingabe im LD-Feld NAME bleibt im Fall von Bibliothekslisten unverändert.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'1C'
	ACC	Subcode zur Zeit nicht genutzt: UNUSE
	LD-RETURN	UP-Zugriffsidentifikation
LD	LD-RETURN	vollständiger DMS-Dateiname in LD
	PASSWORD	Bibliothek-Beschreibung
	LINK	Kennwort lt. PASSWORD-Kommando
	MAX-NAME-LEN	Dateikettungsname
	NAME	Maximale Länge des Bibliotheksnamens
		Bibliothekensname

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

COPY Kopieren eines Elementes

Mit COPY kann ein Element kopiert werden.

Mit ED1 wird das Quellelement, mit LD1 die Quellbibliothek, mit ED2 das Zielelement und mit LD2 die Zielbibliothek angegeben.

LD1 und LD2 dürfen dieselbe Bibliothek bezeichnen. Typ, Name und Version des Zielelementes müssen vollständig angegeben werden.

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB
Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)).

Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED2 ein Datum bzw. eine Uhrzeit eintragen. Soll das Datum und die Uhrzeit des Eingabeelementes übernommen werden, müssen die Felder mit Leerzeichen besetzt sein.

Die Parameter OVERWRITE, DESTROY und PROT-IND werden ausgewertet.

Wird OVERWRITE=EXTEND angegeben, muss gelten:

- In den Elementen dürfen keine ISAM-Schlüssel vorhanden sein.
- Enthalten Ein- und Ausgabeelement Dateiattribute (Satzart 164), müssen diese übereinstimmen.
- Enthält nur eines der Elemente Dateiattribute, ist RECORD-FORMAT=FIXED nicht erlaubt.
- Der CCS-Name des Quellelementes muss mit dem des Zielelementes übereinstimmen.

Überschreiben des Zielnamensraums (OVERWRITE=NAME)

Mit OVERWRITE=NAME 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 Zielelements. D.h. alle Benutzerangaben für das Zielelement (wie z.B. INCP) wirken erst auf den leeren Zielnamensraum. Bei INCP wird z.B. immer die Defaultversion erzeugt.

Einschränkungen

- Eingabe- und Ausgabebibliothek müssen unterschiedlich sein.
- Tritt während des Löschens des Zielnamensraums ein Fehler auf (z.B. Element schreibgeschützt), so wird die COPY-Funktion abgebrochen.

Das Feld ED2.STORE-FORM bestimmt die Speicherungsform für das anzulegende Element. Die Speicherungsform darf den Einstellungen der Typattribute bzw. Bibliotheksattribute nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherungsform haben. Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-Baumes sind.

STORE-FORM=FULL

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

STORE-FORM=STD

Das Element wird gemäß der für den Elementscope geforderten Speicherungsform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

Ist die geforderte Speicherungsform DELTA, wird bei Subcode UNUSE die Basis als Standardbasis bestimmt. Bei allen anderen Subcodes wird die Basis durch die Angabe in ED2.VERSION bestimmt.

STORE-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. Bei Subcode UNUSE muss zusätzlich die Beschreibung ED3 angegeben werden, die das Basiselement bezeichnet. Bei Verwendung des Versionsautomatismus wird die ED3 ignoriert, die Basisversion muss dann in der ED2 angegeben werden.

Behandlung von Delta-Elementen - ohne Versionsautomatismus:

- Soll das Element als Delta-Element aufgenommen werden, muss gelten:
ED2.STORE-FORM=DELTA, ED2.TYP=ED3.TYP, ED2.NAME=ED3.NAME
- Soll das Element als erstes Element eines Delta-Baumes aufgenommen werden, darf ein Element dieses Typs und Namens noch nicht existieren und es muss zusätzlich gelten:
ED2.VERSION=ED3.VERSION
- Soll das Element als weiteres Element eines Delta-Baumes aufgenommen werden, muss mit ED3 das bereits existierende Basiselement angegeben werden und es muss gelten:
ED2.VERSION \neq ED3.VERSION

Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-baumes sind.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0A'
	SUBCODE	Versionsautomatismus (UNUSE oder INCP oder INCB oder HIGP)
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	OVERWRITE	logisches Überschreiben
	PROT-IND	Elementschutz (STD oder SAME)
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD1	PASSWORD	Bibliothek-Beschreibung (Quellbibliothek) Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED1	TYP	Element-Beschreibung (Quellelement) Elementtyp
	NAME	Elementname
	VERSION	Elementversion
LD2	PASSWORD	Bibliothek-Beschreibung (Zielbibliothek) Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname

Parameter- struktur	Feld	Bedeutung
ED2	TYP	Element-Beschreibung (Zielelement) Elementtyp
	NAME	Elementname
	VERSION	Elementversion
	STORE-FORM	Speicherungsform
	USER-DATE	vom Benutzer vergebenes Datum
	USER-TIME	vom Benutzer vergebene Uhrzeit
ED3		Element-Beschreibung (Basiselement) falls ED2.STORE-FORM = DELTA und Subcode=Blank
	TYP	Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD1	NAME	Bibliothek-Beschreibung (Quellbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (Quellelement) Elementversion (falls Aufruf mit *HIGH)
LD2	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED2	VERSION	Element-Beschreibung (Zielelement) Elementversion (falls Aufruf mit *HIGH oder bei Versions- automatismus)

COPYLIB Kopieren einer Bibliothek

Mit COPYLIB wird eine komplette Bibliothek logisch mit allen Bibliotheks-, Typ- und Elementattributen kopiert.

Mit LD1 wird die Quellbibliothek und mit LD2 die Zielbibliothek bestimmt. Die Zielbibliothek darf nicht existieren oder sie muss den FCBTYPE=NONE haben.

Bei COPYLIB wird das Feld ATTR-IND des CB ausgewertet. Bei „S“ (SAME) werden die Dateischutzattribute der Quellbibliothek auf die Zielbibliothek übernommen.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION FUNCTION SUBCODE ACC ATTR-IND LD-RETURN	Funktionskontrollblock Schnittstellenversion Funktionscode X'1B' Subcode zur Zeit nicht genutzt: UNUSE UP-Zugriffsidentifikation Bibliotheksattribute kopieren (STD / SAME) vollständiger DMS-Dateiname in LD
LD1	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Quellbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
LD2	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Zielbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD1	NAME	Bibliothek-Beschreibung (Quellbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
LD2	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

COPYSTR Kopieren eines Delta-Baumes

Mit COPYSTR kann ein Delta-Baum strukturerhaltend kopiert werden.

Mit ED1 wird der zu kopierende Delta-Baum angegeben,
mit LD1 die Bibliothek, in der er sich befindet,
mit ED2 Typ und Name des kopierten Delta-Baums und
mit LD2 die Bibliothek, in die er kopiert werden soll.

LD1 und LD2 dürfen dieselbe Bibliothek bezeichnen.

Typ und Name des Ein- und Ausgabe-Delta-Baums müssen vollständig angegeben werden. Es darf noch kein Element des bei ED2 angegebenen Typs und Namens existieren. Angaben zu Version, vom Benutzer vergebenem Datum und Uhrzeit werden nicht ausgewertet. Sie werden unverändert in den Ausgabe-Delta-Baum übernommen. Es werden alle Versionen kopiert.

Die Parameter DESTROY und PROT-IND werden ausgewertet, damit ein Delta-Element gegebenenfalls physikalisch gelöscht werden kann.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0B'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	PROT-IND	Elementschutz (STD oder SAME)
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD1	PASSWORD	Bibliothek-Beschreibung (Quellbibliothek) Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED1	TYP	Element-Beschreibung (Quellelement) Elementtyp
	NAME	Elementname

Parameter- struktur	Feld	Bedeutung
LD2	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Zielbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED2	TYP NAME	Element-Beschreibung (Zielelement) Elementtyp Elementname

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE LMS-MSG DMS-MSG PLAM-MSG	Funktionskontrollblock Returncode LMS-Meldungsschlüssel DMS-Meldungsschlüssel PLAM-Meldungsschlüssel
LD1	NAME	Bibliothek-Beschreibung (Quellbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
LD2	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

DEL Löschen eines Elementes

Mit DEL kann ein Element gelöscht werden. Mit ED wird das zu löschende Element, mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Der Parameter DESTROY wird ausgewertet.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'07'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

END Beenden eines Unterprogrammzugriffs

END ist der letzte Funktionsaufruf an LMS, der den UP-Zugriff beendet. Alle angeforderten Ressourcen (z.B. Speicherplatz) für diesen Zugriff werden freigegeben. Das Feld ACC des CB wird auf 4 x X'FF' gesetzt. Noch geöffnete Elemente werden implizit mit SUBCODE = RESET geschlossen. Alle noch offenen Bibliotheken werden ebenfalls geschlossen.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	SCBVERSION	Schnittstellenversion
	FUNCTION	Funktionscode X'02'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	ACC	UP-Zugriffsidentifikation
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel

GET Lesen eines Satzes

Mit GET kann ein Satz gelesen werden.

Dem Benutzer stehen zwei Subcodes (SUBCODE) zur Verfügung, einen Satz zum Lesen auszuwählen.

- **SUBCODE=SEQ**
Der nächste Satz wird bezogen auf den letzten GET-Aufruf mit derselben Zugriffspfad-Identifikation in das Feld ER ausgegeben. Es genügt beim Aufruf die Felder REC-ACC-ID (siehe OPENGET) und BUFFER-LEN zu versorgen. Der Satzpuffer muss in der Länge BUFFER-LEN angelegt sein. In die Felder RECORD-TYPE bzw. RECORD-NR des RD wird nach dem Lesen die Satzart bzw. die Satznummer hinterlegt. Ist das Ende des Elementes erreicht, wird im Feld RETURNCODE des CB X'08' hinterlegt. Wird die Satzart gewechselt, erfolgt keine Meldung. Der Wechsel kann aber durch Auswerten der Satzart (RECORD-TYPE) erkannt werden.
- **SUBCODE=DIR**
Es müssen zusätzlich die Felder RECORD-TYPE und RECORD-NR versorgt werden. Der Satz wird ausgegeben, der durch RECORD-TYPE und RECORD-NR des RD (Wiedergewinnungsadresse) gekennzeichnet ist. Ist der Satz nicht vorhanden, wird X'0C' im Feld RETURNCODE des CB hinterlegt. RD und ER bleiben unverändert.

SEQ und DIR können beliebig gemischt werden.

Beim Lesen eines Satzes wird der Satzkopf in ER von LMS versorgt. Der Wert der Satzlänge errechnet sich als Minimum der Satzlänge des gelesenen Satzes und der in RD vom Benutzer angegebenen Pufferlänge. Höchstens in dieser Länge wird der Satz übergeben. Ist er zu lang, wird X'04' im Feld RETURNCODE des CB hinterlegt. Die in ER angegebenen Werte zu Satzlänge und Satzart werden in die entsprechenden Felder von RD übernommen.

Nach dem Lesen steht in den Feldern RECORD-TYPE bzw. RECORD-LEN des RD die Satzart bzw. die Satzlänge und in RECORD-NR die satzartrelative Satznummer. In RD steht die wahre Satzlänge. Im Fehlerfall kann aus dieser Satzlänge die Pufferlänge bestimmt werden.

Mit GET können z.B. abwechselnd Sätze aus verschiedenen Satzarten gelesen werden. Ist ein Element mit OPENGET eröffnet worden, kann das betroffene Element nicht überschrieben werden.

Es können die Satzarten 1-159, 163 und 164 (siehe auch [Seite 141](#)) gelesen werden.

Elemente mit Format-B Sätzen können mit GET nur gelesen werden, wenn beim OPENGET der Subcode EXTRA gesetzt wird.

Wird ein Format-B Satz gelesen, liefert LMS in

1. ER den Format-B Satz ohne vier Byte Satzkopf. Der Puffer ER muss 256 K lang sein.
2. RECORD-LEN die Originallänge des Format-B Satzes
3. RECORD-TYPE die Satzart 160
4. RECORD-NR die satzartrelative Satznummer

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'11'
	ACC	Subcode (DIR oder SEQ) UP-Zugriffsidentifikation
RD	REC-ACC-ID	Satzbeschreibung
	BUFFER-LEN	Zugriffspfad-Identifikation
	RECORD-TYPE	Pufferlänge von ER
	RECORD-NR	Satzart (muss bei SEQ nicht angegeben werden) Satznummer (muss bei SEQ nicht angegeben werden)
ER	-	Element-Record-Area keine Versorgung notwendig

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
RD	RECORD-LEN	Satzbeschreibung Satzlänge
	RECORD-TYPE	Satzart
	RECORD-NR	Satznummer
ER		Element-Record-Area Satzinhalt (incl. 4 Byte Satzkopf)

GSYSELEM Lesen einer Elementangabe aus einer S-Variablen

Mit GSYSELEM wird der Inhalt einer S-Variablen gelesen und als Angabe eines Elements (Bibliothek, Elementname, Version und Typ) interpretiert.

Bibliothek (Bibliotheks- oder Dateikettungsname), Elementtyp und Version können vorbelegt werden (Inhalt ungleich Leerzeichen). Sind in der S-Variablen die entsprechenden Angaben weggelassen (bzw. mit dem Standardwert ***BY-PROGRAM** belegt), bleiben nach der Rückkehr die vorbelegten Werte erhalten. Fehlende Angaben ohne Vorbelegung sind unzulässig. Vorbelegungen müssen im Format des UP erfolgen (z.B. höchstmögliche Version: X'FF' und mindestens ein Leerzeichen).

Der Name der S-Variablen ist im Feld NAME der Struktur ED anzugeben. Gesucht wird zunächst eine prozedurlokale, danach eine taskglobale S-Variable dieses Namens.

Ist der Returncode ungleich LMSOK (X'00'), bleiben die Felder in ED und LD unverändert.

Syntax des Inhalts der S-Variablen (Kleinschreibung wird **nicht** in Großschreibung umgesetzt und wird als Syntaxverstoß betrachtet):

```
*LIBRARY-ELEMENT(...)
```

```
  LIBRARY = *BY-PROGRAM / <full-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 = *BY-PROGRAM / *HIGHEST-EXISTING / *UPPER-LIMIT /  
        <composed-name 1..24 with-under>
```

```
  ,TYPE = *BY-PROGRAM / <alphanum-name 1..8>
```

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'1F'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
LD	LINK	Bibliothek-Beschreibung Dateikettungsname (ggf. Vorbelegung)
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamens
	NAME	Bibliotheksname (ggf. Vorbelegung)
ED	TYP	Element-Beschreibung Elementtyp (ggf. Vorbelegung)
	NAME	Name der S-Variablen
	VERSION	Elementversion (ggf. Vorbelegung)

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
LD	LINK	Bibliothek-Beschreibung Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamens
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

INIT Eröffnen eines Unterprogrammzugriffs

Mit INIT wird ein UP-Zugriff eröffnet. Dieser Aufruf ist der erste Funktionsaufruf an LMS. Im Feld ACC des CB wird die UP-Zugriffsidentifikation zurückgegeben. Diese UP-Zugriffsidentifikation wird bei allen weiteren Funktionsaufrufen benötigt. Das Feld LMSVERSION des CB enthält die aktuelle LMS-Versionsnummer. Die Verarbeitungsoperanden werden auf Standardwerte gesetzt.

Es können mehrere INIT-Aufrufe nacheinander erfolgen, damit parallele UP-Zugriffe realisiert werden können. Für jeden UP-Zugriff muss jeweils ein eigener CB angelegt werden, weil bei jedem INIT-Aufruf eine neue UP-Zugriffsidentifikation im Feld ACC des CB hinterlegt wird.

Ist beim Aufruf von INIT ein Fehler aufgetreten, steht im Feld ACC des CB 4 x X'FF'.

Der UP-Zugriff wird mit END und der Angabe der UP-Zugriffsidentifikation beendet.

Aufrufparameter

Parameterstruktur	Feld	Bedeutung
CB		Funktionskontrollblock
	SCBVERSION	Schnittstellenversion
	FUNCTION	Funktionscode X'01'
	SUBCODE	Subcode wird zur Zeit nicht benutzt: UNUSE

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
	ACC	UP-Zugriffsidentifikation
	LMSVERSION	LMS-Version
	DESTROY	physikalisches Überschreiben
	FCB	Dateiattribute (bei Select)
	RKEY	ISAM-Schlüssel und Dateiattribute
	OVERWRITE	logisches Überschreiben
	COLUMN-P	Spalten pro Zeile
	LINE-P	Zeilen pro Seite
	PROT-IND	Elementschutz
	ATTR-IND	Bibliotheksattribute kopieren
INFO	zu bearbeitender Teilbereich eines Elements	
LD-RETURN	das Feld NAME der LD als Rückkehrparameter für vollständigen DMS-Dateinamen der Bibliothek nutzen	

LOCK Sperren eines Elementes

Mit LOCK kann ein Element explizit gegen Veränderung gesperrt werden. Mit ED wird das zu sperrende Element und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Die Elementsperrung wirkt bis ein UNLOCK oder END eingegeben oder das Programm beendet wird. Sie wird nur im Speicher gehalten.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameterstruktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0C'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliothekensname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

LST Ausgeben eines Elementes nach SYSLST

Mit LST wird ein Element nach SYSLST ausgegeben. Mit ED wird das auszugebende Element und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Bei der Ausgabe nach SYSLST werden die Parameter COLUMN, LINE und INFO berücksichtigt. INFO muss den Wert „S“ (STD Standardbereich bearbeiten) haben. LST ist für alle Elementtypen zulässig. Benutzertypen werden dabei entsprechend ihres Basistyps behandelt.

Bei LST stehen dem Benutzer zwei Subcodes (SUBCODE) zur Verfügung, um die Darstellungsform eines Satzes zu bestimmen:

- SUBCODE=SYM

Die Sätze werden abhängig von ihrem Elementtyp dargestellt:

- Bis auf die Typen P, R, L, C und davon abgeleitete Typen werden die Elemente zeichenweise dargestellt.
- Bei Elementen vom Typ P erfolgt die Darstellung ebenfalls zeichenweise, wobei das erste Zeichen eines jeden Satzes als Vorschubsteuerzeichen ausgewertet wird.
- Bei Elementen vom Typ R werden die ESD-, ISD-, RLD-, TXT-, TXTP-, REP- und END-Informationen in aufbereiteter Form ausgegeben. Sonstige Informationen, wie z.B. LSD oder DSDD werden unaufbereitet ausgegeben, d.h. es wird das Satz-längenfeld und ggf. die Satznummer mit ausgegeben. Fortlaufende Textinformation wird nicht aufgeteilt.
- Elemente vom Typ L (LLM) werden in aufbereiteter Form dargestellt.
- Bei Elementen vom Typ C (Phasen) erfolgt die Darstellung zeichenweise und sedezial nebeneinander.

- SUBCODE=HEX

Die Darstellung erfolgt zeichenweise und sedezial übereinander. D.h. für jeden Elementsatz werden zwei Zeilen ausgegeben: in der ersten Zeile wird der Satz zeichenweise und in der zweiten Zeile sedezial dargestellt.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'14'
	SUBCODE	Ausgabeformat (SYM oder HEX)
	ACC	UP-Zugriffsidentifikation
	COLUMN-P	Spaltenanzahl
	LINE-P	Zeilenanzahl
	INFO	zu bearbeitender Teilbereich eines Elements
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

MODEA Ändern der Elementattribute

Mit MODEA werden die Elementattribute für ein bestimmtes Element eingestellt. Mit ED wird das Element bestimmt und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Datum und Uhrzeit können nur gemeinsam geändert werden. Das bedeutet:

- ist das USER-DATE in der Beschreibung EA ein Blank, so wird die USER-TIME in EA ignoriert und die alten Elementangaben bleiben erhalten.
- ist das USER-DATE in der Beschreibung EA ungleich einem Blank, so wird USER-TIME aus EA unverändert übernommen.

MOD-DATE-IND steuert, ob im Element die aktuelle Uhrzeit und das aktuelle Datum als Zeitpunkt der letzten Änderung eingetragen werden.

Ein Element kann als ausgeliehen oder als wieder zurückgegeben gekennzeichnet werden (siehe [Seite 21](#)).

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'1A'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Parameter- struktur	Feld	Bedeutung
EA	USER-DATE USER-TIME CCS-NAME HOLD-STATE MOD-DATE-IND	Elementattribute – Benutzerdatum – Uhrzeit – Coded-Character-Set-Name – Hold Flag: Zustand, der dem Element zugeordnet wird. MODIFICATION DATE INDICATOR steuert die Aktualisierung des Änderungsdatums

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE LMS-MSG DMS-MSG PLAM-MSG	Funktionskontrollblock Returncode LMS-Meldungsschlüssel DMS-Meldungsschlüssel PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

MODEP Ändern des Elementschutzes

Mit MODEP wird der Elementschutz für ein bestimmtes Element eingestellt. Mit ED wird das Element bestimmt und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'15'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion
PA	P-TIND-READ	Schutzattribute Es können alle Felder von PA (Seite 41) vorbelegt werden
	...	

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

MODLA Ändern der Bibliotheksattribute

Mit MODLA werden die Attribute für eine Bibliothek bestimmt. Mit LD wird die Bibliothek angegeben und mit LA deren Attribute festgelegt. Es können ein Administrationsrecht für die Bibliothek und Voreinstellungen für den Elementschutz für neue Elemente vergeben werden. Es kann die für die Elemente der Bibliothek zulässige Speicherungsform definiert, das Führen des Zugriffsdatum aktiviert und ein Ausleihverfahren eingeschaltet werden.

Der PA legt die Voreinstellungen für den Elementschutz und den Kreis der Ausleihberechtigten für die Bibliothek fest.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'16'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
LA	P-TIND-ADMI	Bibliotheksattribute Eingabefeld für Verwaltungsrecht
	P-ADMI-OWN	Eingabefeld für Verwaltungsrecht (Eigentümer)
	P-ADMI-GRP	Eingabefeld für Verwaltungsrecht (Gruppe)
	P-ADMI-OTH	Eingabefeld für Verwaltungsrecht (alle anderen)
	P-ADMI-PIND	Eingabefeld für Verwaltungsrecht-Kennwort
	P-ADMI-PSWD	Verwaltungsrecht-Kennwort
	STORE-FORM	Speicherungsform für Bibliothek
	WRITE-CTRL	Write-Control für Bibliothek
	ACCESS-DATE	Zugriffsdatum für Elemente führen

Parameter- struktur	Feld	Bedeutung
PA	P-TIND-READ ...	Schutzattribute Es können alle Felder von PA (Seite 41) vorbelegt werden

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE LMS-MSG DMS-MSG PLAM-MSG	Funktionskontrollblock Returncode LMS-Meldungsschlüssel DMS-Meldungsschlüssel PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

MODTA Ändern der Typattribute

Mit MODTA werden die Typattribute bestimmt. Durch LD wird die Bibliothek angegeben, durch TD der Elementtyp festgelegt. Durch TA wird die geltende (Versions-) Konvention eingestellt und der übergeordnete Typ festgelegt. Es kann die für die Elemente des Typs zulässige Speicherungsform definiert und das Ausleihverfahren eingeschaltet werden. Mit dem Typ-Verwaltungsrecht wird der Kreis der Administrationsberechtigten für diesen Typ festgelegt. Nur diese dürfen Elemente anlegen, löschen und umbenennen. Diese Einstellungen haben Priorität gegenüber der Einstellung durch MODLA.

Die PA legt die Voreinstellungen für den Elementschutz und den Kreis der Ausleihberechtigten für den Typ fest.

Hinweis

- Der durch SUPER-TYPE entstehende Graph muss zyklenfrei (Baum) sein.
- Für Standardtypen (ein Zeichen lang bzw. Typen, die mit \$ oder SYS beginnen) kann kein SUPER-TYPE vereinbart werden.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameterstruktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'18'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksnamen
TD		Typbeschreibung
	TYP	Elementtyp

Parameter- struktur	Feld	Bedeutung
TA	CONVENTION	Typattribute geltende Konvention für den Elementtyp
	V-EXAMPLE	Versionsbeispiel für die Konventionen STD-SEQUENCE und MULTI-SEQUENCE
	P-TIND-ADMI	Eingabefeld für Verwaltungsrecht
	P-ADMI-OWN	Eingabefeld für Verwaltungsrecht (Eigentümer)
	P-ADMI-GRP	Eingabefeld für Verwaltungsrecht (Gruppe)
	P-ADMI-OTH	Eingabefeld für Verwaltungsrecht (alle anderen)
	P-ADMI-PIND	Eingabefeld für Verwaltungsrecht-Kennwort
	P-ADMI-PSWD	Verwaltungsrecht-Kennwort
	STORE-FORM	Speicherungsform für Typ
	WRITE-CTRL	Write-Control für Typ
PA	SUPER-TYPE	Eingabefeld für übergeordneten Typ
	P-TIND-READ ...	Schutzattribute Es können alle Felder von PA (Seite 41) vorbelegt werden

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

OPENGET Eröffnen eines Elementes zum Lesen

Mit OPENGET wird ein Element zum Lesen eröffnet. Mit ED wird das zu lesende Element und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des bereits existierenden Elementes müssen vollständig angegeben werden.

Bei der Eröffnung wird zur Identifizierung des Zugriffspfades eine Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt. Ist bei der Eröffnung ein Fehler aufgetreten, hat sie den Wert 4 x X'FF'.

Ein Element kann mehrfach zum Lesen eröffnet werden, wenn mehrfach OPENGET mit derselben Element- und Bibliotheksbezeichnung aufgerufen wird. Für jeden OPEN sollte ein eigener RD angelegt werden, weil bei jedem OPENGET eine neue Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt wird. Diese Zugriffspfad-Identifikation wird für alle weiteren Zugriffe und zum Schließen des Elementes benötigt.

Ein Element kann nicht gleichzeitig mit OPENGET, OPENPUT und OPENUPD eröffnet werden.

Nach der Eröffnung steht der Lesezeiger (für sequentielles Lesen) auf dem Elementanfang.

Angaben zu USER-DATE, USER-TIME und STORE-FORM des ED werden nicht ausgewertet.

Elemente mit Format-B Sätzen können mit GET nur gelesen werden, wenn beim OPENGET der Subcode EXTRA gesetzt wird.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0E'
	SUBCODE	EXTRA: Format-B Sätze im Element erlaubt UNUSE: sonst
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
RD	-	Satzbeschreibung keine Versorgung notwendig
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
RD	REC-ACC-ID	Satzbeschreibung Zugriffspfad-Identifikation
	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

OPENPUT Eröffnen eines Elementes zum Schreiben

Mit OPENPUT wird ein Element zum Schreiben eröffnet. Mit ED1 wird das zu schreibende Element und mit LD die Bibliothek, in der es sich befindet, angegeben.

Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB

Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)).

Ist die Bibliothek nicht vorhanden, so wird sie eingerichtet. Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED1 ein Datum bzw. eine Uhrzeit eintragen. Soll das aktuelle Datum und die aktuelle Uhrzeit übernommen werden, müssen die Felder mit Leerzeichen besetzt sein.

Die Parameter OVERWRITE und DESTROY werden ausgewertet. OVERWRITE=NAME ist nicht zulässig. Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-Baumes sind.

Bei der Eröffnung wird zur Identifizierung des Zugriffspfades eine Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt. Ist bei der Eröffnung ein Fehler aufgetreten, hat sie den Wert 4 x X'FF'.

Es können nacheinander mehrere OPENPUT-Aufrufe auf verschiedene Elemente, aber nicht auf dasselbe Element erfolgen. Für jeden Aufruf sollte ein eigener RD angelegt werden, weil bei jedem OPENPUT-Aufruf eine neue Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt wird.

Ein Element kann nicht gleichzeitig mit OPENGET, OPENPUT und OPENUPD eröffnet werden.

Die Elementtypen C und L sind nicht erlaubt, da Elemente dieser Typen nicht verarbeitbare Sätze enthalten.

Das Feld ED1.STORE-FORM bestimmt die Speicherungsform für das anzulegende Element. Die Speicherungsform darf den Einstellungen der Typattribute bzw. Bibliotheksattribute nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherungsform haben. Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-Baumes sind.

STORE-FORM=FULL

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

STORE-FORM=STD

Das Element wird gemäß der für den Elementscope geforderten Speicherungsform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts Besonderes gefordert, wird Vollspeicherung gewählt.

Ist die geforderte Speicherungsform DELTA, wird bei Subcode UNUSE die Basis als Standardbasis bestimmt. Bei allen anderen Subcodes wird die Basis durch die Angabe in ED1.VERSION bestimmt.

STORE-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. Bei Subcode UNUSE muss zusätzlich die Beschreibung ED2 angegeben werden, die das Basiselement bezeichnet. Bei Verwendung des Versionsautomatismus wird die ED2 ignoriert, die Basisversion muss dann in der ED1 angegeben werden.

Behandlung von Delta-Elementen - ohne Versionsautomatismus

- Soll das Element als Delta-Element aufgenommen werden, muss gelten:
ED1.STORE-FORM=DELTA, ED1.TYP=ED2.TYP, ED1.NAME=ED2.NAME
- Soll das Element als erstes Element eines Delta-Baums aufgenommen werden, darf ein Element dieses Typs und Namens noch nicht existieren und es muss zusätzlich gelten:
ED1.VERSION=ED2.VERSION
- Soll das Element als weiteres Element eines Delta-Baums aufgenommen werden, muss mit ED2 das bereits existierende Basiselement angegeben werden und es muss gelten:
ED1.VERSION ? ED2.VERSION

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0F'
	SUBCODE	Versionsautomatismus (UNUSE oder INCP oder INCB oder HIGP)
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	OVERWRITE	logisches Überschreiben
	LD-RETURN	vollständiger DMS-Dateiname in LD

Parameter- struktur	Feld	Bedeutung
RD	-	Satzbeschreibung keine Versorgung notwendig
LD	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED1	TYP NAME VERSION STORE-FORM USER-DATE USER-TIME	Element-Beschreibung (Ziel) Elementtyp Elementname Elementversion Speicherungsform vom Benutzer vergebenes Datum vom Benutzer vergebene Uhrzeit
ED2	TYP NAME VERSION	Element-Beschreibung (Basis) falls ED1.STORE-FORM = DELTA Elementtyp Elementname Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
RD	REC-ACC-ID	Satzbeschreibung Zugriffspfad-Identifikation
	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (Zielelement) Elementversion (falls Aufruf mit *HIGH oder bei Versions- automatismus)

OPENUPD Eröffnen eines Elementes zum Lesen und Schreiben

Mit OPENUPD wird ein Element zum Lesen und Schreiben eröffnet. Mit ED wird das zu lesende und schreibende Element und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des bereits existierenden Elementes müssen vollständig angegeben werden. Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED ein Datum bzw. eine Uhrzeit eintragen. Soll das aktuelle Datum und die aktuelle Uhrzeit übernommen werden, müssen die Felder mit Leerzeichen besetzt sein.

Der Parameter DESTROY wird ausgewertet, damit ein Element gegebenenfalls physikalisch gelöscht werden kann.

Bei der Eröffnung wird zur Identifizierung des Zugriffspfades eine Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt. Ist bei der Eröffnung ein Fehler aufgetreten, hat sie den Wert 4 x X'FF'.

Es können nacheinander mehrere OPENUPD-Aufrufe auf verschiedene Elemente, aber nicht auf dasselbe Element erfolgen. Für jeden sollte ein eigener RD angelegt werden, weil bei jedem OPENUPD-Aufruf eine neue Zugriffspfad-Identifikation im Feld REC-ACC-ID des RD hinterlegt wird.

GET- und PUT-Zugriffe werden koordiniert. Es kann noch die alte Variante gelesen werden, während schon eine neue Variante geschrieben wird. GET- und PUT-Aufrufe, die unter derselben Zugriffspfad-Identifikation aufgerufen werden, beeinflussen sich nicht.

Ein Element kann nicht gleichzeitig mit OPENGET, OPENPUT und OPENUPD eröffnet werden.

Das Feld STORE-FORM des ED wird nicht ausgewertet; der alte Wert kann nicht verändert werden.

Der Elementtypen C und L sind nicht erlaubt, da Elemente dieses Typs nicht verarbeitbare Sätze enthalten.

OPENUPD auf Delta-Elemente ist nur dann möglich, wenn sie Blätter eines Deltabaumes sind.

Wenn nur ein Satz verändert werden soll, muss trotzdem das ganze Element gelesen und geschrieben werden.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'10'
	SUBCODE	Subcode wird zur Zeit nicht genutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	LD-RETURN	vollständiger DMS-Dateiname in LD
RD	-	Satzbeschreibung keine Versorgung notwendig
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung (Quelle, Ziel) Elementtyp
	NAME	Elementname
	VERSION	Elementversion
	USER-DATE	vom Benutzer vergebenes Datum
	USER-TIME	vom Benutzer vergebene Uhrzeit

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
RD	REC-ACC-ID	Satzbeschreibung Zugriffspfad-Identifikation
	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung (Quell- und Zielelement) Elementversion (falls Aufruf mit *HIGH oder bei Versions-automatismus)

PROVIDE Reservieren und kopieren eines Elementes

PROVIDE reserviert ein Element einer Quellbibliothek und stellt eine Kopie in einer Ausgabebibliothek zur Verfügung. Bei aktivem Write Control ist ein reserviertes Element vor anderweitiger Weiterentwicklung geschützt. Bei zusätzlich eingestellter Konvention ist der gesamte Versionsraum über der angegebenen Version für den Holder reserviert.

Mit ED1 wird das Quellelement, mit LD1 die Quellbibliothek, mit ED2 das Zielelement und mit LD2 die Zielbibliothek angegeben. LD1 und LD2 dürfen dieselbe Bibliothek bezeichnen. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB
Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)).

Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED2 ein Datum bzw. eine Uhrzeit eintragen. Soll das Datum und die Uhrzeit des Eingabeelementes übernommen werden, müssen die Felder mit Leerzeichen besetzt sein.

Die Parameter OVERWRITE und DESTROY werden ausgewertet.

OVERWRITE=EXTEND und OVERWRITE=NAME sind nicht zulässig.

Das Feld ED2.STORE-FORM bestimmt die Speicherungsform für das anzulegende Element. Die Speicherungsform darf den Einstellungen der Typattribute bzw. Bibliotheksattribute nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherungsform haben. Delta-Elemente können nur dann überschrieben werden, wenn sie Blätter eines Delta-Baumes sind.

STORE-FORM=FULL

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

STORE-FORM=STD

Das Element wird gemäß der für den Elementscope geforderten Speicherungsform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

Ist die geforderte Speicherungsform DELTA, wird bei Subcode UNUSE die Basis als Standardbasis bestimmt. Bei allen anderen Subcodes wird die Basis durch die Angabe in ED2.VERSION bestimmt.

STORE-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. Bei Subcode UNUSE muss zusätzlich die Beschreibung ED3 angegeben werden, die das Basiselement bezeichnet. Bei Verwendung des Versionsautomatismus wird die ED3 ignoriert, die Basisversion muss dann in der ED2 angegeben werden.

Behandlung von Delta-Elementen - ohne Versionsautomatismus:

- Soll das Element als Delta-Element aufgenommen werden, muss gelten:
ED2.STORE-FORM=DELTA, ED2.TYP=ED3.TYP, ED2.NAME=ED3.NAME
- Soll das Element als erstes Element eines Delta-Baumes aufgenommen werden, darf ein Element dieses Typs und Namens noch nicht existieren und es muss zusätzlich gelten:
ED2.VERSION=ED3.VERSION
- Soll das Element als weiteres Element eines Delta-Baumes aufgenommen werden, muss mit ED3 das bereits existierende Basiselement angegeben werden und es muss gelten:
ED2.VERSION ? ED3.VERSION

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'1D'
		Versionsautomatismus (UNUSE oder INCP oder INCB oder HIGP)
	ACC	UP-Zugriffsidentifikation
	DESTROY	physikalisches Überschreiben
	OVERWRITE	logisches Überschreiben
LD1	LD-RETURN	vollständiger DMS-Dateiname in LD
	PASSWORD	Bibliothek-Beschreibung (Quellbibliothek)
	LINK	Kennwort lt. PASSWORD-Kommando
	MAX-NAME-LEN	Dateikettungsname
ED1	NAME	Maximale Länge des Bibliotheksnamen
	VERSION	Bibliothekensname
	TYP	Element-Beschreibung (Quellelement)
	NAME	Elementtyp
	VERSION	Elementname
		Elementversion

Parameter- struktur	Feld	Bedeutung
LD2	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Zielbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED2	TYP NAME VERSION STORE-FORM USER-DATE USER-TIME	Element-Beschreibung (Zielelement) Elementtyp Elementname Elementversion Speicherungsform vom Benutzer vergebenes Datum vom Benutzer vergebene Uhrzeit
ED3	TYP NAME VERSION	Element-Beschreibung (Basiselement) falls ED2.STORE-FORM = DELTA und Subcode=Blank Elementtyp Elementname Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD1	NAME	Bibliothek-Beschreibung (Quellbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (Quellelement) Elementversion (falls Aufruf mit *HIGH)
LD2	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED2	VERSION	Element-Beschreibung (Zielelement) Elementversion (falls Aufruf mit *HIGH oder bei Versions-automatismus)

PUT Schreiben eines Satzes

Mit PUT kann ein Satz geschrieben werden. Die Satzarten können in beliebiger Reihenfolge geschrieben werden. Innerhalb einer Satzart werden sie grundsätzlich sequentiell geschrieben. Satzlänge (RECORD-LEN) und Satzart (RECORD-TYPE) werden dem RD entnommen. Die ersten vier Zeichen im Puffer müssen für den Satzkopf reserviert werden.

Häufige Wechsel der Satzart sind sowohl aus Zeit- als auch aus Speicherplatzgründen nicht sinnvoll.

Es können die Satzarten 1-159, 163 und 164 (siehe auch [Seite 141](#)) geschrieben werden.

Bei Delta-Elementen dürfen die Satzarten nicht gemischt geschrieben werden.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'12'
	ACC	Subcode wird zur Zeit nicht genutzt: UNUSE
RD	REC-ACC-ID	UP-Zugriffsidentifikation
	RECORD-LEN	Satzbeschreibung
	RECORD-TYPE	Zugriffspfad-Identifikation
ER		Satzlänge
		Satzart
		Satzinhalt (incl. 4 Byte Satzkopf)

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock
	LMS-MSG	Returncode
	DMS-MSG	LMS-Meldungsschlüssel
	PLAM-MSG	DMS-Meldungsschlüssel
		PLAM-Meldungsschlüssel

REN Umbenennen eines Elementes

Mit REN kann ein Element umbenannt werden. Mit ED1 wird das umzubennende Element, mit LD die Bibliothek, in der es sich befindet, angegeben. Mit ED2 wird die neue Bezeichnung des Elementes bestimmt. Sowohl beim Eingabeelement als auch beim Ausgabeelement müssen Typ, Name und Version vollständig angegeben werden.

Beim Ausgabeelement muss zusätzlich noch ein gültiges vom Benutzer zu vergebendes Datum und ebenso eine gültige Uhrzeit angegeben werden. Sollen sich Datum und Uhrzeit gegenüber dem Eingabeelement nicht ändern, müssen die entsprechenden Felder mit Leerzeichen besetzt sein. Datum und Uhrzeit können nur gemeinsam geändert werden. Das bedeutet:

- ist das USER-DATE in der Beschreibung ED2 ein Leerzeichen, so wird die USER-TIME in ED2 ignoriert und die alten Elementangaben bleiben erhalten.
- ist das USER-DATE in der Beschreibung ED2 ungleich einem Leerzeichen, so werden USER-DATE und USER-TIME aus ED2 unverändert übernommen.

Die Parameter DESTROY und OVERWRITE werden ausgewertet. OVERWRITE=EXTEND und OVERWRITE=NAME wirken wie OVERWRITE=NO.

Ein Umbenennen von Delta-Elementen ist nicht möglich. Ebenso darf das Ausgabeelement nicht bereits als Delta-Element existieren.

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB
Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)).

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'06'
	ACC	Versionsautomatismus (INCP oder INCB oder INCB oder HIGP)
	DESTROY	UP-Zugriffsidentifikation
	OVERWRITE	physikalisches Überschreiben
	LD-RETURN	logisches Überschreiben
		vollständiger DMS-Dateiname in LD

Parameter- struktur	Feld	Bedeutung
LD	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED1	TYP NAME VERSION	Element-Beschreibung (alt) Elementtyp Elementname Elementversion
ED2	TYP NAME VERSION USER-DATE USER-TIME	Element-Beschreibung (neu) Elementtyp Elementname Elementversion vom Benutzer vergebenes Datum vom Benutzer vergebene Uhrzeit

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE LMS-MSG DMS-MSG PLAM-MSG	Funktionskontrollblock Returncode LMS-Meldungsschlüssel DMS-Meldungsschlüssel PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (alt) Elementversion (falls Aufruf mit *HIGH)

REORGLIB Bibliothek reorganisieren

Mit REORGLIB wird eine Bibliothek reorganisiert. Dabei wird möglichst viel nicht genutzter Speicherplatz am Ende der Bibliotheksdatei freigegeben. Der für eine Bibliothek erforderliche Speicherplatz kann damit z.T. erheblich reduziert werden. Wird diese Reduzierung nicht erreicht, weil z.B. nach Systemfehler hintere Blöcke zwar bereits als belegt markiert, aber noch nicht benutzt sind oder soll die absolute Minimalgröße der Bibliothek erreicht werden, ist das Kopieren über Zwischenspeicher anzuwenden (COPY-LIBRARY).

Mit LD wird die Bibliothek bestimmt.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'06'
	SUBCODE	Subcode wird zur Zeit nicht benutzt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)

RETURN Rückgabe eines Elementes

RETURN kopiert ein Element einer Quellbibliothek in eine Ausgabebibliothek, wenn die zur Zielversion angegebene Basis in der Ausgabebibliothek vom Benutzer reserviert ist. Es löscht das Element in der Quellbibliothek und gibt die Reservierung in der Ausgabebibliothek wieder zurück.

Zur ersten Version eines Elements gibt es noch keine Basis. RETURN erfordert in diesem Falle Administrationsrecht.

Mit ED1 wird das Quellelement, mit LD1 die Quellbibliothek, mit ED2 das Zielelement und mit LD2 die Zielbibliothek angegeben. LD1 und LD2 dürfen dieselbe Bibliothek bezeichnen. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

ED3 bezeichnet die Basis zum Element ED2. Es muss gelten:

```
ED2.TYP=ED3.TYP, ED2.NAME=ED3.NAME
```

Ist keine Version in ED3.VERSION (=Blank) angegeben, so wird die vom Benutzer ausgeliehene Version als Basisversion genommen (Fehler bei Mehrdeutigkeit).

Es sind vier Subcodeangaben möglich: UNUSE, INCP, HIGP und INCB

Bei Subcodeangaben ungleich UNUSE wird die Versionsangabe besonders behandelt (siehe [Seite 24](#)). Die Versionsangabe in ED3 wird ignoriert.

Zusätzlich kann der Benutzer in die Felder USER-DATE bzw. USER-TIME des ED2 ein Datum bzw. eine Uhrzeit eintragen. Soll das Datum und die Uhrzeit des Eingabeelementes übernommen werden, müssen die Felder mit Leerzeichen besetzt sein.

Die Parameter OVERWRITE und DESTROY werden ausgewertet.

OVERWRITE=EXTEND und OVERWRITE=NAME sind nicht zulässig.

Die Speicherungsform des Zielelements (ED2) ist durch die Basis (ED3) bestimmt.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION FUNCTION SUBCODE ACC DESTROY OVERWRITE LD-RETURN	Funktionskontrollblock Schnittstellenversion Funktionscode X'1E' Versionsautomatismus (UNUSE oder INCP oder INCB oder HIGP) UP-Zugriffsidentifikation physikalisches Überschreiben logisches Überschreiben vollständiger DMS-Dateiname in LD
LD1	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Quellbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED1	TYP NAME VERSION	Element-Beschreibung (Quellelement) Elementtyp Elementname Elementversion
LD2	PASSWORD LINK MAX-NAME-LEN NAME	Bibliothek-Beschreibung (Zielbibliothek) Kennwort lt. PASSWORD-Kommando Dateikettungsname Maximale Länge des Bibliotheksnamen Bibliotheksname
ED2	TYP NAME VERSION STORE-FORM USER-DATE USER-TIME	Element-Beschreibung (Zielelement) Elementtyp Elementname Elementversion Speicherungsform vom Benutzer vergebenes Datum vom Benutzer vergebene Uhrzeit

Parameter- struktur	Feld	Bedeutung
ED3		Element-Beschreibung (Basiselement)
	TYP	Elementtyp = ED2.TYP
	NAME	Elementname = ED2.NAME
	VERSION	Elementversion (Basisversion)

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD1	NAME	Bibliothek-Beschreibung (Quellbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED1	VERSION	Element-Beschreibung (Quellelement) Elementversion (falls Aufruf mit *HIGH)
LD2	NAME	Bibliothek-Beschreibung (Zielbibliothek) vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED2	VERSION	Element-Beschreibung (Zielelement) Elementversion (bei Versionsautomatismus und *HIGH)

SEL Ausgeben eines Elementes in eine Datei

Mit SEL wird ein Element in eine Datei ausgegeben. Mit ED wird das auszugebende Element, mit LD die Bibliothek, in der es sich befindet, und mit FD die Datei, in die es ausgegeben werden soll, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden.

Die Parameter FCBTYPE, OVERWRITE, INFO und PROT-IND werden ausgewertet.

Das Katalogattribut CCS wird aus dem Element übernommen. Bei OVERWRITE=EXTEND muss der CCS-Name des Elementes mit dem der Datei übereinstimmen.

INFO=STD wirkt wie INFO=TXT. Der eigentliche Text, d.h. Satzart 1 wird ausgegeben. Bei INFO=COM wird der separat gespeicherte Kommentar, d.h. Satzart 2 ausgegeben.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'09'
	ACC	Subcode wird zur Zeit nicht unterstützt: UNUSE
	FCB	UP-Zugriffsidentifikation
	OVERWRITE	Dateiattribute
	INFO	logisches Überschreiben
	PROT-IND	zu bearbeitender Teilbereich eines Elements
	LD-RETURN	Dateischutz
LD	LD-RETURN	vollständiger DMS-Dateiname in LD
	PASSWORD	Bibliothek-Beschreibung
	LINK	Kennwort lt. PASSWORD-Kommando
	MAX-NAME-LEN	Dateikettungsname
ED	NAME	Maximale Länge des Bibliotheksnamen
	VERSION	Bibliotheksnamen
	TYP	Element-Beschreibung
ED	NAME	Elementtyp
	VERSION	Elementname
	VERSION	Elementversion

Parameter- struktur	Feld	Bedeutung
FD		Datei-Beschreibung
	PASSWORD	Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	NAME	Dateiname

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
LD	PLAM-MSG	PLAM-Meldungsschlüssel
	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH oder bei Versions- automatismus)

SHOWLA Anzeigen der Bibliotheksattribute

Mit SHOWLA werden die Attribute einer Bibliothek ausgegeben. Mit LD wird die Bibliothek bestimmt. Mit LI wird die gewünschte Information bereitgestellt.

Die aktuellen Einstellungen des Administrationsrechts und die Initialwerte für den Elementschutz werden ausgegeben. Kennwörter werden nicht ausgegeben, sondern es wird nur angezeigt, dass ein Kennwort vergeben ist.

Zusätzlich werden die Bibliotheksgröße, die freien 2K-Seiten, das Bibliotheksformat und ein UPAM-Protection-Indikator ausgegeben.

Angezeigt werden die für die Elemente der Bibliothek zulässige Speicherungsform, Indikatoren für das Ausleihverfahren und das Führen des Zugriffsdatums für Elemente.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'17'
	ACC	Subcode wird zur Zeit nicht unterstützt: UNUSE
	LD-RETURN	UP-Zugriffsidentifikation
LD	LD-RETURN	vollständiger DMS-Dateiname in LD
	PASSWORD	Bibliothek-Beschreibung
	LINK	Kennwort lt. PASSWORD-Kommando
	MAX-NAME-LEN	Dateikettungsname
LI	NAME	Maximale Länge des Bibliotheksnamen
	-	Bibliotheksinformation
		keine Versorgung notwendig

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
LI	P-TIND-ADMI ...	Felder des LI (siehe Seite 38)

SHOWTA Anzeigen der Typattribute

Mit SHOWTA wird die aktuelle Einstellung der Typattribute, des Verwaltungsrechts und die Initialwerte des Elementschutzes ausgegeben. Kennwörter werden nicht angezeigt. Es wird jedoch ausgegeben, ob ein Kennwort vergeben ist. Ausgegeben werden ferner die geltende (Versions-) Konvention, der übergeordnete Typ, der Basistyp, die für Elemente des Typs zulässige Speicherungsform und das Attribut, das das Ausleihverfahren beschreibt.

Durch LD wird die Bibliothek bestimmt, durch TD der Elementtyp. In TI wird das Ergebnis zurückgegeben.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'19'
	SUBCODE	Subcode wird zur Zeit nicht unterstützt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliotheks-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
TD		Typbeschreibung
	TYP	Elementtyp
TI	-	Typinformation keine Versorgung notwendig

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
TI	TYP ...	Typinformation Felder des TI (siehe Seite 48)

TOC Fortsetzen von TOCPRIM oder TOCSEC

TOC setzt eine der Funktionen TOCPRIM oder TOCSEC fort. Welche Funktion TOC fortsetzt, wird über die TOC-Identifikation bestimmt, die vorher in einer TOCPRIM- oder TOCSEC-Funktion vereinbart wurde. Die gleiche TOC-Identifikation kann beliebig oft verwendet werden. Für eine TOC-Identifikation gilt immer die letzte Zuordnung.

Solange es Elementeinträge gibt, die den bei EM in der TOCPRIM- oder TOCSEC-Funktion angegebenen Kriterien entspricht, wird die Elementinformation in die Felder von EI geschrieben. Das Ausgabeformat entspricht der beim TOCPRIM bzw. TOCSEC mit dem Subcode ausgewählten Elementinformation (siehe Funktionen TOCPRIM bzw. TOCSEC).

Entspricht kein weiteres Element den angegebenen Kriterien, wird im Feld RETURNCODE des CB der Code EOF eingetragen. Die Felder von EI bleiben unverändert.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameterstruktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'05'
	ACC	Subcode wird zur Zeit nicht unterstützt: UNUSE UP-Zugriffsidentifikation
TID	-	TOC-Identifikation keine Versorgung notwendig
EI	-	Elementinformation keine Versorgung notwendig

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
EI	TYP	Elementinformation Felder des EI (siehe Seite 26)
	...	gefüllt gemäß Subcode bei TOCPRIM bzw. TOCSEC

TOCPRIM Suchen eines Elementes im primären Inhaltsverzeichnis

TOCPRIM liefert Informationen über die Elementeinträge in der über LD vereinbarten Bibliothek. Dazu wird im primären Inhaltsverzeichnis alphabetisch nach den Elementeinträgen (TYP NAME VERSION) gesucht.

Bei jedem Aufruf von TOCPRIM muss eine zulässige TOC-Identifikation angegeben werden, damit sich ein nachfolgendes TOC auf diese Identifikation beziehen kann. Für die TOC-Identifikation dürfen die Werte 1,...,10 angegeben werden. Bei TOCPRIM-Funktionen mit gleichen Werten für die TOC-Identifikation gilt die zuletzt getroffene Zuordnung.

Für das erste Element, das den in EM angegebenen Kriterien entspricht, wird die Elementinformation in die Felder von EI geschrieben.

Entspricht kein Element den angegebenen Kriterien, wird im Feld RETURNCODE des CB EOF eingetragen. Die Felder von EI bleiben unverändert.

Ein Element entspricht den Kriterien, wenn es zu der in EM angegebenen Maske passt, d.h. jedes Feld des Elementes muss zu dem entsprechenden Feld bzw. bei der Bereichsangabe für die Elementgröße zu den entsprechenden Feldern der Maske passen.

Auswertung der Stringfelder der Maske EM:

Jedes Stringfeld wird bis zum ersten Leerzeichen ausgewertet. Ein Stringfeld, das mit einem Leerzeichen beginnt, wird als leer angenommen. Ein Leerzeichen am Anfang der Maske bedeutet das gleiche wie die Angabe '*', das Elementfeld passt immer zu dem Maskenfeld.

Auswertung der Felder E-SIZE-MIN und E-SIZE-MAX:

Es passen alle Elemente mit einer Größe (in PAM-Seiten) für die gilt:
 $E-SIZE-MIN \leq \text{Größe} \leq E-SIZE-MAX$. Durch die Angabe von X'00000000' bei E-SIZE-MIN und X'FFFFFFFF' bei E-SIZE-MAX können auch beliebige Werte zur Auswahl zugelassen werden.

Im Feld VERSION des EM können zusätzlich zwei Schlüsselwörter eingegeben werden:

- *HIGH bzw. *HIGH|prefix

Es wird nur nach der höchsten existierenden Version bzw. nach der höchsten existierenden Version mit angebbarem Präfix gesucht.

- *LOW bzw. *LOW|prefix

Es wird nur nach der niedrigsten existierenden Version bzw. nach der niedrigsten existierenden Version mit angebbarem Präfix gesucht.

Die Eingaben müssen links im Feld VERSION des EM beginnen und durch ein Leerzeichen beendet werden. Ein zusätzlich in diesem Feld angegebenes Zeichen entwertet die Schlüsselwörter. Soll nach einer Version gesucht werden, die mit HIGH endet, genügt die Angabe „**HIGH“.

Für die Ausgabe gibt es zwei Formate (Subcodes) zur Auswahl:

- SHORT
Die zurückgegebene Elementinformation beinhaltet nur die Elementbezeichnung (TYP, NAME, VERSION) und die Speicherungsform (STORE-FORM). In den übrigen Feldern befinden sich Leerzeichen.
Diese Ausgabeform eignet sich beim Programmieren von Schleifen im rufenden Programm und wenn bei EM nur Bedingungen für die Elementbezeichnung angegeben wurden.
- LONG
Die Elementinformation wird mit der Erweiterung 1 und 3 ausgegeben. Es werden keine Sekundärnamen ausgegeben. Die Felder SEC-NAME und SEC-ATTRIBUTE des EM werden nicht ausgewertet.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameter- struktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock
	FUNCTION	Schnittstellenversion
	SUBCODE	Funktionscode X'03'
	ACC	Subcode (SHORT oder LONG)
	LD-RETURN	UP-Zugriffsidentifikation vollständiger DMS-Dateiname in LD
TID	-	TOC-Identifikation keine Versorgung notwendig
EI	-	Elementinformation keine Versorgung notwendig
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname

Parameter- struktur	Feld	Bedeutung
EM		Element-Maske (Such-Muster)
	TYP	Elementtyp
	NAME	Elementname
	VERSION	Elementversion
	STORE-FORM	Speicherungsform
	USER-DATE	vom Benutzer vergebenes Datum
	USER-TIME	vom Benutzer vergebene Uhrzeit
	CREATION-DATE	Datum der Erzeugung des Elementes
	CREATION-TIME	Uhrzeit der Erzeugung des Elementes
	MODIFI-DATE	Datum der letzten Änderung
	MODIFI-TIME	Uhrzeit der letzten Änderung
	P-TIND-READ	Anzeigefeld für Leserecht
	P-READ-OWN	Anzeigefeld für Leserecht (Eigentümer)
	P-READ-GRP	Anzeigefeld für Leserecht (Gruppe)
	P-READ-OTH	Anzeigefeld für Leserecht (alle anderen)
	P-READ-PIND	Anzeigefeld für Leserecht-Kennwort
	P-TIND-WRIT	Anzeigefeld für Schreibrecht
	P-WRIT-OWN	Anzeigefeld für Schreibrecht (Eigentümer)
	P-WRIT-GRP	Anzeigefeld für Schreibrecht (Gruppe)
	P-WRIT-OTH	Anzeigefeld für Schreibrecht (alle anderen)
	P-WRIT-PIND	Anzeigefeld für Schreibrecht-Kennwort
	P-TIND-EXEC	Anzeigefeld für Ausführungsrecht
	P-EXEC-OWN	Anzeigefeld für Ausführungsrecht (Eigentümer)
	P-EXEC-GRP	Anzeigefeld für Ausführungsrecht (Gruppe)
	P-EXEC-OTH	Anzeigefeld für Ausführungsrecht (alle anderen)
	P-EXEC-PIND	Anzeigefeld für Ausführungsrecht-Kennwort
	P-GUARD-READ	Leseguard
	P-GUARD-WRIT	Schreibguard
	P-GUARD-EXEC	Ausführungsguard
	CCS-NAME	Coded-Character-Set-Name
P-TIND-HOLD	Anzeigefeld für Ausleihrecht	
P-HOLD-OWN	Anzeigefeld für Ausleihrecht (Eigentümer)	

Parameter- struktur	Feld	Bedeutung
	P-HOLD-GRP	Anzeigefeld für Ausleihrecht (Gruppe)
	P-HOLD-OTH	Anzeigefeld für Ausleihrecht (alle anderen)
	P-HOLD-PIND	Anzeigefeld für Ausleihrecht-Kennwort
	P-GUARD-HOLD	Ausleihguard
	HOLD-STATE	Hold Flag: Elementzustand
	ACCESS-DATE	Datum des letzten Zugriffs für das Element
	ACCESS-TIME	Uhrzeit des letzten Zugriffs für das Element
	E-SIZE-MIN	Untergrenze der Elementgrößenwahl (PAM-Seiten, 2K-Einheit)
	E-SIZE-MAX	Obergrenze der Elementgrößenwahl (PAM-Seiten, 2K-Einheit)

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
LD	PLAM-MSG	PLAM-Meldungsschlüssel
	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
EI		Elementinformation
	TYP ...	Felder des EI (siehe Seite 26) gefüllt gemäß Subcode bei TOCPRIM bzw. TOCSEC

TOCSEC Suchen eines Elementes im sekundären Inhaltsverzeichnis

TOCSEC liefert Informationen über die Elementeinträge in der über LD vereinbarten Bibliothek. Dazu wird im sekundären Inhaltsverzeichnis alphabetisch nach den Elementeinträgen (TYP SEC-NAME SEC-ATTRIBUTE NAME VERSION) gesucht.

Bei jedem Aufruf von TOCSEC muss eine zulässige TOC-Identifikation angegeben werden, damit sich ein nachfolgendes TOC auf diese Funktion beziehen kann. Für die TOC-Identifikation dürfen die Werte 1,...,10 angegeben werden. Bei TOCSEC-Funktionen mit gleichen Werten für die TOC-Identifikation gilt die zuletzt gemachte TOCSEC-Funktion.

Für das erste Element, das den in EM angegebenen Kriterien entspricht, wird die Elementinformation in die Felder von EI geschrieben. Entspricht kein Element den angegebenen Kriterien, wird im Feld RETURNCODE des CB EOF eingetragen. Die Felder von EI bleiben unverändert.

Ein Element entspricht den Kriterien, wenn es zu der in EM angegebenen Maske passt, d.h. jedes Feld des Elementes muss zu dem entsprechenden Feld bzw. der Bereichsangabe für die Elementgröße zu den entsprechenden Feldern der Maske passen.

Auswertung der Stringfelder der Maske EM:

Jedes Stringfeld wird bis zum ersten Leerzeichen ausgewertet. Ein Stringfeld, das mit einem Leerzeichen beginnt, wird als leer angenommen. Ein Leerzeichen am Anfang der Maske bedeutet das gleiche wie die Angabe "*", das Elementfeld passt immer zu dem Maskenfeld.

Auswertung der Felder E-SIZE-MIN und E-SIZE-MAX:

Es passen alle Elemente mit einer Größe (in PAM-Seiten) für die gilt:
 $E-SIZE-MIN \leq \text{Größe} \leq E-SIZE-MAX$. Durch die Angabe von X'00000000' bei E-SIZE-MIN und X'FFFFFFF' bei E-SIZE-MAX können auch beliebige Werte zur Auswahl zugelassen werden.

Zusätzlich zu den Maskenfeldern der TOCPRI-M-Funktion werden die Felder SEC-NAME und SEC-ATTRIBUTE des EM als Suchkriterium herangezogen. CSECT-Namen mit mehr als 32 Zeichen werden vor dem Mustervergleich auf 32 Zeichen abgeschnitten.

Im Feld VERSION des EM können zwei Schlüsselwörter eingegeben werden:

- *HIGH
Es wird nur nach der höchsten existierenden Version gesucht.
- *LOW
Es wird nur nach der niedrigsten existierenden Version gesucht.

Die Eingaben müssen links im Feld VERSION des EM beginnen und durch ein Leerzeichen beendet werden.

Ein zusätzlich in diesem Feld angegebenes Zeichen entwertet die Schlüsselwörter. Soll nach einer Version gesucht werden, die mit HIGH endet, genügt die Angabe „**HIGH“.

Für die Ausgabe gibt es zwei Formate zur Auswahl:

- SHORT
Die zurückgegebene Elementinformation beinhaltet nur die Elementbezeichnung (TYP, NAME, VERSION), den Sekundärnamen (SEC-NAME), das Sekundärattribut (SEC-ATTRIBUTE) und die Speicherungsform (STORE-FORM). In den übrigen Feldern befinden sich Leerzeichen.
Diese Ausgabeform eignet sich beim Programmieren von Schleifen im rufenden Programm und wenn bei EM nur Bedingungen für die Elementbezeichnung, den Sekundärnamen und das Sekundärattribut (SEC-ATTRIBUTE) angegeben wurden.
- LONG
Die Elementinformation wird mit den Erweiterungen 1, 2 und 3 ausgegeben. CSECT-Namen mit mehr als 32 Zeichen werden zur Ausgabe im Feld SEC-NAME auf 32 Zeichen abgeschnitten. Werden solche Namen vollständig benötigt, so müssen die Sekundärsätze des Elementes mit GET gelesen werden.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameterstruktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'04'
	SUBCODE	Subcode (SHORT oder LONG)
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
TID	-	TOC-Identifikation keine Versorgung notwendig
EI	-	Elementinformation keine Versorgung notwendig
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname

Parameter- struktur	Feld	Bedeutung
EM		Element-Maske (Such-Muster)
	TYP	Elementtyp
	NAME	Elementname
	VERSION	Elementversion
	STORE-FORM	Speicherungsform
	USER-DATE	vom Benutzer vergebenes Datum
	USER-TIME	vom Benutzer vergebene Uhrzeit
	CREATION-DATE	Datum der Erzeugung des Elementes
	CREATION-TIME	Uhrzeit der Erzeugung des Elementes
	MODIFI-DATE	Datum der letzten Änderung
	MODIFI-TIME	Uhrzeit der letzten Änderung
	SEC-NAME	Referenzname
	SEC-ATTRIBUTE	Referenzattribut
	P-TIND-READ	Anzeigefeld für Leserecht
	P-READ-OWN	Anzeigefeld für Leserecht (Eigentümer)
	P-READ-GRP	Anzeigefeld für Leserecht (Gruppe)
	P-READ-OTH	Anzeigefeld für Leserecht (alle anderen)
	P-READ-PIND	Anzeigefeld für Leserecht-Kennwort
	P-TIND-WRIT	Anzeigefeld für Schreibrecht
	P-WRIT-OWN	Anzeigefeld für Schreibrecht (Eigentümer)
	P-WRIT-GRP	Anzeigefeld für Schreibrecht (Gruppe)
	P-WRIT-OTH	Anzeigefeld für Schreibrecht (alle anderen)
	P-WRIT-PIND	Anzeigefeld für Schreibrecht-Kennwort
	P-TIND-EXEC	Anzeigefeld für Ausführungsrecht
	P-EXEC-OWN	Anzeigefeld für Ausführungsrecht (Eigentümer)
	P-EXEC-GRP	Anzeigefeld für Ausführungsrecht (Gruppe)
	P-EXEC-OTH	Anzeigefeld für Ausführungsrecht (alle anderen)
	P-EXEC-PIND	Anzeigefeld für Ausführungsrecht-Kennwort
	P-GUARD-READ	Leseguard
	P-GUARD-WRIT	Schreibguard
	P-GUARD-EXEC	Ausführungsguard
CCS-NAME	Coded-Character-Set-Name	

Parameter- struktur	Feld	Bedeutung
	P-TIND-HOLD	Anzeigefeld für Ausleihrecht
	P-HOLD-OWN	Anzeigefeld für Ausleihrecht (Eigentümer)
	P-HOLD-GRP	Anzeigefeld für Ausleihrecht (Gruppe)
	P-HOLD-OTH	Anzeigefeld für Ausleihrecht (alle anderen)
	P-HOLD-PIND	Anzeigefeld für Ausleihrecht-Kennwort
	P-GUARD-HOLD	Ausleihguard
	HOLD-STATE	Hold Flag: Elementzustand
	HOLDER	USERID des Holders
	ACCESS-DATE	Datum des letzten Zugriffs für das Element
	ACCESS-TIME	Uhrzeit des letzten Zugriffs für das Element
	E-SIZE-MIN	Untergrenze der Elementgrößenwahl (PAM-Seiten, 2K-Einheit)
	E-SIZE-MAX	Obergrenze der Elementgrößenwahl (PAM-Seiten, 2K-Einheit)

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB		Funktionskontrollblock
	RETURNCODE	Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD		Bibliothek-Beschreibung
	NAME	vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
EI		Elementinformation
	TYP	Felder des EI (siehe Seite 26)
	...	gefüllt gemäß Subcode bei TOCPRIM bzw. TOCSEC

UNLOCK Aufheben einer Elementsperr

Mit UNLOCK kann die mit LOCK gesetzte Sperre für ein Element aufgehoben werden. Mit ED wird das Element, dessen Sperre aufgehoben werden soll, und mit LD die Bibliothek, in der es sich befindet, angegeben. Typ, Name und Version des Elementes müssen vollständig angegeben werden. Die Elementsperr kann nur innerhalb des UP-Zugriffs aufgehoben werden, in dem sie gesetzt wurde.

Aufrufparameter

Die Parameterstrukturen müssen beim UP-Aufruf in dieser Reihenfolge stehen.

Parameterstruktur	Feld	Bedeutung
CB	SCBVERSION	Funktionskontrollblock Schnittstellenversion
	FUNCTION	Funktionscode X'0D'
	SUBCODE	Subcode wird zur Zeit nicht unterstützt: UNUSE
	ACC	UP-Zugriffsidentifikation
	LD-RETURN	vollständiger DMS-Dateiname in LD
LD	PASSWORD	Bibliothek-Beschreibung Kennwort lt. PASSWORD-Kommando
	LINK	Dateikettungsname
	MAX-NAME-LEN	Maximale Länge des Bibliotheksnamen
	NAME	Bibliotheksname
ED	TYP	Element-Beschreibung Elementtyp
	NAME	Elementname
	VERSION	Elementversion

Rückkehrparameter

Parameter- struktur	Feld	Bedeutung
CB	RETURNCODE	Funktionskontrollblock Returncode
	LMS-MSG	LMS-Meldungsschlüssel
	DMS-MSG	DMS-Meldungsschlüssel
	PLAM-MSG	PLAM-Meldungsschlüssel
LD	NAME	Bibliothek-Beschreibung vollständiger DMS-Dateiname der Bibliothek (abhängig vom Feld LD-RETURN des CB)
ED	VERSION	Element-Beschreibung Elementversion (falls Aufruf mit *HIGH)

4.3 Programmierhilfen

Das nachfolgende Kapitel gibt Empfehlungen, welche symbolischen Namen verwendet werden können und beschreibt Sätze der Satzarten 163 und 164.

4.3.1 Symbolische Namen

Für die Verarbeitungsoperanden, Funktionscodes, Subcodes, Returncodes und für die Speicherform der Elemente stehen eine Reihe von symbolischen Namen bzw. Equates zur Verfügung.

LMS Verarbeitungsparameter-Werte

Symbol	Wert	Bedeutung
		--- zur allgemeinen Verwendung ---
YES	'Y'	YES
NO	'N'	NO
NONE	'N'	NONE
ANY	' '	ANY
UNCH	' '	UNCHANGED
SAME	'M'	SAME
STD	'S'	STD
		--- für CB-Feld FCB ---
ISAM	'I'	ISAM
SAM	'Q'	SAM
CAT	'C'	CAT
		--- für CB-Feld OV ---
EXT	'E'	EXTEND
ONLY	'O'	ONLY
NAME	'A'	NAME
		--- für CB-Feld INFO ---
TXT	X'01'	TEXT ONLY
COM	X'02'	COMMENT / DOCUMENTATION

Funktionscodes

Symbol	Wert	Bedeutung	Aufrufparameter
INIT	X'01'	INIT	CB
END	X'02'	END	CB
TOCP	X'03'	TOCPRIM	CB, TID,EI, LD, EM
TOCS	X'04'	TOCSEC	CB, TID,EI, LD, EM
TOC	X'05'	TOC	CB, TID,EI
REN	X'06'	REN	CB, LD,ED1,ED2
DEL	X'07'	DEL	CB, LD, ED
ADD	X'08'	ADD	CB, FD, LD, ED1 [,ED2]
SEL	X'09'	SEL	CB, LD, ED, FD
COPY	X'0A'	COPY	CB, LD1,ED1,LD2,ED2 [,ED3]
COPST	X'0B'	COPYSTR	CB, LD1,ED1,LD2,ED2
LOCK	X'0C'	LOCK	CB, LD, ED
UNLK	X'0D'	UNLOCK	CB, LD, ED
OPENG	X'0E'	OPEN GET	CB, RD, LD, ED
OPENP	X'0F'	OPEN PUT	CB, RD, LD, ED1 [,ED2]
OPENU	X'10'	OPEN UPD	CB, RD, LD, ED
GET	X'11'	GET	CB, RD, ER
PUT	X'12'	PUT	CB, RD, ER
CLOSE	X'13'	CLOSE	CB, RD
LST	X'14'	LIST ELEMENT	CB, LD, ED
MEP	X'15'	MODIFY ELEMENT PROTECTION	CB, LD, ED, PA
MLA	X'16'	MODFIY LIBRARY ATTRIBUTES	CB, LD, LA, PA
SLA	X'17'	SHOW LIBRARY ATTRIBUTES	CB, LD, LI
MTA	X'18'	MODFIY TYPE ATTRIBUTES	CB, LD, TD, TA, PA
STA	X'19'	SHOW TYPE ATTRIBUTES	CB, LD, TD, TI
MEA	X'1A'	MODIFY ELEMENT ATTRIBUTES	CB, LD, ED, EA
COPLB	X'1B'	COPY LIBRARY	CB, LD1,LD2
CLOLB	X'1C'	CLOSE LIBRARY	CB, LD
PROVI	X'1D'	PROVIDE ELEMENT	CB, LD1,ED1,LD2,ED2 [,ED3]
RETUR	X'1E'	RETURN ELEMENT	CB, LD1,ED1,LD2,ED2,ED3
GSYSE	X'1F'	GET SYSELEM	CB, LD, ED
REOLB	X'20'	REORGANIZE LIB	CB, LD

Subcodes

Symbol	Wert	Bedeutung
UNUSE	' '	DEFAULT
SHORT	'S'	TOC SHORT
LONG	'L'	TOC LONG
DIR	'D'	READ DIRECT
SEQ	'S'	READ SEQUENTIAL
WRITE	'W'	CLOSE OUTPUT ELEMENT FOR WRITE
RESET	'R'	FORGET OUTPUT ELEMENT
SYM	'S'	SHOW-ELEMENT SYMBOLIC
HEX	'H'	SHOW-ELEMENT ALPHA+HEX
INCP	'P'	INCREMENT WITH PREFIX
INCB	'B'	INCREMENT WITH BASE
HIGP	'H'	HIGHEST EXISTING WITH PREFIX
EXTRA	'X'	FORMAT-B RECORDS ALLOWED

Returncodes

Symbol	Wert	Bedeutung
OK	X'00'	OK
TRUNC	X'04'	RECORD TRUNCATED
EOF	X'08'	END OF GET/TOC
LMSE	X'0C'	LMS ERROR
PARER	X'14'	PARAMETER ERROR
SEQR	X'18'	SEQUENCE ERROR
INTER	X'1C'	LMS INTERNAL ERROR

Symbole für verschiedene Felder

Symbol	Wert	Bedeutung
		--- SPEICHERFORM ---
FULL	'V'	Voll-Element
DELTA	'D'	Delta-Element
		--- KONVENTIONEN ---
CNONE	'N'	NONE
CSEQ	'S'	STD-SEQUENCE
CMSEQ	'M'	MULTI-SEQUENCE
CTREE	'T'	STD-TREE
		--- PROTECTION INDICATORS ---
PNONE	'N'	NONE
PSTD	'Y'	STD-PROTECTION
PGD	'G'	PROTECTION BY GUARD
		--- SOURCE CODE CONTROL ---
FREE	'F'	FREE
INHLD	'H'	IN HOLD
		--- WRITE-CTRL ---
ACTIV	'A'	ACTIVATE ACTIVE
DEACT	'D'	DEACTIVATE DEACTIVATED
		--- ACCESS-DATE ---
STD	'S'	kein Zugriffsdatum führen
KEEP	'K'	Zugriffsdatum führen
		--- MODIFICATION DATE INDICATOR ---
OLD	'O'	BY-SOURCE
SDAT	'S'	NEW (SYSTEM DATE)

4.3.2 Aufbau des Sekundärsatzes (Satzart 163)

Durch Schreiben von Sekundärsätzen werden Einträge in das Sekundärverzeichnis einer PLAM-Bibliothek vorgenommen. Die Sekundärsätze müssen folgenden Aufbau haben:

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PM163LL	Satzlänge	2	0
	reserviert	1	0
PM163RT	Satzart	1	163
PMSECNA	Sekundärname	32	32 x X'40'
PMSECAT	Attribut zum Sekundärnamen '0' - CSECT '1' - ENTRY	8	8 x X'40'
PMSFIND	Format Indikator	1	0
PMSNAML	LONG SEC NAME: - 32K-45	variabel	1 x X'40'

4.3.3 Aufbau des Attributsatzes (Satzart 164)

Der Attributsatz hat, wenn er vorhanden ist, immer die Satzart 164. Im Attributsatz sind die Attribute der Datei gespeichert, die ursprünglich die Daten enthielt.

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PMRECLEN	Satzlänge	2	540
	reserviert	1	0
PMRECID	IDENTIFICATION OF PLAM RECORD	1	164
PMVERS	VERSION OF SPECIFIED PLAM RECORD	1	2
PMRECNUM	RECORD DESCRIBES FORMAT OF PLAM RECORD OF RECORD TYPE WITH SPECIFIED NUMBER	1	1
PMFNAME	FILE NAME TAKEN FROM FCB	54	54 x X'40'
PMFTYPE	FCBTYPE (SET/RESET) PMFTYPES=X'C0' : R SAM PMFTYPEI=X'40' : R ISAM PMFTYPEP=X'C0' : S PAM PMVMIN =X'01' * R VALPROP MIN. F. PMVMAX =X'01' * S VALPROP MAX. F.	1	1 x X'00'
PMSHARE	SHARE (SET/RESET) CAT PMSHAREY=X'04' : S YES PMACCESS=X'08' : S ACCESS=READ PMSHCCNO=X'C0' : R NO CONTROL CHAR PMSHCCM =X'40' : S MACHINE CODE CONTROL CHAR PMSHCCA =X'C0' : S ASA CONTROL CHAR	1	1x X'00'
PMSIZE	FILE SIZE	3	3 x X'00'
PMSALL	SECONDARY ALLOCATION	2	2 x X'00'
PMREFC	RECFORM (SET RESET) PMRECFE =X'04' : S FIXED PMRECFV =X'02' : S VARIABLE PMRECFU =X'06' : S UNDEFINED	1	1 x X'00'
PMBLKSIZ	BLKSIZE	2	0
PMRECSIZ	RECSIZE	2	0
PMKEYPOS	KEYPOS	2	0
PMKEYLEN	KEYLEN	1	1 x X'00'
PMPAD	PAD	1	1 x X'00'
PMLOGLN	LOGLEN	1	1 x X'00'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PMVALLN	VALLEN	1	1 x X'00'
PMKEY	DOES KEY EXISTS IN MEMBER? PMKEYY =C'Y' : YES PMKEYN =C'N' : NO	1	1 x X'40'
PMCFID	CFID (unbenutzt)	4	4 x X'00'
PMCTRLI	BLKCTRL-INDICATOR PMCTRLN =X'80' : S BLKCTRL=NO PMCTRLP =X'40' : S BLKCTRL=PAMKEY PMCTRLD =X'20' : S BLKCTRL=DATA PMCTRL0 =X'10' : S BLKCTRL=NULL PMCTRLR =X'F0' : R BLKCTRL NOT SPECIF. PMBCF4K =X'08' : S BLOCK CTRL 4K PMBCF2K =X'04' : S BLOCK CTRL 2K PMCTRLU =X'03' : R BLKCTRL=RESERVED,0!	1	1 x X'00'
PMPERF	IOPERF-INDICATOR PMPFVH =X'03' : S IOPERF=VERY-HIGH PMPFHI =X'02' : S IOPERF=HIGH PMPFST =X'01' : S IOPERF=STD PMPFNS =X'00' : S IOPERF NOT SPECIF.	1	1 x X'00'
PMUSAG	IOUSAGE-INDICATOR PMUSRW =X'03' : S IOUSAGE=RDWRT PMUSWR =X'02' : S IOUSAGE=WRITE PMUSRD =X'01' : S IOUSAGE=READ PMUSNS =X'00' : S IOUSAGE NOT SPECIF.	1	1 x X'00'
PMEDMS3	CATALOG-INDICATOR (IDCEX) PMESPEC =X'08' : S PLAM FILE INDICATOR	1	1 x X'00'
	reserviert (muss 0 sein)	1	1 x X'00'
PMAIXCNT	ALTERNATE INDEX COUNT PMAIXMAX=30: MAX. NR. AIX ENTRIES	2	0
PMFSIZ	FILE SIZE >= 32 GB PMSIZE muss X'FFFFFF' sein	4	4 x X'00'
	reserviert (muss 0 sein)		86 x X'00'

Feld	Bedeutung	Länge in Byte	Inhalt nach der Initialisierung
PMAIXNAM	KEYNAME	8	8 x X'00' *
PMAIXKPO	KEYPOS	2	0 *
PMAIXKLE	KEYLEN	1	0 *
PMAIXIND	INDICATOR PMAIXIDK=X'80' : S DUPKEY=YES : R DUPKEY=NO PMAIX# =12 : LENGTH OF AIX ENTRY		1 x X'00' *

* 30 mal (AIXMAX)

5 COBOL-Schnittstelle

5.1 Der Verbindungsmodul LMSUP1

Zum Aufruf der LMS-Funktionen aus COBOL-Programmen steht der Modul LMSUP1 zur Verfügung. Sein Aufruf erfolgt über:

```
CALL "LMSUP1" USING parameterliste.
```

Für die Einträge in die Parameterliste siehe Beschreibung der jeweiligen Funktion ab [Seite 17](#). Der Modul LMSUP1 wird aus der Bibliothek SYSLNK.LMS.034 zum Hauptprogramm hinzugebunden.

5.2 Generierung der Parameterstrukturen für COBOL

Um die Parameterstrukturen für COBOL generieren zu können, muss zunächst die Bibliothek SYSLIB.LMS.034, in der sich die COPY-Elemente befinden, zugewiesen werden. Die Zuweisung kann mit folgendem Kommando erfolgen:

```
/ADD-FILE-LINK FILE-NAME = $.SYSLIB.LMS.034, LINK-NAME = COBLIB
```

Der Installationsort dieser Bibliothek ist über IMON frei wählbar.

Mit Hilfe der Built-In-Funktion INSTALLATION-PATH kann der Installationsort von SYSLIB.LMS.034 ermittelt und in eine S-Variable gespeichert werden:

```
/SET-VARIABLE LIBRARY-NAME =INSTALLATION-PATH           -  
                (LOGICAL-ID = 'SYSLIB'                 -  
                ,INSTALLATION-UNIT = 'LMS'             -  
                ,VERSION = '3.4'                       -  
                ,DEFAULT-PATH-NAME = '$.SYSLIB.LMS.034')
```

Anschließend kann die Bibliothek folgendermaßen zugewiesen werden:

```
/ADD-FILE-LINK FILE-NAME=&(LIBRARY-NAME), LINK-NAME=COBLIB
```

Für die Parameterstrukturen TID und ER stehen keine COPY-Elemente zur Verfügung.

LMSCOBCB

LMSCOBCB generiert den Funktionskontrollblock.

COPY LMSCOBCB.

Auflösung von LMSCOBCB

```

01  LMSUP-SCB.
02  SCBVERSION PIC X(2)      VALUE "04".
02  FUNC       PIC X(1)      VALUE LMSUP-UPINIT.
02  SUBCODE    PIC X(1)      VALUE SPACE.
02  ACC       PIC S9(9) COMP VALUE -1.
02  RETURNCODE PIC X(1)      VALUE LOW-VALUE.
   88 LMSRET-OK          VALUE LMSUP-OK.
   88 LMSRET-TRUNC      VALUE LMSUP-TRUNC.
   88 LMSRET-EOF        VALUE LMSUP-EOF.
   88 LMSRET-LMSERR     VALUE LMSUP-LMSERR.
   88 LMSRET-PARERR     VALUE LMSUP-PARERR.
   88 LMSRET-SEQERR     VALUE LMSUP-SEQERR.
   88 LMSRET-INTERR     VALUE LMSUP-INTERR.
02  FILLER    PIC X(1)      VALUE LOW-VALUE.
02  LMS-MSG   PIC 9(4) COMP  VALUE 0.
02  DMS-MSG   PIC 9(4) COMP  VALUE 0.
02  PLAM-MSG  PIC 9(4) COMP  VALUE 0.
02  LMSVERSION PIC X(12)    VALUE SPACES.
02  FILLER    PIC 9(5) COMP  VALUE 0.

***
***** LMS  PARAMETERS *****
***

02  DESTROY   PIC X(1)      VALUE SPACE.
02  FCB      PIC X(1)      VALUE SPACE.
02  RKEY     PIC X(1)      VALUE SPACE.
02  OVERWRITE PIC X(1)      VALUE SPACE.
02  COLUMN-P PIC 9(4) COMP  VALUE 0.
02  LINE-P   PIC 9(4) COMP  VALUE 0.
02  PROT-IND PIC X(1)      VALUE SPACE.
02  ATTR-IND PIC X(1)      VALUE SPACE.
02  INFO     PIC X(1)      VALUE SPACE.
02  FILLER   PIC X(5)      VALUE SPACES.

***** END OF COPY ELEMENT LMSCOBCB *****

```

LMSCOBEA

LMSCOBEA generiert die Elementattribute.

```
COPY LMSCOBEA.
```

Auflösung von LMSCOBEA

```
01 LMSUP-EA.
   02 USER-DATE      PIC X(14)      VALUE SPACES.
   02 USER-TIME      PIC X(8)       VALUE SPACES.
   02 CCS-NAME       PIC X(8)       VALUE SPACES.
   02 HOLD-STATE     PIC X(1)       VALUE SPACES.
   *                 HOLD FLAG:      '- ' : FREE
   *                                     'H' : INHOLD
   *                                     ' ' : UNCHANGE
   02 FILLER         PIC X(8)       VALUE SPACES.
   02 MOD-DATE-IND   PIC X(1)       VALUE "0".
   *                 MODIFICATION DATE INDICATOR: 'O' : OLD
   *                                               'S' : SYSTEM DATE
   02 FILLER         PIC X(56)     VALUE SPACES.
```

```
***** END OF COPY ELEMENT LMSCOBEA *****
```

LMSCOBED

LMSCOBED generiert die Elementbeschreibung.

```
COPY LMSCOBED.
```

Auflösung von LMSCOBED

```
01 LMSUP-ED.
   02 LMSUP-ED-ELEM.
   03 TYP            PIC X(8)       VALUE SPACES.
   03 NAME          PIC X(64)     VALUE SPACES.
   03 VERSION       PIC X(24)     VALUE SPACES.
   *
   02 STORE-FORM    PIC X         VALUE "V".
   02 USER-DATE     PIC X(14)     VALUE SPACES.
   02 USER-TIME     PIC X(8)      VALUE SPACES.
```

```
***** END OF COPY ELEMENT LMSCOBED *****
```

LMSCOBEI

LMSCOBEI generiert die Elementinformation.

COPY LMSCOBEI.

Auflösung von LMSCOBEI

```

01 LMSUP-EI.
  02 LMSUP-EI-ED.
    03 LMSUP-EI-ED-ELEM.
      04 TYP          PIC X(8)      VALUE SPACES.
      04 NAME        PIC X(64)     VALUE SPACES.
      04 VERSION     PIC X(24)     VALUE SPACES.
    *
      03 STORE-FORM  PIC X(1)      VALUE SPACES.
      03 USER-DATE   PIC X(14)     VALUE SPACES.
      03 USER-TIME   PIC X(8)      VALUE SPACES.
    *
      02 CREATION-DATE PIC X(14)   VALUE SPACES.
      02 CREATION-TIME PIC X(8)    VALUE SPACES.
      02 MODIFI-DATE  PIC X(14)    VALUE SPACES.
      02 MODIFI-TIME  PIC X(8)     VALUE SPACES.
      02 SEC-NAME     PIC X(32)    VALUE SPACES.
      02 SEC-ATTRIBUTE PIC X(8)    VALUE SPACES.
      02 FILLER       PIC X(5)     VALUE SPACES.

***** PROTECTION ATTRIBUTES *****
      02 P-TIND-READ  PIC X(1)     VALUE SPACES.
      02 P-READ-OWN   PIC X(1)     VALUE SPACES.
      02 P-READ-GRP   PIC X(1)     VALUE SPACES.
      02 P-READ-OTH   PIC X(1)     VALUE SPACES.
      02 P-READ-PIND  PIC X(1)     VALUE SPACES.
      02 FILLER       PIC S9(9) COMP VALUE 0.

      02 P-TIND-WRIT  PIC X(1)     VALUE SPACES.
      02 P-WRIT-OWN   PIC X(1)     VALUE SPACES.
      02 P-WRIT-GRP   PIC X(1)     VALUE SPACES.
      02 P-WRIT-OTH   PIC X(1)     VALUE SPACES.
      02 P-WRIT-PIND  PIC X(1)     VALUE SPACES.
      02 FILLER       PIC S9(9) COMP VALUE 0.

      02 P-TIND-EXEC  PIC X(1)     VALUE SPACES.
      02 P-EXEC-OWN   PIC X(1)     VALUE SPACES.
      02 P-EXEC-GRP   PIC X(1)     VALUE SPACES.
      02 P-EXEC-OTH   PIC X(1)     VALUE SPACES.
      02 P-EXEC-PIND  PIC X(1)     VALUE SPACES.
      02 FILLER       PIC S9(9) COMP VALUE 0.

```

```

02 P-GUARD-READ PIC X(18) VALUE SPACES.
02 P-GUARD-WRIT PIC X(18) VALUE SPACES.
02 P-GUARD-EXEC PIC X(18) VALUE SPACES.
02 CCS-NAME PIC X(8) VALUE SPACES.

02 P-TIND-HOLD PIC X(1) VALUE SPACES.
02 P-HOLD-OWN PIC X(1) VALUE SPACES.
02 P-HOLD-GRP PIC X(1) VALUE SPACES.
02 P-HOLD-OTH PIC X(1) VALUE SPACES.
02 P-HOLD-PIND PIC X(1) VALUE SPACES.
02 FILLER PIC S9(9) COMP VALUE 0.
02 P-GUARD-HOLD PIC X(18) VALUE SPACES.

02 HOLD-STATE PIC X(1) VALUE SPACES.
* HOLD FLAG: ' ' : FREE
* 'H' : INHOLD
02 HOLDER PIC X(8) VALUE SPACES.
02 ACCESS-DATE PIC X(14) VALUE SPACES.
02 ACCESS-TIME PIC X(8) VALUE SPACES.
02 FILLER PIC X(1) VALUE SPACES.
02 ELEMENT-SIZE PIC S9(9) COMP VALUE 0.
02 FILLER PIC X(40) VALUE SPACES.

***** END OF COPY ELEMENT LMSCOBEI *****

```

LMSCOBEM

LMSCOBEM generiert die Elementmaske.

COPY LMSCOBEM.

Auflösung von LMSCOBEM

```

01 LMSUP-EM.
  02 TYP          PIC X(20)      VALUE SPACES.
  02 NAME        PIC X(132)     VALUE SPACES.
  02 VERSION     PIC X(52)      VALUE SPACES.
  02 STORE-FORM  PIC X(6)       VALUE SPACES.
  02 USER-DATE   PIC X(32)     VALUE SPACES.
  02 USER-TIME   PIC X(20)     VALUE SPACES.
  02 CREATION-DATE PIC X(32)   VALUE SPACES.
  02 CREATION-TIME PIC X(20)   VALUE SPACES.
  02 MODIFI-DATE PIC X(32)     VALUE SPACES.
  02 MODIFI-TIME PIC X(20)     VALUE SPACES.
  02 SEC-NAME    PIC X(68)     VALUE SPACES.
  02 SEC-ATTRIBUTE PIC X(20)   VALUE SPACES.
  02 FILLER      PIC X(14)     VALUE SPACES.

*****      PROTECTION ATTRIBUTES      *****
  02 P-TIND-READ PIC X(1)      VALUE SPACES.
  02 P-READ-OWN  PIC X(1)      VALUE SPACES.
  02 P-READ-GRP  PIC X(1)      VALUE SPACES.
  02 P-READ-OTH  PIC X(1)      VALUE SPACES.
  02 P-READ-PIND PIC X(1)      VALUE SPACES.
  02 FILLER      PIC S9(9) COMP VALUE 0.

  02 P-TIND-WRIT PIC X(1)      VALUE SPACES.
  02 P-WRIT-OWN  PIC X(1)      VALUE SPACES.
  02 P-WRIT-GRP  PIC X(1)      VALUE SPACES.
  02 P-WRIT-OTH  PIC X(1)      VALUE SPACES.
  02 P-WRIT-PIND PIC X(1)      VALUE SPACES.
  02 FILLER      PIC S9(9) COMP VALUE 0.

  02 P-TIND-EXEC PIC X(1)      VALUE SPACES.
  02 P-EXEC-OWN  PIC X(1)      VALUE SPACES.
  02 P-EXEC-GRP  PIC X(1)      VALUE SPACES.
  02 P-EXEC-OTH  PIC X(1)      VALUE SPACES.
  02 P-EXEC-PIND PIC X(1)      VALUE SPACES.
  02 FILLER      PIC S9(9) COMP VALUE 0.

  02 P-GUARD-READ PIC X(40)    VALUE SPACES.
  02 P-GUARD-WRIT PIC X(40)    VALUE SPACES.
  02 P-GUARD-EXEC PIC X(40)    VALUE SPACES.
  02 CCS-NAME     PIC X(20)     VALUE SPACES.

```

```
02 P-TIND-HOLD PIC X(1) VALUE SPACES.
02 P-HOLD-OWN PIC X(1) VALUE SPACES.
02 P-HOLD-GRP PIC X(1) VALUE SPACES.
02 P-HOLD-OTH PIC X(1) VALUE SPACES.
02 P-HOLD-PIND PIC X(1) VALUE SPACES.
02 FILLER PIC S9(9) COMP VALUE 0.
02 P-GUARD-HOLD PIC X(40) VALUE SPACES.

02 HOLD-STATE PIC X(1) VALUE SPACES.
* HOLD FLAG: ' ' : FREE
* 'H' : INHOLD
* ' ' : ANY

02 HOLDER PIC X(20) VALUE SPACES.
02 ACCESS-DATE PIC X(32) VALUE SPACES.
02 ACCESS-TIME PIC X(20) VALUE SPACES.
02 FILLER PIC X(3) VALUE SPACES.
02 E-SIZE-MIN PIC S9(9) COMP VALUE 0.
02 E-SIZE-MAX PIC S9(9) COMP VALUE -1.
02 FILLER PIC X(64) VALUE SPACES.

***** END OF COPY ELEMENT LMSCOBEM *****
```

LMSCOBFDF

LMSCOBFDF generiert die Dateibeschreibung.

```
COPY LMSCOBFDF.
```

Auflösung von LMSCOBFDF

```

01 LMSUP-FD.
   02 PASSWORD      PIC S9(9) COMP VALUE 0.
   02 PASSWORD-X    REDEFINES PASSWORD  PIC X(4).
   02 LINK          PIC X(8)          VALUE SPACES.
   02 NAME          PIC X(54)        VALUE SPACES.
***** END OF COPY ELEMENT  LMSCOBFDF *****

```

LMSCOBLA

LMSCOBLA generiert die Bibliotheksattribute.

```
COPY LMSCOBLA.
```

Auflösung von LMSCOBLA

```

01 LMSUP-LA.
   02 P-TIND-ADMI   PIC X(1)          VALUE SPACES.
   02 P-ADMI-OWN   PIC X(1)          VALUE SPACES.
   02 P-ADMI-GRP   PIC X(1)          VALUE SPACES.
   02 P-ADMI-OTH   PIC X(1)          VALUE SPACES.
   02 P-ADMI-PIND  PIC X(1)          VALUE SPACES.
   02 P-ADMI-PSWD  PIC S9(9) COMP VALUE 0.
   02 P-ADMI-PSWD-X REDEFINES P-ADMI-PSWD PIC X(4).
   02 P-GUARD-ADMI PIC X(18)         VALUE SPACES.
   02 STORE-FORM   PIC X(1)          VALUE SPACES.
*      STORAGE FORM FOR LIBRARY 'S' : STD ( FULL OR DELTA )
*                                          'V' : FULL ELEMENT
*                                          'D' : DELTA ELEMENT
   02 WRITE-CTRL   PIC X(1)          VALUE SPACES.
*      WRITE-CONTROL FOR LIBRARY 'A' : ACTIVATE
*                                          'D' : DEAVTIVATE
   02 ACCESS-DATE  PIC X(1)          VALUE SPACES.
   02 FILLER       PIC X(34)         VALUE SPACES.
***** END OF COPY ELEMENT  LMSCOBLA *****

```

LMSCOBLD

LMSCOBLD generiert die Bibliotheksbeschreibung.

```
COPY LMSCOBLD.
```

Auflösung von LMSCOBLD

```
001 LMSUP-LD.  
    02 PASSWORD      PIC S9(9) COMP VALUE 0.  
    02 PASSWORD-X    REDEFINES PASSWORD  PIC X(4).  
    02 LINK           PIC X(8)          VALUE SPACES.  
    02 FILLER        PIC S9(9) COMP VALUE 0.  
    02 FILLER        PIC S9(9) COMP VALUE 0.  
    02 MAX-NAME-LEN  PIC 9(4) COMP  VALUE 54.  
    02 NAME          PIC X(54)        VALUE SPACES.  
***** END OF COPY ELEMENT  LMSCOBLD *****
```

Hinweis

Bei der Kennwort-Vergabe ist darauf zu achten, dass das Kennwort nicht kürzer als 4 Byte ist. Der Grund dafür ist, dass COBOL bei einem MOVE die Daten von links nach rechts in das Empfangsfeld überträgt und die dabei verbleibenden Stellen (rechts) mit Leerzeichen auffüllt. Leerzeichen stellen aber im BS2000 eine gültige Kennwort-Kombination dar.

LMSCOBLI

LMSCOBLI generiert die Bibliotheksinformation.

COPY LMSCOBLI.

Auflösung von LMSCOBLI

```

1  LMSUP-LI.
   02 P-TIND-ADMI  PIC X(1)      VALUE SPACES.
   02 P-ADMI-OWN  PIC X(1)      VALUE SPACES.
   02 P-ADMI-GRP  PIC X(1)      VALUE SPACES.
   02 P-ADMI-OTH  PIC X(1)      VALUE SPACES.
   02 P-ADMI-PIND PIC X(1)      VALUE SPACES.
   02 FILLER      PIC S9(9) COMP VALUE 0.
   02 P-GUARD-ADMI PIC X(18)     VALUE SPACES.

   02 STORE-FORM  PIC X(1)      VALUE SPACES.
   *             STORAGE FORM FOR LIBRARY 'S' : STD ( FULL OR DELTA )
   *
   *             'V' : FULL ELEMENT
   *             'D' : DELTA ELEMENT
   02 WRITE-CTRL  PIC X(1)      VALUE SPACES.
   *             WRITE-CONTROL FOR LIBRARY 'A' : ACTIVE
   *             'D' : DEAVTIVATED
   02 ACCESS-DATE PIC X(1)      VALUE SPACES.
   02 FILLER      PIC X(24)     VALUE SPACES.
   02 LIB-FORM    PIC X(1)      VALUE SPACES.
   *             LIBRARY FORMAT (NK2/NK4) '2' : NK2 LIBRARY FORMAT
   *             '4' : NK4 LIBRARY FORMAT
   02 UPAM-PROT   PIC X(1)      VALUE SPACES.
   *             UPAM PROTECTED (YES/NO) 'Y' : LIB IS UPAM PROTECTED
   *             'N' : LIB IS NOT UPAM PROT.
   02 FILE-SIZE   PIC S9(9) COMP VALUE 0.
   02 FREE-SIZE   PIC S9(9) COMP VALUE 0.

   02 P-TIND-READ PIC X(1)      VALUE SPACES.
   02 P-READ-OWN  PIC X(1)      VALUE SPACES.
   02 P-READ-GRP  PIC X(1)      VALUE SPACES.
   02 P-READ-OTH  PIC X(1)      VALUE SPACES.
   02 P-READ-PIND PIC X(1)      VALUE SPACES.
   02 FILLER      PIC S9(9) COMP VALUE 0.

   02 P-TIND-WRIT PIC X(1)      VALUE SPACES.
   02 P-WRIT-OWN  PIC X(1)      VALUE SPACES.
   02 P-WRIT-GRP  PIC X(1)      VALUE SPACES.
   02 P-WRIT-OTH  PIC X(1)      VALUE SPACES.
   02 P-WRIT-PIND PIC X(1)      VALUE SPACES.
   02 FILLER      PIC S9(9) COMP VALUE 0.

```

```
02 P-TIND-EXEC PIC X(1) VALUE SPACES.
02 P-EXEC-OWN PIC X(1) VALUE SPACES.
02 P-EXEC-GRP PIC X(1) VALUE SPACES.
02 P-EXEC-OTH PIC X(1) VALUE SPACES.
02 P-EXEC-PIND PIC X(1) VALUE SPACES.
02 FILLER PIC S9(9) COMP VALUE 0.

02 P-GUARD-READ PIC X(18) VALUE SPACES.
02 P-GUARD-WRIT PIC X(18) VALUE SPACES.
02 P-GUARD-EXEC PIC X(18) VALUE SPACES.

02 P-TIND-HOLD PIC X(1) VALUE SPACES.
02 P-HOLD-OWN PIC X(1) VALUE SPACES.
02 P-HOLD-GRP PIC X(1) VALUE SPACES.
02 P-HOLD-OTH PIC X(1) VALUE SPACES.
02 P-HOLD-PIND PIC X(1) VALUE SPACES.
02 FILLER PIC S9(9) COMP VALUE 0.
02 P-GUARD-HOLD PIC X(18) VALUE SPACES.
02 FILLER PIC X(68) VALUE SPACES.
```

```
***** END OF COPY ELEMENT LMSCOBLI *****
```

LMSCOBPA

LMSCOBPA generiert die Schutzattribute.

COPY LMSCOBPA.

Auflösung von LMSCOBPA

```

01  LMSUP-PA.
    02  P-TIND-READ  PIC X(1)      VALUE SPACES.
    02  P-READ-OWN  PIC X(1)      VALUE SPACES.
    02  P-READ-GRP  PIC X(1)      VALUE SPACES.
    02  P-READ-OTH  PIC X(1)      VALUE SPACES.
    02  P-READ-PIND PIC X(1)      VALUE SPACES.
    02  P-READ-PSWD PIC S9(9) COMP VALUE 0.
        02  P-READ-PSWD-X REDEFINES P-READ-PSWD PIC X(4).

    02  P-TIND-WRIT  PIC X(1)      VALUE SPACES.
    02  P-WRIT-OWN  PIC X(1)      VALUE SPACES.
    02  P-WRIT-GRP  PIC X(1)      VALUE SPACES.
    02  P-WRIT-OTH  PIC X(1)      VALUE SPACES.
    02  P-WRIT-PIND PIC X(1)      VALUE SPACES.
    02  P-WRIT-PSWD PIC S9(9) COMP VALUE 0.
        02  P-WRIT-PSWD-X REDEFINES P-WRIT-PSWD PIC X(4).

    02  P-TIND-EXEC  PIC X(1)      VALUE SPACES.
    02  P-EXEC-OWN  PIC X(1)      VALUE SPACES.
    02  P-EXEC-GRP  PIC X(1)      VALUE SPACES.
    02  P-EXEC-OTH  PIC X(1)      VALUE SPACES.
    02  P-EXEC-PIND PIC X(1)      VALUE SPACES.
    02  P-EXEC-PSWD PIC S9(9) COMP VALUE 0.
        02  P-EXEC-PSWD-X REDEFINES P-EXEC-PSWD PIC X(4).

    02  P-GUARD-READ PIC X(18)     VALUE SPACES.
    02  P-GUARD-WRIT PIC X(18)     VALUE SPACES.
    02  P-GUARD-EXEC PIC X(18)     VALUE SPACES.

    02  P-TIND-HOLD  PIC X(1)      VALUE SPACES.
    02  P-HOLD-OWN  PIC X(1)      VALUE SPACES.
    02  P-HOLD-GRP  PIC X(1)      VALUE SPACES.
    02  P-HOLD-OTH  PIC X(1)      VALUE SPACES.
    02  P-HOLD-PIND PIC X(1)      VALUE SPACES.
    02  P-HOLD-PSWD PIC S9(9) COMP VALUE 0.
        02  P-HOLD-PSWD-X REDEFINES P-HOLD-PSWD PIC X(4).
    02  P-GUARD-HOLD PIC X(18)     VALUE SPACES.
    02  FILLER      PIC X(84)     VALUE SPACES.

***** END OF COPY ELEMENT  LMSCOBPA *****

```

LMSCOBRD

LMSCOBRD generiert die Satzbeschreibung.

COPY LMSCOBRD.

Auflösung von LMSCOBRD

```
01  LMSUP-RD.
    02  REC-ACC-ID      PIC S9(9) COMP VALUE -1.
    02  BUFFER-LEN     PIC 9(9) COMP  VALUE 0.
    02  RECORD-LEN     PIC 9(9) COMP  VALUE 0.
    02  FILLER         PIC X(3)      VALUE LOW-VALUE.
    02  RECORD-TYPE    PIC X          VALUE LMSUP-ONE.
    02  RECORD-NR      PIC 9(9) COMP  VALUE 0.
    02  FILLER         PIC S9(9) COMP VALUE 0.
    02  FILLER         PIC S9(9) COMP VALUE 0.
***** END OF COPY ELEMENT  LMSCOBRD *****
```

LMSCOBTA

LMSCOBTA generiert die Typattribute.

COPY LMSCOBTA.

Auflösung von LMSCOBTA

```

01  LMSUP-TA.
    02  CONVENTION      PIC X(1)      VALUE SPACES.
    02  FILLER          PIC X(3)      VALUE SPACES.
    02  V-EXAMPLE       PIC X(24)     VALUE SPACES.

    02  P-TIND-ADMI     PIC X(1)      VALUE SPACES.
    02  P-ADMI-OWN      PIC X(1)      VALUE SPACES.
    02  P-ADMI-GRP      PIC X(1)      VALUE SPACES.
    02  P-ADMI-OTH      PIC X(1)      VALUE SPACES.
    02  P-ADMI-PIND     PIC X(1)      VALUE SPACES.
    02  P-ADMI-PSWD     PIC S9(9) COMP VALUE 0.
    02  P-ADMI-PSWD-X REDEFINES P-ADMI-PSWD PIC X(4).
    02  P-GUARD-ADMI    PIC X(18)     VALUE SPACES.

    02  STORE-FORM      PIC X(1)      VALUE SPACES.
    *      STORAGE FORM FOR TYPE      'N' : NONE
    *                                          'S' : STD ( FULL OR DELTA )
    *                                          'V' : FULL ELEMENT
    *                                          'D' : DELTA ELEMENT
    02  WRITE-CTRL      PIC X(1)      VALUE SPACES.
    *      WRITE-CONTROL FOR TYPE      'A' : ACTIVATE
    *                                          'D' : DEAVTIVATE
    02  SUPER-TYPE      PIC X(8)      VALUE SPACES.
    02  FILLER          PIC X(47)     VALUE SPACES.

***** END OF COPY ELEMENT  LMSCOBTA *****

```

LMSCOBTD

LMSCOBTD generiert die Typbeschreibung.

```
COPY LMSCOBTD.
```

Auflösung von LMSCOBTD

```

01 LMSUP-TD.
   02 TYP          PIC X(8)      VALUE SPACES.
   02 FILLER       PIC X(8)      VALUE SPACES.

***** END OF COPY ELEMENT LMSCOBTD *****

```

LMSCOBTI

LMSCOBTI generiert die Typinformation.

```
COPY LMSCOBTI.
```

Auflösung von LMSCOBTI

```

01 LMSUP-TI.
   02 TYP          PIC X(8)      VALUE SPACES.
   02 FILLER       PIC X(8)      VALUE SPACES.
   02 CONVENTION   PIC X(1)      VALUE SPACES.
   02 FILLER       PIC X(3)      VALUE SPACES.
   02 V-EXAMPLE    PIC X(24)     VALUE SPACES.

   02 P-TIND-ADMI  PIC X(1)      VALUE SPACES.
   02 P-ADMI-OWN   PIC X(1)      VALUE SPACES.
   02 P-ADMI-GRP   PIC X(1)      VALUE SPACES.
   02 P-ADMI-OTH   PIC X(1)      VALUE SPACES.
   02 P-ADMI-PIND  PIC X(1)      VALUE SPACES.
   02 FILLER       PIC S9(9) COMP VALUE 0.
   02 P-GUARD-ADMI PIC X(18)     VALUE SPACES.

   02 STORE-FORM   PIC X(1)      VALUE SPACES.
*      STORAGE FORM FOR TYPE      'N' : NONE
*                                     'S' : STD ( FULL OR DELTA )
*                                     'V' : FULL ELEMENT
*                                     'D' : DELTA ELEMENT

   02 WRITE-CTRL   PIC X(1)      VALUE SPACES.
*      WRITE-CONTROL FOR TYPE      'A' : ACTIVE
*                                     'D' : DEACTIVATED

   02 SUPER-TYPE   PIC X(8)      VALUE SPACES.
   02 BASIS-TYPE   PIC X(8)      VALUE SPACES.

```

```

02 FILLER          PIC X(39)      VALUE SPACES.

*****          PROTECTION ATTRIBUTES          *****
02 P-TIND-READ    PIC X(1)        VALUE SPACES.
02 P-READ-OWN     PIC X(1)        VALUE SPACES.
02 P-READ-GRP     PIC X(1)        VALUE SPACES.
02 P-READ-OTH     PIC X(1)        VALUE SPACES.
02 P-READ-PIND    PIC X(1)        VALUE SPACES.
02 FILLER         PIC S9(9) COMP  VALUE 0.

02 P-TIND-WRIT    PIC X(1)        VALUE SPACES.
02 P-WRIT-OWN     PIC X(1)        VALUE SPACES.
02 P-WRIT-GRP     PIC X(1)        VALUE SPACES.
02 P-WRIT-OTH     PIC X(1)        VALUE SPACES.
02 P-WRIT-PIND    PIC X(1)        VALUE SPACES.
02 FILLER         PIC S9(9) COMP  VALUE 0.

02 P-TIND-EXEC    PIC X(1)        VALUE SPACES.
02 P-EXEC-OWN     PIC X(1)        VALUE SPACES.
02 P-EXEC-GRP     PIC X(1)        VALUE SPACES.
02 P-EXEC-OTH     PIC X(1)        VALUE SPACES.
02 P-EXEC-PIND    PIC X(1)        VALUE SPACES.
02 FILLER         PIC S9(9) COMP  VALUE 0.

02 P-GUARD-READ   PIC X(18)       VALUE SPACES.
02 P-GUARD-WRIT   PIC X(18)       VALUE SPACES.
02 P-GUARD-EXEC   PIC X(18)       VALUE SPACES.

02 P-TIND-HOLD    PIC X(1)        VALUE SPACES.
02 P-HOLD-OWN     PIC X(1)        VALUE SPACES.
02 P-HOLD-GRP     PIC X(1)        VALUE SPACES.
02 P-HOLD-OTH     PIC X(1)        VALUE SPACES.
02 P-HOLD-PIND    PIC X(1)        VALUE SPACES.
02 FILLER         PIC S9(9) COMP  VALUE 0.
02 P-GUARD-HOLD   PIC X(18)       VALUE SPACES.
02 FILLER         PIC X(52)       VALUE SPACES.

*****          END OF COPY ELEMENT          LMSCOBTI          *****

```

5.3 Programmierhilfen

Symbolische Namen LMSCOBEQ

LMSCOBEQ muss in der SYMBOLIC CHARACTERS Klausel des SPECIAL-NAMES Paragraphen angegeben werden. Es vereinbart symbolische Namen, die als Programmierhilfe für Verarbeitungsoperandenwerte, Funktionscodes, Subcodes, Returncodes und Speicherform der Elemente dienen.

Beim Einsatz von LMSCOBEQ ist zu beachten, dass der SPECIAL-NAMES Paragraph ein Eingabesatz ist. Sollte LMSCOBEQ der einzige Eintrag im SPECIAL-NAMES Paragraph sein, muss im Programm der abschließende Punkt explizit gesetzt werden.

COPY LMSCOBEQ.

Auflösung von LMSCOBEQ

```

***** LMSUP PARAMETER VALUES *****
      LMSUP-YES      IS 233
      LMSUP-NO       IS 214
      LMSUP-NONE     IS 214
      LMSUP-ANY      IS 65
      LMSUP-UNCHANGE IS 65
      LMSUP-SAME     IS 213
      LMSUP-STD      IS 227

***** FOR: CBFEB *****
      LMSUP-ISAM     IS 202
      LMSUP-SAM      IS 217
      LMSUP-CATA     IS 196

***** FOR: CBOV *****
      LMSUP-EXT      IS 198
      LMSUP-ONLY     IS 215
      LMSUP-NAME     IS 194

***** FOR: INFO *****
      LMSUP-TXT      IS 2
      LMSUP-COM      IS 3

***** LMSUP FUNCTION VALUES *****
      LMSUP-UPINIT   IS 02
      LMSUP-UPEND    IS 03
      LMSUP-TOCPRIM  IS 04
      LMSUP-TOCSEC   IS 05
      LMSUP-TOC      IS 06
      LMSUP-REN      IS 07

```

LMSUP-DEL	IS	08
LMSUP-ADD	IS	09
LMSUP-SEL	IS	10
LMSUP-COPY	IS	11
LMSUP-COPSTRUC	IS	12
LMSUP-LOCK	IS	13
LMSUP-UNLOCK	IS	14
LMSUP-OPEN-GET	IS	15
LMSUP-OPEN-PUT	IS	16
LMSUP-OPEN-UPD	IS	17
LMSUP-GET	IS	18
LMSUP-PUT	IS	19
LMSUP-CLOSE	IS	20
LMSUP-LIST	IS	21
LMSUP-MOD-EL-PROT	IS	22
LMSUP-MOD-LIB-A	IS	23
LMSUP-SHOW-LIB-A	IS	24
LMSUP-MOD-TYP-A	IS	25
LMSUP-SHOW-TYP-A	IS	26
LMSUP-MOD-EL-A	IS	27
LMSUP-COPY-LIB	IS	28
LMSUP-CLOSE-LIB	IS	29
LMSUP-PROVIDE	IS	30
LMSUP-RETURN	IS	31
LMSUP-GET-SYS-EL	IS	32
LMSUP-REORG-LIB	IS	33

***** LMSUP SUBCODE VALUES

LMSUP-UNUSED	IS	65
LMSUP-SHORT	IS	227
LMSUP-LONG	IS	212
LMSUP-DIR	IS	197
LMSUP-SEQ	IS	227
LMSUP-WRITE	IS	231
LMSUP-RESET	IS	218
LMSUP-SYM	IS	227
LMSUP-HEX	IS	201
LMSUP-INC-PRE	IS	216
LMSUP-INC-BASE	IS	195
LMSUP-HIGH-PRE	IS	201
LMSUP-EXTRA	IS	232

***** LMSUP RETURN CODE VALUES

LMSUP-OK	IS	1
LMSUP-TRUNC	IS	5
LMSUP-EOF	IS	9
LMSUP-LMSERR	IS	13
LMSUP-PARERR	IS	21
LMSUP-SEQERR	IS	25
LMSUP-INTERR	IS	29

```
***** STORAGE FORM VALUES *****
      LMSUP-FULL   IS  230
      LMSUP-DELTA IS  197

***** CONVENTIONS *****
      LMSUP-CNONE IS  214
      LMSUP-CSEQ  IS  227
      LMSUP-CMSEQ IS  213
      LMSUP-CTREE IS  228

***** PROTECTION INDICATORS *****
      LMSUP-PNONE IS  214
      LMSUP-PSTD  IS  233
      LMSUP-PGUARD IS  200

***** SOURCE CODE CONTROL *****
      LMSUP-FREE  IS  97
      LMSUP-INHOLD IS  201

***** WRITE CONTROL *****
      LMSUP-ACTIV IS  194
      LMSUP-DEACTIV IS  197

***** ACCESS DATE *****
      LMSUP-KEEP  IS  211

***** MODIFICATION DATE INDICATOR *****
      LMSUP-OLD   IS  215
      LMSUP-SYS-DATE IS  227

***** OTHER VALUES *****
      LMSUP-ONE   IS  2
      LMSUP-TWO   IS  3

***** END OF COPY ELEMENT LMSCOBEQ *****
```

Aufbau eines Satzes der Satzart 163

Nachfolgend ist der Satz LMSCO163 als Beispiel für einen Satz der Satzart 163 beschrieben (siehe [Abschnitt „Aufbau des Sekundärsatzes \(Satzart 163\)“ auf Seite 141](#)).

```
COPY LMSCO163.
```

Auflösung von LMSCO163

```
01 LMS-REC163.
  02 LEN          PIC 9(4) COMP VALUE 0.
  02 FILLER       PIC X          VALUE LOW-VALUE.
  02 REC-TYPE     PIC X          VALUE SA-163.
  02 SEC-NAME     PIC X(32)     VALUE SPACES.
  02 SEC-ATTRIBUTE PIC X(8)     VALUE SPACES.
*               '0'           ' : - CSECT
*               '1'           ' : - ENTRY
  02 SEC-ATTR-BIT REDEFINES SEC-ATTRIBUTE PIC X.
  02 SEC-FORMAT-IND PIC X      VALUE LOW-VALUE.
  02 SEC-NAME-LONG.
  03 SEC-NAME-BYTE1 PIC X      VALUE SPACE.
***
***** END OF COPY ELEMENT LMSCO163 *****
```

Aufbau eines Satzes der Satzart 164

Nachfolgend ist der Satz LMSCO164 als Beispiel für einen Satz der Satzart 164 beschrieben (siehe [Abschnitt „Aufbau des Attributsatzes \(Satzart 164\)“](#) auf Seite 142).

COPY LMSCO164.

Auflösung von LMSCO164

```

01  LMS-REC164.
    02  LEN                PIC 9(4) COMP  VALUE 540.
    02  FILLER             PIC X          VALUE LOW-VALUE.
    02  REC-TYPE          PIC X          VALUE SA-164.
    02  VERSION           PIC X          VALUE LMSUP-TWO.
    02  REC-NUMBER       PIC X          VALUE LMSUP-ONE.
    02  FILENAME         PIC X(54)      VALUE SPACES.
    02  FCBTYPE-FIELD   PIC X          VALUE LOW-VALUE.
*      FCBTYPE (SET/RESET)
*      X'CO'              R SAM
*      X'40'              S ISAM
*      X'CO'              S PAM
*      X'01'              R VALPROP MIN. FUNCT.
*      X'01'              S VALPROP MAX. FUNCT.
    02  SHARE-FIELD     PIC X          VALUE LOW-VALUE.
*      SHARE (SET,RESET)
*      X'04'              S YES
*      X'08'              S ACCESS=READ
*      X'CO'              R NO CONTROL CHAR
*      X'40'              S MACHINE CODE CONTROL CHAR
*      X'CO'              S ASA CONTROL CHAR
    02  FILESIZE         PIC X(3)      VALUE LOW-VALUE.
*      FILESIZE < 32GB
    02  SEC-ALLOC        PIC 9(4) COMP  VALUE 0.
    02  RECFORM-FIELD   PIC X          VALUE LOW-VALUE.
*      RECFORM (SET,RESET)
*      X'04'              S FIXED
*      X'02'              S VARIABLE
*      X'06'              S UNDEFINED
    02  BLKSIZE          PIC 9(4) COMP  VALUE 0.
    02  RECSIZE          PIC 9(4) COMP  VALUE 0.
    02  KEYPOS           PIC 9(4) COMP  VALUE 0.
    02  KEYLEN           PIC X          VALUE LOW-VALUE.
    02  PAD              PIC X          VALUE LOW-VALUE.
    02  LOGLEN           PIC X          VALUE LOW-VALUE.
    02  VALLLEN          PIC X          VALUE LOW-VALUE.
    02  KEY-EX           PIC X          VALUE SPACE.
*      KEY EXISTS IN MEMBER ? (YES/NO)
    88  KEY-YES          VALUE LMSUP-YES.
    88  KEY-NO           VALUE LMSUP-NO.
    02  CFID             PIC S9(9) COMP VALUE 0.

```

```

02 BLKCTRL-FIELD PIC X          VALUE LOW-VALUE.
*      BLKCTRL-INDICATOR
*      X'80'          S BLKCTRL=NO
*      X'40'          S BLKCTRL=PAMKEY
*      X'20'          S BLKCTRL=DATA
*      X'10'          S BLKCTRL=NULL
*      X'F0'          R BLKCTRL=NOT SPECIFIED
*      X'08'          S BLOCK CONTROL FIELD 4K
*      X'04'          S BLOCK CONTROL FIELD 2K
*      X'03'          R — RESERVED, MUST BE 0 —
02 PMPERF          PIC X          VALUE LOW-VALUE.
*      IOPERF-INDICATOR
*      X'03'          S IOPERF=VERY-HIGH
*      X'02'          S IOPERF=HIGH
*      X'01'          S IOPERF=STD
*      X'00'          S IOPREF NOT SPECIFIED
02 PMUSAG          PIC X          VALUE LOW-VALUE.
*      IOUSAGE-INDICATOR
*      X'03'          S IOUSAGE=RDWRT
*      X'02'          S IOUSAGE=WRITE
*      X'01'          S IOUSAGE=READ
*      X'00'          S IOUSAGE NOT SPECIFIED
02 PMEDMS3        PIC X          VALUE LOW-VALUE.
*      CATALOG-INDIC (IDCEX)
*      X'08'          S PLAM FILE
*
02 FILLER          PIC X          VALUE LOW-VALUE.
02 A-IND-COUNT    PIC 9(4) COMP  VALUE 0.
02 FILESIZE-32   PIC X(4)        VALUE LOW-VALUE.
    >= 32GB, FILESIZE MUST BE X'FFFFFF'
02 FILLER          PIC X(86)      VALUE LOW-VALUE.
*
02 AIX            OCCURS 30 TIMES.
    03 KEYNAME     PIC X(8)        VALUE LOW-VALUE.
    03 A-KEYPOS    PIC 9(4) COMP  VALUE 0.
    03 A-KEYLEN    PIC X          VALUE LOW-VALUE.
    03 INDICATOR   PIC X          VALUE LOW-VALUE.
*      X'80'          S DUPKEY=YES
*                      R DUPKEY=NO
***
***** END OF COPY ELEMENT LMSCO164 *****

```

5.4 Beispiel

Das folgende COBOL-Programm beinhaltet folgende Funktionen:

1. Eröffnen eines Unterprogrammzugriffs (INIT)
2. Aufnehmen einer Datei als Element (ADD)
3. Suchen eines Elementes in einem Inhaltsverzeichnis (TOCPRIM)
4. Eröffnen eines Elementes (OPENGET)
5. Satzweise Lesen eines Elementes (GET)
6. Schließen des Elementes (CLOSE)
7. Beenden des Unterprogrammzugriffs (END)

Zum besseren Verständnis wurden Kommentare in das Beispiel aufgenommen.

```

IDENTIFICATION DIVISION.
*****
PROGRAM-ID.          LMSUPCOB.
*****

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    TERMINAL IS MONITOR
    SYMBOLIC CHARACTERS
        COPY LMSCOBEQ.
    .

*   DER OBIGE PUNKT MUSS GESCHRIEBEN WERDEN, UM DEN           *
*   SPECIAL-NAMES PARAGRAPHEN ABZUSCHLIESSEN                 *
/
DATA DIVISION.
WORKING-STORAGE SECTION.
*****
*****
**
** DIE KONTROLLBLOECKE FUER DAS ARBEITEN MIT LMS ALS          **
** UNTERPROGRAMM LIEGEN IN DER BIBLIOTHEK SYSLIB.LMS.<VERS>  **
** ALS COPY-ELEMENTE VOR                                     **
*****
*****

COPY LMSCOBCB.

COPY LMSCOBED.

COPY LMSCOBEI.

COPY LMSCOBEM.

```

COPY LMSCOBF.D.

COPY LMSCOBL.D.

COPY LMSCOBR.D.

```
*****
*   TOC-IDENTIFICATION                               *
*****
01 LMSUP-TID                                PIC 9(08) BINARY VALUE 1.
```

```
*****
*   ELEMENT RECORD (ZUM UEBERGEHEN VON ELEMENTEN MIT PUT/GET) *
*****
01 LMSUP-ER.
   05 SATZKOPF.
       10 SATZLAENGE                                PIC S9(04) BINARY.
       10 FILLER                                    PIC X(02).
   05 SATZPUFFER                                PIC X(256).
```

```
*****
*   HILFSFELDER, KONSTANTENDEFINITIONEN FUER DAS PROGRAMM *
*****
01 H01-HILFSFELDER.
   05 H01-DATEI-LINKNAME                          PIC X(08) VALUE "FILELINK".
   05 H01-BIBLIOTHEK-LINKNAME                     PIC X(08) VALUE "LIBLINK".
   05 H01-ELEMENT-NAME                            PIC X(54) VALUE "PROBEELEM".
   05 H01-ELEMENT-TYP                             PIC X VALUE "S".
   05 H01-ELEMENT-VERSION                         PIC X(24) VALUE "1".
   05 H01-PUFFER-LAENGE                           PIC 9(09) BINARY VALUE 260.
```

```
*****
*   AUSGABEBEREICHE FUER ELEMENTSATZ                *
*****
01 A01-AUSGABE-FELDER.
   05 A01-SATZLAENGE                                PIC 9(04) BINARY.
   05 A01-AUSGABE-SATZ.
       10 FILLER                                    PIC X OCCURS 1 TO 256
                                                DEPENDING ON
                                                A01-SATZLAENGE.
```

/

```
*****
PROCEDURE DIVISION.
*****
```

```
STEUER SECTION.
ST-ANFANG.
  PERFORM LMS-INITIALISIEREN.
  IF LMSRET-OK
```

```
        THEN
            PERFORM LMS-AUFNEHMEN
            PERFORM LMS-INHALT
            PERFORM LMS-ELEM-BEARBEITEN
            PERFORM LMS-BEENDEN
        END-IF.
    ST-ENDE.
    STOP RUN.

/
    LMS-INITIALISIEREN SECTION.
    LMS-INIT-ANFANG.

    *****
    * CONTROL BLOCK VORBEREITEN FUER DIE INITIALISIERUNG *
    *****
        MOVE LMSUP-UPINIT TO FUNC    IN LMSUP-SCB.
        MOVE LMSUP-UNUSED TO SUBCODE IN LMSUP-SCB.
        CALL "LMSUP1"    USING LMSUP-SCB.

    *****
    * RETURN-CODE AUSWERTEN *
    *****
        IF LMSRET-OK
            THEN
                DISPLAY "INITIALISIEREN DURCHGEFUEHRT"
                UPON MONITOR
            ELSE
                DISPLAY "FEHLER BEIM INITIALISIEREN AUFGETRETEN"
                UPON MONITOR

        END-IF.
    LMS-INIT-ENDE.
    EXIT.

/
    LMS-AUFNEHMEN SECTION.
    LMS-AUF-ANFANG.
```

```

*****
* DIE DATEI MIT "DATEI-LINKNAME" WIRD UNTER "ELEMENT-NAME"      *
* IN DIE BIBLIOTHEK MIT "BIBLIOTHEK-LINKNAME" EINGETRAGEN      *
*                                                                *
* CONTROL BLOCK, FILE DESCRIPTION, LIBRARY DESCRIPTION UND      *
* ELEMENT DESCRIPTION VORBEREITEN FUER DAS AUFNEHMEN           *
* EINES ELEMENTES                                             *
*                                                                *
* (DIE ANDEREN FELDER BLEIBEN WIE BEIM INIT)                   *
*****

```

```

MOVE LMSUP-ADD           TO FUNC           IN LMSUP-SCB.
MOVE LMSUP-UNUSED       TO SUBCODE        IN LMSUP-SCB.
MOVE LMSUP-YES          TO OVERWRITE      IN LMSUP-SCB.
MOVE H01-DATEI-LINKNAME TO LINK           IN LMSUP-FD.
MOVE H01-BIBLIOTHEK-LINKNAME TO LINK      IN LMSUP-LD.
MOVE H01-ELEMENT-TYP    TO TYP           IN LMSUP-ED.
MOVE H01-ELEMENT-NAME   TO NAME          IN LMSUP-ED.
MOVE H01-ELEMENT-VERSION TO VERSION       IN LMSUP-ED.
CALL "LMSUP1" USING     LMSUP-SCB,
                        LMSUP-FD,
                        LMSUP-LD,
                        LMSUP-ED.

```

```

*****
* RETURN-CODE AUSWERTEN                                       *
*****

```

```

IF LMSRET-OK
THEN
    DISPLAY "AUFNEHMEN DES ELEMENTS " H01-ELEMENT-NAME
           " DURCHGEFUEHRT "
           UPON MONITOR
ELSE
    DISPLAY "FEHLER BEIM AUFNEHMEN EINES ELEMENTES"
           UPON MONITOR

END-IF.
LMS-AUF-ENDE.
EXIT.

/
LMS-INHALT SECTION.
LMS-INHALT-ANFANG.

```

```

*****
* ES WIRD NACH EINEM BESTIMMTEN ELEMENT MIT "ELEMENT-NAME"      *
* UND "ELEMENT-TYP" GESUCHT.                                     *
*                                                                 *
* CONTROL BLOCK UND ELEMENT MASK VORBEREITEN ZUM SUCHEN        *
* EINES ELEMENTES (TOCPRIM)                                     *
* (LIBRARY DEFINITION BLEIBT WIE VORBELEGT)                    *
*****

```

```

      MOVE LMSUP-TOCPRIM      TO FUNC      IN LMSUP-SCB.
      MOVE LMSUP-LONG        TO SUBCODE   IN LMSUP-SCB.
      MOVE H01-ELEMENT-TYP   TO TYP       IN LMSUP-EM.
      MOVE H01-ELEMENT-NAME  TO NAME      IN LMSUP-EM.
      CALL "LMSUP1" USING    LMSUP-SCB,
                             LMSUP-TID,
                             LMSUP-EI,
                             LMSUP-LD,
                             LMSUP-EM.

```

```

*****
* RETURN-CODE AUSWERTEN,                                       *
* DRUCKAUFBEREITETE AUSGABE DER ELEMENTINFORMATION            *
*****

```

```

      IF LMSRET-OK
      THEN
          DISPLAY "SUCHEN EINES ELEMENTES DURCHGEFUEHRT: "
          UPON MONITOR
          DISPLAY "TYP      ", TYP          IN LMSUP-EI
          UPON MONITOR
          DISPLAY "NAME     ", NAME        IN LMSUP-EI
          UPON MONITOR
          DISPLAY "VERSION  ", VERSION    IN LMSUP-EI
          UPON MONITOR
          DISPLAY "FORMAT   ", STORE-FORM  IN LMSUP-EI
          UPON MONITOR
          DISPLAY "USER-DATE ", USER-DATE  IN LMSUP-EI
          UPON MONITOR
          DISPLAY "USER-TIME ", USER-TIME  IN LMSUP-EI
          UPON MONITOR
          DISPLAY "CR-DATE  ", CREATION-DATE IN LMSUP-EI
          UPON MONITOR
          DISPLAY "CR-TIME  ", CREATION-TIME IN LMSUP-EI
          UPON MONITOR
          DISPLAY "MOD-DATE ", MODIFI-DATE IN LMSUP-EI
          UPON MONITOR
          DISPLAY "MOD-TIME ", MODIFI-TIME IN LMSUP-EI
          UPON MONITOR

```

```

        DISPLAY "SEC-NAME  ", SEC-NAME      IN LMSUP-EI
        UPON MONITOR
        DISPLAY "SEC-ATTR  ", SEC-ATTRIBUTE IN LMSUP-EI
        UPON MONITOR
    ELSE
        DISPLAY "FEHLER BEIM SUCHEN EINES ELEMENTES"
        UPON MONITOR
    END-IF.
LMS-INHALT-ENDE.
EXIT.

```

/

```

LMS-ELEM-BEARBEITEN SECTION.
LMS-ELEM-BEA-ANFANG.

```

```

*****
* EIN ELEMENT WIRD ZUR BEARBEITUNG GEOEFFNET, SATZWEISE          *
* GELESEN UND WIEDER GESCHLOSSEN                                *
*****
*****
* CONTROL BLOCK UND ELEMENT DESCRIPTION VORBEREITEN FUER DAS    *
* OEFFNEN EINES ELEMENTES.                                     *
* (LIBRARY DESCRIPTION BLEIBT WIE BISHER, MUSS ABER VERSORGT    *
* SEIN. DAS ELEMENT MUSS DURCH DIE DREI EIGENSCHAFTEN TYP,     *
* NAME UND VERSION EINDEUTIG BESCHRIEBEN SEIN)                 *
*****
        MOVE LMSUP-OPEN-GET      TO FUNC      IN LMSUP-SCB.
        MOVE LMSUP-UNUSED       TO SUBCODE   IN LMSUP-SCB.
        MOVE H01-ELEMENT-NAME   TO NAME      IN LMSUP-ED.
        MOVE H01-ELEMENT-TYP    TO TYP       IN LMSUP-ED.
        MOVE H01-ELEMENT-VERSION TO VERSION  IN LMSUP-ED.
        CALL "LMSUP1" USING      LMSUP-SCB,
                                LMSUP-RD,
                                LMSUP-LD,
                                LMSUP-ED.

```

```

*****
* RETURN-CODE AUSWERTEN,                                     *
* WENN KEIN FEHLER AUFGETRETEN IST, WIRD GELESEN           *
*****
      IF LMSRET-OK
      THEN

*****
* CONTROL BLOCK UND RECORD DESCRIPTION VORBEREITEN FUER     *
* DAS SATZWEISE LESEN EINES ELEMENTES.                     *
* (MIT PUFFER-LAENGE WIRD DIE MAX. ERWARTETE SATZLAENGE    *
* ANGEGEBEN)                                               *
*****
      MOVE LMSUP-GET          TO FUNC          IN LMSUP-SCB
      MOVE LMSUP-SEQ          TO SUBCODE       IN LMSUP-SCB
      MOVE HO1-PUFFER-LAENGE TO BUFFER-LEN    IN LMSUP-RD
*****
* LESESCHLEIFE BIS ELEMENTENDE                             *
*****
      PERFORM UNTIL NOT LMSRET-OK
          CALL "LMSUP1" USING LMSUP-SCB,
                               LMSUP-RD,
                               LMSUP-ER

*****
* RETURN-CODE AUSWERTEN                                     *
*****
      EVALUATE TRUE
      WHEN LMSRET-OK
          SUBTRACT 4 FROM SATZLAENGE IN LMSUP-ER
          GIVING A01-SATZLAENGE
          MOVE SATZPUFFER TO A01-AUSGABE-SATZ
          DISPLAY A01-AUSGABE-SATZ UPON MONITOR
      WHEN LMSRET-EOF
          DISPLAY "ELEMENTENDE ERREICHT"
          UPON MONITOR
      WHEN LMSRET-TRUNC
          DISPLAY "SATZPUFFER ZU KURZ ANGEGEBEN"
          UPON MONITOR
      WHEN OTHER
          DISPLAY "FEHLER BEIM LESEN EINES SATZES"
          UPON MONITOR
      END-EVALUATE
      END-PERFORM

```

```

*****
* CONTROL BLOCK UND RECORD DESCRIPTION VORBEREITEN FUER          *
* DAS SCHLIESSEN EINES ELEMENTES.                               *
*****
      MOVE LMSUP-CLOSE   TO FUNC    IN LMSUP-SCB
      MOVE LMSUP-UNUSED TO SUBCODE IN LMSUP-SCB
      CALL "LMSUP1"     USING  LMSUP-SCB,
                          LMSUP-RD

*****
* RETURN-CODE AUSWERTEN                                         *
*****
      IF LMSRET-OK
      THEN
          DISPLAY "ELEMENT ' " H01-ELEMENT-NAME " ' WURDE "
                  "GESCHLOSSEN"
                  UPON MONITOR
      ELSE
          DISPLAY "FEHLER BEIM SCHLIESSEN EINES ELEMENTES"
                  UPON MONITOR
      END-IF
      ELSE
          DISPLAY "FEHLER BEIM OEFFNEN EINES ELEMENTES"
                  UPON MONITOR
      END-IF.
      LMS-ELEM-BEA-ENDE.
      EXIT.

/
      LMS-BEENDEN SECTION.
      LMS-BEENDEN-ANFANG.

*****
* CONTROL BLOCK VORBEREITEN FUER DAS BEENDEN DES LMS          *
*****
      MOVE LMSUP-UPEND TO FUNC    IN LMSUP-SCB.
      MOVE LMSUP-UNUSED TO SUBCODE IN LMSUP-SCB.
      CALL "LMSUP1" USING LMSUP-SCB.

```

```
*****
* RETURN-CODE AUSWERTEN                                     *
*****
    IF LMSRET-OK
    THEN
        DISPLAY "LMS BEENDET"
            UPON MONITOR
    ELSE
        DISPLAY "FEHLER BEIM BEENDEN DES LMS"
            UPON MONITOR
    END-IF.
LMS-BEENDEN-ENDE.
EXIT.
```

6 C-Schnittstelle

6.1 Der Verbindungsmodul LMSUP1

Zum Aufruf der LMS-Funktionen aus C-Programmen steht der Modul LMSUP1 zur Verfügung. Sein Aufruf erfolgt über:

LMSUP1 (parameterliste)

Für die Einträge in die Parameterliste siehe [Seite 17](#), Beschreibung der Funktionen. Der Modul LMSUP1 wird aus der Bibliothek SYSLNK.LMS.034 zum Hauptprogramm hinzugebunden.

6.2 Include-Elemente für den C/C++-Compiler

Um die Parameterstrukturen für C generieren zu können, muss zunächst die Bibliothek SYSLIB.LMS.034, in der sich das Include-Element LMS.H befindet, zugewiesen werden. Die Zuweisung kann mit folgender Anweisung für den C/C++-Compiler erfolgen:

```
//MODIFY-INCLUDE-LIBRARIES -  
//      STD-INCLUDE-LIBRARY = (... , $.SYSLIB.LMS.034)
```

Der Installationsort dieser Bibliothek ist über IMON frei wählbar.

Mit Hilfe der Built-In-Funktion INSTALLATION-PATH kann der Installationsort von SYSLIB.LMS.034 ermittelt und in eine S-Variable gespeichert werden:

```
/SET-VARIABLE LIBRARY-NAME =INSTALLATION-PATH      -  
              (LOGICAL-ID = `SYSLIB`             -  
              ,INSTALLATION-UNIT = `LMS`         -  
              ,VERSION = `3.4`                  -  
              ,DEFAULT-PATH-NAME = `$.SYSLIB.LMS.034`)
```

Anschließend kann die Bibliothek folgendermaßen zugewiesen werden:

```
/START-CPLUS-COMPILER
//MODIFY-INCLUDE-LIBRARIES -
//          STD-INCLUDE-LIBRARY = (... , &(LIBRARY-NAME))
```

Include-Element LMS.H

```
/* LMS.H          @>V3.3000 1997-11-06<@ */

#ifdef WAS_LMS
#else

#ifdef __cplusplus
extern "C" {
#endif

/*****
/*          L M S U P          *****/
/*****

#define WAS_LMS
#define BLANK4  ' ',' ',' ',' '
#define BLANK8  ' ',' ',' ',' ',' ',' ',' '
#define BLANK18 BLANK8,BLANK8,' ',' '
#define BLANK24 BLANK8,BLANK8,BLANK8
#define BLANK64 BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8
#define ZERO4   {'\0','\0','\0','\0'}
#define X0_8    '\0','\0','\0','\0','\0','\0','\0','\0'
#define X0_32   X0_8,X0_8,X0_8,X0_8

/*****
/*          Function Control Block          *****/
/*****

struct lmsc_cb
{
    char  scbvers[2];          /* interface-version          in */
    char  function;           /* function-code               in */
    char  subcode;            /* function-subcode            in */
    int   acc;                /* up access-identification    inout */
    char  retcode;            /* returncode                   out */
    char  filler1;            /* reserved                      */
    short lms_msg;            /* lms message-code            out */
    short dms_msg;            /* dms-message-code            out */
    short plam_msg;           /* plam-message-code           out */
}
```

```

char  lmsvers[12];      /* version of lms                out */
int   filler2;         /* reserved                      */
/***** lms parameters *****/
char  destroy;        /* element-destroy-option        inout */
char  fcb;            /* fcbtype for file              inout */
char  rkey;           /* key-handling                  inout */
char  overwrite;     /* overwrite-option              inout */
short column;        /* lms parameter: column         inout */
short line;          /* lms parameter: line           inout */
char  prot_ind;      /* lms par: protection ind.      inout */
char  attr;          /* copylib with attributes       inout */
char  info;          /* record types for ....         inout */
char  ld_return;     /* DMS-filename in ld           inout */
char  filler3[4];    /* reserved                      */
};
#define LMSC_CB_PROTO \
{ \
    {'\xF0', '\xF4'},          /* scbvers                */ \
    '\x01',                    /* function                */ \
    ' ',                        /* subcode                 */ \
    -1,                        /* acc                     */ \
    '\0',                      /* retcode                 */ \
    '\0',                      /* filler1                 */ \
    0,                         /* lms_msg                 */ \
    0,                         /* dms_msg                 */ \
    0,                         /* plam_msg                */ \
    {BLANK8, BLANK4},          /* lmsvers                 */ \
    0,                         /* filler2                 */ \
    ' ',                        /* destroy                 */ \
    ' ',                        /* fcbtype                 */ \
    ' ',                        /* rkey                    */ \
    ' ',                        /* overwrite                */ \
    0,                         /* column                  */ \
    0,                         /* line                    */ \
    ' ',                        /* prot_ind                */ \
    ' ',                        /* attr                    */ \
    ' ',                        /* rec_range               */ \
    ' ',                        /* ld_return               */ \
    {BLANK4}                    /* filler3                 */ \
}
static struct lmsc_cb lmsc_cb_proto = LMSC_CB_PROTO;

/*****
/*                               Element Attributes                               *****/
/*****
struct lmsc_ea

```

```

{
    char  user_date[14];    /* date from user          in */
    char  user_time[8];    /* time from user hh:mm:ss in */
    char  ccs_name[8];     /* ccs from user          in */
                                /*      V03.00 */
    char  hold_state;      /* hold flag: '-' : free   in */
                                /*      'H' : in hold     */
                                /*      ' ' : unchange    */
    char  filler1[8];      /* reserved (holder userid) in */
    char  mod_date_ind;    /* modification date indicator in */
                                /*      'O' : old         */
                                /*      'S' : system date */
    char  filler2[56];     /* reserved                in */
};
#define LMSC_EA_PROTO \
{\
    {BLANK8,BLANK4,' ',' '},          /* user-date          */\
    {BLANK8},                        /* user-time          */\
    {BLANK8},                        /* ccs-name           */\
    ' ',                              /* hold-state        */\
    {BLANK8},                        /* reserved (holder) */\
    'O',                              /* mod-dat-ind       */\
    {BLANK24,BLANK24,BLANK8}         /* reserved           */\
}
static struct lmsc_ea lmsc_ea_proto = LMSC_EA_PROTO;

/*****
/*          Element Description          *****/
/*****
struct lmsc_ed
{
    char  typ[8];            /* typ of element          in */
    char  name[64];         /* name of element         in */
    char  version[24];      /* version fo element      in */
    char  store_form;       /* storage form            in */
    char  user_date[14];    /* date from user          in */
    char  user_time[8];     /* time from user hh:mm:ss in */
};
#define LMSC_ED_PROTO \
{\
    {BLANK8},                /* typ                */\
    {BLANK64},              /* name                */\
    {BLANK8,BLANK8,BLANK8}, /* version            */\
    'V',                    /* stroage-form       */\
    {BLANK8,BLANK4,' ',' '}, /* user-date          */\
    {BLANK8}                /* user-time          */\
}

```

```

}
static struct lmsc_ed lmsc_ed_proto = LMSC_ED_PROTO;

/*****
/*                               Element Information                               *****/
/*****
struct lmsc_ei
{
    char   typ[8];                /* type of element                out */
    char   name[64];              /* name of element                 out */
    char   version[24];          /* version fo element              out */
    char   store_form;           /* storgae form                    out */
    char   user_date[14];        /* date from user                  out */
    char   user_time[8];         /* time from user                  out */
    char   crea_date[14];        /* creation date of element        out */
    char   crea_time[8];         /* creation time                   out */
    char   modif_date[14];       /* date of last modification       out */
    char   modif_time[8];        /* time of last modifcation        out */
    char   sec_name[32];         /* secondary name                  out */
    char   sec_attr[8];          /* attribute of secondary name     out */
    char   filler1[5];           /* reserved                         */
    /***** protection attributes *****/
    char   p_tind_read;          /* prot. type indi. for read       out */
    char   p_read_own;          /* read indicator for owner         out */
    char   p_read_grp;          /* read indicator for group         out */
    char   p_read_oth;          /* read indicator for others        out */
    char   p_read_pind;         /* read password indicator         out */
    char   filler2[4];          /* reserved                         */
    char   p_tind_writ;          /* prot. type indi. for write      out */
    char   p_writ_own;          /* write indicator for owner        out */
    char   p_writ_grp;          /* write indicator for group        out */
    char   p_writ_oth;          /* write indicator for others       out */
    char   p_writ_pind;         /* write password indicator         out */
    char   filler3[4];          /* reserved                         */
    char   p_tind_exec;          /* prot. type indi. for exec       out */
    char   p_exec_own;          /* exec indicator for owner         out */
    char   p_exec_grp;          /* exec indicator for group         out */
    char   p_exec_oth;          /* exec indicator for others        out */
    char   p_exec_pind;         /* exec password indicator         out */
    char   filler4[4];          /* reserved                         */
    char   p_read_guard[18];     /* read guard                       out */
    char   p_writ_guard[18];     /* write guard                      out */
    char   p_exec_guard[18];     /* exec guard                       out */
    char   ccs_name[8];         /* ccs-name                        out */
                                /* V03.00: source code control    */
    char   p_tind_hold;         /* prot. type indi. for holder     out */

```

```

char p_hold_own;      /* holder indicator for owner   out */
char p_hold_grp;     /* holder indicator for group   out */
char p_hold_oth;     /* holder indicator for others  out */
char p_hold_pind;    /* holder password indicator    out */
char filler5[4];     /* reserved                     */
char p_hold_guard[18]; /* holder guard                  out */
                        /*                               */
char hold_state;     /* hold flag: '-' : free        out */
                        /*                               */
                        /*                               */
                        /*                               */
char holder[8];      /* holder userid                 out */
char access_date[14]; /* access date                   out */
char access_time[ 8]; /* access time                   out */
char filler8[ 1];    /* reserved                     */
unsigned int element_size; /* element size                out */
char filler9[40];    /* reserved                     */
};
#define LMSC_EI_PROTO \
{ \
    {BLANK8},          /* typ                          */
    {BLANK64},        /* name                          */
    {BLANK8,BLANK8,BLANK8}, /* version                      */
    ' ',              /* storage form                  */
    {BLANK8,BLANK4,' ',' '}, /* user date                    */
    {BLANK8},         /* user time                     */
    {BLANK8,BLANK4,' ',' '}, /* creation date                */
    {BLANK8},         /* creation time                 */
    {BLANK8,BLANK4,' ',' '}, /* modification date            */
    {BLANK8},         /* modification time            */
    {BLANK8,BLANK8,BLANK8,BLANK8}, /* secondary name              */
    {BLANK8},         /* secondary attr                */
    {BLANK4,' '},     /* reserved                      */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for read               */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for write              */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for exec               */
    {BLANK18},        /* read guard                    */
    {BLANK18},        /* write guard                   */
    {BLANK18},        /* exec guard                    */
    {BLANK8},         /* ccs-name                      */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for holder            */
    {BLANK18},        /* holder guard                  */
    ' ',              /* hold-state                    */
    {BLANK8},         /* holder                        */
    {BLANK8,BLANK4,' ',' '}, /* access date                  */
    {BLANK8},         /* access time                   */
    ' ',              /* reserved                      */
    0,                /* element size                  */
    {BLANK18,BLANK18,BLANK4} /* reserved                      */
}

```

```

static struct lmsc_ei lmsc_ei_proto = LMSC_EI_PROTO;

/*****
/*                               Element Mask                               *****/
/*****
struct lmsc_em
{
    char   typ[20];                /* type of element                in */
    char   name[132];             /* name of element                in */
    char   version[52];          /* version of element             in */
    char   store_form[6];        /* storage form                   in */
    char   user_date[32];        /* date from user                 in */
    char   user_time[20];        /* time from user hh:mm:ss       in */
    char   crea_date[32];        /* creation date of element       in */
    char   crea_time[20];        /* creation time of element       in */
    char   modif_date[32];       /* date of last modification     in */
    char   modif_time[20];       /* time of last modification     in */
    char   sec_name[68];         /* secondary name                 in */
    char   sec_attr[20];         /* attribute of secondary name   in */
    char   filler1[14];          /* reserved                       */
    /***** protection attributes *****/
    char   p_tind_read;          /* prot. type indi. for read     in */
    char   p_read_own;          /* read indicator for owner      in */
    char   p_read_grp;          /* read indicator for group      in */
    char   p_read_oth;          /* read indicator for others     in */
    char   p_read_pind;         /* read password indicator      in */
    char   filler2[4];           /* reserved                       */
    char   p_tind_writ;         /* prot. type indi. for write   in */
    char   p_writ_own;          /* write indicator for owner     in */
    char   p_writ_grp;          /* write indicator for group     in */
    char   p_writ_oth;          /* write indicator for others   in */
    char   p_writ_pind;         /* write password indicator     in */
    char   filler3[4];           /* reserved                       */
    char   p_tind_exec;         /* prot. type indi. for exec    in */
    char   p_exec_own;          /* exec indicator for owner     in */
    char   p_exec_grp;          /* exec indicator for group     in */
    char   p_exec_oth;          /* exec indicator for others    in */
    char   p_exec_pind;         /* exec password indicator     in */
    char   filler4[4];           /* reserved                       */
    char   p_read_guard[40];     /* read guard                    in */
    char   p_writ_guard[40];     /* write guard                   in */
    char   p_exec_guard[40];     /* exec guard                    in */
    char   ccs_name[20];        /* ccs-name                     in */
                                /* V03.00: source code control  */
    char   p_tind_hold;         /* prot. type indi. for holder  in */
    char   p_hold_own;          /* holder indicator for owner    in */

```

```

char p_hold_grp; /* holder indicator for group in */
char p_hold_oth; /* holder indicator for others in */
char p_hold_pind; /* holder password indicator in */
char filler5[4]; /* reserved */
char p_hold_guard[40]; /* holder guard in */
/*
char hold_state; /* hold flag: '-' : free in */
/* 'H' : inhold */
/* ' ' : any */
char holder[20]; /* holder userid in */
char access_date[32]; /* access date in */
char access_time[20]; /* access time in */
char filler8[ 3]; /* reserved */
unsigned int e_size_min; /* min element size in */
unsigned int e_size_max; /* max element size in */
char filler9[64]; /* reserved */
};
#define LMSC_EM_PROTO \
{\
    {BLANK8,BLANK8,BLANK4}, /* type */
    {BLANK64,BLANK64,BLANK4}, /* name */
    {BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK4}, /* vers */
    {BLANK4,' ',' '}, /* storage form */
    {BLANK8,BLANK8,BLANK8,BLANK8}, /* user date */
    {BLANK8,BLANK8,BLANK4}, /* user time */
    {BLANK8,BLANK8,BLANK8,BLANK8}, /* creation date */
    {BLANK8,BLANK8,BLANK4}, /* creation time */
    {BLANK8,BLANK8,BLANK8,BLANK8}, /* modification date */
    {BLANK8,BLANK8,BLANK4}, /* modification time */
    {BLANK64,BLANK4}, /* secondary name */
    {BLANK8,BLANK8,BLANK4}, /* secondary attr */
    {BLANK8,BLANK4,' ',' '}, /* reserved */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for read */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for write */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for exec */
    {BLANK18,BLANK18,BLANK4}, /* read guard */
    {BLANK18,BLANK18,BLANK4}, /* write guard */
    {BLANK18,BLANK18,BLANK4}, /* exec guard */
    {BLANK8,BLANK8,BLANK4}, /* ccs-name */
    ' ',' ',' ',' ',' ',' ',ZER04, /* prot for holder */
    {BLANK18,BLANK18,BLANK4}, /* holder guard */
    ' ', /* hold-state */
    {BLANK18,' ',' '}, /* holder */
    {BLANK8,BLANK8,BLANK8,BLANK8}, /* access date */
    {BLANK8,BLANK8,BLANK4}, /* access time */
    {' ',' ',' ',' '}, /* reserved */
    0, /* min element size */
    0xFFFFFFFF, /* max element size */

```

```

        {BLANK64}                                /* reserved          */\
    }
static struct lmsc_em lmsc_em_proto = LMSC_EM_PROTO;

/*****
/*                               File Description          *****/
/*****
struct lmsc_fd
{
    char  password[4];          /* password              in */
    char  link[8];             /* linkname for file     in */
    char  name[54];            /* filename              inout */
};
#define LMSC_FD_PROTO \
{\
    ZER04,                    /* password              */\
    {BLANK8},                 /* linkname              */\
    {BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK4,' ',' ' } \
}
static struct lmsc_fd lmsc_fd_proto = LMSC_FD_PROTO;

/*****
/*                               Library Attributes        *****/
/*****
struct lmsc_la
{
    char  p_tind_admin;        /* prot. type indi. for admin  in */
    char  p_admin_own;         /* admin indicator for owner    in */
    char  p_admin_grp;         /* admin indicator for group    in */
    char  p_admin_oth;         /* admin indicator for others   in */
    char  p_admin_pind;        /* admin password indicator     in */
    char  p_admin_pswd[4];     /* admin password              in */
    char  p_admin_guard[18];   /* admin guard                 in */
                                /* V03.00 */
    char  store_form;          /* ind. for holder-authorization in */
                                /* 'S': STD fully or delta stored */
                                /* 'V': FULL all el. fully stored */
                                /* 'D': DELTA all el. fully stored */
    char  write_ctrl;          /* write-control for library    in */
                                /* 'A': activated              */
                                /* 'D': deactivated            */
    char  access_date;         /* access date                  in */
                                /* 'N': NONE (no KEEP)        */
                                /* 'K': KEEP                   */
    char  filler2[34];         /* reserved                     */
}

```

```

};
#define LMSC_LA_PROTO \
{\
    ' ',' ',' ',' ',' ',' ',ZER04,          /* prot for Adm   in */\
    {BLANK18},                               /* admin guard   */\
    ' ',' ',' ',' ',' ',                    /* store-form/wr-ctrl*/\
    {BLANK24,BLANK8,' ',' ' }                /* reserved      */\
}
static struct lmsc_la lmsc_la_proto = LMSC_LA_PROTO;

/*****
/*          Library Description          *****/
/*****
struct lmsc_ld
{
    char password[4];          /* password for library   in */
    char link[8];             /* linkname                in */
    char reserv1[8];          /*                          */
    short max_name_len;       /* max.length of library_name in */
    char name[54];            /* name of library        inout */
};
#define LMSC_LD_PROTO \
{\
    ZER04,                      /* password             */\
    {BLANK8},                    /* linkname              */\
    {'\0','\0','\0','\0','\0','\0','\0','\0'},/* reserv1              */\
    (short)54,                   /* max length name      */\
    {BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK4,' ',' ' } \
}
static struct lmsc_ld lmsc_ld_proto = LMSC_LD_PROTO;

/*****
/*          Library Information          *****/
/*****
struct lmsc_li
{
    char p_tind_admin;         /* prot. type indi. for admin out */
    char p_admin_own;         /* admin indicator for owner   out */
    char p_admin_grp;         /* admin indicator for group   out */
    char p_admin_oth;         /* admin indicator for others  out */
    char p_admin_pind;        /* admin password indicator    out */
    char filler1[4];          /* reserved                     */
    char p_admin_guard[18];    /* admin guard                  */
    char store_form;          /* ind. for holder-authorization out */
}

```

```

/* 'S': STD   fully or delta stored */
/* 'V': FULL  all el. fully stored  */
/* 'D': DELTA all el. fully stored  */
char write_ctrl; /* write-control for library      out */
/* 'A': activated                      */
/* 'D': deactivated                      */
char access_date; /* access date                      out */
/* 'N': NONE (no KEEP)                  */
/* 'K': KEEP                             */
char filler2[24]; /* reserved                          */
char lib_form; /* library format (NK2/NK4)          out */
/* '2' : NK2 library format             */
/* '4' : NK4 library format             */
char upam_prot; /* UPAM protected (yes/no)          out */
/* 'Y' : lib is UPAM protected          */
/* 'N' : lib is not UPAM protected     */
int file_size; /* file size                          out */
int free_size; /* free size                           out */
/***** protection attributes *****/
char p_tind_read; /* prot. type indi. for read        out */
char p_read_own; /* read indicator for owner          out */
char p_read_grp; /* read indicator for group          out */
char p_read_oth; /* read indicator for others         out */
char p_read_pind; /* read password indicator          out */
char filler4[4]; /* reserved                          */
char p_tind_writ; /* prot. type indi. for write       out */
char p_writ_own; /* write indicator for owner         out */
char p_writ_grp; /* write indicator for group         out */
char p_writ_oth; /* write indicator for others        out */
char p_writ_pind; /* write password indicator          out */
char filler5[4]; /* reserved                          */
char p_tind_exec; /* prot. type indi. for exec        out */
char p_exec_own; /* exec indicator for owner          out */
char p_exec_grp; /* exec indicator for group          out */
char p_exec_oth; /* exec indicator for others         out */
char p_exec_pind; /* exec password indicator          out */
char filler6[4]; /* reserved                          */
char p_read_guard[18]; /* read guard                        out */
char p_writ_guard[18]; /* write guard                       out */
char p_exec_guard[18]; /* exec guard                        out */
/* V03.00 */
char p_tind_hold; /* prot. type indi. for holder      out */
char p_hold_own; /* holder indicator for owner        out */
char p_hold_grp; /* holder indicator for group        out */
char p_hold_oth; /* holder indicator for others       out */
char p_hold_pind; /* holder password indicator         out */
char filler7[4]; /* reserved                          */
char p_hold_guard[18]; /* holder guard                      out */

```

```

        char filler8[68];        /* reserved */
};
#define LMSC_LI_PROTO \
{
    ' ',' ',' ',' ',' ',' ',' ',ZER04,        /* prot for admin */\
    {BLANK18},                                /* admin guard */\
    ' ',' ',' ',' ',' ',                /* store-form/wr-ctrl */\
    {BLANK24},                                /* reserved */\
    ' ',' ',' ', 0, 0,                    /* fomat + size */\
    ' ',' ',' ',' ',' ',' ',ZER04,          /* prot for read */\
    ' ',' ',' ',' ',' ',' ',ZER04,          /* prot for write */\
    ' ',' ',' ',' ',' ',' ',ZER04,          /* prot for exec */\
    {BLANK18},                                /* read guard */\
    {BLANK18},                                /* write guard */\
    {BLANK18},                                /* exec guard */\
    ' ',' ',' ',' ',' ',' ',ZER04,          /* holder */\
    {BLANK18},                                /* holder guard */\
    {BLANK64,BLANK4}                          /* reserved */
}
static struct lmsc_li lmsc_li_proto = LMSC_LI_PROTO;

```

```

/*****
/*
Protection Attributes
*****/
/*****
struct lmsc_pa
{
    char p_tind_read;        /* prot. type indi. for read in */
    char p_read_own;        /* read indicator for owner in */
    char p_read_grp;        /* read indicator for group in */
    char p_read_oth;        /* read indicator for others in */
    char p_read_pind;       /* read password indicator in */
    char p_read_pswd[4];    /* read password in */
    char p_tind_writ;       /* prot. type indi. for write in */
    char p_writ_own;        /* write indicator for owner in */
    char p_writ_grp;        /* write indicator for group in */
    char p_writ_oth;        /* write indicator for others in */
    char p_writ_pind;       /* write password indicator in */
    char p_writ_pswd[4];    /* write password in */
    char p_tind_exec;       /* prot. type indi. for exec in */
    char p_exec_own;        /* exec indicator for owner in */
    char p_exec_grp;        /* exec indicator for group in */
    char p_exec_oth;        /* exec indicator for others in */
    char p_exec_pind;       /* exec password indicator in */
    char p_exec_pswd[4];    /* exec password in */
    char p_read_guard[18];  /* read guard in */
    char p_writ_guard[18];  /* write guard in */
}

```

```

    char p_exec_guard[18]; /* exec guard          in */
                          /* V03.00 */
    char p_tind_hold;     /* prot. type indi. for holder in */
    char p_hold_own;     /* holder indicator for owner  in */
    char p_hold_grp;     /* holder indicator for group  in */
    char p_hold_oth;     /* holder indicator for others in */
    char p_hold_pind;    /* holder password indicator  in */
    char p_hold_pswd[4]; /* holder password            in */
    char p_hold_guard[18]; /* holder guard                in */
    char filler1[84];    /* reserved                     */
};
#define LMSC_PA_PROTO \
{ \
    ' ',' ',' ',' ',' ',' ',' ',ZER04, /* prot for read */
    ' ',' ',' ',' ',' ',' ',' ',ZER04, /* prot for write */
    ' ',' ',' ',' ',' ',' ',' ',ZER04, /* prot for exec */
    {BLANK18}, /* read guard */
    {BLANK18}, /* write guard */
    {BLANK18}, /* exec guard */
    ' ',' ',' ',' ',' ',' ',' ',ZER04, /* prot for holder */
    {BLANK18}, /* holder guard */
    {BLANK64,BLANK18,' ',' ' } /* reserved */
}
static struct lmsc_pa lmsc_pa_proto = LMSC_PA_PROTO;

/*****
/*                               Record Description                               */
/*****
struct lmsc_rd
{
    int  rec_acc_id; /* element access id      inout */
    int  buffer_len; /* length of recordbuffer  in */
    int  record_len; /* length of record        inout */
    char filler1[3]; /* reserved                */
    char record_type; /* record type             inout */
    int  record_num; /* number of record        inout */
    char filler2[8]; /* reserved                */
};
#define LMSC_RD_PROTO \
{ \
    -1, /* rec_acc_id */
    0, /* buffer_len */
    0, /* record_len */
    {'\0','\0','\0'}, /* reserv1 */
    '\x01', /* record_type */
    0, /* record_num */
}

```

```

        {'\0','\0','\0','\0','\0','\0','\0','\0'} /* reserv2          */\
    }
    static struct lmsc_rd lmsc_rd_proto = LMSC_RD_PROTO;

    /*****
    /*                                Type Attributes                                *****/
    /*****
    struct lmsc_ta
    {
        char  convention;          /* type convention          in */
        char  filler1[3];         /* reserved                  */
        char  v_example[24];      /* version example (f. seq) in */
                                   /* V03.00 */
        char  p_tind_admin;       /* prot. type indi. for admin in */
        char  p_admin_own;        /* admin indicator for owner in */
        char  p_admin_grp;        /* admin indicator for group in */
        char  p_admin_oth;        /* admin indicator for others in */
        char  p_admin_pind;       /* admin password indicator in */
        char  p_admin_pswd[4];    /* admin password           in */
        char  p_admin_guard[18]; /* admin guard              in */
                                   /* V03.00 */
        char  store_form;         /* ind. for holder-authorization in */
                                   /* 'S': STD fully or delta stored */
                                   /* 'V': FULL all el. fully stored */
                                   /* 'D': DELTA all el. fully stored */
        char  write_ctrl;         /* write-control for type    in */
                                   /* 'A': activated           */
                                   /* 'D': deactivated        */
        char  super_type[8];      /* super type                in */
        char  filler2[47];       /* reserved                  */
    };
    #define LMSC_TA_PROTO \
    {\
        ' ',                          /* convention          */\
        {' ', ' ', ' ', ' ', ' ', ' ', /* reserved          */\
        {BLANK24},                     /* v_example          */\
        ' ', ' ', ' ', ' ', ' ', ' ', /* prot for Adm    in */\
        {BLANK18},                     /* admin guard       */\
        ' ', ' ', ' ', ' ', ' ', ' ', /* store-form/wr-ctrl */\
        {BLANK8},                      /* super type        */\
        {BLANK24,BLANK18,BLANK4,' ' } /* reserved          */\
    }
    static struct lmsc_ta lmsc_ta_proto = LMSC_TA_PROTO;

```

```

/*****
/*                               Type Description                               *****/
/*****
struct lmsc_td
{
    char   typ[8];                /* element type                in */
    char   filler1[8];           /* reserved                     */
};
#define LMSC_TD_PROTO \
{\
    {BLANK8},                    /* typ                          */\
    {BLANK8}                     /* reserved                      */\
}
static struct lmsc_td lmsc_td_proto = LMSC_TD_PROTO;

```

```

/*****
/*                               Type Information                               *****/
/*****
struct lmsc_ti
{
    char   typ[8];                /* element type                out */
    char   filler1[8];           /* reserved                     */
    char   convention;           /* type convention              out */
    char   filler2[3];           /* reserved                     */
    char   v_example[24];        /* version example (f. seq)     out */
                                        /* V03.00 */
    char   p_tind_admin;         /* prot. type indi. for admin   out */
    char   p_admin_own;         /* admin indicator for owner     out */
    char   p_admin_grp;         /* admin indicator for group     out */
    char   p_admin_oth;         /* admin indicator for others    out */
    char   p_admin_pind;        /* admin password indicator     out */
    char   filler3[4];           /* reserved                     */
    char   p_admin_guard[18];    /* admin guard                   out */
                                        /* V03.00 */
    char   store_form;          /* ind. for holder-authorization out */
                                        /* 'S': STD fully or delta stored */
                                        /* 'V': FULL all el. fully stored */
                                        /* 'D': DELTA all el. fully stored */
    char   write_ctrl;          /* write-control for library     out */
                                        /* 'A': activated                */
                                        /* 'D': deactivated              */
    char   super_type[8];        /* super type                    out */
    char   basis_type[8];        /* basis type                    out */
    char   filler5[39];          /* reserved                     */
    /***** protection attributes *****/
    char   p_tind_read;         /* prot. type indi. for read    out */

```



```

#define LMSUP_MOD_EL_PROT '\x15' /* modify element protection */
#define LMSUP_MOD_LIB_A '\x16' /* modify library attributes */
#define LMSUP_SHOW_LIB_A '\x17' /* show library attributes */
#define LMSUP_MOD_TYP_A '\x18' /* modify type attributes */
#define LMSUP_SHOW_TYP_A '\x19' /* show type attributes */
#define LMSUP_MOD_EA '\x1A' /* modify element attributes */
#define LMSUP_COPY_LIB '\x1B' /* copy library */
#define LMSUP_CLOSE_LIB '\x1C' /* close library */
#define LMSUP_PROVIDE '\x1D' /* provide element */
#define LMSUP_RETURN '\x1E' /* return element */
#define LMSUP_GET_SYS_EL '\x1F' /* get variable syslmslem */
#define LMSUP_REORG_LIB '\x20' /* reorganize library */
/* ----- subcodes ----- */
#define LMSUP_UNUSED ' ' /* subcode unused (default) */
#define LMSUP_SHORT 'S' /* toc short */
#define LMSUP_LONG 'L' /* toc long */
#define LMSUP_DIR 'D' /* direct read */
#define LMSUP_SEQ 'S' /* sequential read */
#define LMSUP_WRITE 'W' /* close output element */
#define LMSUP_RESET 'R' /* reset output element */
#define LMSUP_SYM 'S' /* show element symbolic */
#define LMSUP_HEX 'H' /* show element alpha + hex */
#define LMSUP_INC_PRE 'P' /* increment with prefix */
#define LMSUP_INC_BASE 'B' /* increment with baset */
#define LMSUP_HIGH_PRE 'H' /* highest with prefix */
#define LMSUP_EXTRA 'X' /* Format-B records allowed */
/* ----- returncodes ----- */
#define LMSUP_OK '\0' /* OK */
#define LMSUP_TRUNC '\x04' /* REC TRUNC */
#define LMSUP_EOF '\x08' /* EOF (GET/TOC) */
#define LMSUP_LMSERR '\x0C' /* LMS ERROR */
#define LMSUP_PARERR '\x14' /* PARAMETER ERROR */
#define LMSUP_SEQERR '\x18' /* SEQUENCE ERROR */
#define LMSUP_INTERR '\x1C' /* LMS INTERNAL ERROR */
/* ----- storage form ----- */
#define LMSUP_FULL 'V' /* fully stored */
#define LMSUP_DELTA 'D' /* delta stored */
/* ----- conventions ----- */
#define LMSUP_CNONE 'N' /* none */
#define LMSUP_CSEQ 'S' /* std-sequence */
#define LMSUP_CMSEQ 'M' /* multi-sequence */
#define LMSUP_CTREE 'T' /* std-tree */
/* ----- protection indicators ----- */
#define LMSUP_PNONE 'N' /* none */
#define LMSUP_PSTD 'Y' /* standard protection */
#define LMSUP_PGUARD 'G' /* protection by guard */
/* ----- source code control ----- */
#define LMSUP_FREE '-' /* free */

```

```

#define LMSUP_INHOLD      'H'  /* in hold                                */
/* ----- write control ----- */
#define LMSUP_ACTIV      'A'  /* active                                */
#define LMSUP_DEACTIV    'D'  /* inactive                              */
/* ----- access date ----- */
#define LMSUP_KEEP       'K'  /* keep access date                      */
/* ----- modification date indicator ----- */
#define LMSUP_OLD        'O'  /* by-source                             */
#define LMSUP_SYS_DATE   'S'  /* new (system date)                    */
/* ----- */

#ifdef C_V1
extern void lmsup1 ();
#else
extern void lmsup1( struct lmsc_cb *cb, ...);
#endif

#ifdef __cplusplus
}
#endif

#endif

```

Include-Element LMSREC.H

Dieses Element enthält die Definitionen für den Aufbau spezieller Sätze (Satzart 163, 164, siehe [Abschnitt „Aufbau des Sekundärsatzes \(Satzart 163\)“ auf Seite 141](#) und [Abschnitt „Aufbau des Attributsatzes \(Satzart 164\)“ auf Seite 142](#)).

```

/* LMSREC.H    @>V3.1000 1995-04-26<@ */

#ifdef WAS_LMSREC
#else

/*****
/*                L M S R E C                *****/
/*****

#define WAS_LMSREC
#define BLANK4   ' ',' ',' ',' '
#define BLANK8   ' ',' ',' ',' ',' ',' ',' '
#define BLANK18  BLANK8,BLANK8,' ',' '
#define BLANK24  BLANK8,BLANK8,BLANK8

```

```

#define BLANK64 BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8,BLANK8
#define ZERO4  {'\0','\0','\0','\0'}
#define X0_4   {'\0','\0','\0','\0'}
#define X0_8   {'\0','\0','\0','\0','\0','\0','\0','\0'}
#define X0_32  X0_8,X0_8,X0_8,X0_8

/*****
/*          RECORD TYPE : 1 6 3          *****/
/*****
struct lmsc_rec163
{
    short rec_len;          /* record length          */
    char  filler1;         /* reserved              */
    char  rec_type;        /* identification of plam record */
    char  sec_name[32];    /* secondary name        */
    char  sec_attr[8];     /* secondary attribute    */
                                /* '0' : - CSECT          */
                                /* '1' : - ENTRY          */
    char  format_ind;     /* format indicator      */
    char  sec_name_long[1]; /* long secondary name: - 32K-45 */
};
#define LMSC_REC163_PROTO  \
{\
    0,          /* record length          */
    '\0',       /* reserved              */
    163,        /* identification of plam record */
    {BLANK24,BLANK8}, /* secondary name        */
    {BLANK8},   /* secondary attribute    */
    0,          /* format indicator      */
    ' '        /* long secondary name: - 32K-45 */
}
static struct lmsc_rec163 lmsc_rec163_proto = LMSC_REC163_PROTO;

/*****
/*          RECORD TYPE : 1 6 4          *****/
/*****
#define LMSREC_AIXMAX      30 /* max nr. of aix entries */
struct lmsc_rec164
{
    short rec_len;          /* record length          */
    char  filler1;         /* reserved              */
    char  rec_type;        /* identification of plam record */
    char  version;         /* version of specified PLAM record*/
    char  rec_num;         /* record describes format of
                           PLAM record of record type
                           with specified number          */
};

```

```

char filename[54]; /* file name taken from FCB */
char fcb; /* fcbtype (SET/RESET) */
char share; /* share (SET,RESET) */
char size[3]; /* file size < 32GB */
char sec_alloc[2]; /* secondary allocation (byte align)*/
char recform; /* recform (SET,RESET) */
short blksize; /* blksize */
short recsize; /* recsize */
short keypos; /* keypos */
char keylen; /* keylen */
char pad; /* pad */
char loglen; /* loglen */
char vallen; /* vallen */
char key_ind; /* does key exists in member */
char cfid[4]; /* cfid */
char blkctrl_ind; /* BLKCTRL-indicator */
char ioperf_ind; /* IOPERF-indicator */
char iousage_ind; /* IOusage-indicator */
char catalog_ind; /* catalog-indicator */
char filler2; /* reserved, must be 0 */
short a_ind_count; /* alternate index count */
char filesize_32[4]; /* file size >= 32GB,
                        filesize must be X'FFFFFF' */
char filler3[86]; /* reserved, must be 0 */
struct
{
    char keyname[8]; /* keyname */
    short keypos; /* keypos */
    char keylen; /* keylen */
    char indicator; /* dupkey=yes/no */
} aix[LMSREC_AIXMAX];
};
#define LMSC_AIX_PROTO {X0_8,0,'\0','\0'}
#define LMSC_REC164_PROTO \
{\
    540, /* record length */
    '\0', /* reserved */
    164, /* identification of plam record */
    '\2', /* version of specified PLAM record*/
    '\1', /* record describes format of \
            PLAM record of record type \
            with specified number */
    {BLANK18,BLANK18,BLANK18}, /* file name taken from FCB */
    '\0', /* fcbtype (SET/RESET) */
    '\0', /* share (SET,RESET) */
    {'\0','\0','\0'}, /* file size < 32GB */
    {'\0','\0'}, /* secondary allocation */
    '\0', /* recform (SET,RESET) */
    0, /* blksize */
};

```

```

0, /* recsize */
0, /* keypos */
'\0', /* keylen */
'\0', /* pad */
'\0', /* loglen */
'\0', /* vallen */
' ', /* does key exists in member */
ZER04, /* cfid */
'\0', /* BLKCTRL-indicator */
'\0', /* IOPERF-indicator */
'\0', /* IOusage-indicator */
'\0', /* catalog-indicator */
'\0', /* reserved, must be 0 */
0, /* alternate index count */
{'\0', '\0', '\0', '\0'}, /* file size >= 32GB */
{X0_32, X0_32, X0_8, X0_8, /* reserved, must be 0
X0_4, '\0', '\0'}, \
{LMSC_AIX_PROTO, LMSC_AIX_PROTO, LMSC_AIX_PROTO, \
LMSC_AIX_PROTO, LMSC_AIX_PROTO, \
LMSC_AIX_PROTO, LMSC_AIX_PROTO, LMSC_AIX_PROTO, \
LMSC_AIX_PROTO, LMSC_AIX_PROTO} \
}
static struct lmsc_rec164 lmsc_rec164_proto = LMSC_REC164_PROTO;

/* ----- fcctype (SET/RESET) ----- */
#define LMSREC_FTYPES '\xC0' /* R SAM */
#define LMSREC_FTYPEI '\x40' /* S ISAM */
#define LMSREC_FTYPEP '\xC0' /* S PAM */
#define LMSREC_VMIN '\x01' /* R VALPROP MIN. FUNCT. */
#define LMSREC_VMAX '\x01' /* S VALPROP MAX. FUNCT. */
/* ----- share (SET, RESET) ----- */
#define LMSREC_SHAREY '\x04' /* S YES */
#define LMSREC_ACCESS '\x08' /* S ACCESS=READ */
#define LMSREC_SHCCNO '\xC0' /* R NO CONTROL CHAR */
#define LMSREC_SHCCM '\x40' /* S MACHINE CODE CONTROL CHAR */
#define LMSREC_SHCCA '\xC0' /* S ASA CONTROL CHAR */
/* ----- recform (SET, RESET) ----- */
#define LMSREC_RECFF '\x04' /* S FIXED */
#define LMSREC_RECFV '\x02' /* S VARIABLE */
#define LMSREC_RECFU '\x06' /* S UNDEFINED */

```

```

/* ----- does key exists in member ? ----- */
#define LMSREC_KEYY      'Y'  /* YES */
#define LMSREC_KEYN      'N'  /* NO  */
/* ----- BLKCTRL-indicator ----- */
#define LMSREC_CTRLN     '\x80' /* S BLKCTRL=NO */
#define LMSREC_CTRLP     '\x40' /* S BLKCTRL=PAMKEY */
#define LMSREC_CTRLD     '\x20' /* S BLKCTRL=DATA */
#define LMSREC_CTRL0     '\x10' /* S BLKCTRL=NULL */
#define LMSREC_CTRLR     '\xF0' /* R BLKCTRL=NOT SPECIFIED */
#define LMSREC_BCF4K     '\x08' /* S BLOCK CONTROL FIELD 4K */
#define LMSREC_BCF2K     '\x04' /* S BLOCK CONTROL FIELD 2K */
#define LMSREC_CTRLU     '\x03' /* R -- RESERVED, MUST BE 0 -- */
/* ----- IOPERF-indicator ----- */
#define LMSREC_PFVH     '\x03' /* S IOPERF=VERY-HIGH */
#define LMSREC_PFHI     '\x02' /* S IOPERF=HIGH */
#define LMSREC_PFST     '\x01' /* S IOPERF=STD */
#define LMSREC_PFNS     '\x00' /* S IOPERF NOT SPECIFIED */
/* ----- IOUSAGE-indicator ----- */
#define LMSREC_USRW     '\x03' /* S IOUSAGE=RDWRT */
#define LMSREC_USWR     '\x02' /* S IOUSAGE=WRITE */
#define LMSREC_USRD     '\x01' /* S IOUSAGE=READ */
#define LMSREC_USNS     '\x00' /* S IOUSAGE NOT SPECIFIED */
/* ----- CATALOG-indicator (IDCEX) ----- */
#define LMSREC_ESPEC     '\x08' /* S PLAM FILE INDICATOR */
/* ----- BLKCTRL-indicator ----- */
#define LMSREC_AIXIDK     '\x80' /* S DUPKEY = YES */
                                /* R DUPKEY = NO */
/* ----- */
#endif

```

6.3 Beispiel

Das folgende C-Programm beinhaltet folgende Funktionen:

1. Eröffnen eines Unterprogrammzugriffs (INIT)
2. Aufnehmen einer Datei als Element (ADD)
3. Suchen eines Elementes in einem Inhaltsverzeichnis (TOCPRIM)
4. Eröffnen eines Elementes (OPENGET)
5. Satzweise Lesen eines Elementes (GET)
6. Schließen des Elementes (CLOSE)
7. Beenden des Unterprogrammzugriffs (END)

Zum besseren Verständnis wurden Kommentare in das Beispiel aufgenommen.

```

/* ***** */
/*
/*          BEISPIEL FUER LMS ALS UNTERPROGRAMM
/*
/* ***** */

#include <stdlib.h>
#include <stdio.h>
#include <string.h> /* include-Element für Kopierfunktionen u.a */
#include <lms.h>    /* include-Element für die LMS-Strukturen */

main()
{
    /* Bereitstellen und Initialisieren der Parameterstrukturen */

    struct lmsc_cb cb; /* Bereitstellen der Parameterstrukturen für cb */
    struct lmsc_ed ed; /* Bereitstellen der Parameterstrukturen für ed */
    struct lmsc_em em; /* Bereitstellen der Parameterstrukturen für em */
    struct lmsc_ei ei; /* Bereitstellen der Parameterstrukturen für ei */
    struct lmsc_fd fd; /* Bereitstellen der Parameterstrukturen für fd */
    struct lmsc_ld ld; /* Bereitstellen der Parameterstrukturen für ld */
    struct lmsc_rd rd; /* Bereitstellen der Parameterstrukturen für rd */

    int tid=1;          /* Initialisieren der TOC-Identifikation */
    char buffer[200];  /* Pufferlänge von ER; wird bei GET benötigt */
    char temp[100];    /* Puffer zum Umkopieren fuer Ausgabe */
    char * ptemp;      /* Zeiger in diesen Puffer */

    cb = lmsc_cb_proto; /* Kontrollblock CB initialisieren */
    ed = lmsc_ed_proto; /* Elementbeschreibung initialisieren */
    em = lmsc_em_proto; /* Elementmaske initialisieren */
    ei = lmsc_ei_proto; /* Elementinformation initialisieren */
    fd = lmsc_fd_proto; /* Dateiinformatio initialisieren */
    ld = lmsc_ld_proto; /* Bibliotheksbeschreibung initialisieren */
    rd = lmsc_rd_proto; /* Satzbeschreibung initialisieren */

    /* ***** */
    /*
    /* 1. Eröffnen des Unterprogrammzugriffs mit INIT
    /*
    /* ***** */

    cb.function = LMSUP_INIT ;
    cb.subcode = LMSUP_UNUSED ;
    lmsup1(&cb);

    /* Abfrage des Returncodes */

    if (cb.retcode != LMSUP_OK)
    {
        /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

```

```

        printf("Fehler beim Initialisieren\n");
        exit(1);
    }
    else
        printf("Initialisieren erfolgreich beendet \n");

/* ***** */
/*
/* 2. Aufnehmen eines Elementes mit ADD
/*
/*
/* ***** */

cb.function=LMSUP_ADD;
cb.subcode =LMSUP_UNUSED;
cb.overwrite =LMSUP_YES;

/*    Bereitstellen der benötigten Werte für ADD
/*
/* Es ist zu beachten, dass die zu kopierenden Argumente mit einem
/* Blank enden müssen, falls sie kürzer als das Zielfeld sind.

strfill(ld.name,"#BSP.LIB.C",sizeof(ld.name));
/* Bibliotheksname bereitstellen
strfill(fd.name,"#BSP.IN.INPUT",sizeof(fd.name));
/* Dateiname bereitstellen
strfill(ed.name,"BSP.ELEMENT",sizeof(ed.name));
/* Elementname bereitstellen
strfill(ed.version,"001",sizeof(ed.version));
/* Elementversion bereitstellen
strfill(ed.typ,"S",sizeof(ed.typ));
/* Elementtyp bereitstellen

/* Programmaufruf */

lmsup1(&cb,&fd,&ld,&ed);

/* Abfrage des Returncodes */

if (cb.retcode != LMSUP_OK)
{
    /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

    printf("Fehler beim Element aufnehmen \n");
    exit(1);
}
else
    printf("Element aufgenommen \n");

```

```

/* ***** */
/*
/* 3. Suchen eines Elementes im Inhaltsverzeichnis mit TOCPRIM
/*
/* ***** */

cb.function = LMSUP_TOCPRIM;
cb.subcode = LMSUP_LONG;

/*      Bereitstellen der benötigten Werte für TOCPRIM
/*
/* Es ist zu beachten, dass die zu kopierenden Argumente mit einem
/* Blank enden müssen, falls sie kürzer als das Zielfeld sind.

strfill(em.name,"BSP.ELEMENT",sizeof(em.name));
/* Elementname bereitstellen
strfill(em.version,"001",sizeof(em.version));
/* Elementversion bereitstellen
strfill(em.typ,"S",sizeof(em.typ));
/* Elementtyp bereitstellen

/* Programmaufruf */

lmsup1(&cb,&tid,&ei,&ld,&em);

/* Abfrage des Returncodes */

if (cb.retcode != LMSUP_OK)
{
    /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

    printf("Fehler beim Element suchen \n");
    exit(1);
}
else
{
    printf("Elementsuche erfolgreich:\n");

    strncpy(temp,ei.typ,sizeof(ei.typ));
    ptemp = index(temp,' ');
    if (ptemp) *ptemp = '\0';
    printf("typ          %s\n",temp);

    strncpy(temp,ei.name,sizeof(ei.name));
    ptemp = index(temp,' ');
    if (ptemp) *ptemp = '\0';
    printf("name          %s\n",temp);

    /* display Typ
    /* display Name
    /* display Version

```

```

        strncpy(temp,ei.version,sizeof(ei.version));
        ptemp = index(temp,' ');
        if (ptemp) *ptemp = '\\0';
        printf("version      %s\\n",temp);
                                           /* display Date      */
        strncpy(temp,ei.user_date,sizeof(ei.user_date));
        ptemp = index(temp,' ');
        if (ptemp) *ptemp = '\\0';
        printf("user-date   %s\\n\\n",temp);
    }

/* ***** */
/*                                               */
/* 4. Eröffnen eines Elementes mit OPENGET      */
/*                                               */
/* ***** */

cb.function = LMSUP_OPEN_GET;
cb.subcode  = LMSUP_UNUSED ;

/*      Bereitstellen der benötigten Werte für OPENGET      */
/*                                               */
/* Es ist zu beachten, dass die zu kopierenden Argumente mit einem */
/* Blank enden müssen, falls sie kürzer als das Zielfeld sind.   */

    strncpy(ed.name,"#BSP.LIB.C",sizeof(ed.name));
                                           /* Bibliotheksname bereitstellen */
    strncpy(ed.name,"BSP.ELEMENT",sizeof(ed.name));
                                           /* Elementname bereitstellen   */
    strncpy(ed.version,"001",sizeof(ed.version));
                                           /* Elementversion bereitstellen */
    strncpy(ed.typ,"S",sizeof(ed.typ));
                                           /* Elementtyp bereitstellen    */

/* Programmaufruf */

lmsup1(&cb,&rd,&ld,&ed);

/* Abfrage des Returncodes */

if (cb.retcode != LMSUP_OK)
{
    /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

    printf("Fehler beim Element oeffnen \\n");
    exit(1);
}

```

```
    }
    else
        printf("Element eroeffnet \n");

/* ***** */
/*                                             */
/* 5. Lesen eines Satzes mit GET in Schleife */
/*                                             */
/* ***** */

do
{
    cb.function = LMSUP_GET;
    cb.subcode = LMSUP_SEQ;

    /* Bereitstellen der benoetigten Werte fuer GET */

    rd.buffer_len = sizeof(buffer)-1;

    /* Programmaufruf */

    lmsupl(&cb,&rd,buffer);

    switch (cb.retcode) /* Abfrage des Returncodes */
    {
        case LMSUP_OK: /* Ausgeben des Satzes */
            buffer[rd.record_len]='\0';
            printf("%s\n",buffer+4);
            break;

        case LMSUP_TRUNC: /* Satz abgeschnitten */
            printf("Satzpuffer zu kurz\n");
            break;

        case LMSUP_EOF: /* Element Ende */
            break;

        default: /* Es ist ein Fehler aufgetreten - Meldung */
            printf("Fehler beim Satz lesen \n");
            break;
    }
}
while (cb.retcode == LMSUP_OK);
```

```
/* ***** */
/*                                             */
/* 6. Schließen eines Elementes mit CLOSE      */
/*                                             */
/* ***** */

    cb.function = LMSUP_CLOSE;
    cb.subcode = LMSUP_UNUSED;

    /* Programmaufruf */

    lmsup1(&cb,&rd);

    /* Abfrage des Returncodes */

    if (cb.retcode != LMSUP_OK)
    {
        /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

        printf("Fehler beim Element schließen \n");
        exit(1);
    }
    else
        printf("Element wieder geschlossen \n");

/* ***** */
/*                                             */
/* 7. Beenden des Unterprogrammzugriffs mit END */
/*                                             */
/* ***** */

    cb.function = LMSUP_END;
    cb.subcode = LMSUP_UNUSED ;

    /* Programmaufruf */

    lmsup1(&cb);

    /* Abfrage des Returncodes */

    if (cb.retcode != LMSUP_OK)
    {
        /* Es ist ein Fehler aufgetreten – Meldung und Programm beenden */

        printf("Fehler beim Beenden \n");
        exit(1);
    }
}
```

```
        else
            printf("Unterprogrammzugriff beendet \n");

    } /* Ende des Hauptprogramms */
```

7 Assembler-Schnittstelle

7.1 Der Verbindungsmodul LMSUP1

Zum Aufruf der LMS-Funktionen aus Assembler-Programmen steht der Modul LMSUP1 zur Verfügung. Sein Aufruf erfolgt über:

Name	Operation	Operanden
	L BALR	15,=V(LMSUP1) 14,15

Für die Einträge in die Parameterliste siehe Beschreibung der Funktionen ab [Seite 17](#).

Der Modul LMSUP1 wird aus der Bibliothek SYSLNK.LMS.034 zum Hauptprogramm hinzugebunden.

Beim Unterprogrammprung nach LMS müssen die Register wie folgt geladen sein:

Register 1: Adresse der Parameterliste

Register 13: Adresse der Savearea

Register 14: Rücksprungadresse

Register 15: Einsprungadresse

Parameterliste

Die Parameterliste enthält eine Folge von Adressen, die vom Rufer angegeben werden müssen. Die für einen Aufruf relevanten Adressen hängen von der aufgerufenen Funktion und den dabei verlangten Kontrollblöcken ab.

Savearea

Die Savearea ist ein Register-Sicherstellungsbereich aus 18 Worten, der vom Rufer zur Verfügung gestellt werden muss.

Rücksprungadresse

An diese Adresse im Hauptprogramm springt LMS zurück, nachdem eine Funktion ausgeführt wurde.

Einsprungadresse:

LMSUP1

7.2 Generierung der Parameterstrukturen für Assembler

Um die Parameterstrukturen für Assembler generieren zu können, muss zunächst die Bibliothek SYSLIB.LMS.034, in der sich die Makros befinden, zugewiesen werden. Die Zuweisung kann mit folgendem Kommando erfolgen:

```
/ADD-FILE-LINK FILE-NAME = $.SYSLIB.LMS.034, LINK-NAME = ALTLIB
```

Der Installationsort dieser Bibliothek ist über IMON frei wählbar.

Mit Hilfe der Built-In-Funktion INSTALLATION-PATH kann der Installationsort von SYSLIB.LMS.034 ermittelt und in eine S-Variable gespeichert werden:

```
/SET-VARIABLE LIBRARY-NAME =INSTALLATION-PATH           -  
                    (LOGICAL-ID = `SYSLIB`             -  
                    ,INSTALLATION-UNIT = `LMS`         -  
                    ,VERSION = `3.4`                  -  
                    ,DEFAULT-PATH-NAME = `$.SYSLIB.LMS.034`)
```

Anschließend kann die Bibliothek folgendermaßen zugewiesen werden:

```
/ADD-FILE-LINK FILE-NAME=&(LIBRARY-NAME), LINK-NAME=ALTLIB
```

Für die Parameterstrukturen TID und ER stehen keine Makros zur Verfügung.

LMSASSCB

LMSASSCB generiert den Funktionskontrollblock.

Name	Operation	Operanden
name	LMSASSCB	[P=prefix,] MF = $\left\{ \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge CB.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSCB

```

*****
*
*          FUNCTION CONTROL BLOCK (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.SCBV  DS    CL2          VERSION OF INTERFACE MACROS  IN  CL2'04'
&P.FUNC  DS    XL1          FUNCTION CODE                   IN  XL1'01'
&P.SUBC  DS    CL1          SUBCODE                               IN  CL1' '
&P.ACC   DS    F            INIT ACCESS ID                       INOUT X'FFFFFFFF'
&P.RTC   DS    XL1          MAIN RETURNCODE                       OUT  XL1'00'
&P.RES1  DS    XL1          FREE                                  XL1'00'
&P.LMSM  DS    H            LMS-MSG-CODE                          OUT  H'0'
&P.DMSM  DS    H            DMS-MSG-CODE                          OUT  H'0'
&P.PLAM  DS    H            PLAM-MSG-CODE                         OUT  H'0'
&P.LMSV  DS    CL12         LMS VERSION                           OUT  CL12' '
&P.RES2  DS    XL4          FREE                                  XL4'00'
*
```

```
*****
LMS PARAMETER *****
*
&P.DEST DS CL1 DESTROY INOUT CL1 ' '
&P.FCB DS CL1 FCBTYP E INOUT CL1 ' '
&P.KEY DS CL1 KEY INOUT CL1 ' '
&P.OV DS CL1 OVERWRITE INOUT CL1 ' '
&P.COL DS H COLUMN INOUT H'O'
&P.LINE DS H LINE INOUT H'O'
&P.PI DS CL1 PROTECTION INDICATOR INOUT CL1 ' '
&P.ATTR DS CL1 COPYLIB WITH ATTRIBUTES INOUT CL1 ' '
&P.INFO DS CL1 RECORD TYPE FOR .... INOUT CL1 ' '
&P.LDRT DS CL1 DMS-FILENAME IN LD INOUT CL1 ' '
&P.RES3 DS CL4 FREE CL4 ' '
&P.PLNG EQU *-&P.SCBV CB LENGTH
```

LMSASSEA

LMSASSEA generiert die Elementattribute.

Name	Operation	Operanden
name	LMSASSEA	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge EA.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSEA

```

*****
*
*          ELEMENT ATTRIBUTES (DSECT)
*
*****
SPACE
&NAME      DSECT
&P.UDAT   DS   CL14      USER DATE                IN  CL14' '
&P.UTIM   DS   CL8       USER TIME                 IN  CL8'  '
&P.CCSN   DS   CL8       CODED CHARACTER SET NAME   IN  CL8'  '
*
*****          SOURCE CODE CONTROL          *****
*
&P.HOSTA  DS   CL1       HOLD FLAG                   IN  CL1'  '
.*
.*        ' ': UNCHANGE
.*
.*        '-': FREE
.*
&P.RES1   DS   CL8       'H': IN HOLD
.*        FREE (HOLDER USERID)             IN  CL8'  '
*
&P.MDATI  DS   CL1       MODIFICATION DATE INDICATOR IN  CL1'0'
.*
.*        'O': OLD
.*
.*        'S': SYSTEM DATE
&P.RES2   DS   CL56      FREE                               CL56'  '
*
&P.PLNG   EQU   *-&P.UDAT      EA LENGTH

```

LMSASSED

LMSASSED generiert die Elementbeschreibung.

Name	Operation	Operanden
name	LMSASSED	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge ED.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke
verwendet werden können.

Auflösung von LMSASSED

```

*****
*
*          ELEMENT DESCRIPTION (DSECT)
*
*****
          SPACE
&NAME   DSECT
&P.ELEM DS   OCL96      ELEMENT IDENTIFIER      IN
&P.TYPE DS    CL8       ELEMENT TYPE             IN CL8' '
&P.NAME DS   CL64      ELEMENT NAME              IN CL64' '
&P.VERS DS   CL24      ELEMENT VERSION           IN CL24' '
&P.STOR DS   CL1       STORAGE FORM             IN CL1'V'
&P.UDAT DS   CL14      USER DATE                 IN CL14' '
&P.UTIM DS   CL8       USER TIME                 IN CL8' '
&P.PLNG EQU  *-&P.ELEM  ED LENGTH

```

LMSASSEI

LMSASSEI generiert die Elementinformation.

Name	Operation	Operanden
name	LMSASSEI	[P=prefix,] MF = $\left. \begin{matrix} D \\ C \\ L \end{matrix} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge EI.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSEI

```

*****
*
*           ELEMENT INFORMATION (DSECT)
*
*****
SPACE
&NAME     DSECT
&P.ELEM   DS    0CL96      ELEMENT IDENTIFIER      OUT
&P.TYPE   DS     CL8       ELEMENT TYPE             OUT CL8' '
&P.NAME   DS    CL64       ELEMENT NAME             OUT CL64' '
&P.VERS   DS    CL24       ELEMENT VERSION           OUT CL24' '
&P.STOR   DS     CL1       STORAGE FORM                 OUT CL1' '
&P.UDAT   DS    CL14       USER DATE                   OUT CL14' '
&P.UTIM   DS     CL8       USER TIME                     OUT CL8' '
&P.CDAT   DS    CL14       CREATION DATE                 OUT CL14' '
&P.CTIM   DS     CL8       CREATION TIME                  OUT CL8' '
&P.MDAT   DS    CL14       MODIFICATION DATE          OUT CL14' '
&P.MTIM   DS     CL8       MODIFICATION TIME             OUT CL8' '
&P.SECN   DS    CL32       SECONDARY NAME                 OUT CL32' '
&P.SECA   DS     CL8       SECONDARY ATTRIBUTE           OUT CL8' '
&P.RES1   DS     CL5       FREE                           OUT CL5' '
*

```

```

*****
*
***** PROTECTION ATTRIBUTES *****
*
&P.PTRD DS CL1 PROT. TYPE INDI. FOR READ OUT CL1' '
.* 'N': NO SPECIAL PROTECTION
.* 'Y': SPECIAL PROTECTION
.* 'G': PROTECTION BY GUARD
&P.PRDU DS CL1 READ INDICATOR FOR OWNER OUT CL1' '
&P.PRDG DS CL1 READ INDICATOR FOR GROUP OUT CL1' '
&P.PRDO DS CL1 READ INDICATOR FOR OTHERS OUT CL1' '
&P.PRDP DS CL1 READ PASSWORD INDICATOR OUT CL1' '
&P.RES2 DS FL4 FL4'0'
*
&P.PTWR DS CL1 PROT. TYPE INDI. FOR WRITE OUT CL1' '
&P.PWRU DS CL1 WRITE INDICATOR FOR OWNER OUT CL1' '
&P.PWRG DS CL1 WRITE INDICATOR FOR GROUP OUT CL1' '
&P.PWRO DS CL1 WRITE INDICATOR FOR OTHERS OUT CL1' '
&P.PWRP DS CL1 WRITE PASSWORD INDICATOR OUT CL1' '
&P.RES3 DS FL4 FL4'0'
*
&P.PTEX DS CL1 PROT. TYPE INDI. FOR EXEC OUT CL1' '
&P.PEXU DS CL1 EXEC INDICATOR FOR OWNER OUT CL1' '
&P.PEXG DS CL1 EXEC INDICATOR FOR GROUP OUT CL1' '
&P.PEXO DS CL1 EXEC INDICATOR FOR OTHERS OUT CL1' '
&P.PEXP DS CL1 EXEC PASSWORD INDICATOR OUT CL1' '
&P.RES4 DS FL4 FL4'0'
*
&P.PGRD DS CL18 READ GUARD OUT CL18' '
&P.PGWR DS CL18 WRITE GUARD OUT CL18' '
&P.PGEX DS CL18 EXEC GUARD OUT CL18' '
&P.CCSN DS CL8 CODED CHARACTER SET NAME OUT CL8' '
*
***** SOURCE CODE CONTROL *****
*
&P.PTHO DS CL1 IND. FOR HOLDER-AUTHORIZATION UT CL1' '
.* 'N': NO SPECIAL AUTHORIZATION
.* 'Y': SPECIAL AUTHORIZATION
.* 'G': AUTHORIZATION BY GUARD
&P.PHOU DS CL1 HOLDER INDICATOR FOR OWNER OUT CL1' '
&P.PHOG DS CL1 HOLDER INDICATOR FOR GROUP OUT CL1' '
&P.PHOO DS CL1 HOLDER INDICATOR FOR OTHERS OUT CL1' '
&P.PHOP DS CL1 HOLDER PASSWORD INDICATOR OUT CL1' '
&P.RES5 DS FL4 FL4'0'
&P.PGHO DS CL18 HOLDER GUARD OUT CL18' '
*
&P.HOSTA DS CL1 HOLD FLAG OUT CL1' '
.* '-' : FREE
.* 'H' : IN HOLD
&P.HOLD DS CL8 HOLDER USERID OUT CL8' '
&P.ADAT DS CL14 ACCESS DATE OUT CL14' '
&P.ATIM DS CL8 ACCESS TIME OUT CL8' '
*
&P.RES6 DS CL1 CL1' '

```

&P.ESIZE DS	F	ELEMENT-SIZE	OUT	FL4'0'
*				
&P.RES7 DS	CL40	FREE		CL40' '
&P.PLNG EQU	*-&P.ELEM	EI LENGTH		

LMSASSEM

LMSASSEM generiert die Elementmaske.

Name	Operation	Operanden
name	LMSASSEM	[P=prefix,] MF = $\left. \begin{matrix} D \\ C \\ L \end{matrix} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge EM.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke
verwendet werden können.

Auflösung von LMSASSEM

```

*****
*
*          ELEMENT MASK (DSECT)
*
*****
      SPACE
&NAME  DSECT
&P.TYPE DS   CL20      ELEMENT TYPE      IN  CL20' '
&P.NAME DS  CL132     ELEMENT NAME      IN  CL132' '
&P.VERS DS   CL52     ELEMENT VERSION   IN  CL52' '
&P.STOR DS   CL6      STORAGE FORM        IN  CL6' '
&P.UDAT DS   CL32     USER DATE           IN  CL32' '
&P.UTIM DS   CL20     USER TIME           IN  CL20' '
&P.CDAT DS   CL32     CREATION DATE        IN  CL32' '
&P.CTIM DS   CL20     CREATION TIME        IN  CL20' '
&P.MDAT DS   CL32     MODIFICATION DATE    IN  CL32' '
&P.MTIM DS   CL20     MODIFICATION TIME    IN  CL20' '
&P.SECN DS   CL68     SECONDARY NAME       IN  CL68' '
&P.SECA DS   CL20     SECONDARY ATTRIBUTE  IN  CL20' '
&P.RES1 DS   CL14     FREE                  CL14' '
*
    
```

```

*****
*
***** PROTECTION ATTRIBUTES *****
*
&P.PTRD DS CL1 PROT. TYPE INDICATOR FOR READ IN CL1' '
.* 'A': ANY PROTECTION
.* 'N': NO SPECIAL PROTECTION
.* 'Y': SPECIAL PROTECTION
.* 'G': PROTECTION BY GUARD
&P.PRDU DS CL1 READ INDICATOR FOR OWNER IN CL1' '
&P.PRDG DS CL1 READ INDICATOR FOR GROUP IN CL1' '
&P.PRDO DS CL1 READ INDICATOR FOR OTHERS IN CL1' '
&P.PRDP DS CL1 READ PASSWORD INDICATOR IN CL1' '
&P.RES2 DS FL4 FL4'O'
*
&P.PTWR DS CL1 PROT. TYPE IND. FOR WRITE IN CL1' '
&P.PWRU DS CL1 WRITE INDICATOR FOR OWNER IN CL1' '
&P.PWRG DS CL1 WRITE INDICATOR FOR GROUP IN CL1' '
&P.PWRO DS CL1 WRITE INDICATOR FOR OTHERS IN CL1' '
&P.PWRP DS CL1 WRITE PASSWORD INDICATOR IN CL1' '
&P.RES3 DS FL4 FL4'O'
*
&P.PTEX DS CL1 PROT. TYPE INDICATOR FOR EXEC IN CL1' '
&P.PEXU DS CL1 EXEC INDICATOR FOR OWNER IN CL1' '
&P.PEXG DS CL1 EXEC INDICATOR FOR GROUP IN CL1' '
&P.PEXO DS CL1 EXEC INDICATOR FOR OTHERS IN CL1' '
&P.PEXP DS CL1 EXEC PASSWORD INDICATOR IN CL1' '
&P.RES4 DS FL4 FL4'O'
*
&P.PGRD DS CL40 READ GUARD IN CL40' '
&P.PGWR DS CL40 WRITE GUARD IN CL40' '
&P.PGEX DS CL40 EXEC GUARD IN CL40' '
&P.CCSN DS CL20 CODED CHARACTER SET NAME IN CL20' '
*
***** SOURCE CODE CONTROL *****
*
&P.PTHO DS CL1 IND. FOR HOLDER-AUTHORIZATION IN CL1' '
.* 'N': NO SPECIAL AUTHORIZATION
.* 'Y': SPECIAL AUTHORIZATION
.* 'G': AUTHORIZATION BY GUARD
&P.PHOU DS CL1 HOLDER INDICATOR FOR OWNER IN CL1' '
&P.PHOG DS CL1 HOLDER INDICATOR FOR GROUP IN CL1' '
&P.PHOO DS CL1 HOLDER INDICATOR FOR OTHERS IN CL1' '
&P.PHOP DS CL1 HOLDER PASSWORD INDICATOR IN CL1' '
&P.RES5 DS FL4 FL4'O'
&P.PGHO DS CL40 HOLDER GUARD IN CL40' '
*
&P.HOSTA DS CL1 HOLD FLAG IN CL1' '
.* ' ': ANY
.* '-': FREE
.* 'H': IN HOLD
&P.HOLD DS CL20 HOLDER USERID IN CL20' '
&P.ADAT DS CL32 ACCESS DATE IN CL32' '
&P.ATIM DS CL20 ACCESS TIME IN CL20' '

```

*							
&P.RES6	DS	CL3					CL3' '
&P.ESMIN	DS	F	ELEMENT-SIZE	MINIMUM		IN	FL4'0'
&P.ESMAX	DS	F	ELEMENT-SIZE	MAXIMUM		IN	X'FFFFFFFF'
*							
&P.RES7	DS	CL64		FREE			CL64' '
&P.PLNG	EQU	*-&P.TYPE		EM LENGTH			

LMSASSFD

LMSASSFD generiert die Dateibeschreibung.

Name	Operation	Operanden
name	LMSASSFD	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge FD.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke
verwendet werden können.

Auflösung von LMSASSFD

```
*****
*
*          FILE DESCRIPTION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.PSWD  DS      F           DMS PASSWORD           IN  F'0'
&P.LINK  DS      CL8        LINK NAME              IN  CL8' '
&P.NAME  DS      CL54       FILE NAME              INOUT CL54' '
&P.PLNG  EQU     *-&P.PSWD  FD LENGTH
```

LMSASSLA

LMSASSLA generiert das Administrationsrecht für die Bibliothek.

Name	Operation	Operanden
name	LMSASSLA	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge LA.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSLA

```

*****
*
*          LIBRARY ATTRIBUTES (DSECT)
*
*****
          SPACE
&NAME   DSECT
&P.PTAD DS   CL1          PROT. TYPE INDI. FOR ADMIN   IN  CL1' '
.*
.*
.*
&P.PADU DS   CL1          ADMIN INDICATOR FOR OWNER   IN  CL1' '
&P.PADG DS   CL1          ADMIN INDICATOR FOR GROUP   IN  CL1' '
&P.PADO DS   CL1          ADMIN INDICATOR FOR OTHERS  IN  CL1' '
&P.PADP DS   CL1          ADMIN PASSWORD INDICATOR   IN  CL1' '
&P.PADW DS   FL4          ADMIN PASSWORD               IN  FL4'O'
&P.PGAD DS   CL18        ADMIN GUARD                   IN  CL18' '
*
&P.STOR DS   CL1          STORAGE FORM FOR LIBRARY   IN  CL1' '
.*
.*
.*
.*
&P.WRCT DS   CL1          WRITE-CONTROL FOR LIBRARY   IN  CL1' '
.*
.*
.*
&P.ADAT DS   CL1          ACCESS DATE                   IN  CL1' '
.*
.*
.*
&P.RES1 DS   CL34        FREE                           CL34' '
&P.PLNG EQU   *-&P.PTAD  LA LENGTH

```

LMSASSLD

LMSASSLD generiert die Bibliotheksbeschreibung.

Name	Operation	Operanden
name	LMSASSLD	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge LD.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke
verwendet werden können.

Auflösung von LMSASSLD

```
*****
*
*          LIBRARY DESCRIPTION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.PSWD  DS    F           DMS PASSWORD           IN  F'0'
&P.LINK  DS   CL8         LINK NAME           IN  CL8' '
&P.RES1  DS   2F         FREE
&P.LEN   DS    Y         MAX. LENGTH OF LIBRARY NAME IN  Y(54)
&P.NAME  DS   CL54       LIBRARY NAME           INOUT CL54' '
&P.PLNG  EQU   *-&P.PSWD LD LENGTH
```

LMSASSLI

LMSASSLI generiert die Bibliotheksinformation.

Name	Operation	Operanden
name	LMSASSL	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge LI.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSLI

```

*****
*
*          LIBRARY INFORMATION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.PTAD  DS    CL1          PROT. TYPE INDI. FOR ADMIN   OUT  CL1' '
.*
.*          'N': NO SPECIAL PROTECTION
.*          'Y': SPECIAL PROTECTION
.*          'G': PROTECTION BY GUARD
&P.PADU  DS    CL1          ADMIN INDICATOR FOR OWNER   OUT  CL1' '
&P.PADG  DS    CL1          ADMIN INDICATOR FOR GROUP   OUT  CL1' '
&P.PADO  DS    CL1          ADMIN INDICATOR FOR OTHERS  OUT  CL1' '
&P.PADP  DS    CL1          ADMIN PASSWORD INDICATOR    OUT  CL1' '
&P.RES1  DS    FL4          FL4'O'
&P.PGAD  DS    CL18        ADMIN GUARD                OUT  CL18' '
*
&P.STOR  DS    CL1          STORAGE FORM FOR LIBRARY   OUT  CL1' '
.*
.*          'S': STD ( FULL OR DELTA )
.*          'V': FULL ELEMENT
.*          'D': DELTA ELEMENT
&P.WRCT  DS    CL1          WRITE-CONTROL FOR LIBRARY   OUT  CL1' '
.*
.*          'A': ACTIVE
.*          'D': DEACTIVATED
&P.ADAT  DS    CL1          ACCESS DATE                OUT  CL1' '

```

```

.*          'N': NONE (DONT KEEP)
.*          'K': KEEP
&P.RES2 DS CL24 FREE CL24' '
&P.LFORM DS CL1 LIBRARY FORMAT (NK2/NK4) OUT CL1' '
*          '2' : NK2 LIBRARY FORMAT
*          '4' : NK4 LIBRARY FORMAT
&P.UPROT DS CL1 UPAM PROTECTED (YES/NO) OUT CL1' '
*          'Y' : LIB IS UPAM PROTECTED
*          'N' : LIB IS NOT UPAM PROTECTED
&P.FILSZ DS F FILE SIZE OUT F'0'
&P.FRESZ DS F FREE SIZE OUT F'0'
*
***** PROTECTION ATTRIBUTES *****
*
&P.PTRD DS CL1 PROT. TYPE INDI. FOR READ OUT CL1' '
&P.PRDU DS CL1 READ INDICATOR FOR OWNER OUT CL1' '
&P.PRDG DS CL1 READ INDICATOR FOR GROUP OUT CL1' '
&P.PRDO DS CL1 READ INDICATOR FOR OTHERS OUT CL1' '
&P.PRDP DS CL1 READ PASSWORD INDICATOR OUT CL1' '
&P.RES3 DS FL4 FL4'0'
*
&P.PTWR DS CL1 PROT. TYPE INDI. FOR WRITE OUT CL1' '
&P.PWRU DS CL1 WRITE INDICATOR FOR OWNER OUT CL1' '
&P.PWRG DS CL1 WRITE INDICATOR FOR GROUP OUT CL1' '
&P.PWRO DS CL1 WRITE INDICATOR FOR OTHERS OUT CL1' '
&P.PWRP DS CL1 WRITE PASSWORD INDICATOR OUT CL1' '
&P.RES4 DS FL4 FL4'0'
*
&P.PTEX DS CL1 PROT. TYPE INDI. FOR EXEC OUT CL1' '
&P.PEXU DS CL1 EXEC INDICATOR FOR OWNER OUT CL1' '
&P.PEXG DS CL1 EXEC INDICATOR FOR GROUP OUT CL1' '
&P.PEXO DS CL1 EXEC INDICATOR FOR OTHERS OUT CL1' '
&P.PEXP DS CL1 EXEC PASSWORD INDICATOR OUT CL1' '
&P.RES5 DS FL4 FL4'0'
*
&P.PGRD DS CL18 READ GUARD OUT CL18' '
&P.PGWR DS CL18 WRITE GUARD OUT CL18' '
&P.PGEX DS CL18 EXEC GUARD OUT CL18' '
*
&P.PTHO DS CL1 IND. FOR HOLDER-AUTHORIZATION CL1' '
.*          'N': NO SPECIAL AUTHORIZATION
.*          'Y': SPECIAL AUTHORIZATION
.*          'G': AUTHORIZATION BY GUARD
&P.PHOU DS CL1 HOLDER INDICATOR FOR OWNER OUT CL1' '
&P.PHOG DS CL1 HOLDER INDICATOR FOR GROUP OUT CL1' '
&P.PHOO DS CL1 HOLDER INDICATOR FOR OTHERS OUT CL1' '
&P.PHOP DS CL1 HOLDER PASSWORD INDICATOR OUT CL1' '
&P.RES6 DS FL4 FL4'0'
&P.PGHO DS CL18 HOLDER GUARD OUT CL18' '
*
&P.RES7 DS CL68 FREE CL68' '
&P.PLNG EQU *-&P.PTAD LI LENGTH

```

LMSASSPA

LMSASSPA generiert das Verwaltungsrecht und die Standardwerte für den Elementschutz für die Bibliothek.

Name	Operation	Operanden
name	LMSASSPA	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge PA.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSPA

```

*****
*
*      PROTECTION ATTRIBUTES (DSECT)
*
*****
      SPACE
&NAME  DSECT
&P.PTRD DS  CL1      PROT. TYPE INDI. FOR READ      IN  CL1' '
.*
.*
.*
&P.PRDU DS  CL1      READ INDICATOR FOR OWNER      IN  CL1' '
&P.PRDG DS  CL1      READ INDICATOR FOR GROUP      IN  CL1' '
&P.PRDO DS  CL1      READ INDICATOR FOR OTHERS      IN  CL1' '
&P.PRDP DS  CL1      READ PASSWORD INDICATOR      IN  CL1' '
&P.PRDW DS  FL4      READ PASSWORD                      IN  FL4'O'
*
&P.PTWR DS  CL1      PROT. TYPE INDI. FOR WRITE      IN  CL1' '
&P.PWRU DS  CL1      WRITE INDICATOR FOR OWNER      IN  CL1' '
&P.PWRG DS  CL1      WRITE INDICATOR FOR GROUP      IN  CL1' '
&P.PWRO DS  CL1      WRITE INDICATOR FOR OTHERS      IN  CL1' '
&P.PWRP DS  CL1      WRITE PASSWORD INDICATOR      IN  CL1' '
&P.PWRW DS  FL4      WRITE PASSWORD                      IN  FL4'O'
*
&P.PTEX DS  CL1      PROT. TYPE INDI. FOR EXEC      IN  CL1' '

```

&P.PEXU	DS	CL1	EXEC INDICATOR FOR OWNER	IN	CL1' '
&P.PEXG	DS	CL1	EXEC INDICATOR FOR GROUP	IN	CL1' '
&P.PEXO	DS	CL1	EXEC INDICATOR FOR OTHERS	IN	CL1' '
&P.PEXP	DS	CL1	EXEC PASSWORD INDICATOR	IN	CL1' '
&P.PEXW	DS	FL4	EXEC PASSWORD	IN	FL4'0'
*					
&P.PGRD	DS	CL18	READ GUARD	IN	CL18' '
&P.PGWR	DS	CL18	WRITE GUARD	IN	CL18' '
&P.PGEX	DS	CL18	EXEC GUARD	IN	CL18' '
*					
*****		SOURCE CODE CONTROL	*****		
*					
&P.PTHO	DS	CL1	IND. FOR HOLDER-AUTHORIZATION	IN	CL1' '
.*			'N': NO SPECIAL AUTHORIZATION		
.*			'Y': SPECIAL AUTHORIZATION		
.*			'G': AUTHORIZATION BY GUARD		
&P.PHOU	DS	CL1	HOLDER INDICATOR FOR OWNER	IN	CL1' '
&P.PHOG	DS	CL1	HOLDER INDICATOR FOR GROUP	IN	CL1' '
&P.PHOO	DS	CL1	HOLDER INDICATOR FOR OTHERS	IN	CL1' '
&P.PHOP	DS	CL1	HOLDER PASSWORD INDICATOR	IN	CL1' '
&P.PHOW	DS	FL4	HOLDER PASSWORD	IN	FL4'0'
&P.PGHO	DS	CL18	HOLDER GUARD	IN	CL18' '
&P.RES1	DS	CL84	FREE		CL84' '
&P.PLNG	EQU	*-&P.PTRD	PA LENGTH		

LMSASSRD

LMSASSRD generiert die Satzbeschreibung.

Name	Operation	Operanden
name	LMSASSRD	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge RD.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSRD

```

*****
*
*          RECORD DESCRIPTION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.ACC   DS    F          READ ACCESS ID      INOUT  X'FFFFFFFF'
&P.BLEN  DS    F          BUFFER LENGTH      IN    F'0'
&P.RLEN  DS    F          RECORD LENGTH      INOUT  F'0'
&P.RES1  DS    XL3        FREE                XL3'000000'
&P.RECID DS    OXL5       RECORD ID          INOUT
&P.RTYP  DS    XL1        RECORD TYPE       INOUT  XL1'01'
&P.RNUM  DS    F          RECORD NUMBER      INOUT  F'0'
&P.RES2  DS    2F        FREE                2F'0'
&P.PLNG  EQU    *-&P.ACC  RD LENGTH

```

LMSASSTA

LMSASSTA generiert die Typattribute.

Name	Operation	Operanden
name	LMSASSTA	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge TA.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSTA

```

*****
*
*          TYPE ATTRIBUTES (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.CONV  DS    CL1          TYPE CONVENTION          IN  CL1' '
.*
.*          'N' - NONE
.*          'S' - STD-SEQUENCE
.*          'M' - MULTI-SEQUENCE
.*          'T' - STD-TREE
&P.RES1  DS    CL3          FREE                      CL3' '
&P.EXPL  DS    CL24         VERSION EXAMPLE (F. SEQ.)  IN  CL24' '
*
&P.PTAD  DS    CL1          PROT. TYPE INDI. FOR ADMIN  IN  CL1' '
.*          'N': NO SPECIAL PROTECTION
.*          'Y': SPECIAL PROTECTION
.*          'G': PROTECTION BY GUARD
&P.PADU  DS    CL1          ADMIN INDICATOR FOR OWNER   IN  CL1' '
&P.PADG  DS    CL1          ADMIN INDICATOR FOR GROUP   IN  CL1' '
&P.PADO  DS    CL1          ADMIN INDICATOR FOR OTHERS  IN  CL1' '
&P.PADP  DS    CL1          ADMIN PASSWORD INDICATOR     IN  CL1' '
&P.PADW  DS    FL4          ADMIN PASSWORD                   IN  FL4'0'
&P.PGAD  DS    CL18         ADMIN GUARD                      IN  CL18' '
*

```

&P.STOR	DS	CL1	STORAGE FORM FOR LIBRARY	IN	CL1' '
.*			' ': UNCHANGE		
.*			'N': NONE		
.*			'S': STD (FULL OR DELTA)		
.*			'V': FULL ELEMENT		
.*			'D': DELTA ELEMENT		
&P.WRCT	DS	CL1	WRITE-CONTROL FOR LIBRARY	IN	CL1' '
.*			' ': UNCHANGE		
.*			'N': NONE		
.*			'A': ACTIVATE		
.*			'D': DEACTIVATE		
&P.STYP	DS	CL8	SUPER TYPE	IN	CL8' '
*					
&P.RES2	DS	CL47	FREE		CL47' '
&P.PLNG	EQU	*-&P.CONV	TA LENGTH		

LMSASSTD

LMSASSTD generiert den Elementtyp.

Name	Operation	Operanden
name	LMSASSTD	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardwert ist die Zeichenfolge TD.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke
verwendet werden können.

Auflösung von LMSASSTD

```

*****
*
*          TYPE DESCRIPTION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.TYPE  DS    CL8          ELEMENT TYPE      IN  CL8' '
&P.RES1  DS    CL8          FREE                CL8' '
&P.PLNG  EQU   *-&P.TYPE    TD LENGTH

```

LMSASSTI

LMSASSTI generiert die Typinformation.

Name	Operation	Operanden
name	LMSASSTI	[P=prefix,] MF = $\left. \begin{array}{c} D \\ C \\ L \end{array} \right\}$

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen. Standardwert ist die Zeichenfolge TI.

D Ein Pseudoabschnitt (DSECT) wird generiert.

C Ein Speicherbereich wird generiert (ohne CSECT-Anweisung).

L Es werden Konstanten definiert, die für die Initialisierung der Kontrollblöcke verwendet werden können.

Auflösung von LMSASSTI

```

*****
*
*          TYPE INFORMATION (DSECT)
*
*****
          SPACE
&NAME    DSECT
&P.TYPE  DS    CL8          ELEMENT TYPE          OUT  CL8' '
&P.RES1  DS    CL8          FREE                      CL8' '
*
&P.CONV  DS    CL1          TYPE CONVENTION        OUT  CL1' '
.*
.*          'N' - NONE
.*          'S' - STD-SEQUENCE
.*          'M' - MULTI-SEQUENCE
.*          'T' - STD-TREE
&P.RES2  DS    CL3          FREE                      CL3' '
&P.EXPL  DS    CL24         VERSION EXAMPLE (F. SEQ.) OUT  CL24' '
*
&P.PTAD  DS    CL1          PROT. TYPE INDI. FOR ADMIN OUT  CL1' '
.*
.*          'N': NO SPECIAL PROTECTION
.*          'Y': SPECIAL PROTECTION
.*          'G': PROTECTION BY GUARD
&P.PADU  DS    CL1          ADMIN INDICATOR FOR OWNER  OUT  CL1' '
&P.PADG  DS    CL1          ADMIN INDICATOR FOR GROUP  OUT  CL1' '
&P.PADO  DS    CL1          ADMIN INDICATOR FOR OTHERS OUT  CL1' '
&P.PADP  DS    CL1          ADMIN PASSWORD INDICATOR  OUT  CL1' '
&P.RES3  DS    FL4
          FL4'0'

```

&P.PGAD	DS	CL18	ADMIN GUARD	OUT	CL18'	'
*						
&P.STOR	DS	CL1	STORAGE FORM FOR TYPE	OUT	CL1'	'
.*			'N': NONE			
.*			'S': STD (FULL OR DELTA)			
.*			'V': FULL ELEMENT			
.*			'D': DELTA ELEMENT			
&P.WRCT	DS	CL1	WRITE-CONTROL FOR TYPE	OUT	CL1'	'
.*			'N': NONE			
.*			'A': ACTIVE			
.*			'D': DEACTIVATED			
&P.STYP	DS	CL8	SUPER TYPE	OUT	CL8'	'
&P.BTYP	DS	CL8	BASIS TYPE	OUT	CL8'	'
*						
&P.RES4	DS	CL39	FREE		CL39'	'
*						
*****		PROTECTION ATTRIBUTES	*****			
*						
&P.PTRD	DS	CL1	PROT. TYPE INDI. FOR READ	OUT	CL1'	'
.*			'N': NO SPECIAL PROTECTION			
.*			'Y': SPECIAL PROTECTION			
.*			'G': PROTECTION BY GUARD			
&P.PRDU	DS	CL1	READ INDICATOR FOR OWNER	OUT	CL1'	'
&P.PRDG	DS	CL1	READ INDICATOR FOR GROUP	OUT	CL1'	'
&P.PRDO	DS	CL1	READ INDICATOR FOR OTHERS	OUT	CL1'	'
&P.PRDP	DS	CL1	READ PASSWORD INDICATOR	OUT	CL1'	'
&P.RES5	DS	FL4			FL4'0'	
*						
&P.PTWR	DS	CL1	PROT. TYPE INDI. FOR WRITE	OUT	CL1'	'
&P.PWRU	DS	CL1	WRITE INDICATOR FOR OWNER	OUT	CL1'	'
&P.PWRG	DS	CL1	WRITE INDICATOR FOR GROUP	OUT	CL1'	'
&P.PWRO	DS	CL1	WRITE INDICATOR FOR OTHERS	OUT	CL1'	'
&P.PWRP	DS	CL1	WRITE PASSWORD INDICATOR	OUT	CL1'	'
&P.RES6	DS	FL4			FL4'0'	
*						
&P.PTEX	DS	CL1	PROT. TYPE INDI. FOR EXEC	OUT	CL1'	'
&P.PEXU	DS	CL1	EXEC INDICATOR FOR OWNER	OUT	CL1'	'
&P.PEXG	DS	CL1	EXEC INDICATOR FOR GROUP	OUT	CL1'	'
&P.PEXO	DS	CL1	EXEC INDICATOR FOR OTHERS	OUT	CL1'	'
&P.PEXP	DS	CL1	EXEC PASSWORD INDICATOR	OUT	CL1'	'
&P.RES7	DS	FL4			FL4'0'	
*						
&P.PGRD	DS	CL18	READ GUARD	OUT	CL18'	'
&P.PGWR	DS	CL18	WRITE GUARD	OUT	CL18'	'
&P.PGEX	DS	CL18	EXEC GUARD	OUT	CL18'	'
*						
&P.PTHO	DS	CL1	IND. FOR HOLDER-AUTHORIZATION	OUT	CL1'	'
.*			'N': NO SPECIAL AUTHORIZATION			
.*			'Y': SPECIAL AUTHORIZATION			
.*			'G': AUTHORIZATION BY GUARD			
&P.PHOU	DS	CL1	HOLDER INDICATOR FOR OWNER	OUT	CL1'	'
&P.PHOG	DS	CL1	HOLDER INDICATOR FOR GROUP	OUT	CL1'	'

&P.PH00	DS	CL1	HOLDER INDICATOR FOR OTHERS	OUT	CL1' '
&P.PH0P	DS	CL1	HOLDER PASSWORD INDICATOR	OUT	CL1' '
&P.RES8	DS	FL4			FL4'0'
&P.PGH0	DS	CL18	HOLDER GUARD	OUT	CL18' '
&P.RES9	DS	CL52	FREE		CL52' '
&P.PLNG	EQU	*-&P.TYPE	TA LENGTH		

7.3 Programmierhilfen

Symbolische Namen LMSASSEQ

LMSASSEQ generiert eine Reihe von Equates, die als Programmierhilfe für Verarbeitungsoperandenwerte, Funktionscodes, Subcodes, Returncodes und Speicherungsform der Elemente dienen.

Die Werte der Verarbeitungsoperanden haben die gleiche Bedeutung wie in [1] beschrieben.

Name	Operation	Operanden
name	LMSASSEQ	[P=prefix]

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardmäßig haben die Feldnamen keinen Prefix.

Auflösung von LMSASSEQ

```

***** LMS PARAMETER VALUES *****
*
&P.YES EQU 'Y' YES
&P.NO EQU 'N' NO
&P.NONE EQU 'N' NONE
&P.ANY EQU ' ' ANY
&P.UNCH EQU ' ' UNCHANGED
&P.SAME EQU 'M' SAME
&P.STD EQU 'S' STD
***** FOR: CBFCB *****
&P.ISAM EQU 'I' ISAM
&P.SAM EQU 'Q' SAM
&P.CAT EQU 'C' CAT
***** FOR: CBOV *****
&P.EXT EQU 'E' EXTEND
&P.ONLY EQU 'O' ONLY
&P.NAME EQU 'A' NAME
***** FOR: CBINFO *****
&P.TXT EQU X'01' TEXT ONLY
&P.COM EQU X'02' COMMENT / DOCUMENTATION ONLY
*
***** FUNCTION CODES *****
*
&P.INIT EQU X'01' INIT ( CB )
&P.END EQU X'02' END ( CB )
&P.TOCP EQU X'03' TOCPRIM ( CB, TID,EI, LD, EM )

```

```

&P.TOCS EQU X'04' TOCSEC ( CB, TID,EI, LD, EM )
&P.TOC EQU X'05' TOC ( CB, TID,EI )
&P.REN EQU X'06' REN ( CB, LD, ED1,ED2 )
&P.DEL EQU X'07' DEL ( CB, LD, ED )
&P.ADD EQU X'08' ADD ( CB, FD, LD, ED1 [,ED2] )
&P.SEL EQU X'09' SEL ( CB, LD, ED, FD )
&P.COPY EQU X'0A' COPY ( CB, LD1,ED1,LD2,ED2 [,ED3] )
&P.COPST EQU X'0B' COPYSTR ( CB, LD1,ED1,LD2,ED2 )
&P.LOCK EQU X'0C' LOCK ( CB, LD, ED )
&P.UNLK EQU X'0D' UNLOCK ( CB, LD, ED )
&P.OPENG EQU X'0E' OPEN GET ( CB, RD, LD, ED )
&P.OPENP EQU X'0F' OPEN PUT ( CB, RD, LD, ED1 [,ED2] )
&P.OPENU EQU X'10' OPEN UPD ( CB, RD, LD, ED )
&P.GET EQU X'11' GET ( CB, RD, ER )
&P.PUT EQU X'12' PUT ( CB, RD, ER )
&P.CLOSE EQU X'13' CLOSE ( CB, RD )
&P.LST EQU X'14' LIST ELEMENT ( CB, LD, ED )
&P.MEP EQU X'15' MODIFY PROTECTION ( CB, LD, ED, PA )
&P.MLA EQU X'16' MODIFY LIB. ATTR. ( CB, LD, LA, PA )
&P.SLA EQU X'17' SHOW LIB. ATTR. ( CB, LD, LI )
&P.MTA EQU X'18' MODIFY TYPE ATTR. ( CB, LD, TD, TA, PA )
&P.STA EQU X'19' SHOW TYPE ATTR. ( CB, LD, TD, TI )
&P.MEA EQU X'1A' MODIFY ELEM. ATTR.( CB, LD, ED, EA )
&P.COPLB EQU X'1B' COPY LIBRARY ( CB, LD1,LD2 )
&P.CLOLB EQU X'1C' CLOSE LIBRARY ( CB, LD )
&P.PROVI EQU X'1D' PROVIDE ELEMENT ( CB, LD1,ED1,LD2,ED2 )
&P.RETUR EQU X'1E' RETURN ELEMENT ( CB, LD1,ED1,LD2,ED2,ED3 )
&P.GSYSE EQU X'1F' GET SYSELEM ( CB, LD, ED )
&P.REOLB EQU X'20' REORGANIZE LIB ( CB, LD )

```

*

```

***** SUBCODES *****

```

*

```

&P.UNUSE EQU ' ' SUBCODE UNUSED (DEFAULT)
&P.SHORT EQU 'S' TOC SHORT
&P.LONG EQU 'L' TOC LONG
&P.DIR EQU 'D' READ DIRECT
&P.SEQ EQU 'S' READ SEQUENTIAL
&P.WRITE EQU 'W' CLOSE OUTPUT ELEMENT FOR WRITE
&P.RESET EQU 'R' FORGET OUTPUT ELEMENT
&P.SYM EQU 'S' SHOW ELEMENT SYMBOLIC
&P.HEX EQU 'H' SHOW-ELEMENT ALPHA+HEX
&P.INCP EQU 'P' INCREMENT WITH PREFIX
&P.INCB EQU 'B' INCREMENT WITH BASE
&P.HIGP EQU 'H' HIGHEST EXISTING WITH PREFIX
&P.EXTRA EQU 'X' FORMAT-B RECORDS ALLOWED

```

*

```

***** RETURNCODES *****
*
&P.OK EQU X'00' OK
&P.TRUNC EQU X'04' RECORD TRUNCATED
&P.EOF EQU X'08' END OF GET/TOC
&P.LMSER EQU X'0C' LMS ERROR
&P.PARER EQU X'14' PARAMETER ERROR
&P.SEQER EQU X'18' SEQUENCE ERROR
&P.INTER EQU X'1C' LMS INTERNAL ERROR
*
***** STORAGE FORM *****
*
&P.FULL EQU 'V' FULL ELEMENT
&P.DELTA EQU 'D' DELTA ELEMENT
*
***** CONVENTIONS *****
*
&P.CNONE EQU 'N' NONE
&P.CSEQ EQU 'S' STD-SEQUENCE
&P.CMSEQ EQU 'M' MULTI-SEQUENCE
&P.CTREE EQU 'T' STD-TREE
*
***** PROTECTION INDICATORS *****
*
&P.PNONE EQU 'N' NONE
&P.PSTD EQU 'Y' STD-PROTECTION
&P.PGD EQU 'G' PROTECTION BY GUARD
*
***** SOURCE CODE CONTROL *****
*
&P.FREE EQU '-' FREE
&P.INHLD EQU 'H' IN HOLD
*
***** WRITE CONTROL *****
*
&P.ACTIV EQU 'A' ACTIVATED
&P.DEACT EQU 'D' DEACTIVATED
*
***** ACCESS DATE *****
*
&P.KEEP EQU 'K' KEEP
*
***** MODIFICATION DATE INDICATOR *****
*
&P.OLD EQU 'O' BY-SOURCE
&P.SDAT EQU 'S' NEW (SYSTEM DATE)
.*

```

Aufbau eines Satzes der Satzart 163

Nachfolgend ist der Satz LMSAS163 als Beispiel für einen Satz der Satzart 163 beschrieben.

Name	Operation	Operanden
name	LMSAS163	[P=prefix]

prefix Max. 3 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardmäßig haben die Feldnamen keinen Prefix.

Auflösung von LMSAS163

```

*****
*
*          RECORD TYPE : 1 6 3 (DSECT)
*
*
*****
      SPACE
&NAME   DSECT
&P.163LL DS   H          RECORD LENGTH          H'0'
          DC   AL1(0)          AL1(0)
&P.163RT DC   AL1(163)       RECORD TYPE          AL1(163)
&P.SECNA DC   CL32' '       SECONDARY NAME : 1 - 32    CL32' '
&P.SECAT DC   CL8' '        SECONDARY ATTRIBUTE : 0 - 8    CL8' '
*          '0          ' : - CSECT
*          '1          ' : - ENTRY
&P.SFIND DC   AL1(0)        FORMAT INDICATOR          AL1(0)
&P.SNAML DC   CL1' '        LONG SEC NAME : - 32K-45    CL1' '
&P.163LG EQU  *-&P.163LL

```

Aufbau eines Satzes der Satzart 164

Nachfolgend ist der Satz LMSAS164 als Beispiel für einen Satz der Satzart 164 beschrieben.

Name	Operation	Operanden
name	LMSAS164	[P=prefix]

prefix Max. 2 Zeichen, die den Feldnamen vorangestellt werden sollen.
Standardmäßig haben die Feldnamen keinen Prefix.

Auflösung von LMSAS164

```

*****
*
*          RECORD TYPE : 1 6 4 (DSECT)
*
*
*****
      SPACE
&NAME  DSECT
&P.RECLEN DS   H           LENGTH OF PLAM RECORD           H'540'
          DC     AL1(0)
&P.RECID  DC   AL1(164)   IDENTIFICATION OF PLAM RECORD           AL1(164)
&P.VERS   DC   AL1(2)    VERSION OF SPECIFIED PLAM RECORD         AL1(2)
&P.RECNUM DS   AL1       RECORD DESCRIBES FORMAT           AL1(1)
*
*          OF PLAM RECORD OF RECORD TYPE
*          WITH SPECIFIED NUMBER
&P.FNAME  DS   CL54      FILE NAME           TAKEN FROM FCB       CL54' '
&P.FTYPE  DS   X         FCBTYPE (SET/RESET)           )   X'00'
&P.FTYPES EQU  X'CO'    R SAM                               )
&P.FTYPEI EQU  X'40'    S ISAM                              )
&P.FTYPEP EQU  X'CO'    S PAM                               )
&P.VMIN   EQU  X'01'    R VALPROP MIN. FUNCT.           )
&P.VMAX   EQU  X'01'    S VALPROP MAX. FUNCT.           )
&P.SHARE  DS   X         SHARE (SET,RESET)           CAT   X'00'
&P.SHAREY EQU  X'04'    S YES                               )
&P.ACCESS EQU  X'08'    S ACCESS=READ                   )
&P.SHCCNO EQU  X'CO'    R NO CONTROL CHAR               )
&P.SHCCM  EQU  X'40'    S MACHINE CODE CONTROL CHAR     )
&P.SHCCA  EQU  X'CO'    S ASA CONTROL CHAR               )
&P.SIZE   DS   XL3      FILE SIZE <32GB,                )   XL3'00'
&P.SALL   DS   XL2      SECONDARY ALLOCATION              )   XL2'00'
&P.RECF   DS   X        RECFORM (SET,RESET)             )   X'00'
&P.RECFF  EQU  X'04'    S FIXED                           )
&P.RECFV  EQU  X'02'    S VARIABLE                       )
&P.RECFU  EQU  X'06'    S UNDEFINED                      )
&P.BLKSIZ DS   H        BLKSIZE                          )   H'0'

```

&P.RECSIZ	DS	H	RECSIZE)	H'0'
&P.KEYPOS	DS	H	KEYPOS)	H'0'
&P.KEYLEN	DS	X	KEYLEN)	X'00'
&P.PAD	DS	X	PAD	FCB	X'00'
&P.LOGLN	DS	X	LOGLEN)	X'00'
&P.VALLN	DS	X	VALLEN)	X'00'
&P.KEY	DS	C	DOES KEY EXISTS IN MEMBER ?		C' '
&P.KEYY	EQU	C'Y'	YES		
&P.KEYN	EQU	C'N'	NO		
&P.CFID	DS	XL4	CFID)	XL4'00'
&P.CTRLI	DS	X	BLKCTRL-INDICATOR)	X'00'
&P.CTRLN	EQU	X'80'	S BLKCTRL=NO)	
&P.CTRLP	EQU	X'40'	S BLKCTRL=PAMKEY)	
&P.CTRLD	EQU	X'20'	S BLKCTRL=DATA)	
&P.CTRLR	EQU	X'10'	S BLKCTRL=NULL)	
&P.CTRLR	EQU	X'F0'	R BLKCTRL=NOT SPECIFIED)	
&P.BCF4K	EQU	X'08'	S BLOCK CONTROL FIELD 4K)	
&P.BCF2K	EQU	X'04'	S BLOCK CONTROL FIELD 2K)	
&P.CTRLU	EQU	X'03'	R — RESERVED, MUST BE 0 —)	
*					
&P.PERF	DS	X	IOPERF-INDICATOR)	X'00'
&P.PFVH	EQU	X'03'	S IOPERF=VERY-HIGH)	
&P.PFHI	EQU	X'02'	S IOPERF=HIGH)	
&P.PFST	EQU	X'01'	S IOPERF=STD)	
&P.PFNS	EQU	X'00'	S IOPERF NOT SPECIFIED)	
&P.USAG	DS	X	IOUSAGE-INDICATOR)	X'00'
&P.USRW	EQU	X'03'	S IOUSAGE=RDWRT)	
&P.USWR	EQU	X'02'	S IOUSAGE=WRITE)	
&P.USRD	EQU	X'01'	S IOUSAGE=READ)	
&P.USNS	EQU	X'00'	S IOUSAGE NOT SPECIFIED)	
&P.EDMS3	DS	X	CATALOG-INDIC)	X'00'
&P.ESPEC	EQU	X'08'	S PLAM FILE INDICATOR)	
*					
	DS	X	FREE (0))	X'00'
&P.AIXCNT	DS	H	ALTERNATE INDEX COUNT)	H'0'
&P.AIXMAX	EQU	30	MAX. NR. OF AIX ENTRIES)	
&P.FSIZ	DS	XL4	FILE SIZE >= 32GB,		
*			&P.SIZE MUST BE X'FFFFFF'		
	DS	CL86	RESERVED, MUST BE 0		XL86'00'
*					
&P.AIXNAM	DS	CL8	KEYNAME)	XL8'00'
&P.AIXKPO	DS	H	KEYPOS)	H'0'
&P.AIXKLE	DS	AL1	KEYLEN)	AL1(0)
&P.AIXIND	DS	XL1	INDICATOR)	XL1'0'
&P.AIXIDK	EQU	X'80'	S DUPKEY=YES)	
*			R DUPKEY=NO)	
&P.AIX#	EQU	*-&P.AIXNAM	LENGTH OF AIX ENTRY)	
	DS	(&P.AIXMAX-1)CL(&P.AIX#)			29XL12'00'
*					
&P.LEN	EQU	*-&P.RECLEN	LENGTH OF PLAM RECORD		

7.4 Beispiel

Das folgende Assembler-Programm beinhaltet folgende Funktionen:

- Eröffnen eines Unterprogrammzugriffs (INIT)
- Aufnehmen einer Datei als Element (ADD)
- Suchen eines Elementes in einem Inhaltsverzeichnis (TOCPRIM)
- Eröffnen eines Elementes (OPENGET)
- Lesen des 1. Satzes (GET)
- Schließen des Elementes (CLOSE)
- Beenden des Unterprogrammzugriffs (END)
- zum besseren Verständnis: Kommentieren durch Meldungen

```

*****
*
*           BEISPIEL FUER LMS ALS UNTERPROGRAMM           *
*
*****
*
LMSUP      CSECT
R1         EQU 1           ADRESSE DER PARAMETERLISTE
R2         EQU 2           TEMPORAERES ARBEITSREGISTER
R3         EQU 3           TEMPORAERES ARBEITSREGISTER
R4         EQU 4           TEMPORAERES ARBEITSREGISTER
R5         EQU 5           TEMPORAERES ARBEITSREGISTER
R10        EQU 10          BASISREGISTER
R11        EQU 11          BASISREGISTER
R13        EQU 13          ADRESSE DER SAVEAREA
R14        EQU 14          RUECKSPRUNGADRESSE
R15        EQU 15          EINSPRUNGADRESSE
*
*
*           BALR R10,0
*           USING *,R10,R11
BASIS      LA R11,BASIS+4095      2. BASISREGISTER
*           LA R11,1(R11)
*           LA R13,SAVEAREA

```

```

*****
*
*          AUFBRUF VON INIT
*
*****
*
*          ADRESSE LMSASSCB IN DAS 1. WORT DER PARAMETERLISTE
*
*          LA   R1,SBCB
*          ST   R1,PARAM1
*
*          CB VORBEREITEN FUER AUFBRUF INIT
*
*          MVC  SBCB(CBPLNG),DEFGB
*
*          FUNKTION INIT UND SUBCODE UNUSED IMPLIZIT EINGESTELLT
*
*          LA   R1,PARAM
*          L    R15,=V(LMSUP1)
*          BALR R14,R15
*
*          RETURNCODE-ABFRAGE
*
INITCL  CLI   CBRTC,OK
        BNE  RCPROC
*
*****
*
*          AUFBRUF VON ADD
*
*****
*
*          CB VORBEREITEN FUER AUFBRUF ADD
*
*          MVI  CBFUNC,ADD           FUNKTIONSCODE
*          MVI  CBSUBC,UNUSE        SUBCODE UNUSED (DEFAULT)
*          MVI  CBOV,YES            OVERWRITE=YES
*
*          ANDERE FELDER BLEIBEN WIE BEI INIT
*
*          FD VORBEREITEN FUER AUFBRUF ADD
*
*          MVC  SBFD(FDPLNG),DEFFD   BEREICH FD DEFINIEREN
*          MVC  FDLINK,FILELINK      LINKNAME NACH FD
*
*          LD VORBEREITEN FUER AUFBRUF ADD
*
*          MVC  SBLD(LDPLNG),DEFDLD  BEREICH LD DEFINIEREN
*          MVC  LDLINK,LIBLINK       LINKNAME NACH LD
*
*          ED VORBEREITEN FUER AUFBRUF ADD
*

```

```

MVC SBED(EDPLNG),DEFED          BEREICH ED DEFINIEREN
MVI EDTYPE,'S'                 ABLAGE ALS TYP S
MVC EDNAME(9),ELNAME           ELEMENTNAME
MVC EDVERS(1),ELVERS           ELEMENTVERSION
*
*           PARAMETERLISTE VERSORGEN; PARAM1 ENTHAELT A(CB)
*
LA R1,SBFD                      A(FD)
ST R1,PARAM2
LA R1,SBLD                      A(LD)
ST R1,PARAM3
LA R1,SBED                      A(ED)
ST R1,PARAM4
*
LA R1,PARAM
L R15,=V(LMSUP1)
BALR R14,R15
*
*           RETURNCODE-ABFRAGE
*
ADDCL CLI CBRTC,OK
      BNE RCPROC
*
*****
*
*           AUFRUF VON TOC
*
*****
*           CB VORBEREITEN FUER AUFRUF TOCPRIM
*
MVI CBFUNC,TOCP                 FUNKTIONSCODE
MVI CBSUBC,LONG                 ERWEITERTE ELEMENTINFO
*
*           LD-FELDER BLEIBEN WIE VORBELEGT
*
*           EM VORBEREITEN FUER AUFRUF TOCPRIM
*
LA R2,SBEM                      ZIELADRESSE
LA R3,EMPLNG                    UEBERTRAGUNGSLAENGE
LA R4,DEFEM                      SENDEADRESSE
LR R5,R3
MVCL R2,R4
*
MVI EMTYPE,'S'
MVC EMNAME(9),ELNAME
*
*           PARAMETERLISTE VERSORGEN; PARAM1 ENTHAELT A(CB)
*

```

```
LA    R1,SBTID           A(TID)
ST    R1,PARAM2
LA    R1,SBEI           A(EI)
ST    R1,PARAM3
LA    R1,SBLD           A(LD)
ST    R1,PARAM4
LA    R1,SBEM           A(EM)
ST    R1,PARAM5
*
LA    R1,PARAM
L     R15,=V(LMSUP1)
BALR R14,R15
*
*           RETURNCODE-ABFRAGE
*
TOCCL CLI  CBRTC,OK
      BNE  RCPROC
*
*           ERHALTENE INFO AUSGEBEN
*
MVC   OTYPF,EITYPE
WROUT OTYP,TERM
*
MVC   ONAMEF,EINAME
WROUT ONAME,TERM
*
MVC   OVERSF,EIVERS
WROUT OVERS,TERM
*
MVC   ODATEF,EIUDAT
WROUT ODATE,TERM
*
```

```

*****
*
*          AUFRUF VON OPENGET
*
*****
*          CB VORBEREITEN FUER AUFRUF OPENGET
*
*          MVI   CBFUNC,OPENG          FUNKTIONSCODE
*          MVI   CBSUBC,UNUSE         SUBCODE UNUSED (DEFAULT)
*
*          LD-FELDER BLEIBEN WIE VORBELEGT
*
*          ED-FELDER BLEIBEN WIE VORBELEGT
*
*          RD VORBEREITEN FUER AUFRUF OPENGET
*
*          MVC   SBRD(RDPLNG),DEFRD    BEREICH RD DEFINIEREN
*
*          PARAMETERLISTE VERSORGEN; PARAM1 ENTHAELT A(CB)
*
*          LA    R1,SBRD                A(RD)
*          ST    R1,PARAM2
*          LA    R1,SBLD                A(LD)
*          ST    R1,PARAM3
*          LA    R1,SBED                A(ED)
*          ST    R1,PARAM4
*
*          LA    R1,PARAM
*          L     R15,=V(LMSUP1)
*          BALR R14,R15
*
*          RETURNCODE-ABFRAGE
*
*          OPENGCL CLI   CBRTC,OK
*          BNE   RCPROC
*

```

```

*****
*
*          AUFRUF VON GET, SAETZE IN SCHLEIFE LESEN
*
*****
*          CB VORBEREITEN FUER AUFRUF GET
*
GETLOOP  MVI   CBFUNC,GET          FUNKTIONSCODE
          MVI   CBSUBC,SEQ        SEQUENTIELLES LESEN
*
*          RD VORBEREITEN FUER AUFRUF GET
*
          MVC   RDBLEN,PLENGTH    PUFFERLAENGE EINGABEBEREICH
*
          PARAMETERLISTE VERSORGEN; PARAM1 ENTHAELT A(CB)
*
          A(RD) WURDE SCHON BEI OPENGET VERSORGT
*
          LA    R1,SBER            A(ER)
          ST    R1,PARAM3
*
          LA    R1,PARAM
          L     R15,=(LMSUP1)
          BALR  R14,R15
*
*          RETURNCODE-ABFRAGE / SATZ AUSGEBEN
*
GETCL    CLI   CBRTC,OK
          BNE  GETEND
*
*          SATZAUSGABE OHNE SPALTE 1 (STEUERZEICHEN)
*
          WROUT SBER,TERM
          B     GETLOOP
*
*          END OF ELEMENT ABFRAGEN
*
GETEND   CLI   CBRTC,EOF
          BNE  RCPROC
*

```

```
*****
*
*          AUFRUF VON CLOSE
*
*****
*          CB VORBEREITEN FUER AUFRUF CLOSE
*
*          MVI   CBFUNC,CLOSE          FUNKTIONSCODE
*          MVI   CBSUBC,UNUSE          SUBCODE UNUSED (DEFAULT)
*
*          RD-FELDER BLEIBEN WIE VORBELEGT
*
*          PARAMETERLISTE VERSORGEN; PARAM1 ENTHAELT A(CB)
*
*          A(RD) WURDE SCHON BEI OPENGET VERSORGT
*
*          LA    R1,PARAM
*          L     R15,=V(LMSUP1)
*          BALR R14,R15
*
*          RETURNCODE-ABFRAGE
*
CLOSECL  CLI   CBRTC,OK
          BNE  RCPROC
```

```

*****
*
*           AUFRUF VON END
*
*****
*           CB VORBEREITEN FUER AUFRUF END
*
*           MVI   CBFUNC,END           FUNKTIONSCODE
*           MVI   CBSUBC,UNUSE        SUBCODE UNUSED (DEFAULT)
*
*           LA    R1,PARAM
*           L     R15,=V(LMSUP1)
*           BALR  R14,R15
*
*           RETURNCODE-ABFRAGE
*
ENDCL      CLI   CBRTC,OK
           BNE  RCPROC
           TERM
*
*****
*
*           FEHLER-PROZEDUR
*
*****
RCPROC    EQU   *
           WROUT MESSAGE,FEHLER
*
FEHLER    EQU   *
TERM      TERM

```

```
*****
*
*           KONSTANTEN-DEFINITIONEN
*
*****
DEF CB      LMSASSCB MF=L           KONSTANTEN FUER CB
*
DEF FD      LMSASSFD MF=L          KONSTANTEN FUER FD
*
DEF LD      LMSASSLD MF=L          KONSTANTEN FUER LD
*
DEF ED      LMSASSED MF=L          KONSTANTEN FUER ED
*
DEF EI      LMSASSEI MF=L          KONSTANTEN FUER EI
*
DEF EM      LMSASSEM MF=L          KONSTANTEN FUER EM
*
DEF RD      LMSASSRD MF=L          KONSTANTEN FUER RD
*
FILELINK DC 'FILELINK'
LIBLINK  DC 'LIBLINK '
PLENGTH DC A(L'SBER)
ELNAME   DC 'PROBEELEM'
ELVERS   DC '1'
*
MESSAGE  DC Y(MESSAGEE-MESSAGE)
          DS CL2
          DC X'40'
          DC 'FUNKTION FEHLERHAFT'
MESSAGEE EQU *
*
```

```

*****
*
*           SPEICHERBEREICHE
*
*****
SBCB      LMSASSCB MF=C           SPEICHERBEREICH FUER CB
*
SBFD      LMSASSFD MF=C          SPEICHERBEREICH FUER FD
*
SBLD      LMSASSLD MF=C          SPEICHERBEREICH FUER LD
*
SBED      LMSASSED MF=C          SPEICHERBEREICH FUER ED
*
SBEI      LMSASSEI MF=C          SPEICHERBEREICH FUER EI
*
SBEM      LMSASSEM MF=C          SPEICHERBEREICH FUER EM
*
SBRD      LMSASSRD MF=C          SPEICHERBEREICH FUER RD
*
SBER      DS      CL256           SATZPUFFER
SBTID     DC      F'1'            TOC-IDENTIFIKATION
*
OTYP      DC      Y(OTYPE-OTYP)   FUER TYP-AUSGABE
          DS      CL2
          DC      X'40'
          DC      'TYPE           '
OTYPF     DC      CL(L'EITYPE)' '
OTYPE     EQU     *
*
ONAME     DC      Y(ONAMEE-ONAME)  FUER NAMENS-AUSGABE
          DS      CL2
          DC      X'40'
          DC      'NAME           '
ONAMEF    DC      CL(L'EINAME)' '
ONAMEE    EQU     *
*
OVERS     DC      Y(OVERSE-OVERS)  FUER TYP-AUSGABE
          DS      CL2
          DC      X'40'
          DC      'VERSION       '
OVERSF    DC      CL(L'EIVERS)' '
OVERSE    EQU     *
*
ODATE     DC      Y(ODATEE-ODATE)  FUER DATUMS-AUSGABE
          DS      CL2
          DC      X'40'
          DC      'USER-DATE     '
ODATEF    DC      CL(L'EIUDAT)' '
ODATEE    EQU     *

```

```

*****
*
*          PARAMETERLISTE
*
*****
PARAM   DS   OF
PARAM1  DS   F          A(LMSASSCB)
PARAM2  DS   F
PARAM3  DS   F
PARAM4  DS   F
PARAM5  DS   F
PARAM6  DS   F
*****
*
*          SAVEAREA
*
*****
SAVEAREA DS   18F
*****
END

```

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)
SDF-Format
Benutzerhandbuch
- [2] **XHCS** (BS2000/OSD)
8-bit-Code-Verarbeitung im BS2000/OSD
Benutzerhandbuch

Stichwörter

A

- ADD-Funktion 58
- Administrationsrecht, siehe Verwaltungsrecht
- Adressierungsmodus des Hauptprogramms 15
- aktiver Write Control 107
- Ändern
 - der Bibliotheksattribute 93
 - der Elementattribute 89
 - der Typattribute 95
 - des Elementschutzes 91, 93
- Änderungsdatum aktualisieren 21
- Anzeigen
 - der Bibliotheksattribute 121
 - der Typattribute 123
- Assembler-Schnittstelle 209
- Attributsatz Aufbau 142
- Aufbau
 - der Parmeterstrukturen 17
 - des Attributsatzes 142
 - des Sekundärsatzes 141
- Aufheben einer Elementsperrre 135
- Aufnehmen einer Datei 58
- Aufrufparameter versorgen 15
- Aufrufvorbereitungen 15
- Ausgeben eines Elementes 86, 119
- Ausleihverfahren einschalten 93

B

- Beenden eines UP-Zugriffs 76
- Bibliothek
 - bestimmen 37
 - Inhalt auflisten 26, 127
 - kopieren 70
 - schließen 64, 76
 - UPAM-Schutz 38
 - Verwaltungsrecht vergeben 34
- Bibliotheksattribute
 - anzeigen 121
 - bestimmen 93
- Bibliotheksbeschreibung 37
- Bibliotheksgröße ausgeben 38

C

- C-Schnittstelle 177
- CB - Control Block 18
- CCS-Name 21, 66
- CLOSE-Funktion 62
- COBOL-Schnittstelle 145
- COPY-Element 17
- COPY-Funktion 64, 66
- COPYLIB-Funktion 70
- COPYSTR-Funktion 72

D

Darstellungsform eines Satzes 86
Datei aufnehmen 58
Dateiattribute 58
Dateibeschreibung 34
Datum ändern 89
Datum eintragen
 bei ADD 58
 bei COPY 66
DEL-Funktion 74
Delta-Baum kopieren 72
Delta-Element aufnehmen 58, 67, 108

E

EA - Element Attributes 21
ED - Element Description 23
EI - Element Information 26
Eintragen aktuelle Uhrzeit 58, 66
Eintragen aktuelles Datum 58, 66
Element
 Ausgabeformat 53
 ausgeben 86, 119
 im primären Inhaltsverzeichnis suchen 127
 im sekundären Inhaltsverzeichnis
 suchen 131
 kopieren 66
 löschen 74
 reserviertes 107
 reserviertes zurückgeben 116
 schließen 62
 sperrern 84
 suchen 12
 umbenennen 112
 zum Lesen eröffnen 97
 zum Lesen/Schreiben eröffnen 104
 zum Schreiben eröffnen 100
Element einer Quellbibliothek
 kopieren 116
 reservieren 107
Elementattribute
 ändern 89
 einstellen 21
Elementbeschreibung 23
Elementinformation 26

Elementmaske 29
Elementsatz übergeben 33
Elementschutz
 ändern 91, 93
 anzeigen 123
 festlegen 41
Elementsperre aufheben 135
Elementtyp
 ausgeben 48
 bestimmen 48
 Konvention 46
EM - Element Mask 29
END-Funktion 76
ER - Element Record 33
Eröffnen
 eines Elementes zum Lesen 97
 eines Elementes zum Lesen/Schreiben 104
 eines Elementes zum Schreiben 100
 eines UP-Zugriffs 82

F

FD - File Description 34
Fortsetzen von TOCPRIM 125
Fortsetzen von TOCSEC 125
Funktion der Parameterstrukturen 17
Funktionen
 für das Suchen nach Elementen 12
 für den Elementschutz 13
 für den Versionsautomatismus 13
 für die Unterprogramm-Schnittstelle 11
 für die XHCS-Unterstützung 13
 für Lesen/Schreiben von Elementen 12

G

Generierung
 der Parameterstrukturen für Assembler 210
 der Parameterstrukturen für C 177
 der Parameterstrukturen für COBOL 146
GET-Funktion 77
GSYSELEM-Funktion 80

H

Hauptprogramm
 Adressierungsmodus 15
 Datenübergabe 17
 Rücksprung 210

I

INCLUDE-Element 17
INIT-Funktion 82

J

Julianisches Datum 32

K

Konvention für Elementtyp 46
Kopieren einer Bibliothek 70
Kopieren eines Delta-Baumes 72
Kopieren eines Elementes 64, 66

L

LA - Library Attributes 34
LD - Library Description 37
Lesen
 eines Satzes 77
 Elementangabe aus S-Variable 80

LI - Library Information 38

LMS als Unterprogramm 7

LMSAS163 239
LMSAS164 240
LMSASSCB 211
LMSASSEA 213
LMSASSED 214
LMSASSEI 215
LMSASSEM 218
LMSASSEQ 236
LMSASSFD 221
LMSASSLA 222
LMSASSLD 224
LMSASSLI 225
LMSASSPA 227
LMSASSRD 229
LMSASSTA 230
LMSASSTD 232
LMSASSTI 233

LMSCO163 165
LMSCO164 166
LMSCOBCB 147
LMSCOBEA 148
LMSCOBED 148
LMSCOBEI 149
LMSCOBEM 151
LMSCOBEQ 162
LMSCOBFD 153
LMSCOBLA 153
LMSCOBLD 154
LMSCOBLI 155
LMSCOBPA 157
LMSCOBRD 158
LMSCOBTA 159
LMSCOBTD 160
LMSCOBTI 160
LMSUP1

 für die Assembler-Schnittstelle 209

 für die C-Schnittstelle 177

 für die COBOL-Schnittstelle 145

LOCK-Funktion 84

Löschen eines Elementes 74

LST-Funktion 86

M

Makros 17
Maskenfelder 29
MODEA-Funktion 89
MODEP-Funktion 91
MODLA-Funktion 93
MODTA-Funktion 95

N

Name symbolischer 137

O

OPENGET-Funktion 97
OPENPUT-Funktion 100
OPENUPD-Funktion 104

P

PA - Protection Attributes 41
Parameterstrukturen
 für Assembler generieren 210
 für C generieren 177
 für COBOL generieren 146
Parameterstrukturen (Funktion, Aufbau) 17
Programmierhilfen 137
 für COBOL 162
PROVIDE-Funktion 107
PUT-Funktion 111

R

RD - Record Description 45
Readme-Datei 9
Reihenfolge der Parameter 51
REN-Funktion 112
REORGLIB 114
Ressourcen freigeben 76
RETURN-Funktion 116
Returncodes 16
Rückkehrparameter 15

S

S-Variable Inhalt lesen 80
Satz
 darstellen 86
 lesen 77
 schreiben 111
 verändern 104
Satzart 77
 Reihenfolge 111
Satzart 163 (Aufbau) 141
Satzart 163 für ASSEMBLER 239
Satzart 163 für COBOL 165
Satzart 164 (Aufbau) 142
Satzart 164 für ASSEMBLER 240
Satzart 164 für COBOL 166
Satzkopf 111
Satzlänge 111
Satznummer 77
Schließen eines Elementes 62
Schreiben eines Satzes 111
Schutzattribute 41

Sekundärsatz Aufbau 141
SEL-Funktion 119
SHOWLA-Funktion 121
SHOWTA-Funktion 123
Spalte pro Zeile 86
Speicherungsform 24
 definieren 93
Sperren eines Elementes 84
Standardwerte für Elementschutz
 anzeigen 121
Subcode LMSUP-DIR 77
Subcode LMSUP-HEX 86
Subcode LMSUP-RESET 62
Subcode LMSUP-SEQ 77
Subcode LMSUP-SYM 86
Subcode LMSUP-WRITE 62
Subcode LONG 128, 132
Subcode SHORT 128, 132
Subcodes
 für den Versionsautomatismus 24
 Übersicht 53
Suchen eines Elementes
 im primären Inhaltsverzeichnis 127
 im sekundären Inhaltsverzeichnis 131
symbolische Namen 137

T

TA - Type Attributes 46
TD - Type Description 48
TI - Type Information 48
TID - TOC-Identifikation 50
TOC-Funktion 125
 bestimmen 50
TOC-Identifikation 125
TOCPRIM fortsetzen 125
TOCPRIM-Funktion 127
TOCSEC fortsetzen 125
TOCSEC-Funktion 131
Typattribute 46
 ändern 95
 anzeigen 123
Typbeschreibung 48
Typinformation 48

U

- Uhrzeit ändern 89
- Uhrzeit eintragen
 - bei ADD 58
 - bei COPY 66
- Umbenennen eines Elementes 112
- UNLOCK-Funktion 135
- UP-Funktionen 11
- UP-Funktionen (Übersicht) 55
- UP-Zugriff beenden 76
- UP-Zugriff eröffnen 82
- UPAM-Schutz 38

V

- Versionsangabe 23
- Versionsautomatismus 13, 24
 - steuern 53
- Versorgen der Aufrufparameter 15
- Verwaltungsrecht anzeigen 121
- Verwaltungsrecht für eine Bibliothek 34
- Voreinstellungen für Elementenschutz 41
- Vorschubsteuerzeichen 86

W

- Wiedergewinnungsadresse 45
- Write Control aktiv 107

X

- XHCS-Unterstützung 13

Z

- Zeichenfolge ersetzen 31
- Zeichensatz
 - einem Element zuordnen 21
 - Name ausgegeben 26
- Zeichensatz-Unterstützung 13
- Zeilen pro Seite 86
- Zugriffsdatum aktivieren 93
- Zugriffsidentifikation 11
- Zugriffspfad-Identifikation 12
 - hinterlegen 104

