



Das Dokumentations -

und

Source - Verwaltungssystem

Bedienungsanleitung

Version: A.03.01

Stand: 17. Juni 2008

Hinweis

Die SSD IT Consulting GmbH haftet nicht für etwaige Fehler in dieser Dokumentation. Eine Haftung für mittelbare und unmittelbare Schäden, die im Zusammenhang mit der Lieferung oder dem Gebrauch dieser Dokumentation entstehen, ist ausgeschlossen, soweit dies gesetzlich zulässig ist.

Diese Dokumentation enthält urheberrechtlich geschützte Informationen.

Alle Rechte, insbesondere das Recht auf Vervielfältigung und Verbreitung sowie der Übersetzung, bleiben vorbehalten. Kein Teil der Dokumentation darf in irgendeiner Form (durch Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne vorherige schriftliche Zustimmung der SSD IT Consulting GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Copyright:

SSD IT Consulting GmbH

**Gütschstrasse 11
CH 6404 Greppen**

Telefon: +41 (0)41 390 31 92
FAX: +41 (0)41 390 31 93
Natel: +41 (0)79 286 54 13
EMail: Support@ssd-it.ch

Drucklegende

Version A.02.00	April 1988
Version A.02.01	Dezember 1988
Version A.02.02	April 1989
Version A.02.03	September 1989
Version A.02.04	Dezember 1990
Version A.02.05	September 1991
Version A.02.06	Februar 1993
Version A.02.07	Juni 1994
Version A.02.08	Juni 1996
Version A.02.09	Oktober 1997
Version A.02.09.01	Mai 2000
Version A.03.00	August 2002
Version A.03.01	Juni 2008

1. Aufgaben von SD/3000

Das **Software-Dictionary-System** basiert auf 3 Grundgedanken der modernen Software-Entwicklung:

1. strukturierte Programmierung
2. Standard-Modulköpfe (=Programmdokumentation) in der Source
3. automatisch gepflegte Dokumentationsdatenbank

Diese Grundgedanken ziehen sich durch das Gesamtpaket SD/3000. Darüber hinaus will SD/3000 dem Entwickler möglichst einfach zu handhabende Hilfsmittel an die Hand geben, damit er seine tägliche Arbeit einfacher und schneller erledigen kann.

Den DV-/Projekt-Leiter unterstützt SD/3000 durch eine bessere Übersicht über die Projekte und einen systematischen Aufbau der Systemdokumentation über das Gesamtsystem.

Die Version A.01.00 von SD/3000 deckte die folgenden Funktionen ab:

- ONLINE-Definition und -Verwaltung von Projekten und Sachgebieten
- Programmdokumentation im Quelltext nach einfachen, vorgegebenen Mustern (unabhängig von der Programmiersprache)
- Automatische Erzeugung einer Dokumentationsdatenbank aus dem Quelltext
- Automatische Erzeugung von Referenzierungen (z.B. Datei -> Programm, COMMON -> Programm, Programm -> Datei, Programm -> Maske, Programm -> COPYLIB)
- Referenzierung von frei definierbaren Schlagworten
- Automatische Verwaltung der Änderungshistorie von Programmen
- Automatische Generierung von Übersetzungs- und Binde-Jobs.
- Standardisierte INCLUDE-Bibliotheken für alle Programmiersprachen
- Dokumentenverwaltung

Darüber hinaus enthält das Softwarepaket SD/3000 eine Reihe von allgemein verwendbaren Utilities, die die Arbeit der Softwareerstellung erheblich erleichtern.

Dazu gehören:

- das SPERR-System für Segmenter-Bibliotheken und sonstige MPE-Dateien,
- ein universell nutzbares Listprogramm, das auch komprimierte, QEDIT und Squished Dateien direkt andrucken kann,
- der zentrale COMPILER-Preprozessor zum Auflösen von \$INCLUDE's.

Ab der Version **A.02.05** steht ein neuer, vom Anwender modifizierbarer Jobgenerator im Native-Mode zur Verfügung. Außerdem wurden die COPYLIB-Generatoren für V/3000 und IMAGE-Satzbeschreibungen, globale IMAGE-Definitionen und für Listenlayouts integriert.

Mit der Version **A.02.06** ist die Sourcecode-Verwaltung über die SDSLIB implementiert. Der Generator unterstützt zusätzlich die Preprozessoren für Oracle sowie INCLUDE-Gruppen.

Mit der Version **A.02.07** sind die meisten SD-Programme als NM-Programme implementiert worden. Ab Version **A.02.08** kann SD/3000 nur noch auf MPE/iX Systemen verwendet werden.

2. Das Verwaltungsmodul SDMAINT

Das Verwaltungsmodul dient zur Eingabe und zum Ändern der Angaben in der SD/3000-Datenbank. Mit diesem Modul werden die JOB-Definitionen, die Steuerinformationen sowie die SD/3000-internen Dateien und Texte zur Dokumentation verwaltet. Der Name der Datenbank ist:

SD3000.DATABASE.GLOBAL

Das Verwaltungsprogramm wird gestartet mit

:M {auswahl} (als UDC/XEQ-Datei) oder mit
:RUN SDMAINT.PUB.GLOBAL [;PARM={auswahl}]

Soll nur in einer Maske gearbeitet werden, so kann beim Aufruf durch Angabe der Auswahlnummer direkt in diese Maske gesprungen werden. Es erscheint nicht das Menü. Durch Drücken der EXIT-Funktionstaste wird das Programm sofort wieder verlassen.

Zusätzlich steht ein TDP/EDITOR/QEDIT-Interface für den Aufruf zur Verfügung. Durch Eingabe des entsprechenden PROC-Befehls wird SDMAINT ebenfalls gestartet. Hierzu ist es erforderlich, daß in dem jeweiligen EDITOR eine Arbeitsdatei angelegt wurde. Der Aufruf ist z.B.:

PROC SDM,*,* in TDP
P SDM,S,*,* in EDITOR/QEDIT

Das gesamte Verwaltungsmodul ist mit der Masken-Software V/3000 geschrieben. Die Funktionstasten-Belegung ist in allen Funktionen gleich. Wenn neuere HP-Bildschirme der 262x/239x/700-Serien verwendet werden, so wird die Funktionstasten-Belegung am Bildschirm angezeigt.

Sie hat grundsätzlich folgenden Aufbau:

generieren eines MPE/V-Jobs in COMPILE/LINK-Maske
einfrieren von Versionen in Sachgebietenmaske
löschen des derzeitigen Satzes
drucken der Maske mit Inhalt
Refresh
zurück oder generieren eines MPE/iX-NM-Jobs
weiter oder SDPRINT
Hilfe
Exit

Das Drucken einer Maske () geschieht:

- a) wenn ein FILE-Befehl für SDLISTE vorhanden ist auf diesen File oder
- b) auf den dem Terminal zugeordneten voreingestellten Drucker lt. SD/3000-Konfiguration.

A c h t u n g

Es darf kein FILE-Befehl für **SDTERM** bestehen. Sollte dies der Fall sein, so wird SDMAINT mit einem VOPENTERM-Fehler abgebrochen.

2.0.1 Das Verwaltungsmenue

Die Auswahlen 1, 2, 11, 12, 20 und 21

- 1 Übersetzungsjobs definieren
- 2 Bindejobs definieren
- 11 Datei Definitionen
- 12 COMMON Definitionen
- 20 Dokument Einträge
- 21 COPYLIB/INCLUDE-Gruppe Definition

sind allen Benutzern zugänglich.

Die Auswahlen 3 bis 10 und 13 bis 19 und 22

- 3 Projekte
- 4 Sachgebiete
- 5 Jobids
- 6 -
- 7 -
- 8 Schlagworte
- 9 Intrinsic Definition
- 10 Voreinstellungen
- 13 Bildschirm Definition
- 14 Drucker Definition
- 15 Install-Account Verwaltung
- 16 Dokumentations Phasen
- 17 Dokumentations Typen
- 18 Dokumenten Stati
- 19 Dokument Typen
- 22 Datei Zugriffsroutinen

sind nur Anwendern mit der SM-Berechtigung bzw. Anwendern im GLOBAL-Account zugänglich.

2.0.2 Standard - Dialogablauf

Nach Eingabe des Schlüsselbegriffs und ENTER wird die Maske mit den vorhandenen Angaben gefüllt und kann korrigiert werden. Ein zweites ENTER veranlaßt die Übernahme der aktuellen Feldinhalte.

Ist ein Satz neu einzufügen, so erscheint nach dem ersten ENTER die Meldung: "... *bisher unbekannt*". Ein weiteres ENTER veranlaßt die Übernahme des Maskeninhaltes in die Datenbank.

Eine Änderung zwischen erstem und zweitem ENTER unterbricht den Ablauf und startet mit der neuen Angabe den Ablauf.

Das Löschen ist nur über die Funktionstaste möglich und muß durch ein zweites Drücken dieser Taste bestätigt werden.

Das Löschen von COMPILE-Einträgen ist nicht möglich, wenn noch Source-Dateien in der SD-Source-Bibliothek gespeichert sind.

2.1 Übersetzungsjobs definieren

Mit dieser Maske wird ein Übersetzungsjob definiert, der nachher mit einem Generator-Befehl **Gxx** erzeugt werden kann. Die Felder dieser Maske im einzelnen:

2.1.1 Schlüsselbegriffe

Modul

Ein maximal 16-stelliger Name, der neben Buchstaben und Zahlen auch die Sonderzeichen ‘ und _ enthalten darf. Diese Zeichen sind für SPL- und PASCAL-Programme erforderlich. Ein Modulname muß mit einem Buchstaben beginnen und kann von Buchstaben, Zahlen oder den oben aufgeführten Sonderzeichen gefolgt werden. Der Modulname wird grundsätzlich linksbündig und in Großbuchstaben gespeichert.

Projekt

Ein maximal 4-stelliges Projektkürzel. Das Kürzel besteht aus einem Buchstaben und bis zu 3 weiteren Buchstaben und/oder Zahlen. Das Kürzel wird intern linksbündig und in Großbuchstaben gespeichert. Innerhalb eines Projekts müssen die Sachgebiets- und Modulnamen eindeutig vergeben werden.

Sachgebiet

Ein maximal 2-stelliges Sachgebietskürzel. Das Kürzel besteht aus einem Buchstaben und evtl. einem weiteren Buchstaben oder einer Zahl. Es wird intern linksbündig und in Großbuchstaben gespeichert.

2.1.2 Folgejob

Folgebefehl

Soll nach erfolgreicher Abarbeitung eines mit GC(x) gestarteten Jobs ein Folgejob gestartet werden, so ist hier der entsprechende G-Befehl einzutragen. Zulässig sind: GC, GCX, GL und GLX. Beim Starten von Jobs über die GCL/GPR-Befehle werden die Folgejobs nicht mit gestartet.

Folgemodul

Wenn ein Folgebefehl eingetragen wurde, so muß hier der zugehörige Modulname eingetragen werden. Dieser Modulname wird hier ausnahmsweise nicht überprüft, ob er vorhanden ist. Ein entsprechender Fehler tritt erst bei Ausführung des Folgebefehls im Job auf.

Folgeprojekt

Wenn ein Folgebefehl eingetragen wurde, so kann hier der zugehörige Projektname eingetragen werden. Wird dies Feld leer gelassen, wird das im Account voreingestellte Projekt genutzt.

Folgeoptionen

Wenn ein Folgebefehl eingetragen wurde, so können hier zugehörige Generatoroptionen für den Start des Folgejobs eingetragen werden. Wenn es sich bei dem Account um einen INSTALL-Account handelt, so wird die evtl. eingetragene I-Option bei Aufruf des Generators ohne I auch für den Folgejob ausgeblendet.

2.1.3 Herkunft (Source)

Sourcedatei

Der Name der entsprechenden Compiler-Eingabe-Datei oder der Datei für die Sonderbehandlung (Source-Typ = STR, CAT, GCAT, HELP oder VIEW). Normale Sourcedateien sollten nicht qualifiziert, d.h. ohne Gruppe- und Accountangabe, angegeben werden. Nur Sourcedateien vom Typ STR müssen voll qualifiziert eingetragen werden.

Sourcetyp

Der Typ der Source, ist entweder ein Compiler:

FTN4	FORTRAN-4/3000
FTN7	FORTRAN-77/3000 oder FTNXL
SPL	SPL/3000
TPL	TRANSACT/3000 oder TRANXL
TPLF	Fastran
RPL	REPORT/3000
PAS	PASCAL/3000 oder PASXL
COB2	COBOLII/3000 oder COB74XL
COBX	COBOLiX/3000 oder COB85XL
CC	CCS/C Compiler oder CCXL
OC	Oracle C Preprozessor und CCXL
OCOB	Oracle COBOL Preprozessor und COBOLiX/COB85XL
OFOR	Oracle FORTRAN Preprozessor und FTN/FTNXL
PC	SQL-C Preprozessor und CCXL
PCOB	SQL-COBOL Preprozessor und COBOLiX/COB85XL
PFOR	SQL-FORTRAN Preprozessor und FTN/FTNXL
PPAS	SQL-PASCAL Preprozessor und PASCAL/PASXL

oder eine Datei für besondere Behandlung:

CAT	Meldungskatalog(MAKECAT-Format)
GCAT	Meldungskatalog (GENCAT-Format)
HELP	Hilfekatalog im MAKECAT-Format
VIEW	eine V/3000 Maskendatei
BRW	Business Report Writer
JOB	ein Job (z.B. QUERY)
QDD	ein COGNOS Dictionary
QUIK	ein QUICK-Programm
QUIZ	ein QUIZ-Programm
STR	Executejob ohne JOB- und EOJ-Statement

Preprozessor Optionen

Bei HP-SQL-Preprozessoren (PC, PCOB, PFOR und PPAS) muß hier der Name des DB-Environment eingetragen werden. Wenn es sich um eine Source vom Typ STR handelt, so kann hier der echte Dateiname der Source eingetragen werden. In diesem Fall finden später der Kopflader und die Source-Bibliothek die Source.

Compiler Optionen

Für alle Compiler können Optionen angegeben werden. Sie entsprechen bei den Standardcompilern den Angaben im \$CONTROL-Befehl bzw. den sonstigen Preprozessor Befehlen (z.B. \$SET). Durch das **SDINCLUDE-JCW** wird gesteuert, ob der Compiler-Aufruf ein Standard-MPE Compileraufruf ist (SDINCLUDE = 0) oder über UDC's (z.B. mit SDINCL) geschieht.

Der CCS/C Compiler bringt standardmäßige keine Liste. Hierzu sollte hier eine der List-Optionen eingetragen werden. Bei BRW-Übersetzungen kann hier das Selektionsset angegeben werden.

2.1.4 Ziel

Zieldatei

Die Zieldatei ist abhängig vom Zieltyp eine USL, RL, SL, XL, OBJ, CREL, CLIB oder ein ausführbares Programm. Der Name der Zieldatei sollte in der Regel bei Programmen mit PUB qualifiziert werden. Eine Zuordnung in einen speziellen Account sollte vermieden werden.

Bei CCS/C Programmen kann zusätzlich ein CLINK Transferdatei angegeben werden. Bei den anderen Compilern kann über die Angabe einer LINK-Folgemaske gezielt in den LINK-Step eingegriffen werden. Ist in diesem Fall eine Zieldatei angegeben, so wird sie vor Aufruf des SEGMENTER/LINKEDIT/FASTSEG-Programms gelöscht.

Ist eine MPE-Bibliothek nicht vorhanden, so wird sie mit 5000 Sätzen in max. 16 Extents neu angelegt.

Zieltyp

USL *Zieldatei* ist eine USL bzw. eine MPE/iX-RL bei Nutzung des Compatibility-Features.

Segment ist das Ziel-Segment
Modul ist der Modulname
als *PREP-Option* ist nur zulässig:

LISTUSL <segmentname>

OBJ *Zieldatei* ist eine MPE/iX-Objektdatei.

RL *Zieldatei* ist eine MPE/V-RL bzw. eine MPE/iX-RL
Modul ist der Modulname,
als *PREP-Option* ist nur **LISTRL** zulässig.

SL *Zieldatei* ist eine MPE/V-SL bzw. eine MPE/iX-XL bei Nutzung des Compatibility-Features.

Modul ist der Modulname
Segment ist das Ziel-Segment
als *PREP-Option* ist nur zulässig:

LISTSL <segmentname>

XL *Zieldatei* ist eine MPE/iX-XL,
Modul ist der Modulname
Segment ist ein Locality-Set
als *PREP-Option* sind alle LINKEDIT-ADDXL Parameter zulässig

PROG *Zieldatei* ist eine Programmdatei
Einheit ist die Klasse der Zielplatteneinheit,
Dateischutz für die Programmdatei

PREP-Optionen können alle benutzt werden,
Bei CCS/C Programmen ist dies eine *CLINK-Option*.

RL kann genutzt werden.
Bei CCS/C Programmen ist dies eine **CLIB**.

PCOD *RAPID/3000* Programmcode (TPL oder RPL)

- CREL** *Zieldatei* ist eine CCS/C CREL-Datei
Dateischutz für die CREL-Datei
- CLIB** *Zieldatei* ist eine CCS/C CLIB-Datei
- CTR** Mit Hilfe des CCS/C CLINK Programms wird eine Zieldatei erzeugt.
Der Typ wird in der Transferdatei festgelegt.
- LINK** Auf einer Folgemaske können SEGMENTER/LINKEDIT/FASTSEG - Befehle für diesen Job
manuell eingegeben werden. Diese Folgemaske erscheint nach jedem internen Update dieser
Maske.

Bei den Sourcetypen **xCAT, HELP, VIEW, BRW, QRY, QDD, QUIK, QUIZ** und **STR** ist der Zieltyp
nicht erforderlich.

Zieleinheit

Wenn im SEGMENTER/LINK-Step des Jobs eine Datei angelegt wird, so wird diese auf der hier spezifizier-
ten Einheit angelegt. Für dieses Feld ist eine Voreinstellung in der Datenbank hinterlegt und sollte in der
Regel nicht überschrieben werden.

Segment

Bei Zieldateien vom Typ **USL, SL** wird eine Resegmentierung in das angegebene Segment durchgeführt. Als
Segmentname ist auch **\$ALL** zulässig. In diesem Fall werden alle in der Source-USL vorhandenen UNIT's in
die Zieldatei übertragen. Bei Zieldateien vom Typ MPE/V-RL werden bei Angabe von **\$ALL** alle UNIT's in
die RL kopiert.

Dateischutz

Hier kann der Zugriffsschutz der Zieldatei entweder in Form einer Securitymatrix (ALTSEC-Format) oder als
ACD angegeben werden bzw. der Befehl RELEASE.

LINK/PREP-Option

abhängig vom Typ der Zieldateien:

USL	LISTUSL <segmentname>
RL	LISTRL
SL	LISTSL <segmentname>
OBJ	LISTOBJ
XL	alle möglichen LINK-Parameter
PROG	alle möglichen PREP/LINK-Parameter (Bei C-Programmen CLINK Parameter)
PCOD	keine
CLIB	keine
CREL	keine
CTR	keine
LINK	keine

RL

Bei Zieldateien vom Typ **PROG** kann hier die zu nutzende RL eingetragen werden. Wird die RL hier anstelle
im PREP-Feld eingetragen, so wird bei Nutzung der GROUP/INSTALL-Optionen auch jeweils die richtige
RL angezogen.

Bei CCS/C Programmen ist diese Datei eine CLIB.

2.1.5 Job Parameter

JOBID

Kennung, unter welchem Benutzer/Account und in welcher Gruppe die Übersetzung bei der Installation laufen soll.

INPRI

Spezielle INPUT-SPOOL-Priorität, sie entspricht dem INPRI-Parameter im JOB-Statement. Dies Feld sollte in der Regel nicht angegeben werden.

Zulässig sind die Werte 0 bis 14. INPRI=14 wird vom Generator in HIPRI übersetzt.

RUNPRI

Spezielle RUN-Priorität, dies entspricht dem PRI-Parameter im JOB-Statement. Dies Feld sollte in der Regel nicht angegeben zu werden.

Zulässig sind die Angaben BS, CS, DS und ES.

2.1.6 Änderungshistorie

Änderungshistorie

Bei jeder Änderung eines Eintrages in der Datenbank wird das Datum, die Uhrzeit und der Benutzer automatisch registriert.

2.1.7 Jobtexte

Texte

Zu einem COMPILE-Job können 2 Zeilen a 72 Zeichen Hinweistexte hinterlegt werden. Dieser Text wird bei einem späteren Generator-Aufruf mit angezeigt.

2.2 Bindejobs definieren

Die Angaben in der Maske entsprechen im wesentlichen denen in der Maske für Übersetzungsjobs. Es können hier jedoch nur Angaben zu den SEGMENTER/PREP/LINK-Abläufen hinterlegt werden. Diese Maske sollte ferner dazu benutzt werden um spezielle Jobs z.B. zum Umkopieren von Programmdateien, von Installationsjobs oder von Recoveryjobs in SD/3000 einzutragen.

2.2.1 Schlüsselbegriffe

Modul

Ein maximal 16-stelliger Name, der neben Buchstaben und Zahlen auch die Sonderzeichen ‘ und _ enthalten darf. Diese Zeichen sind für SPL- und PASCAL-Programme erforderlich. Ein Modulname muß mit einem Buchstaben beginnen und kann von Buchstaben, Zahlen oder den oben aufgeführten Sonderzeichen gefolgt werden. Der Modulname wird grundsätzlich linksbündig und in Großbuchstaben gespeichert.

Projekt

Ein maximal 4-stelliges Projektkürzel. Das Kürzel besteht aus einem Buchstaben und bis zu 3 weiteren Buchstaben und/oder Zahlen. Das Kürzel wird intern linksbündig und in Großbuchstaben gespeichert. Innerhalb eines Projekts müssen die Sachgebiets- und Modulnamen eindeutig vergeben werden.

Sachgebiet

Ein maximal 2-stelliges Sachgebietskürzel. Das Kürzel besteht aus einem Buchstaben und evtl. einem weiteren Buchstaben oder einer Zahl. Es wird intern linksbündig und in Großbuchstaben gespeichert.

2.2.2 Folgejob

Folgebefehl

Soll nach erfolgreicher Abarbeitung eines mit GL(x) gestarteten Jobs ein Folgejob gestartet werden, so ist hier der entsprechende G-Befehl einzutragen. Zulässig sind: GC, GCX, GL und GLX. Beim Starten von Jobs über die GCL/GPR-Befehle werden die Folgejobs nicht mit gestartet.

Folgemodul

Wenn ein Folgebefehl eingetragen wurde, so muß hier der zugehörige Modulname eingetragen werden. Dieser Modulname wird hier ausnahmsweise nicht überprüft, ob er vorhanden ist. Ein entsprechender Fehler tritt erst bei Ausführung des Folgebefehls im Job auf.

Folgeprojekt

Wenn ein Folgebefehl eingetragen wurde, so kann hier der zugehörige Projektname eingetragen werden. Wird dies Feld leer gelassen, wird das im Account voreingestellte Projekt genutzt.

Folgeoptionen

Wenn ein Folgebefehl eingetragen wurde, so können hier zugehörige Generatoroptionen für den Start des Folgejobs eingetragen werden. Wenn es sich bei dem Account um einen INSTALL-Account handelt, so wird die evtl. eingetragene I-Option bei Aufruf des Generators ohne I auch für den Folgejob ausgeblendet.

2.2.3 Herkunft (Source)

Sourcedatei

Der Name der entsprechenden Sourcedatei oder der Datei für die Sonderbehandlung (Source-Typ = STR, oder JOB).

Normale Sourcedateien sollten nicht qualifiziert, d.h. ohne Gruppe- und Accountangabe, angegeben werden. Nur Sourcedateien vom Typ **STR** müssen voll qualifiziert eingetragen werden.

Sourcetyp

Der Typ der Source, ist eine gültige Eingabedatei für SEGMENTER / FASTSEG / LINKEDIT / CLINK:

USL	USL
USLF	USL mit FASTRAN Programmen
RL	RL (nur bei XL-Version)
OBJ	MPE/iX Object File
CREL	CCS/C relocatable Datei

oder eine Datei für besondere Behandlung:

JOB	ein Job (z.B. QUERY)
STR	Executejob ohne JOB- und EOJ-Statement

2.2.4 Ziel

Zieldatei

Die Zieldatei ist abhängig vom Zieltyp eine SL, RL, USL, XL, CLIB oder ein ausführbares Programm. Der Name der Zieldatei sollte in der Regel bei Programmen mit PUB qualifiziert werden. Eine Zuordnung in einen speziellen Account sollte vermieden werden.

Bei Sourcedateien vom Typ CREL ist zusätzlich eine CLINK Transferdatei möglich. Die weiteren Angaben zur Art des Linkens stehen in dieser Datei. Bei den anderen Sourcetypen kann über die Angabe einer LINK-Folgemaske gezielt in den LINK-Step eingegriffen werden. Ist in diesem Fall eine Zieldatei angegeben, so wird sie vor Aufruf des SEGMENTER/ LINKEDIT/FASTSEG-Programms gelöscht.

Ist die Zielbibliothek nicht vorhanden, so wird sie mit 5000 Sätzen in max. 16 Extents neu angelegt.

Zieltyp

USL	<i>Zieldatei</i> ist eine USL
RL	<i>Zieldatei</i> ist eine RL
SL	<i>Zieldatei</i> ist eine SL oder XL
XL	<i>Zieldatei</i> ist eine XL
PROG	<i>Zieldatei</i> ist eine Programmdatei <i>Zieleinheit, PREP-Option</i> und <i>RL</i> sind zulässig
CLIB	<i>Zieldatei</i> ist eine CCS/C Library
CTR	<i>Zieldatei</i> ist eine CCS/C-CLINK-Transferdatei
LINK	Auf einer Folgemaske können SEGMENTER/LINKEDIT/FASTSEG - Befehle für diesen Job manuell eingegeben werden. Diese Folgemaske erscheint nach jedem internen Update dieser Maske.

Zieleinheit

Wenn die Zieldatei vom Typ **PROG** ist, so kann hier eine spezielle Platteneinheit angegeben werden. Die Voreinstellung ist in der Datenbank hinterlegt und sollte in der Regel nicht überschrieben werden.

Segment

Bei Zieldateien vom Typ **SL** wird ein Segment kopiert. Wird kein Segmentname angegeben, so wird der Modulname als Segmentname übernommen. Als Segmentname ist auch **\$ALL** zulässig. In diesem Fall werden alle in der Source-USL vorhandenen UNIT's in die Zieldatei übertragen.

Dateischutz

Hier kann der Zugriffsschutz der Zieldatei entweder in Form einer Securitymatrix (ALTSEC-Format) oder als ACD angegeben werden bzw. der Befehl RELEASE.

LINK/PREP-Optionen

abhängig vom Typ der Zieldateien:

SL	LISTSL <segmentname>
XL	alle möglichen Parameter
PROG	alle möglichen PREP/LINK-Parameter (Bei CREL-Sources: CLINK Befehle)
CLIB	keine
CTR	keine
LINK	keine

RL

Bei Zieldateien vom Typ **PROG** und **XL** kann hier die zu nutzende **RL** eingetragen werden. Wird die **RL** hier anstelle im PREP-Feld eingetragen, so wird bei Nutzung der GROUP/INSTALL-Optionen auch jeweils die richtige **RL** angezogen.

Bei Sourcedateien vom Typ **CREL** ist dies eine **CLIB**.

2.2.5 Job Parameter**JOBID**

Kennung, unter welchem Benutzer/Account und in welcher Gruppe die Übersetzung bei der Installierung laufen soll.

INPRI

Spezielle INPUT-SPOOL-Priorität, sie entspricht dem INPRI-Parameter im JOB-Statement. Dies Feld sollte in der Regel nicht angegeben werden. Zulässig sind die Werte 0 bis 14. INPRI=14 wird vom Generator in HIPRI übersetzt.

RUNPRI

Spezielle RUN-Priorität, dies entspricht dem PRI-Parameter im JOB-Statement. Dies Feld sollte in der Regel nicht angegeben zu werden. Zulässig sind die Angaben BS, CS, DS und ES.

2.2.6 Änderungshistorie

Bei jeder Änderung eines Eintrages in der Datenbank wird das Datum, die Uhrzeit und der Benutzer automatisch registriert.

2.2.7 Jobtexte**Texte**

Zu einem LINK-Job können 2 Zeilen a 72 Zeichen Hinweistexte hinterlegt werden. Dieser Text wird bei einem späteren Generator-Aufruf mit angezeigt.

2.3 Projekte

Projektkürzel sind maximal 4-stellig, beginnen mit einem Buchstaben und können von Buchstaben und/oder Zahlen gefolgt werden. In dieser Maske muß zu jedem Projekt ein beschreibender Text eingegeben werden.

Um die Arbeit mit SD/3000 zu erleichtern sollten Projekte nicht zu groß (d.h. mit zu vielen Modulen und Sachgebieten) definiert werden. Von SD/3000 her besteht keine Limitation für die Größe eines Projekts.

Das Arbeiten mit kleineren Projekten und Sachgebieten bietet sich des weiteren an damit die Generator-Befehle und Kopf-Lade-Befehle für komplette Sachgebiete oder Projekte häufiger genutzt werden können.

Wenn die Funktionen von SDTRANS genutzt werden sollen, so kann selektiert werden, welche Teile für den Update von Fremdsystemen transferiert werden sollen.

2.4 Sachgebiete

Um ein neues Sachgebiet eintragen zu können, muß das Projekt in SD/3000 bekannt sein. Sachgebiete werden durch eine 2-stellige Kennung, die mit einem Buchstaben beginnt und von einem Buchstaben und/oder einer Zahl gefolgt wird, identifiziert.

Zu jedem Sachgebiet kann ein Verantwortlicher und sein Vertreter festgelegt werden. In dem Feld Kürzel wird zusätzlich das entsprechende Benutzerkürzel im System angegeben. Wenn der entsprechende Account in der INSTALL-Account Maske separat geschützt wird, so können danach nur noch die hier eingetragenen Personen Installationen durchführen. Bei Einsatz der SD-Source-Bibliothek haben diese Personen ebenfalls besondere Berechtigungen. Detail hierzu finden Sie in Kapitel 9.

Wenn ein Sachgebiet komplett installiert und abgenommen ist, kann es durch den SD-Manager für Programmänderungen und/oder SD-Änderungen gesperrt werden. Dem SD-Manager ist es ebenfalls möglich Sachgebiete wieder zu entsperren. Alle SPERR/ENTSPERR-Aktionen werden in SD protokolliert.

Einem gesamten Sachgebiet kann eine Versionsnummer zugeordnet werden. Mit dieser Möglichkeit ist nachträglich jederzeit zu bestimmen, zu welchem Zeitpunkt welche Moduln aktiv waren. Die entsprechenden Sources können in der SD-Sourcebibliothek gespeichert werden.

Durch Drücken der Funktionstaste werden die aktuellen Versionsnummern der zugeordneten Moduln gespeichert. Sind bereits unter dieser Versionsnummer Einträge vorhanden, so werden sie gelöscht.

Wird das Feld Versionsnummer gelöscht und danach die Funktionstaste gedrückt, so werden alle gespeicherten Versionen zu diesem Sachgebiet gelöscht.

2.5 JOBID (Job-Identifikation)

Ein JOBID ist ein 8-stelliges Kürzel. Es beginnt mit einem Buchstaben und kann von Buchstaben und/oder Zahlen gefolgt werden. Es dient dazu, die MPE-Anmeldung flexibel zu handhaben.

Um einen Job eindeutig identifizieren zu können, sind immer die 3 Angaben:

Benutzerkürzel
Accountname und
Gruppenname

erforderlich.

Nachricht

Im Feld Nachricht kann **X**, für den Erhalt von positiven Fertigmeldungen aus dem Job eingetragen werden. Fehlermeldungen im Compile/Link-Step der Jobs werden immer an den Benutzer, der die Anwahl vorgenommen hat, abgegeben.

INPRI

Dies Feld entspricht der MPE INPRI-Option, die für jede Anmeldung gilt. INPRI=14 wird in HIPRI übersetzt.

RUNPRI

Entsprechendes gilt für dies Feld, das die MPE Priority-Angabe BS, CS, DS oder ES enthält.

Diese Felder sollten nur bei Abweichungen vom Standard (INPRI=8, RUNPRI=DS, Nachricht=X) ausgefüllt werden.

Das Löschen von JOBID's ist ab der Version A.02.02 nicht mehr möglich, wenn das JOBID in COMPILE/LINK-Definitionen verwendet wurde.

2.6 Schlagworte

Ab der Version A.02.03 können freie Schlagworte in SD/3000 definiert werden. Damit können projektübergreifende Referenzen für Moduln etabliert werden.

Ein Schlagwort beginnt mit einem Buchstaben und wird von Buchstaben, Zahlen, ‘, - oder _ gefolgt. Die Vergabe von Schlagworten sollte zentral und streng kontrolliert erfolgen, damit sie auch effektiv genutzt werden können.

Zu jedem Schlagwort muß ein beschreibender Text eingegeben werden. Dieser Text wird in den Referenzlisten mit ausgegeben.

2.7 INTRINSIC's

Diese Maske dient zur Verwaltung von Standard-Systemroutinen. Intrinsic's sind entweder von HP zur Verfügung gestellte Routinen, eigene Unterprogramme oder auch allgemeine Programme. Eigene Unterprogramme müssen in SD/3000 im Projekt INTR definiert sein.

Der **Name eines INTRINSIC's** muß mit einem Buchstaben beginnen und kann von Buchstaben, Zahlen oder ‘ und _ gefolgt werden.

Die Typ des INTRINSIC's kann sein:

HPxx - HP Standard Intrinsic / Unterprogramm
xxxx - Eigenentwicklung bzw. zu dokumentierendes Programm

Die HP-Intrinsic's sollten nicht verändert werden. Eigene Routinen sollten unter einem eigenen Typ definiert werden. Bei eigenen Routinen bzw. Standardroutinen kann die Dokumentation entweder in einem Modulkopf im Projekt INTR hinterlegt werden oder in Form einer maximal 5-zeiligen Kurzdokumentation in dieser Maske.

Bei der Installation von SD/3000 sind die folgenden Typen definiert:

HP21	PLOT/21 Unterprogramme	HP62	LU6.2 API Intrinsic's
HPAI	Architected Interface Intrinsic's	HPBR	BRW Intrinsic's
HPCL	Compiler Library	HPD	DIALIB Intrinsic's
HPDS	DS/3000 und DS/1000 Intrinsic's	HPFS	Filesystem Intrinsic's
HPGR	DSG/3000 Intrinsic's	HPIF	IMF/3000 Intrinsic's
HPIM	IMAGE/Turbo-IMAGE Intrinsic's	HPIN	Standard HP Intrinsic's
HPIP	IPC Intrinsic's	HPKS	KSAM Intrinsic's
HPME	MERGE/3000 Intrinsic's	HPNC	NC Intrinsic's
HPNL	NLS Intrinsic's	HPNR	NRJE Intrinsic's
HPNS	NS/3000 Intrinsic's	HPPS	PSP Intrinsic's
HPSD	System Dictionary Intrinsic's	HPSL	Scientific Library
HPSO	SORT/3000 Intrinsic's	HPSQ	SQL Intrinsic's
HPV	V/3000 Intrinsic's	HPXL	MPE/iX Intrinsic's
HPXX	sonstige HP Intrinsic's	HPWD	HPWORD Intrinsic's
PROG	Standard Programme in PUB.SYS		

2.8 Voreinstellungen

In dieser Maske werden die systemweiten Voreinstellungen sowohl für alle Moduln des SD/3000 definiert.

Voreinstellung für:

z.Zt. ist nur die Voreinstellung 1 zugelassen.

Projekt:

Wenn in den Account-Definitionen keine speziellen Voreinstellungen eingetragen sind, so wird bei allen Anwahlen von SD/3000-Programmen dies Projekt als Voreinstellung genommen.

SPERR-System:

Durch Eingabe eines **X** werden wahlweise Aufrufe für das Datei SPERR-System bei der Generierungen von Jobs mit generiert.

Notify-Option:

Voreinstellung für das Eintragen von Jobs im Verwaltungsprogramm. Bei Eingabe eines **X** erhalten die Benutzer eine positive Meldung am Ende des Jobs.

Input-Priorität:

Voreinstellung für das Eintragen von Jobs im Verwaltungsprogramm. Dies Feld entspricht dem INPRI-Parameter der Jobkarte.

RUN-Priorität:

Voreinstellung für das Eintragen von Jobs im Verwaltungsprogramm. Dies Feld entspricht dem PRI-Parameter der Jobkarte.

Programm-Platteneinheit:

Voreinstellung für das Eintragen von Jobs im Verwaltungsprogramm. Sollen auf dem System aus Performancegründen die Programmdateien auf bestimmten Laufwerken liegen, so kann der Name des Laufwerks hier eingetragen werden.

Drucker:

Voreinstellung für das Verwaltungsprogramm. Wenn kein FILE-Befehl für den File **SDLISTE** vorhanden ist, so wird dieser SD/3000-Drucker für den Andruck von Bildschirmmasken verwendet.

Änderungshistorie:

Alle Änderungen in dieser Maske werden mit Datum, Uhrzeit und Benutzer festgehalten.

2.9 Dateinamen

Jede im Dokumentationsteil von SD/3000 verwendete Datei sollte mit ihrer Beschreibung in diesem Set eingetragen sein. Eine Datei ist grundsätzlich einem Projekt und einem Sachgebiet zugeordnet. Beide müssen vorher definiert sein.

Die Identifikation einer Datei bei der Dokumentation geschieht über Dateiname und Projekt. Das Sachgebiet ist nur Kommentar.

Der **Dateityp** wird durch ein 4-stelliges Kürzel angegeben:

BASE	TurboIMAGE Datenbank	AUTO	TurboIMAGE Automatic Master
MAST	TurboIMAGE Manual Master	DETL	TurboIMAGE Detail
SQLC	SQL DBEnvironment	SQLS	SQL DBSet
SQLT	SQL Table	SQLV	SQL View
MPEC	MPE Circular Datei	MPEF	normale MPE-Datei z.b. sequentiell
MPEM	MPE Message Datei	MPER	RIO Datei
KSAM	KSAM/3000 Datei	VPLS	V/3000 Maskendatei
FILE	ein formaler MPE-Filename		
REC	Datensatz innerhalb einer Datei		

Als **Status** ist zulässig:

PERM	für permanent und	TEMP	für temporär.
-------------	-------------------	-------------	---------------

Als **Beschreibung** können bis zu 72 Zeichen Text eingegeben werden. Dieser Text wird in Dokumentationen angedruckt.

2.10 Common

Diese Maske dient zur Verwaltung von Beschreibungen zu globalen Datenbereichen. Sie hat nur Bedeutung im Dokumentationsteil. Ein COMMON wird eindeutig identifiziert durch COMMON-Name und Projekt. Die Zuordnung zu einem Sachgebiet dient als Kommentar. Ein COMMON-Name beginnt mit einem Buchstaben und kann von Buchstaben, Zahlen, -, _ oder ' gefolgt werden. Er kann maximal 16 Stellen lang werden.

Der **Typ des COMMON** kann sein:

FCOM	Fortran COMMON Block	DSEG	HP 3000 Extra Datensegment
GLOB	Globale Variable (SPL/3000)	JCW	Job Control Word
JVAR	MPE/iX Job Control Variable	VAR	Variable
RIN	Resource Identification Number	SOCK	NetIPC Socketname
PORT	AIF Portname		

Als Beschreibung können bis zu 72 Zeichen eingegeben werden.

Als COMMON werden alle globalen Datenbereiche, die zwischen verschiedenen Programmen oder Moduln ausgetauscht werden definiert. Dazu gehören z.B. standardisierte LINKAGE-SECTION in COBOL oder auch andere normierte Datenblöcke zwischen den Programmen. Wird in FORTRAN-Programmen die Referenzierung der einzelnen Variablen in COMMON-Blöcken gewünscht, so muß jede Variable als ein COMMON-Block eingetragen werden.

2.11 Bildschirm Definition

Alle Bildschirme des Systems, an denen das SD/3000-Paket benutzt werden darf, müssen mit dieser Maske dem SD/3000 bekannt gemacht werden. Dies ist u.a. aus Sicherheitsgründen erforderlich. Die Definition eines Bildschirms kann mit Hilfe des JCW's SDLU überschrieben werden. Details hierzu sind bei den entsprechenden Programmen beschrieben.

Bildschirm LU:

Die MPE logische Einheitennummer des Bildschirms

Bildschirm Typ:

Die HP-Typnummer des Bildschirms. Gültig sind:

239n, 262n, 264n, 700.

zugeordneter Drucker:

Der diesem Bildschirm zugeordnete Drucker. Es handelt sich um einen SD/3000-internen Drucker, der vorher definiert sein muß.

Beschreibung:

Die Standortbeschreibung des Bildschirms. Sie darf bis zu 72 Zeichen lang sein.

Änderungshistorie:

Alle Änderungen in dieser Maske werden mit Datum, Uhrzeit und Benutzer festgehalten.

2.12 Drucker Definition

In verschiedenen Moduln des SD/3000-Pakets werden Ausgaben auf den File **SDLISTE** erzeugt. Die hierfür erforderliche Beschreibung des Files ist in dem SD/3000-internen Druckernamen hinterlegt.

Alle hier eingetragenen Drucker können von dem Druckmodul angesprochen werden. Sie werden dort durch den OUT-Befehl angezeigt. Als Sonderdefinition für das Druckprogramm muß ein Satz mit der Klasse **\$STDLIST** eingetragen sein. Der entsprechende Druckername wird als Äquivalent für den Bildschirm benutzt.

Die Felder im einzelnen:

Drucker Name:

Der SD/3000-interne Name des Druckers. Ein Druckername besteht aus einem Buchstaben die von Buchstaben und/oder Zahlen gefolgt werden können. Der Name muß eindeutig sein.

Drucker LU:

Die MPE logische Einheitennummer des Druckers. Dies Feld ist z.Zt. nur Kommentar. Die Generierung der entsprechenden FILE-Befehle erfolgt nur über den Klassennamen.

Drucker Klasse:

Der MPE logische Klassenname des Druckers. Dies Feld muß richtig eingetragen werden. Es wird für alle Generierungen benutzt.

Drucker Typ:

Die HP-Typnummer des Druckers. Die folgenden Typen sind gültig:

2235, 256n, 2608, 263n, 268n, 293n

FILE-Statement nach dem Teil: FILE <druckername>;

Sind für den FILE-Befehl spezielle Optionen wie z.B. mehrere Kopien, eine spezielle Priorität oder eine **ENVIRONMENT/TT**-Datei erforderlich, so muß der FILE-Befehl hier vollständig incl. dem Klassennamen eingetragen werden.

Standort:

Die Beschreibung des Standorts des Druckers. Dies Feld ist nur Kommentar. Es darf bis zu 72 Zeichen lang sein.

Änderungshistorie:

Alle Änderungen in dieser Maske werden mit Datum, Uhrzeit und Benutzer festgehalten.

2.13 Install-Account Verwaltung

Mit dieser Maske können jedem Account spezifische Daten zugeordnet werden. In SD/3000 ist es normalerweise einem normalen Benutzer nicht möglich, Jobs in fremden Accounts anzusprechen. In dieser Maske kann jedoch über die Zuordnung PROD-Account -> TEST-Account diese Begrenzung für bestimmte Accounts aufgehoben werden. Zusätzlich können für den eigentlichen Produktionsaccount spezielle Schutzmechanismen aktiviert werden.

Produktionsaccount:

Der Name des PROD-Accounts, auf den die eingetragenen TEST-Accounts zugreifen dürfen. Hier kann auch ein Account eingetragen werden um die Voreinstellungen zu nutzen.

Testaccounts:

Alle hier eingetragenen Accounts haben Zugriff auf die Daten des Produktionsaccounts.

Voreingestelltes Projekt:

Das in dem Produktionsaccount sowie allen Testaccounts voreingestellte Projektkürzel.

INCLUDE-Bibliothek:

Die Gruppe (und der Account) mit den INCLUDE-Bibliotheken und dem Data-Dictionary. Der Name der Bibliothek ist der Name des Compilers. Wenn der Jobgenerator mit der GROUP-Option aufgerufen wird oder die JVAR SDLOCALCOPYLIB vorhanden ist, so kann eine COPYLIB in der lokalen Gruppe die Standard COPYLIB ersetzen.

INSTALL-Account:

Besonders schützenswerte Account können mit einem **X** in diesem Feld zu einem INSTALL-Account gemacht werden. Beim generieren von Jobs ist dann die G-Option voreingestellt. Alle Jobs die mit der I-Option generiert werden, werden in der Logdatei protokolliert.

Sachgebiets Verantwortlicher:

Ist ein Account besonders schützenswert, so kann zusätzlich zur INSTALL-Option diese Option aktiviert werden. Danach dürfen nur noch die im Sachgebiet eingetragenen Personen Compile/Link-Läufe starten.

Änderungshistorie:

Alle Änderungen in dieser Maske werden mit Datum, Uhrzeit und Benutzer festgehalten.

2.14 Dokumentations Phasen

Die Dokumentenverwaltung in SD/3000 erlaubt die freie Definition einer Dokumentationsstruktur. Der oberste Begriff im Rahmen der Dokumentenverwaltung ist die Dokumentationsphase. Die folgenden Phasen könnten z.B. vergeben werden:

PLANUNG	ENTWURF
PROGR	BETRIEB
WARTUNG	

Der **Name einer Dokumentationsphase** beginnt mit einem Buchstaben und darf von Buchstaben, Zahlen oder den Sonderzeichen ‘ und _ gefolgt werden. Er ist maximal 8-stellig.

Die **Priorität einer Dokumentationsphase** gibt die logische Folge im Projektfortschritt an. Die Nummern sollten mit entsprechenden Abständen vergeben werden um evtl. später zusätzliche Phasen einfügen zu können. Die Priorität wird für die logisch richtige Reihenfolge des Andrucks der Dokumente benötigt.

Die **Beschreibung der Phase** dient zu SD/3000 internen Dokumentationszwecken.

2.15 Dokumentations Typen

Der Dokumentationstyp beschreibt die während einer Dokumentationsphase anfallenden Unterlagen. Gleichzeitig werden die unterschiedlichen Typen einem Level zugeordnet. Die Priorität beschreibt auf der Dokumentationsphasenebene die logische Folge der Typen innerhalb einer Phase. Der Name ist die Beschreibung des Typs. Ein Beispiel für eine Dokumentationsstruktur:

Phase	Typ	LEVEL	PRIO	Beschreibung
PLANUNG	IDEEN	SYSTEM	100	Ideenpapier
	MACH	SYSTEM	200	Machbarkeitsstudie
	RP	SYSTEM	300	Rahmenplanung
ENTWURF	DETAILPL	PROJECT	1000	Detailplanung
	DBDESIGN	PROJECT	1100	DB-Design
	SCHNITT	PROJECT	1200	Schnittstellen Beschreibung
PROGR	MODDES	MODUL	2000	Modul-Design
	PROGDOK	CLASS	2100	Programm Dokumentation
	BEDIEN	PROJECT	2200	Bedienungsanleitung
	OPERHB	PROJECT	2300	Operating Handbuch

2.16 Dokument Stati

Der Dokumentstatus beschreibt den Status eines Dokuments. Der Status hat z.Zt. nur eine beschreibende Funktion. Er soll in einer späteren Ausbaustufe von SD/3000 Steuerungsfunktionen erhalten. Der **Status eines Dokuments** besteht aus einem Buchstaben und wird von Buchstaben oder Zahlen gefolgt. Der Status ist maximal 4-stellig. Der **Name des Status** ist ein Text zur internen Dokumentation.

Beispiel für die Statusvergabe sind:

PLAN	geplantes Dokument	ERST	in Erstellung
PRUF	in Prüfung	FERT	Fertig

2.17 Dokumentation Dateitypen

Ein Typ beschreibt die Art einer Dokumentdatei und definiert den Server für die Verarbeitung dieser Art von Dokumenten. Ein **Dokumentations Dateityp** beginnt mit einem Buchstaben und wird von Buchstaben, Zahlen oder den Sonderzeichen ‘ und _ gefolgt. Der **Server Name** ist der Name einer Prozedur in SL.PUB.GLOBAL die eine Datei von diesem Typ verarbeiten (ausgeben) kann. Die folgenden Server sind in Version A.02.08 implementiert:

<u>Typ</u>	<u>Server Name</u>
ASCII	SD'P'ASCII

Die Tabelle **gültige Ausgabeeinheiten** enthält die Drucker- und Bildschirm Typen auf dem diese Art von Dokument ausgeben werden kann. z.Zt. sind die folgenden Typen zulässig:

TERM	Standard HP Bildschirm HP2392 kompatibel
LP	Standard HP Lineprinter ohne Grafik
PP	HP2680/HP2688 Laserdrucker
LJET	Laserjet mit mindestens PCL-Level 2

2.18 Dokument Einträge

Auf dieser Maske wird eine Dokumentationsdatei in SD/3000 eingetragen. Gleichzeitig wird sie einem Dokumentationstyp zugeordnet und ihr Status festgelegt. Über den Dokumenttyp wird die Art der Aufbereitung bei der Ausgabe festgelegt.

Entsprechend dem **Dokumentationslevel** können Modul-, Projekt- und Sachgebiet-Name eingetragen werden.

Die **Kurzbeschreibung** enthält eine kurze Inhaltsangabe des Dokuments.

Wenn das Dokument auf Band ausgelagert sein sollte, so beschreiben **Einheitentyp** (TAPE, DISC), **Einheit** (Bandname) und **Position** wo sich das Dokument genau befindet.

2.19 COPYLIB Definition

Wenn die Referenzierung für COPYLIB-Member in SD/3000 verwendet werden soll, ist es erforderlich, alle COPYLIB's bzw. INCLUDE-Gruppen über diese Maske zu definieren. Der Name der COPYLIB sollte immer voll qualifiziert angegeben werden. Bei INCLUDE-Gruppen ist als Dateiname ein @ einzugeben.

Jede COPYLIB/INCLUDE-Gruppe muß einem Projekt und einem Sachgebiet zugeordnet werden. Außerdem muß ihr Typ festgelegt werden.

Gültige Typen sind:

COB	für alle COBOL Compiler	FTN	für alle FORTRAN Compiler
PAS	für PASCAL	RPL	für REPORT
SPL	für SPL	TPL	für Transact und Fastran

Zu jeder COPYLIB/INCLUDE-Gruppe muß eine einzeilige Beschreibung eingegeben werden. Diese Beschreibung wird auf den Dokumentationen mit angedruckt.

2.20 Datei Zugriffsroutinen

Wenn der COBOL-Source Analyser installiert ist, können nicht nur die normalen Zugriffe auf IMAGE / Filesystem usw. erkannt werden, sondern auch vom Anwender erstellte zentrale Zugriffsroutinen. Diese Zugriffsroutinen müssen in dieser Maske definiert werden. In SD/3000 können bis zu 10 Zugriffsroutinen eingetragen werden.

Der **Name der Routine** ist der Name des Unterprogramms das mit **CALL** aufgerufen wird. Die Definition einer Zugriffsroutine in dieser Maske ersetzt nicht die Definition des Moduls in der COMPILE-Maske. Diese Maske dient allein der Steuerung des COBOL-Source-Analysers.

Die **Parameternummer** dient der Ermittlung des Dateinamens der MPE-Datei oder des IMAGE-Datasets. Aus diesem Grunde muß der entsprechende Parameter auch eine Variable mit dem Dateinamen enthalten und nicht etwa eine DATASET-Nummer o.ä..

Ab der Version A.02.03 wird bei der Art des Dateizugriffs nicht nur nach INPUT und OUTPUT unterschieden, sondern es werden Kennbuchstaben für den Zugriff verwendet (z.B. G für GET oder D für DELETE). Das Feld Accesscode definiert den Kennbuchstaben, der bei dieser Zugriffsroutine in den Kopf eingetragen wird.

Der COBOL-Source-Analyser vergibt die folgenden Zugriffskodes:

C	für CONTROL	D	für DELETE
G	für GET	I	für INFO
L	für LOCK	M	für MERGE
O	für OPEN	P	für PUT
S	für SORT	U	für UPDATE

3. Der Jobgenerator SDGEN

Das Generatorprogramm (SDGEN.PUB.GLOBAL) unterscheidet 6 verschiedene Versionen des Aufrufs. Sie werden über die PARM-Option angewählt. Zum leichteren Aufruf sind dafür UDC's bzw. XEQ-Dateien installiert. Sind für ein zu startendes Modul Texte definiert, so werden Sie beim Gxx-Aufruf angezeigt.

Die Aufrufsyntax ist für alle Versionen gleich:

:Gxx {modul}[,{proj}[,{opt1},{optn},..]]

{modul}

Das zu bearbeitende Modul bzw. die Executedatei bei den GxF-Aufrufen.

{proj}

Das Projekt, zu dem das Modul gehört. Voreinstellung ist *. Hierbei wird das in der Datenbank voreingestellte systemweite, oder wenn vorhanden, das accountweite Projekt angezeigt und benutzt.

{optn}

In beliebiger Reihenfolge bis zu 4 Optionen, die durch einen Buchstaben gekennzeichnet werden.

Die verschiedenen Aufrufe im einzelnen:

:GC (Übersetzung)

Bei diesem Kommando werden die in der Maske **Übersetzungsjobs** eingestellten Angaben zum Compilieren verwendet.

:GL (Binden)

Bei diesem Befehl werden die Angaben aus der Maske **Bindejobs** verwendet um aus einer USL/OBJ/RL ein lauffähiges Programm zu machen bzw. einen XEQ-Job, der als STR-Typ deklariert wurde, zu starten.

:GCL (Sachgebiet komplett übersetzen)

Durch dieses Kommando kann ein gesamtes Sachgebiet eines Projektes übersetzt und gebunden werden. Statt der Modulbezeichnung wird hier das Sachgebiet angegeben. Es werden zuerst die Übersetzungsjobs gestartet, dann die Bindejobs. **Achtung:** Bei diesem Aufruf werden evtl. definierte Folgejobs nicht mit generiert.

:GPR (Projekt komplett übersetzen)

Der Parameter MODUL ist hier nicht erforderlich. Es wird direkt mit dem Projektparameter begonnen. Dieser Befehl beinhaltet immer das Übersetzen und Binden aller Moduln. Es werden zuerst die Übersetzungsjobs gestartet, dann die Bindejobs. **Achtung:** Bei diesem Aufruf werden evtl. definierte Folgejobs nicht mit generiert.

:GCF (Übersetzung aus Execute-Datei)

Bei diesem Kommando werden die zu übersetzenden Moduln aus einer unnummeriert abgespeicherten Editordatei gelesen. Dieser Aufruf entspricht einem GC-Aufruf für jede Zeile der Editordatei.

:GLF (Binden aus Execute-Datei)

Bei diesem Kommando werden die zu bindenden Moduln aus einer unnummeriert abgespeicherten Editordatei gelesen. Dieser Aufruf entspricht einem GL-Aufruf für jede Zeile der Editordatei.

M P E / i X - E r w e i t e r u n g

Um MPE/iX Native-Mode Compiler zu benutzen muß an den Aufruf ein **X** angefügt werden. Die Aufrufe heißen dann:

:GCX, :GLX, :GCLX, :GPRX, :GCFX, :GLFX.

Aufruf von SDGEN aus anderen Subsystemen

Es steht ein TDP/EDITOR/QEDIT-Interface für den Aufruf zur Verfügung. Durch Eingabe des PROC-Befehls

PROC SDG,,*	in TDP
P SDG,S,*	in QEDIT/EDITOR

wird ebenfalls das Modul SDGEN gestartet. Die einzelnen Parameter für den Aufruf werden vom Benutzer erfragt.

Hierzu ist es erforderlich, daß in dem Editor eine Arbeitsdatei angelegt ist. Dies geschieht z.B. durch TEXT auf eine Datei.

M P E / i X - B e s o n d e r h e i t

Wenn der entsprechende Native-Mode Compiler benutzt werden soll, muß vorher manuell das JCW **SDXL** auf 1 gesetzt werden.

3.1 Optionen für die Generator-Aufrufe

A für Account

Der Job wird nicht mit dem JOBID aus der Datenbank generiert, sondern mit dem Account- und Benutzernamen aus der laufenden Sitzung. Die Dateinamen werden unverändert übernommen. Mit dieser Optionen können Spiegelaccounts implementiert werden.

C für Compile-only

Es erfolgt nur eine Übersetzung, kein LINK o.ä.. Die laufende Software wird nicht geändert.

G für Group

Der Job wird nicht mit dem JOBID aus der Datenbank generiert, sondern mit den Angaben aus der laufenden Session. Alle Dateinamen werden ohne Qualifizierung generiert. Diese Option ist bei JOBID's, die den PROD-Account betreffen die Voreinstellung. Wenn in der lokalen Gruppe eine COPYLIB vorhanden ist, so wird diese benutzt.

I für Installieren

Die Angabe dieser Option führt im PROD-Account zur Nutzung der Angaben aus JOBID. Der Aufruf mit dieser Option wird von SD protokolliert. Nach erfolgreicher Übersetzung wird ein Aufruf für SDKOPF generiert, um den aktuellen Kopf in die Datenbank zu laden.

L für List

Es wird der durch das Kommando angewählte Job generiert und parallel auf dem Bildschirm angelistet. Diese Option dient zum Testen von SD/3000.

N für Nachricht

Wenn in dem JOBID-Eintrag das Feld NACHRICHT auf N steht, so wird hiermit temporär die Funktion aktiviert. Es erfolgt bei positivem Jobende eine Meldung.

P für Print

Für das angegebene Modul wird die Source ausgedruckt. V/3000-Dateien werden mit FORMSPEC angedruckt. Sonderjobs werden normal angedruckt (keine Sourceliste). SEGMENTER/LINKEDIT-Bibliotheken werden mit dem entsprechenden LISTxxx-Befehl angedruckt.

S für Save

Der Job wird generiert, aber nicht an den Input-Spooler übergeben sondern auf eine permanente Datei mit dem Namen gleich dem Jobnamen geschrieben.

T für Test

Die Jobgenerierung wird auf dem Bildschirm angezeigt. Diese Option dient zu Testzwecken in SD/3000.

Die Optionen S, T und L werden nur ausgeführt, wenn sie im entsprechenden LOGON-Account aufgerufen werden, der in JOBID eingetragen ist oder der Benutzer die SM-Berechtigung hat.

3.2 Sonderoptionen SDGEN

Die Jobnamen der generierten Jobs bestehen immer aus einem Kennbuchstaben an erster Stelle und den ersten 7 alphanumerischen Stellen des Modulnamens. Es gibt folgende Kennbuchstaben:

C	Übersetzungsjobs
L	Bindejobs
P	Druckjobs
X	Executejobs
Y	Compile-only-Jobs
J	Sonderjobs vom Typ JOB

Die \$STDLIST-Ausgaben der Übersetzungs- und Bindejobs sowie die Listausgaben bei der P-Option erfolgen auf die Einheit LP mit der Priorität, die mit Hilfe des **SDLISTE-JCW**'s definiert wurde. Wurde das **SDLISTE-JCW** nicht gesetzt, so ist die voreingestellte Priorität 4.

Alle Listausgaben, die mit der P-Option erzeugt werden, erfolgen auf den MPE-File **SDLISTE**. Er kann durch ein entsprechenden Befehl umgelenkt werden.

Über das **JCW STDLISTDELETE** kann das automatische Löschen von fehlerfreien Compile/Binde-Jobs aktiviert werden. Das JCW ist wie folgt definiert:

STDLISTDELETE	Bedeutung
0	Spooldatei wird nicht gelöscht
1	Spooldatei wird gelöscht wenn fehlerfrei

Über das **JCW SDINCLUDE** wird gesteuert, in welcher Form der Compiler aufgerufen wird:

SDINCLUDE	Bedeutung
<= 0	Compiler Optionen mit INFO übergeben
>= 1	Compiler Optionen als Positionsparameter für ein UDC generieren

Über das **JCW SDXL** kann die Generierung von MPE/iX Native-Mode Compiler Aufrufen angesteuert werden. Steht für die entsprechende Programmiersprache kein NM-Compiler zur Verfügung, so wird nach der CM-Übersetzung das Programm mit dem OCTCOMP übersetzt. Das JCW ist dabei wie folgt definiert:

SDXL	Bedeutung
0	Compatibility oder CLASSIC-Mode Compiler
1	NM-Compiler bzw. OCTCOMP behandelt

Über das **JCW SDCLEAN** kann wahlweise ein CLEANxxx-Befehl in den SEGMENTER/LINKEDIT-Step eingebunden werden. Dieser CLEAN-Befehl wird nach dem Löschen der alten Moduln ausgeführt. Das JCW ist dabei wie folgt definiert:

SDCLEAN	Bedeutung
0	Kein CLEANxxx generieren
1	CLEANxxx Befehl generieren

M P E / i X B e s o n d e r h e i t

Der CLEANxxx-Befehl wird generiert für USL's und SL's im CLASSIC- oder Compatibility-Modus. Bei XL-Native-Mode Compiles wird auch ein CLEANxxx-Befehl für RL's und XL's generiert.

Über die **JVAR SD_PURGELINK** wird bei der Jobgenerierung das Löschen von Zieldateien gesteuert. Ist die Variable vorhanden, wird anstatt mit PURGE durch PURGELINK gelöscht. Das JCW ist dabei wie folgt definiert:

SD_PURGE	Bedeutung
nicht vorhanden	Normaler PURGE
vorhanden	PURGELINK

Über die **JVAR SD_JOBQ** kann ab MPEiX 6.0 eine JobQ für alle SD-Jobs angewählt werden. Die Variable muß den JobQ Namen enthalten.

Über die **JVAR SD_JQM** kann eine JQM Jobklasse und Priorität für alle SD-Jobs angewählt werden. Die Variable muß den JQM-Klasse und wahlweise Priorität enthalten.

3.3 Generator im Background

Wenn viele einzelne Compile oder LINK-Jobs gestartet werden müssen, jedoch nicht der Weg über ein Sachgebiet oder Projekt möglich ist, erfordern die einzelnen Aufrufe des Generators erhebliche Wartezeit. Aus diesem Grunde wurde mit der Version A.02.06 eine Generator-Background-Option implementiert. **Diese Option ist nur für MPE/iX-Systeme verfügbar.** Es müssen die entsprechenden XEQ-Dateien aus der Gruppe CMD.GLOBAL verwendet werden. Die Verwendung der SD-UDC-Dateien ist nicht möglich.

Der Anwender startet hierzu einen Backgroundjob mit dem Befehl:

:GB *{generatorbefehl} [{,optionen}]*

Als Generatorbefehle sind hierbei möglich: GC, GCX, GL und GLX.

Die Anwendung ist beschränkt auf Generatorkaufrufe ohne Nutzung der Install-Option. Bei späteren Gxxx-Aufrufen dürfen keine Optionen angegeben werden. Werden dort Optionen angegeben, so wird der Background-Generator nicht genutzt.

Der Background-Generator legt in der LOGON-Gruppe des Benutzers eine MSG-Datei mit dem Namen **SD<generatorbefehl>IN** an. Derartige Dateien dürfen also in der entsprechenden Gruppe nicht vorhanden sein. Zusätzlich wird beim Start des Batchjobs das Joblimit um 1 erhöht.

Der Background-Monitor kann durch folgenden Befehl wieder beendet werden:

:GBE *{generatorbefehl}*

Als Generatorbefehle sind wiederum möglich: GC, GCX, GL und GLX. Zusätzlich wird beim Beenden des Batchjobs das Joblimit um 1 erniedrigt und die entsprechenden MSG-Dateien werden wieder gelöscht.

3.4 Parameter Modifikation beim Generator

Beim Aufruf des Generators können die im COMPILE/LINK-Eintrag definierten Preprozessor-, Compiler- und LINK/PREP-Parameter durch Setzen einer entsprechenden CI-CHAR-Variablen ergänzt bzw. überschrieben werden. Dieses Feature steht nur auf MPE/iX-Systemen zur Verfügung. Die Namen der Variablen sind:

SD_ADD_POPT	für ADD Preprozessor-Option
SD_ADD_COPT	für ADD Compiler-Option
SD_ADD_LOPT	für ADD Link-Option
SD_REP_POPT	für REPLACE Preprozessor-Option
SD_REP_COPT	für REPLACE Compiler-Option
SD_REP_LOPT	für REPLACE Link-Option

4. Der Jobgenerator SDSTR

Um die Kennworte im System leicht änderbar zu machen, ist es erforderlich, in keiner JOB-Datei die Passworte fest einzubinden. Daneben ist es auch aus Datenschutzgründen nicht wünschenswert, das Passworte in normalen Datei unverschlüsselt gespeichert sind.

Alle Übersetzungs- und Bindejobs des Systems werden mit SD/3000 verwaltet. Theoretisch ist es ebenfalls möglich, die Operating-Jobs in SD/3000 einzutragen und sie dann mit dem Generator SDGEN zu starten. Der Arbeits- und Systemaufwand dafür erscheint jedoch zu hoch. Aus diesem Grunde entstand der Job-Generator SDSTR.

Er dient zum Starten von Jobs, die eine bzw. mehrere normale Jobkarten enthalten können. Beim Starten werden die Kennworte aus der verschlüsselten SD-Datenbank bzw. auf MPE/iX-Systemen aus der System-Directory in die Jobkarte eingefügt.

Die Jobkarte(n) innerhalb eines Jobs werden gekennzeichnet durch das Wort **JOB** in den Spalten 2 bis 4 eines Satzes. Damit ist es auch möglich, daß Jobs wieder andere Jobs erzeugen und dort ebenfalls die Passworte eingesetzt werden. Zusätzlich werden Kennworte in **REMOTE-HELLO** Zeilen eingesetzt.

Der Aufruf dieses Jobgenerators geschieht über das UDC bzw. die XEQ-Datei:

:S {jobdatei} bzw. :S "{jobdatei} {parameter}"

Dabei ist *{jobdatei}* eine normale MPE-ASCII-Datei die numeriert oder unnummeriert gespeichert ist.

Voreingestelltes Steuerkarten-Zeichen ist wie im MPE das !. Soll ein anderes Zeichen verwendet werden, so muß es als Parameter (z.B. , \$) angegeben werden.

Als Parameter sind alle im STREAM-Befehl möglichen Parameter erlaubt. Da die Sonderzeichen , und ; zur UDC-Steuerung verwendet werden (MPE/V Version), ist es erforderlich, daß bei Verwendung von Parametern sowohl der Dateiname als auch die Parameter in doppelte Hochkommata eingeschlossen werden.

Wenn in der Jobdatei bereits Kennworte gespeichert sind, so werden diese vom Generator temporär gelöscht und durch die aktuellen überschrieben.

Das Starten vonn Jobs in fremden Accounts ist nur Anwendern mit der SM-Berechtigung gestattet. Die Verwendung von fremden Benutzernamen erfordert die AM-Berechtigung.

Für das Arbeiten mit Testaccounts wurden die SDSTR_FORCE_xxx JVars implementiert. Bei Verwendung dieser Variablen mit dem Wert 1 werden wahlweise Account, Benutzer und/oder Gruppenname durch die Werte der aktuellen Anmeldung ersetzt. Die folgenden Variablen sind vorhanden:

SDSTR_FORCE_ACCOUNT
SDSTR_FORCE_USER
SDSTR_FORCE_GROUP

5. Das Kopf-Ladeprogramm SDKOPF

Mit Hilfe des Programms SDKOPF.PUB.GLOBAL werden die Angaben aus den Modulköpfen und die Anzahl Sourcezeilen mit ihren Untertypen in die SD/3000-Datenbank geladen. Die entsprechenden Sourcedatei dürfen mit Hilfe von COMPRESS, SQUISHER oder QEDIT komprimiert gespeichert sein und können sich auch in der SD-Source-Bibliothek befinden. Vor dem Laden von Moduln mit diesem Programm müssen die Übersetzungsangaben mit dem Modul SDMAINT eingetragen worden sein. SDKOPF bezieht alle Zuordnungsinformationen zwischen Modul / Sachgebiet / Projekt aus diesem Set.

Moduln vom Typ **BRW**, und **HELP** werden nicht bearbeitet. Die Modulköpfe von COPYLIB / INCLUDE-Gruppen-Membren werden mit einem speziellen Befehl geladen. Bei Moduln vom Typ **STR** wird der in der Preprozessor-Option eingetragene Sourcename verwendet, wenn dort eingetragen, sonst wird nach dem echten Dateinamen der Source gefragt.

Der Aufruf geschieht ähnlich wie bei den Gxx-Aufrufen über:

```
:Kxx {modul} [ , {proj} [ , {opt1} [ , {opt2}... ] ] ]
```

{modul}

Das zu bearbeitende Modul bzw. der Name der Executedatei bei dem KF-Aufrufen.

{proj}

Das Projekt, zu dem das Modul gehört Voreinstellung ist *. Es wird die Voreinstellung aus der Datenbank benutzt.

{optn}

In beliebiger Reihenfolge bis zu 4 Optionen, die durch einen Buchstaben gekennzeichnet werden.

Die verschiedenen Aufrufe im einzelnen:

:KM (einzelnes Modul)

Bei diesem Kommando wird nur ein einzelnes Modul geladen.

:KCL (Sachgebiet komplett laden)

Durch dieses Kommando werden alle Moduln eines Sachgebiets komplett geladen.

:KPR (Projekt komplett laden)

Der Parameter MODUL ist hier nicht erforderlich. Es wird direkt mit dem Projektparameter begonnen. Dieser Befehl beinhaltet immer das Laden aller Moduln.

:KF (Moduln über Execute-Datei)

Bei diesem Kommando werden die Namen der einzelnen Moduln aus einer unnummeriert abgespeicherten Execute-Datei gelesen. Sonst entspricht der Aufruf dem KM-Aufruf.

:KC (COPYLIB / INCLUDE-Gruppe laden)

Bei diesem Kommando werden nur die Member einer COPYLIB / INCLUDE-Gruppe geladen.

:KCF (COPYLIB / INCLUDE-Gruppe über Execute-Datei)

Bei diesem Kommando werden die Namen der einzelnen COPYLIB's / INCLUDE-Gruppen aus einer unnummeriert abgespeicherten Executedatei gelesen. Sonst entspricht der Aufruf dem KC-Aufruf.

Zusätzlich steht ein TDP/QEDIT/EDITOR-Interface für den Aufruf zur Verfügung. Durch Eingabe des PROC-Befehls

PROC SDK,,*	bei TDP
P SDK,S,*	bei QEDIT/EDITOR

wird ebenfalls das Modul SDKOPF gestartet. Die Parameter für den Aufruf werden vom Benutzer erfragt.

Hierzu ist es erforderlich, daß im Editor eine Arbeitsdatei angelegt ist. Dies geschieht z.B. durch TEXT auf eine Datei.

5.1 Optionen für den Aufruf von SDKOPF**K für Kopf drucken**

Es werden keine Angaben in die SD/3000 Datenbank geladen. Alle Köpfe werden über den File SDLISTE ausgedruckt.

L für List

Das Laden wird normal durchgeführt, dabei werden alle gelesenen Zeilen **nach** ihrer Bearbeitung über den File **SDLISTE** angedruckt.

T für Test

Das Laden wird normal durchgeführt. Es wird dabei eine Statistik über die in den jeweiligen Sets gelöschten und neu eingestellten Sätze ausgegeben. Diese Option ist zum Test von SD/3000 und zur Fehlersuche nützlich.

5.2 Ladelogik

5.2.1 Allgemeines Vorgehen

Alle Zeilen werden intern in Großbuchstaben übersetzt. Die einzelnen Zeilen werden anhand eines < und eines Schlüsselwortes erkannt. Neben dem < ist ab der Version A.02.05 das Zeichen ~ für Moduln vom Sourcetyp JOB und STR erforderlich. Die definierten Schlüsselworte lassen sich anpassen bzw. ist es möglich pro Schlüsselwort bis zu 9 alternative Schlüsselworte zu definieren. Die Möglichkeiten hierzu sind im Konfigurationshandbuch beschrieben.

Gültige Schlüsselworte in der deutschen Standardversion sind:

INHALT	BESCHREIB	STATUS	VERSION
MASCHINE	RECHNER	UEBERSETZER	PROGRAM
EINGABE PA	AUSGABE PA	EINGABE DA	AUSGABE DA
DATEI REF	MASKEN REF	COPY MODUL	EXTERNE PR
COMMON	NAME	BEMERKUNG	SCHLAGWORT

Bei Moduln vom Typ JOB werden die ersten 8 Stellen und bei Moduln vom Typ COBx die ersten 6 Stellen überlesen. Leerzeilen werden immer ignoriert. Die Stellen 1-3 jeder Zeile werden ignoriert. Bei Beginn eines Ladelaufes werden zunächst alle Angaben, bis auf die Historie, zu dem Modul aus der Datenbank gelöscht. Danach werden die neuen Sätze eingetragen.

5.2.2 Kopfanfang / Kopfende

Kopfanfang und Kopfende werden in der deutschen Standardversion gekennzeichnet durch 10 * ab Stelle 9 im Satz. Die Zeichen zur Kennzeichnung von Kopfanfang und Kopfende sind konfigurierbar. Entsprechende Details sind im Konfigurationshandbuch beschrieben. Der Kopfanfang muß in den ersten 100 Zeilen gefunden werden.

5.2.3 Modulname

Der Modulname darf gesperrt geschrieben sein. Er sollte mit dem Modulnamen aus dem Übersetzungseintrag übereinstimmen. Stimmt er nicht überein, wird beim Laden des Kopfes eine Warnung ausgegeben. Bei 3 Leerzeichen wird **Ende der Zeile** angenommen.

5.2.4 Inhaltsangabe

Die Inhaltsangabe kann maximal 9.999 Zeilen lang sein. Leerzeilen werden nicht abgespeichert. Führende Leerstellen werden eliminiert. Das Zeilenende wird gekennzeichnet durch einen * oder >>.

5.2.5 Versionen-Zeile

Die Version muß als maximal 10-stelliges Feld eingetragen sein. Eine Versionszeile wird dadurch erkannt, daß die ersten 8 Stellen der Zeile mindestens einen Punkt enthalten und danach ein Datum in der folgenden Form folgt:

$$\{t\} t [. | /] \{m\} m [. | /] \{j\} j j$$

Das Datum wird intern auf Gültigkeit überprüft.

Der Autor der Änderung wird als maximal 20-stelliges Feld abgespeichert.

Nach einer Versionszeile folgt die Änderungsbeschreibung. Nur neue Änderungsbeschreibungen werden in die Datenbank aufgenommen.

5.2.6 Maschinentyp

Der Maschinentyp wird bis zu einem Sonderzeichen oder maximal 10 Stellen als Text übernommen. Es finden keine logischen Überprüfungen dieses Texts statt.

5.2.7 Übersetzer

Die Übersetzerangabe muß mit dem Eintrag im COMPILE-Set der SD-Datenbank übereinstimmen. Richtig erkannt werden:

FORTRAN oder FORTRAN/3000	für FTN4
FORTRAN77, FORTRAN-77 oder FTN	für FTN7
COBOLII, COBOL2, COBOL74	für COB2
COBOLiX, COBOL2X, COBOL85	für COBX
MFCOBOL, MSCOBOL, ACUCOBOL	für COBX
SPL	für SPL
PAS	für PAS
TPL oder TRANSACT	für TPL
FASTRAN, TPL oder TRANSACT	für TPLF
RPL oder REPORT	für RPL
MAKECAT	für CAT
GENCAT	für GCAT
QUERY	für JOB
QDESIGN	für QUIK
QUIZ	für QUIZ
QDD	für QDD
CC	für CC

Wenn im COMPILE-Set ein Sonderjob (Source-Typ = **STR**) angegeben ist, so wird jeder gültige Compiler akzeptiert.

Die verschiedenen COBOL, FORTRAN und TRANSACT-Compiler können genutzt werden ohne den Kopf zu ändern (3 Buchstaben sind signifikant). Als Alternative zu COBOL sind auch MFCOBOL, MSCOBOL und ACUCOBOL erlaubt.

5.2.8 Programmaufruf

Die RUN-Befehle dürfen wahlweise mit einem **:** beginnen. Als gültige Programmaufrufe werden erkannt und umgeschlüsselt:

RUN	in RUN
RUN TRANSACT und TR	in TRUN
RUN REPORT und RR	in RRUN
RUN RAPID	
bei Sourcety=TPL	in TRUN
bei Sourcety=RPL	in RRUN
CALL	in CALL
PROC	in PROC (für Transact Programme)
ON	in INT
CREA	in CRAC
BLOCK	in BLDA
POPN	in POPN
STREAM	in STR

5.2.9 Eingabe-Parameter/Ausgabe-Parameter

Werden nicht geprüft und abgespeichert. Das Ausfüllen ist dennoch für die Programmierer sinnvoll, damit diese Angaben z.B. durch die Ladeoption **K** gesammelt angedruckt werden können. Damit entsteht ein generelles Handbuch für den Aufruf von Unterprogrammen.

5.2.10 Eingabe-Dateien/Ausgabe-Dateien (altes Format)

Es darf maximal 1 Dateinamen pro Zeile eingetragen sein. Ein Dateiname darf aus Buchstaben, Zahlen und den Sonderzeichen + - * / \ ? ' # % & @ . bestehen. Andere Sonderzeichen werden ignoriert. Ein *BLANK* bedeutet: Ende des Dateinamens. Ein Dateiname darf maximal 28 Stellen lang sein.

Wenn der Dateiname in SD dem **Projekt SYS** und dem **Sachgebiet UT** zugeordnet ist, so wird er als System-Utility-File übernommen und braucht nicht in jedem neuen Projekt wieder definiert zu werden (z.B. die Fortran-Logical-Units: FTN01 bis FTN30, CONSOLE, STDLIST, STDIN etc.).

Ist ein Dateiname nur in einem anderen Projekt definiert, so wird der Benutzer gefragt, ob er diesem zugeordnet werden soll.

5.2.11 Dateireferenzen (neues Format)

Es darf maximal 1 Dateinamen pro Zeile eingetragen sein. Ein Dateiname darf aus Buchstaben, Zahlen und den Sonderzeichen + - * / \ ? ' # % & @ . bestehen. Andere Sonderzeichen werden ignoriert. Ein *BLANK* bedeutet: **Ende des Dateinamens**. Ein Dateiname darf maximal 28 Stellen lang sein.

Die Buchstaben nach dem *BLANK* werden als Referenztyp abgespeichert. Maximal können 10 Buchstaben folgen. Diese Buchstaben können vom Anwender frei vergeben werden. Der COBOL-Source-Analyser vergibt die folgenden Referenztypen:

C	für	CONTROL	D	für	DELETE
G	für	GET	I	für	INFO
L	für	LOCK	M	für	MERGE
O	für	OPEN	P	für	PUT
S	für	SORT	U	für	UPDATE

Wenn der Dateiname in SD dem Projekt SYS und dem Sachgebiet UT zugeordnet ist, so wird er als System-Utility-File übernommen und braucht nicht in jedem neuen Projekt wieder definiert zu werden (z.B. die Fortran-Logical-Units: FTN01 bis FTN30, CONSOLE, STDLIST, STDIN etc.).

Ist ein Dateiname nur in einem anderen Projekt definiert, so wird der Benutzer gefragt, ob er diesem zugeordnet werden soll.

5.2.12 Externe Programme

In einer Zeile mit externen Referenzen können mehrere Moduln aufgeführt werden. Die Zeile wird beendet beim Auftreten von >, *, - oder :. Damit können wahlweise zusätzliche Kommentare hinter einzelne Moduln geschrieben werden. Ein externer Modulname darf aus Buchstaben, Zahlen, _ und ' bestehen. Er darf maximal 16 Zeichen lang sein. Wenn das Modul als Intrinsic vom Typ HPxx definiert ist, so wird seine Referenzierung nicht abgespeichert. In diesem Fall erscheint eine besondere Meldung.

Wenn das Modul als sonstiges Intrinsic definiert ist, so wird automatisch eine Referenz in das Projekt INTR erzeugt. In diesem Fall erscheint eine besondere Meldung. Ist das referenzierte Modul in einem anderen Projekt definiert, so wird der Benutzer gefragt, ob er diesem zugeordnet werden soll.

Sollen Programme, die mit CREATE oder CREATEPROCESS aufgerufen werden, dokumentiert werden, so sollten diese Programme als Intrinsic's eines beliebigen Typs mit einem entsprechenden Kurztext definiert werden. Der Intrinsicstyp darf nicht mit **HP** beginnen. In dem aufrufenden Programm wird dann nur der Modulname (ohne Qualifizierung auf Gruppe und Account) als externes Modul eingetragen.

5.2.13 COMMON-Angaben

In einer Zeile darf nur eine COMMON-Referenz stehen. Ein COMMON-Name kann nur aus Buchstaben, Zahlen, _, - oder ' bestehen und darf maximal 16 Zeichen lang sein. Das Auftreten eines Sonderzeichens oder eines *BLANK* bedeutet: Ende des Namens.

Ist der referenzierte COMMON in einem anderen Projekt definiert, so wird der Benutzer gefragt, ob er diesem zugeordnet werden soll.

5.2.14 Schlagworte

In einer Zeile können mehrere Schlagworte aufgeführt werden. Das durchsuchen der Zeile wird beendet beim Auftreten von > oder *. Ein Schlagwort darf aus Buchstaben, Zahlen, -, _ und ' bestehen. Es darf maximal 16 Zeichen lang sein.

Wurde das Schlagwort nicht vorher in SD/3000 definiert, so wird die Referenz nicht abgespeichert. Der Benutzer erhält eine Warnung.

5.2.15 Masken Referenzen

In einer Zeile darf nur eine Maskenreferenz stehen. Die Zeichen bis zum Auftreten des ersten *BLANK* werden als Maskenname interpretiert. Der Maskenname ist maximal 16-stellig und darf Buchstaben, Zahlen und _ enthalten.

Die danach folgenden Zeichen geben den Namen der Sourcemaskendatei an. Der Dateiname ist maximal 28-stellig und darf Buchstaben, Zahlen und . enthalten. Wenn Sie mit NLS-Maskendateinamen arbeiten, geben Sie bitte in allen Programmen einen identischen Namen (z.B. die Nummer 000) an. Damit wird die logische Referenzierung nur auf diese Datei hergestellt, die Referenzen auf die anderen Sprachnummern sind jedoch logisch identisch.

Nur die Kombination Maskenname und Maskendateiname ist eindeutig und wird abgespeichert.

5.2.16 COPY Moduln / INCLUDE-Dateien

In einer Zeile darf nur eine COPYLIB / INCLUDE-Referenz stehen. Die Zeichen bis zum Auftreten des ersten BLANK werden als COPYLIB-Member interpretiert. Der Membername ist maximal 8-stellig und darf Buchstaben, Zahlen und - enthalten. Bei INCLUDE-Dateien muß als MEMBER-Name immer ein @ angegeben werden.

Die danach folgenden Zeichen geben den Namen der COPYLIB an. Der Dateiname ist maximal 28-stellig und darf Buchstaben, Zahlen und . enthalten. Bei INCLUDE-Dateien ist dies der Name der INCLUDE-Datei. Wird der COPYLIB-Dateiname nicht voll qualifiziert angegeben, wird er anhand des JOBID's ergänzt.

Nur die Kombination COPYLIB-Member/INCLUDE-Datei und COPYLIB-Name ist eindeutig und wird abgespeichert.

5.2.17 Bemerkungen

Wenn das Abspeichern von Bemerkungen eingeschaltet wurde, werden die Bemerkungen mit maximal 9.999 Zeilen abgespeichert. Leerzeilen werden nicht abgespeichert. Führende Leerstellen werden eliminiert. Das Zeilenende wird gekennzeichnet durch einen * oder >>.

5.3 Der automatische Kopflader (SDKALL)

Zum automatischen Laden aller Köpfe der Moduln, deren Source seit dem letzten Laden modifiziert wurde (Sourcedatei neu angelegt) dient das Utility SDKALL. Es ruft für jedes modifizierte Modul den Kopflader SDKOPF auf. Moduln vom Typ HELP, BRW und STR werden nicht bearbeitet. Dateien vom Typ VIEW werden geladen, wenn das Änderungsdatum der Maskendatei nicht mit dem Datum der ersten Maske in SD/3000 übereinstimmt.

In der zweiten Phase des automatischen Ladens werden alle modifizierten COPYLIB's geladen. Das Laden der COPYLIB's wird ausgelöst, wenn das Änderungsdatum nicht mit dem Datum des ersten Members in SD/3000 übereinstimmt.

Es sind ein UDC bzw. eine XEQ-Datei für den Aufruf vorbereitet:

:KMOD {projekt}

zum Bearbeiten aller Moduln eines Projekts.

Es bietet sich an, :KMOD für alle Projekte in denen laufend entwickelt wird regelmäßig nachts zeitgesteuert in einem Job aufzurufen. Damit ist dann sichergestellt, das keine Einzelaufrufe mit :KM vom entsprechenden Programmierer vergessen wurden.

6. Das Druckprogramm SDPRINT

6.0.1 Aufruf

Das Programm SDPRINT ist das zentrale Auswertungsprogramm der SD/3000-Datenbank. Es kann wie folgt gestartet werden:

1. vom Betriebssystem mit

:P (als UDC/XEQ-Datei) oder mit
:RUN SDPRINT.PUB.GLOBAL

2. vom Verwaltungsprogramm SDMAINT

über Funktionstaste

3. aus TDP/EDITOR/QEDIT über den PROC-Befehl:

PROC SDP,*,* in TDP
P SDP,S,*,* in QEDIT/EDITOR

6.0.2 Befehlseingabe

Die Listenanwahl erfolgt interaktiv über Kommandos. Die Befehle können mit Ihrem vollen Namen oder einer Abkürzung eingegeben werden. Die gültige Abkürzung ist in der Befehlsübersicht 'HELP' in Großbuchstaben angegeben. Alle Kommandos werden auf den Prompt **SD-P** > eingegeben.

In der Regel ist die Eingabe von Selektionsangaben möglich. Die Selektionsparameter werden in einem listenspezifischen Dialog interaktiv abgefragt. Sie werden immer auf einer Seite 0 der Liste mit angedrückt.

6.0.3 Listenformat

Die Listen haben entweder eine Breite von 80 oder 100 Zeichen. Die Länge einer Druckseite ist konfigurierbar. Sie ist in der deutschen Standardversion mit 65 Zeilen voreingestellt.

Der Listenkopf hat immer einen einheitlichen Aufbau. In der ersten Zeile befindet sich immer links das Kürzel **SD/3000**, in der Mitte der Titel der Liste und rechts die Seitennummer. Die zweite Titelzeile enthält links das Datum und die Uhrzeit der Anwahl, bei einigen Listen in der Mitte einen 2. Titelblock und rechts den Namen des Benutzers, der die Liste angewählt hat.

6.0.4 Schlüsselbegriffe

Fragen nach einem Schlüsselbegriff müssen, falls keine Eingabe im LISTF-Format möglich ist, mit der exakten Eingabe des Begriffes beantwortet werden.

Die Eingabe der Schlüsselbegriffe kann in Groß- oder Kleinbuchstaben erfolgen. Sie werden intern immer in Großbuchstaben übersetzt.

6.0.5 LISTF-Format (generische Eingabe)

Bei vielen Selektionsangaben, Schlüsselbegriffen und auch normalen Auswahlangaben ist die Eingabe eines generischen Schlüssels möglich. Hierfür wurde das auf der HP3000 allgemein bekannte LISTF-Format gewählt.

Beim LISTF-Format können die Zeichen @, # und ? als universell ersetzbare Zeichen verwendet werden. Die Zeichen haben die folgenden Bedeutungen:

- @
ersetzt ein oder mehrere alphanumerische Zeichen. Steht das Zeichen allein, so werden alle vorhandene Informationen ausgegeben.
- #
ersetzt ein numerisches Zeichen
- ?
ersetzt ein alphanumerisches Zeichen

Die Platzhalterzeichen können an beliebiger Stelle eines Begriffes angegeben werden.

6.0.6 Ja/Nein-Fragen

Auswahlfragen, die mit Ja oder Nein beantwortet werden müssen, werden generell nur durch die Eingabe des ersten Buchstabens beantwortet (**J** oder **N**). Die Eingabe kann sowohl in Groß- als auch in Kleinbuchstaben erfolgen.

6.0.7 Voreinstellungen

Das zentrale Druckprogramm SDPRINT bietet dem Anwender wie auch in den sonstigen Modulen des SD/3000-Systems in der Regel Voreinstellungen an, die dem üblichen Tagesanfragen entsprechen.

Die Voreinstellungen werden immer in der Frage durch '**RETURN=xxx**' angezeigt. Durch Drücken der RETURN-Taste werden diese Voreinstellungen als aktuelle Eingabe übernommen.

6.0.8 Abbruch eines Befehls

Der Abbruch eines aufgerufenen Kommandos ist sowohl während der Eingabe der Selektionsparameter, der Such- und Sortierphase, als auch der Ausgabephase jederzeit durch Drücken von **CONTROL-Y** möglich. Intern wird das laufende Kommando sauber abgeschlossen und in den Kommando-Eingabemodus zurückgekehrt.

Das gesamte Druckprogramm darf jederzeit durch Drücken der **BREAK**-Taste abgebrochen werden. Störungen können dadurch nicht auftreten.

6.1 Allgemeine Befehle

6.1.1 Programm beenden

Das Modul SDPRINT wird verlassen durch Eingabe von:

END oder E

Es sind keine weiteren Angaben erforderlich.

Beim Beenden des Programms wird die verbrauchte CPU-Zeit mit ausgegeben.

6.1.2 Hilfe

Im Befehls-Eingabemodus kann eine Übersicht über alle möglich Befehle durch Eingabe von:

HELP oder H

angefordert werden. Durch Eingabe von

Help {befehl}

erfolgt eine ausführliche Erklärung des entsprechenden Befehls. Durch

Help All

wird der komplette Hilfskatalog ausgegeben.

6.1.3 Ausgabeeinheit definieren

Die Ausgabe der Andrucke kann auf allen SD/3000-internen Ausgabeeinheiten erfolgen. Beim Aufruf des Programms ist die als \$STDLIST definierte Ausgabeeinheit voreingestellt.

Durch

OUT {einheit}

kann direkt eine Einheit angewählt werden Wird nur

OUT oder O

eingegeben, so wird eine Liste aller möglichen Ausgabeeinheiten angezeigt. Zusätzlich wird der Name der derzeit aktiven Ausgabeeinheit mit ausgegeben.

Benötigen Sie andere oder zusätzliche Ausgabeeinheiten oder Ausgabeeinheiten mit anderen Parametern (Kopienanzahl, Spoolpriorität, ENVIRONMENT-Datei etc.), so wenden Sie sich bitte an Ihren SD/3000-Manager. Er ist jederzeit in der Lage, neue Ausgabeeinheiten zu definieren.

6.1.4 Vorlaufseite unterdrücken

Die Ausgabe der Vorlaufseite von allen SD/3000 Listanwahlen läßt sich unterdrücken mit:

NOPAGE0

6.1.5 Vorlaufseite aktivieren

Die Ausgabe der Vorlaufseite von allen SD/3000 Listanwahlen läßt sich nach einem NOPAGE0 wieder reaktivieren mit:

PAGE0

6.1.6 Anzahl Zeilen pro Seite einstellen

Die Anzahl Zeilen pro Seite für alle SD/3000 Listanwahlen läßt sich mit dem folgenden Befehl einstellen:

PAGELEN {anzahl zeilen}

Die Anzahl Zeilen darf zwischen 20 und 200 liegen.

6.2 Andrucke der SD/3000 internen Stammdaten

6.2.1 Übersicht der Ausgabeeinheiten

Alle von den SD/3000-Moduln verwendeten Ausgabeeinheiten müssen in der SD/3000-Datenbank definiert sein. Bei dieser Definition werden alle für die aktuelle Ausgabe erforderlichen Parameter des FILE-Befehls der Listendatei hinterlegt. Zusätzlich muß eine Freitextbezeichnung der Ausgabeeinheit eingetragen werden.

Die Liste enthält alle in SD/3000 definierten Ausgabeeinheiten mit dem MPE-Klassennamen, der MPE-LU, dem Typ der Einheit, einem evtl. vorhandenen speziellen FILE-Befehlsteil sowie dem Freitext. Sie wird sortiert nach der Ausgabeeinheit.

Die Übersicht wird aufgerufen durch

PRINTER oder PRI

6.2.2 Übersicht der Bildschirmeinheiten

Alle Bildschirme im System, von denen Moduln des SD/3000-Systems aufgerufen werden dürfen, müssen in der SD/3000-Datenbank definiert sein. Diese Definition kann nur durch das SDLU-JCW überschrieben werden. Zu der Definition gehört ein Freitext, in dem der Name des oder der zugehörigen Benutzer eingetragen werden sollte. Außerdem ist jedem Bildschirm ein voreingestellter Drucker zugeordnet. Diese Voreinstellung wird z.Zt. nur in dem Verwaltungsprogramm SDMAINT beim Gebrauch der '*DRUCKE MASKE*' Funktionstaste genutzt.

Die Liste enthält alle in SD/3000 definierten Bildschirmeinheiten mit der MPE-LU-Nummer, dem Typ, dem voreingestellten Drucker sowie dem zugehörigen Freitext. Sie wird nach LU-Nummern sortiert.

Sie wird aufgerufen durch

TERMINAL oder T

6.2.3 ACCOUNT-Definitionen

Die Liste enthält alle in SD/3000 definierten ACCOUNT's mit dem voreingestellten Projekt, dem Account und der Gruppe für die INCLUDE-Bibliotheken und Hinweisen auf eventuell aktivierte Schutzmechanismen sortiert nach dem Namen des Produktionsaccounts. Sind für einen PRODUKTIONS-Account TEST-Accounts eingetragen, so werden diese ebenfalls mit angedruckt.

Folgende Hinweise auf Schutzmechanismen sind möglich:

- *Account ist ein INSTALL-Account*
- *Programmänderungen nur durch den Sachgebiets-Verantwortlichen.*

Die Liste wird aufgerufen durch

ACCOUNT oder A

6.2.4 Datei Zugriffsroutinen

Die Liste enthält alle in SD/3000 definierten Zugriffsroutinen für den COBOL-Analyser.

Zu jeder Routine wird die Parameternummer des Dateinamens und der zu generierende Zugriffskode mit ausgegeben. Die Liste wird sortiert nach dem Namen der Routinen. Sie wird aufgerufen durch:

ACCESS oder ACCE

6.3 Andrucke der Stammdaten

6.3.1 Projektübersicht

Ein Projekt ist der oberste Begriffe im SD/3000-System. Alle anderen Definitionen (Jobs, Dateien etc.) zu einem in SD/3000 abzubildenden System müssen einem Projekt zugeordnet sein.

Die Projektliste enthält alle Projekte, die in SD/3000 eingetragen sind mit ihrer Kurzbeschreibung sortiert nach dem Projektkürzel. Sie wird aufgerufen durch:

PROJECT oder PR

6.3.2 Sachgebietsübersicht

Das Sachgebiet ist eine Untermenge eines Projekts. Es hat reine Dokumentationsfunktion. In der Regel sind die Eintragungen in SD/3000 einem Sachgebiet zugeordnet. Die Eintragung läßt sich jederzeit problemlos ändern.

Die Sachgebietsübersicht enthält alle Sachgebiete zu einem ausgewählten Projektbereich. Die Liste wird sortiert nach Projekt und Sachgebiet. Sie enthält wahlweise den Sachgebiets-Verantwortlichen und seinen Stellvertreter. Wenn ein Sachgebiet für Programm-Änderungen und/oder SD-Änderungen gesperrt ist, so erfolgt ein separater Hinweis. Bei Projektwechsel wird ein Seitenwechsel durchgeführt. Die Liste wird aufgerufen durch:

CLASS oder CL

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebietsverantwortliche mit andrucken (J / N, Return=N) ?

Bei **J** werden, falls vorhanden, der Sachgebietsverantwortliche und sein Stellvertreter mit Namen und Benutzerkürzel ausgegeben.

Sachgebiets-Sperren mit andrucken (J / N, Return=N) ?

Bei **J** werden, falls vorhanden, Sperren für Programm- und/oder Dokumentationsänderungen mit ausgegeben.

6.3.3 JOBID Verzeichnis

Hinter einem JOBID verbergen sich in SD/3000 die Informationen eines JOB-Befehls für den Generatorteil. Bei der Generierung von Jobs kann der Benutzer jeweils entscheiden, ob er diese Stamminformationen oder die Definitionen der augenblicklichen Anmeldung nutzen will.

Die Liste gibt alle gültigen JOBID's aufsteigend sortiert aus. Sie enthält die Informationen JOBID, Benutzer, Gruppe, Account, NOTIFY-Option, Inpri und Expri. Wenn für Benutzer, Gruppe oder Account ein Kennwort definiert ist, so wird in der Spalte **PASS** ein * angedruckt. Der Andruck von Kennworten ist mit dem Modul SD-P aus Sicherheitsgründen nicht möglich. Die Liste wird aufgerufen durch

JOBID oder J

6.3.4 Dateiverzeichnis sortiert nach Namen

Alle Dateinamen, die mit SD/3000 dokumentiert werden sollen, müssen separat in der SD/3000-Datenbank hinterlegt werden. Diese Eintragung hat reine Dokumentationsfunktionen.

Die Liste enthält die in SD/3000 eingetragenen Dateinamen mit ihrem Typ, ihrem Status, dem zugehörigen Projekt und Sachgebiet sowie einem beschreibenden Freitext. Die Liste wird nach Dateinamen sortiert. Sie wird aufgerufen mit:

FILENAME oder FN

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Folgende Eingaben sind möglich:

- ein Dateiname
- ein Dateiname im LISTF-Format (generischer Name)
- !Dateiname für IMAGE/3000 Dateien, deren Name Sonderzeichen enthalten, die auch im LISTF-Format vorkommen (#, @ oder ?)

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generisches Kürzel im LISTF-Format eingegeben werden.

6.3.5 Dateiverzeichnis sortiert nach Projekten

Alle Dateinamen, die in Systemen verwandt werden, die mit SD/3000 dokumentiert werden sollen, müssen separat in der SD/3000-Datenbank hinterlegt werden. Diese Eintragung hat reine Dokumentationsfunktionen.

Die Liste enthält die in SD/3000 eingetragenen Dateinamen mit ihrem Typ, ihrem Status sowie einem beschreibenden Freitext. Sie wird nach Projekt und Dateinamen sortiert. Bei Wechsel des Projekts erfolgt ein Seitenwechsel. Der Aufruf erfolgt mit:

FILEPROJECT oder FP

Selektionsangaben:

Dateiname (['!'] Datei/LISTF-Format) ?

Folgende Eingaben sind möglich:

- ein Dateiname
- ein Dateiname im LISTF-Format (generischer Name)
- !Dateiname für IMAGE/3000 Dateien, deren Name Sonderzeichen enthalten, die auch im LISTF-Format vorkommen (#, @ oder ?)

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generisches Kürzel im LISTF-Format eingegeben werden.

6.3.6 COMMON Verzeichnis

In SD/3000 können globale Datenbereiche, die ähnliche Bedeutung wie temporäre Dateien haben, dokumentiert werden. Diese Datenbereiche werden in FORTRAN üblicherweise als COMMON's bezeichnet. Die Datenbereiche werden nur zu reinen Dokumentationszwecken genutzt. Sie haben keine Steuerungsfunktionen.

Diese Liste enthält alle in SD/3000 eingetragenen COMMON-Blöcke mit ihrem Typ und der Kurzbeschreibung. Die Liste wird sortiert nach Projektkürzel und COMMON-Name. Bei Wechsel des Projekts erfolgt ein Seitenwechsel. Der Aufruf erfolgt mit:

COMMON oder CO

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

6.3.7 Maskenverzeichnis

Das Maskenverzeichnis enthält zu einer V/3000 Maskendatei alle Masken mit ihrer Kurzbeschreibung. Es wird sortiert nach Maskendateiname und Maskenname. Bei Wechsel der Maskendatei wird ein Seitenwechsel durchgeführt. Die Liste wird aufgerufen durch:

FORMFILE oder FORMF

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein Dateiname oder ein generischer Dateiname im LISTF-Format eingegeben werden.

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebietskürzel oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

6.3.8 COPYLIB / INCLUDE-Gruppen Verzeichnis

Das COPYLIB / INCLUDE-Verzeichnis enthält die in einer COPYLIB / INCLUDE-Gruppe gespeicherten Member mit ihrer Kurzbeschreibung, der aktuellen Versionsnummer, Datum und Autor. Wahlweise kann auch die Versionshistorie mit angedruckt werden. Die Liste wird sortiert nach Bibliotheksname und Membername. Bei Wechsel der COPYLIB / INCLUDE-Gruppe wird ein Seitenwechsel durchgeführt. Die Liste wird aufgerufen durch:

COPYLIB oder COPYL

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein Dateiname oder ein generischer Dateiname im LISTF-Format eingegeben werden. Bei INCLUDE-Gruppen muß hier immer ein ! gefolgt von dem Namen der INCLUDE-Gruppe angegeben werden.

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebietskürzel oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Historie mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so wird die komplette Änderungsdocumentation zu dem Member angedruckt.

6.3.9 Schlagwort Verzeichnis

Das Schlagwortverzeichnis enthält alle in SD/3000 gespeicherten Schlagworte mit ihrer Kurzbeschreibung. Die Liste wird sortiert nach Namen des Schlagworts. Sie wird aufgerufen durch:

KEYWORD oder K

6.3.10 Intrinsic Verzeichnis

INTRINSIC's sind in SD/3000 allgemeine Unterprogramme die

- beim Typ HPxx:
nicht in die Dokumentation übernommen werden sollen
- bei sonstigen Typen:
über einen schnelleren Ladeweg (SDKOPF) mit einem speziellen Projekt in die Dokumentation übernommen werden sollen.

Es werden alle Intrinsic-Typen oder nur die eines ausgewählten Typs ausgegeben. Die Liste wird nach den Intrinsic-Namen sortiert. Sie wird aufgerufen durch:

INTRINSIC oder I

Selektionsangaben:

Intrinsic Typ (<typ> oder @, Return=@) ?

Die Liste kann entweder für nur einen Intrinsic-Typ oder für alle (durch Eingabe von @) ausgewählt werden.

6.4 Andrucke der Generierungsinformationen für Jobs

6.4.1 Übersetzungsjob Verzeichnis

Diese Liste enthält die Angaben zu Moduln, die für die Übersetzung erforderlich sind. Sie wird sortiert nach Projekt, Sachgebiet und Modulname. Sie wird aufgerufen durch:

CMODUL oder CM

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Modulname (Modul / LISTF-Format) ?

Es kann eine Modulbezeichnung oder ein generischer Name im LISTF-Format eingegeben werden.

Zieldatei-Information mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden alle Angaben über die Zieldatei in der Liste unterdrückt.

Zugriffsschutz-Information mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden die Angaben über den Zugriffsschutz der Zieldatei nicht angedruckt.

Optionen mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden Preprozessor/Compile/LINK/PREP-Optionen und RL-Angabe nicht mit angelistet.

Texteinträge mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden die evtl. vorhandenen Texte nicht angedruckt.

6.4.2 Bindejob Verzeichnis

Diese Liste enthält die Angaben zu Moduln, die für das Binden erforderlich sind. Sie wird sortiert nach Projekt, Sachgebiet und Modulname. Sie wird aufgerufen durch:

LMODUL oder LM

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Modulname (Modul / LISTF-Format) ?

Es kann eine Modulbezeichnung oder ein generischer Name im LISTF-Format eingegeben werden.

Zieldatei-Information mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden alle Angaben über die Zieldatei in der Liste unterdrückt.

Zugriffsschutz-Information mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden die Angaben über den Zugriffsschutz der Zieldatei nicht angedruckt.

Optionen mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden LINK/PREP-Optionen und RL-Angabe nicht mit angelistet.

Texteinträge mit andrucken (J/N, Return=J) ?

Wird hier ein **N** eingegeben, so werden die evtl. vorhandenen Texte nicht angedruckt.

6.5 Dokumentationen

6.5.1 Modul Dokumentation

In dieser Liste werden zentral die Informationen aus dem entsprechenden Modulkopf plus die zusätzlichen Informationen über die Referenzbildung dargestellt.

Der Standardteil der Liste enthält:

- den Modulnamen,
- die Beschreibung der letzten Version,
- Source- und RUN-Typ,
- voll qualifizierter Name der Sourcedatei und
- Kurzbeschreibung.

Die folgenden Zusatz Informationen können ausgegeben werden:

- die gesamte Änderungshistorie mit Beschreibung,
- die aufrufenden Moduln wahlweise mit Beschreibung,
- die gerufenen Moduln wahlweise mit Beschreibung,
- die verwendeten Dateien mit Art des Zugriffs und wahlweise der Beschreibung,
- die verwendeten globalen Datenbereiche wahlweise mit Beschreibung,
- die verwendeten V/3000 Masken,
- die angesprochenen COPYLIB-Member / INCLUDE-Dateien,
- die definierten Schlagworte.

Die Liste wird sortiert nach Projektkürzel, Sachgebietskürzel und Modulnamen. Sie wird aufgerufen durch:

MODUL oder M

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Modulname (Modul / LISTF-Format) ?

Es kann eine Modulbezeichnung oder ein generischer Name im LISTF-Format eingegeben werden.

Historie mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so wird die komplette Änderungsdokumentation zu den Moduln gedruckt.

Aufrufende Moduln mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden für alle Moduln BOTTOM-UP Modulreferenzen generiert.

Aufgerufene Moduln mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden für alle Moduln TOP-DOWN Modulreferenzen generiert.

Dateireferenzen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden alle von einem Modul benutzten Dateien mit der Art des Zugriffs (I=Input und O=Output in der Version 02.00 bzw. durch einen Zugriffskode ab der Version A.02.03), dem Dateityp und dem Status angedruckt.

COMMON-Referenzen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden alle von den Moduln verwendeten globalen Datenbereiche sowie ihr Typ mit angedruckt.

Maskenreferenzen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden alle verwendeten V/3000-Masken mit angedruckt.

Copylib-Referenzen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden alle verwendeten COPYLIB-Member / INCLUDE-Dateien mit angedruckt.

Schlagworte mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden alle referenzierten Schlagworte mit angedruckt.

Bemerkungen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so werden die evtl. geladenen Bemerkungen mit angedruckt.

Wurde eine der Fragen nach den Referenzen für Modul-BOTTOM-UP, Modul- TOP-DOWN, Dateinamen, COMMON-Namen, Masken, COPYLIB's oder Schlagworte mit **J** beantwortet, so erfolgt die zusätzliche Frage ob zu diesen Referenzen die Beschreibung mit ausgegeben werden soll.

Beschreibung der Referenzen mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben so werden die Beschreibungen der aufgerufenen Moduln, der aufrufenden Moduln, der Dateien, der globalen Datenbereiche, der Masken, der COPYLIB-Member und der Schlagworte mit ausgegeben.

Seitenvorschub nach jedem Modul (J/N, Return=N) ?

Wird hier ein **J** eingegeben so wird nach jedem angedruckten Modul ein Seitenwechsel durchgeführt.

6.5.2 Bibliotheksübersicht sortiert nach Modulnamen

Diese Liste ist eine SOLL-Übersicht über den Inhalt einer Bibliothek. Sie dient u.a. als Crossreferenz, um in den SEGMENTER/LINKEDIT-Andrucken zurechtzufinden. Sie dient dem Systemmanager oder dem Sachgebietsverantwortlichen bei der Verwaltung der in seinem Bereich vorkommenden Bibliotheken. Die Liste enthält alle Moduln in einer/mehreren Bibliotheken (USL, RL, SL, XL) sortiert nach Bibliothek und Modulnamen. Die Liste wird aufgerufen durch:

LIBM

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein Dateiname oder ein generischer Name im LISTF-Format eingegeben werden. Bei der Selektion wird intern immer ein voll qualifizierter Dateiname aus dem COMPILE-Eintrag und den Angaben des JOBID gebildet.

Modulbeschreibung (J/N, Return=N) ?

Wahlweise kann durch Beantwortung dieser Frage mit **J** die Beschreibung jedes Moduls mit ausgedruckt werden.

6.5.3 Bibliotheksübersicht sortiert nach Segmentnamen

Die Liste ist eine SOLL-Übersicht über den Inhalt einer Bibliothek. Die ist das Äquivalent einer entsprechenden SEGMENTER / LINKEDIT-Liste (z.B. PREP-PMAP) jedoch in sortierter Form. Sie dient dem Systemmanager oder Sachgebietsverantwortlichen zur Verwaltung der in seinem Bereich vorkommenden Bibliotheken. Diese Liste enthält alle Moduln in einer/mehreren Bibliotheken (USL, RL, SL, XL) sortiert nach Bibliothek, Segment- und Modulnamen. Sie wird aufgerufen durch:

LIBS

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein Dateiname oder ein generischer Name im LISTF-Format eingegeben werden. Bei der Selektion wird intern immer ein voll qualifizierter Dateiname aus dem COMPILE-Eintrag und den Angaben des JOBID gebildet.

Modulbeschreibung (J/N, Return=N) ?

Wahlweise kann durch Beantwortung dieser Frage mit **J** die Beschreibung jedes Moduls mit ausgedruckt werden.

6.5.4 Datei Referenzliste

Diese Liste enthält die Referenzen aller Moduln des Gesamtsystems auf eine Datei. Sie ist damit das wichtigste Hilfsmittel bei der Änderung von Datenstrukturen in bestehenden Programmsystemen. Die Liste ist sortiert nach Modulnamen. Sie wird aufgerufen durch:

FILEREF oder FR

Selektionsangaben:

Dateiname (['!'] Datei / LISTF-Format) ?

Folgende Eingaben sind möglich:

- ein Dateiname
- ein Dateiname im LISTF-Format (generischer Name)
- !Dateiname für IMAGE/3000 Dateien, deren Name Sonderzeichen enthält, die auch im LISTF-Format vorkommen (#, @ oder ?)

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann nur ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Modulbeschreibung (J/N, Return=N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.5 COMMON Referenzliste

Diese Liste enthält die Referenzen aller Moduln des Gesamtsystems auf einen globalen Datenbereich. Sie ist damit ein sehr wichtiges Hilfsmittel bei der Änderung an den globalen Datenbereichen innerhalb eines Programmsystems. Die Liste ist sortiert nach COMMON-Namen und Modulnamen. Sie wird aufgerufen durch:

COMMONREF oder CR

Selektionsangaben

COMMON-Name (Common / LISTF-Format) ?

Es kann ein COMMON-Name oder ein generischer Name im LISTF-Format eingegeben werden.

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Modulbeschreibung (J/N, Return =N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.6 Masken Referenzliste

Diese Liste enthält die Referenzen aller Moduln des Gesamtsystems auf V/3000 Bildschirmmasken. Die Liste ist sortiert nach Maskennamen und Modulnamen. Bei Wechsel der Maskendatei erfolgt ein Seitenwechsel. Die Liste wird aufgerufen durch

FORMREF oder FORMR

Selektionsangaben

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein Dateiname oder ein generischer Name im LISTF-Format eingegeben werden.

Maskenname (Name / LISTF-Format) ?

Es kann ein Maskenname oder ein generischer Maskenname im LISTF-Format eingegeben werden.

Modulbeschreibung (J/N, Return =N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.7 COPYLIB-Member / INCLUDE-Datei Referenzliste

Diese Liste enthält die Referenzen aller Moduln des Gesamtsystems auf COPYLIB-Member bzw. INCLUDE - Dateien. Die Liste ist sortiert nach Membernamen und Modulnamen. Bei Wechsel der COPYLIB / INCLUDE - Gruppe erfolgt ein Seitenwechsel. Die Liste wird aufgerufen durch:

COPYREF oder COPYR

Selektionsangaben

Dateiname (['!'] Datei / LISTF-Format) ?

Es kann ein COPYLIB-Name oder ein generischer Name im LISTF-Format eingegeben werden. Bei Verwendung von INCLUDE-Gruppen muß hier der Name der INCLUDE-Gruppe mit vorangestelltem ! eingegeben werden.

Copylib Eintrag (Name / LISTF-Format) ?

Es kann ein Membername oder ein generischer Membername im LISTF-Format eingegeben werden. Bei INCLUDE-Dateien ist dies der Dateiname in der INCLUDE-Gruppe.

Modulbeschreibung (J/N, Return =N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.8 Schlagwort Referenzliste

Diese Liste enthält die Referenzen aller Moduln des Gesamtsystems auf ein Schlagwort. Die Liste ist sortiert nach Modulnamen. Bei Wechsel des Schlagwortes erfolgt ein Seitenwechsel. Der Aufruf erfolgt mit:

KEYWORDREF oder KR

Selektionsangaben

Schlagwort (Schlagwort / LISTF-Format) ?

Es kann ein Schlagwort oder ein generisches Schlagwort im LISTF-Format eingegeben werden.

Modulbeschreibung (J/N, Return =N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.9 Versionsverzeichnis

Diese Liste enthält alle gesicherten Sachgebietsversionen mit den zugehörigen Moduln und deren Versionen. Die Liste ist sortiert nach Projekt, Sachgebiet, Version und Modulname. Bei Wechsel des Sachgebiets oder der Version erfolgt ein Seitenwechsel. Die Liste wird aufgerufen durch

VERSION oder V

Selektionsangaben

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder @ für alle Sachgebiete des Projekts angegeben werden.

Version (Nummer / LISTF-Format) ?

Es kann eine Versionsnummer oder ein @ für alle Versionen eingegeben werden. Damit die Selektion über generische Versionen möglich ist, ist es wichtig den Aufbau der Versionsnummern im Unternehmen eindeutig und sinnvoll festzulegen.

Modulbeschreibung (J/N, Return =N) ?

Wenn diese Frage mit **J** beantwortet wird, so wird zu allen Moduln die Beschreibung mit angedruckt.

6.5.10 Summary

Diese Liste enthält eine Summary eines Projektes.

Im ersten Teil werden pro Sachgebiet die Anzahl Moduln, Codezeilen, Kontrollzeilen, Kommentare, Leerzeilen und Modulreferenzen dargestellt. Im zweiten Teil werden die Anzahl Dateien und die Anzahl Referenzen darauf, die Anzahl Common's und die Referenzen darauf sowie die Anzahl Masken und die Referenzen darauf angedruckt. Die Liste ist sortiert nach Sachgebiet. Die Liste wird aufgerufen durch

SUMMARY oder SUM

Selektionsangaben

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

6.5.11 Change Log

In dieser Liste werden alle Änderungshistorien nach Zeitraum angedruckt.

Die Liste wird sortiert nach Projekt, Datum und Modulnamen. Sie wird aufgerufen durch:

CHANGE oder CH

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Modulname (Modul / LISTF-Format) ?

Es kann eine Modulbezeichnung oder ein generischer Name im LISTF-Format eingegeben werden.

6.6 Dokumenten Verwaltung

6.6.1 Dokumentationsstruktur

In dieser Liste werden die Dokumentationsphasen und die Dokumentationstypen in gemeinsamer Form dargestellt. Diese Dokumentationsstruktur muß zentral gewartet werden. Sie dient für das gesamte SD-System. Die Liste enthält immer alle Phasen und Typen sortiert nach Priorität. Sie wird aufgerufen durch

STRUCTURE oder STR

6.6.2 Dokument Stati Verzeichnis

Diese Liste enthält alle in SD/3000 definierten Dokumentations Stati. Ein Status dient zur Qualifizierung eines Dokuments. Die Dokumentations Stati müssen zentral verwaltet werden. Die Liste enthält alle Stati sortiert nach deren Namen. Sie wird aufgerufen durch

STATE oder STA

6.6.3 Dokumenttypen Verzeichnis

Die in SD/3000 hinterlegten Dokumente (Objekte) müssen mit einem bestimmten Dateityp definiert werden. Dieser Dateityp gibt an, wie die Dokumente ausgegeben werden können (bzw. in welcher Form die Aufbereitung durchgeführt werden soll). Jedem Dokument Typ wird eine Serverprozedur sowie ein Satz von erlaubten Ausgabeinheiten zugeordnet. Das Ausgeben von aufbereiteten Dokumenten ist z.Zt. nur mit dem Server SD'P'ASCII möglich. Änderungen der Dokumenttypen müssen zentral gepflegt werden. Die Liste enthält alle Typen sortiert nach Ihrem Namen. Sie wird aufgerufen durch:

FILETYP oder FT

6.6.4 Dokument Einträge Verzeichnis

Die in SD/3000 eingestellten Dokumentationsdateien können mit dieser Funktion selektiv zusammengestellt und wahlweise nur der Dokumenteneintrag oder auch zusätzlich das gesamte Dokument mit angedruckt werden. Die Liste wird nach Phasenpriorität, Typpriorität, Projekt, Sachgebiet und Modul sortiert. Die Liste wird aufgerufen durch:

DOC oder D

Selektionsangaben:

Projekt (Return={Voreinstellung} / Projekt / LISTF-Format) ?

Es kann ein Projektkürzel oder ein generischer Name im LISTF-Format eingegeben werden.

Sachgebiet (Return=@ / Sachgebiet / LISTF-Format) ?

Es kann ein gültiges Sachgebiet oder ein @ für alle Sachgebiete des Projekts eingegeben werden.

Modulname (Modul / LISTF-Format) ?

Es kann ein Modulname oder ein generischer Modulname im LISTF-Format eingegeben werden.

Dokumentationsphase (Return=@ / Phase / LISTF-Format) ?

Es kann eine Phasenbezeichnung oder ein generischer Phasenname im LISTF-Format eingegeben werden.

Dokumentationstyp (Return=@ / Typ / LISTF-Format) ?

Es kann ein gültiger Typ oder ein generischer Typ im LISTF-Format eingegeben werden.

Dokumenten Status (Return=@ / Status / LISTF-Format) ?

Es kann ein gültiger Status oder ein generischer Status im LISTF-Format eingegeben werden.

Dokumentationsdateien mit andrucken (J/N, Return=N) ?

Wird hier ein **J** eingegeben, so wird das Dokument mit dem entsprechenden Server ausgegeben.

7. Utilities

7.1 Das SPERR-System

Mit dem SPERR-System ist es möglich, Dateien auf einer HP3000 außerhalb eines Programm **global** zu sperren. Dies System wurde zunächst entwickelt, um ein paralleles Arbeiten mit dem SEGMENTER innerhalb von USL's zu ermöglichen. Das System ist jedoch auch zur Synchronisation aller anderen Dateiarten anwendbar.

Das SPERR-System nutzt spezielle Datasets innerhalb der SD/3000 Datenbank.

Beispiel für die Anwendung in einem Compilejob:

```
!job test,mgr.global
!spl test.source
!SPERR USLTEST
!continue
!segmenter
usl usltest
purgerbm unit,test
auxusl $oldpass
copy unit,test
exit
!ENTSPERR USLTEST
!eoj
```

7.1.1 SPERR (Sperren einer Datei)

Zum Sperren einer bestehenden Datei wird das Programm SDSPELL mit dem Parameter 1 und dem Dateinamen aufgerufen. Hierzu steht ein UDC / eine XEQ-Datei mit der folgenden Syntax zur Verfügung:

:SPERR {dateiname}

{dateiname}

ist ein gültiger MPE-Dateiname. Die Datei muß nicht vorhanden sein.

7.1.2 ENTSPERR (Freigeben einer Datei)

Zum Freigeben einer Datei wird das Programm SDSPELL mit dem Parameter 2 und dem Dateinamen aufgerufen. Hierzu steht ein UDC / eine XEQ-Datei mit der folgenden Syntax zur Verfügung:

:ENTSPERR {dateiname}

{dateiname}

ist ein gültiger MPE-Dateiname. Die Datei muß nicht vorhanden sein.

7.1.3 CSPERR (Freigeben einer Datei mit allen wartenden)

Zum Freigeben einer Datei und dem automatische Löschen der gesamten Warteliste wird das Programm SDSPELL mit dem Parameter 3 aufgerufen. Hierzu steht ein UDC / eine XEQ-Datei mit der folgenden Syntax zur Verfügung:

:CSPERR *{dateiname}*

{dateiname}

ist ein gültiger MPE-Dateiname. Die Datei muß nicht vorhanden sein.

7.1.4 LSPERR (Andruck Warteliste)

Zum Andrucken der internen SPERR-Informationen wird das Programm SDSPELL mit dem Parameter 0 aufgerufen. Hierzu steht ein UDC / eine XEQ-Datei mit folgender Syntax zur Verfügung:

:LSPERR

Es werden alle gesperrten Dateien sowie die Warteliste angezeigt. Sind keine Sperren vorhanden, so wird auch keine Liste angedruckt. Sollten noch fehlerhafte Sperren aktiv sein, so muß mit dem ENTSPERR-Befehl die jeweilige Sperre manuell aufgehoben werden.

7.2 Der INCLUDE-Prozessor SDINCL

SD/3000 unterstützt die Verwendung von \$INCLUDE-Befehlen in allen Programmiersprachen (außer RPG, COBOL und C) mit Bibliotheken. Diese Bibliotheken werden z.B. mit dem COBOL-COPYLIB-Editor COBEDIT oder einem anderen COPYLIB-Editor erzeugt und bearbeitet.

Die Include's werden durch einen der Preprozessoren vor dem Compiler aufgelöst.

In der UDC-Datei SDCMDXL.PUB.GLOBAL stehen die UDC's für die Compiler. Die Verwendung der BANNER-Seiten wird über das JCW SDBANNER aktiviert.

CCXL	für	C Native Mode
FORTRAN	für	FORTRAN IV
FTN	für	FORTRAN 77
FTNXL	für	FORTRAN 77 Native Mode
SPL	für	SPL
TPL	für	TRANSACT
TRANXL	für	TRANSACT Native Mode
RPL	für	REPORT
PASCAL	für	PASCAL
PASXL	für	PASCAL Native Mode
FASTCOMP	für	FASTRAN
COBOLII	für	COBOL-74
COBOLiX	für	COBOL-85
COB74XL	für	COBOL-74 Native Mode
COB85XL	für	COBOL-85 Native Mode

Die entsprechenden INCLUDE-Bibliotheken stehen immer in den in den Account-Spezifikationen (siehe Kapitel 2.15) eingetragenen Gruppen und Accounts. Sie haben den Namen des entsprechenden Compilers:

FORTRAN	für	FORTRAN IV
FTN	für	FORTRAN 77
SPL	für	SPL
TPL	für	TRANSACT und FASTRAN
RPL	für	REPORT
COBOL	für	alle COBOL-Compiler (Standard Cobol COPY)
PASCAL	für	PASCAL

Wenn auf einem System standardmäßig die Verwendung von INCLUDE-Bibliotheken aktiviert ist (SDINCLUDE-JCW = 1) und für einen USER/ACCOUNT ausnahmsweise der normale MPE-Include verwendet werden soll, so kann dies durch Setzen des JCW's SDMPINCLUDE = 1 (z.B. im LOGON-UDC des Benutzers) aktiviert werden.

7.3 Das LIST-Programm SDLIST

Das zentrale LIST-Programm dient in allen generierten Jobs zum Ausdruck von Dateien. Daneben ist es ein einfaches aber komfortables Hilfsmittel für den Programmierer zum Ausdruck von Dateien. Innerhalb von SD/3000 wird von anderen SD-Moduln als zentrales Druckprogramm verwendet.

Das LIST-Programm unterstützt Dateien, die mit COMPRESS, SQUISHER oder QEDIT komprimiert wurden. Sie werden automatisch beim Ausdruck entkomprimiert. Die Datei wird dazu nicht verändert.

Der interaktive Modus des LIST-Programms wird gestartet durch:

:LIST (als UDC / XEQ-Datei) oder mit
:RUN SDLIST.PUB.GLOBAL

Zum einfachen Ausdruck nur einer Datei dient die Kurzform:

:L {dateiname} [, {dest}] (als UDC / XEQ.Datei) oder mit
:RUN SDLIST.PUB.GLOBAL; INFO="{befehlszeile}"

Dabei ist *{dateiname}* der Name einer normalen MPE-Datei und *{dest}* der Name des Files (FILE-Statement oder Ausgabeklasse) für die Ausgabe. Der Ausgabefile ist mit TERM voreingestellt.

Die Verwendung von **LISTFILE**-Dateinamen ist erlaubt.

Sind in der Datei **\$PAGE**, **\$page** oder / an der ersten Stelle eines Satzes vorhanden, so wird an deren Stelle ein Seitenvorschub durchgeführt.

Im interaktiven Modus können die Befehle einzeln pro Zeile bzw. mehrere Befehle durch ; getrennt pro Zeile eingegeben werden. Alle Befehle können durch ihren ersten Buchstaben abgekürzt werden. Durch Eingabe von Help kann jederzeit der aktuelle Status der Steuerbefehle abgefragt werden.

Die folgenden Befehle sind zulässig:

Cctl	OFF PRE POST Wahl der Vorschub-Steuerung OFF: Datei enthält keine Steuerzeichen PRE: Steuerzeichen im PRE-Spacing Modus POST: Steuerzeichen im POST-Spacing Modus
Exit	Programm beenden
Help	Hilfe und Statusanzeige
List	<i>{dateiname}</i> [<i>{von}</i> [/ <i>{bis}</i>]] Datei Andrucken wahlweise nur in einem <i>{von}</i> / <i>{bis}</i> - Bereich. Wird bei der CM-Version kein Dateiname eingegeben, so wird die zuletzt angesprochene Datei erneut gedruckt. Die NM-Version erlaubt LISTFILE-Format.
Multiple	<i>{anzahl}</i> Datei mehrfach andrucken
Output	<i>{einheit}</i> <i>{formular}</i> Ausgabeeinheit und Formularname einstellen Die NM-Version unterstützt nur FILE-Namen als Einheit und keine Formulare.
Sequence	ON OFF Zeilennummerierung wird linksbündig mit ausgeben
Title	ON OFF “ <i>{text}</i> “ Voreingestellte Titelzeile andrucken oder freie Titelzeile definieren
Blanks	ON OFF Leerstellen am Satzende entfernen
/	Unterbrechung zum SYSTEM-Modus
Xeq	<i>{dateiname}</i> Befehlsdatei ausführen

7.4 Sourcekomprimierung mit COMPRESS/UNPRESS und SQUISHER

Zur effizienteren Speicherung der Sourcedateien auf der HP3000 dient wahlweise das COMPRESS / UNPRESS-System aus der CSL/3000 oder der SQUISHER. Beide Systeme werden von SD/3000 voll unterstützt.

Sind bei einem COMPILER-Aufruf die Dateien komprimiert, so werden sie entkomprimiert, evtl. vorhandene Include's aufgelöst, der Compiler aufgerufen und nach erfolgreichem Compile wieder komprimiert.

Beim COMPRESS/UNPRESS-System wird eine modifizierte Version der CSL benutzt. Diese Version unterstützt keine Chiffrierung und enthält spezielle Erweiterungen für die SD-Source-Bibliothek. Durch ein UDC bzw. eine XEQ-Datei kann direkt eine Datei verarbeitet werden. Die Befehle haben folgende Syntax:

```
:COMPRESS    {zeile}
:UNPRESS    {zeile}
```

Es kann nur ein Dateinamen bzw. ein generischer Dateinamen im LISTFILE-Format pro *{zeile}* eingegeben werden.

Beim SQUISHER wird die Standard-Version aus der CSL benutzt. Zur leichteren Anwendung sind die folgenden UDC's / XEQ-Dateien vorhanden:

```
:SQ          {zeile}
:UNSQ       {zeile}
```

Als *{zeile}* kann ein Dateiname oder ein generischer Name im LISTF-Format eingegeben werden.

7.5 Der Software Transfer Generator SDTRANS

Wenn von einem Entwicklungssystem regelmäßig Software-Updates an andere Systeme transferiert werden sollen, so kann mit diesem Utility eine Indirektdatei aller zu transferierenden Moduln, Dokumentationen etc. erzeugt werden. Voraussetzung dafür ist, daß diese Funktion bei der Definition des Projekts aktiviert ist. Danach wird bei jedem Generatorkauf (Gxxx) ein Flag bei dem entsprechenden Modul gesetzt. Dieses Flag wird bei Angabe der Option RESET später wieder zurückgesetzt.

Bei der Abarbeitung wird intern zunächst ein KMOD-Befehl für das Projekt durchgeführt. Ist der SD-Transfer für das Projekt aktiviert, wird ein SDUNLOAD des Projekts in die Datei **O{project}** durchgeführt. Diese Datei wird automatisch mit in die Indirektdatei eingestellt.

Werden Zieldateien vom Typ XL/RL/USL mit in die Indirektdatei aufgenommen, so wird ein entsprechender Reinigungslauf für die Datei durchgeführt (USL: SEGMENTER CLEANUSL, XL: LINKEDIT CLEANXL, RL: LINKEDIT CLEANRL).

Mit Hilfe des Typs SDINST ist es möglich, ONLINE-Install Bänder zu erzeugen. Diese Option setzt als Betriebssystem MPE/iX 5.0 voraus. In der dort verwendeten Install-Gruppe SDINST werden die zu installierenden Dateien unter einem internen Namen gespeichert. Parallel dazu wird eine MPE-XEQ-Datei mit dem Namen INSTALL.SDINST erzeugt, die die für die Installation notwendigen Befehle enthält.

Zum einfachen Aufruf steht das folgende UDC zur Verfügung:

:SDTRANS {project} {datei} [TYP={typ}] [{option}]

{project} Wird kein Projekt angegeben, wird das im Account voreingestellte Projekt verwendet.

{datei} Name der Indirektdatei, die erstellt werden soll.

{typ} Transfertyp, möglich sind: DSCOPY, SDINST, STORE und FTP. Voreinstellung ist STORE. Beim Transfertyp SDINST wird **{datei}** nicht benutzt, sondern eine separate Gruppe SDINST im entsprechenden Account erzeugt.

{option}	RESET	Löschen der Transferflags in D-COMPILE
	NORESET	Kein löschen der Transferflags (ist Voreinstellung)
	SOURCE	Nur Sourceübertragung und GC/GL auf Zielsystem machen
	SOURCEX	Nur Sourceübertragung und GCX/GLX auf Zielsystem machen

7.6 Das Projekt-Löschprogramm SDPURGE

Zum Löschen eines kompletten Projekts mit allen zugehörigen Einträgen steht das Utility SDPURGE zur Verfügung. **Achtung:** Projekte, zu denen noch Einträge in der SD-Source-Bibliothek vorhanden sind, können nicht gelöscht werden.

SDPURGE darf nur von Benutzern mit der Systemmanager-Berechtigung aufgerufen werden.

Wenn beim Aufruf von SDPURGE kein Projekt mit angegeben wird, so wird ein interaktiver Dialog durchgeführt. In diesem Dialog wird auch noch eine Eingabe zur Bestätigung der Löschung erwartet.

Wird beim Aufruf direkt ein Projekt angegeben, so wird ohne weitere Prüfung das Projekt mit allen seinen Daten gelöscht.

Zum einfachen Aufruf steht das folgende UDC / die XEQ-Datei zur Verfügung:

:SDPURGE [{projekt}]

Auf die Frage:

Welches Projekt soll gelöscht werden ?

muß ein gültiges Projektkürzel bzw. RETURN zum Beenden des Programms eingegeben werden.

Die Projektbeschreibung wird daraufhin angezeigt. Wird danach auf die Frage:

Projekt {projekt} löschen (J/N) ?

ein J eingegeben, so werden die folgenden Daten gelöscht:

- alle Übersetzungsjobs und Bindejobs mit LINK-Erweiterung,
- alle nicht mehr benötigten JOBID's,
- alle Dateibeschreibungen,
- alle COMMON-Beschreibungen,
- alle Dokumentationen,
- alle Dokumentationstexte,
- alle Stamminformationen über COPYLIB's und Masken,
- alle Referenzen für Moduln, Dateien, COMMON's, Masken, COPYLIB-Einträge, Schlagworte,
- alle Historien und
- alle Dokumenteinträge.

7.7 Das Projekt-Entladeprogramm SDUNLOAD

Zum Entladen eines kompletten Projekts mit allen zugehörigen Einträgen in eine Transferdatei steht das Modul SDUNLOAD zur Verfügung. Das Programm kann entweder direkt aufgerufen oder es kann ein interaktiver Dialog durchlaufen werden. Die komplette Entladung eines Projekts wird durchgeführt, wenn beim Aufruf von SDUNLOAD ein Projekt und der Name einer Entladefile angegeben wird. Diese Datei darf noch nicht vorhanden sein.

Bei Nutzung des interaktiven Dialogs können entweder bestimmte Teile entladen oder auch mehrere Projekte in eine Transferdatei entladen werden. Der interaktive Dialog wird verwendet, wenn beim Aufruf kein Projekt angegeben wird. Wenn nur die COMPILE/LINK-Einträge entladen werden sollen, so kann auch nur ein Sachgebiet angegeben werden.

Zum einfachen Aufruf steht das folgende UDC / die XEQ-Datei zur Verfügung:

:SDUNLOAD [{projekt} , {entladefile}]

Wenn der interaktive Dialog durchlaufen wird, so erfolgt zunächst die Frage:

Name der Entladefile ?

Hier muß ein gültiger MPE-Dateiname einer noch nicht vorhandenen Datei eingegeben werden. In diese Transferdatei werden alle später aufgeführten Projekte entladen.

Auf die Frage

Welches Projekt soll entladen werden (RETURN=Ende) ?

muß ein gültiges Projektkürzel bzw. RETURN zum Beenden des Programms eingegeben werden. Die Projektbeschreibung wird angezeigt. Wird danach auf die Frage:

Projekt {projekt} entladen (J/N * Return=J) ?

ein **J** eingegeben, so werden alle Übersetzungsjobs, Bindejobs, Dateibeschreibungen und COMMON-Beschreibungen sowie die benötigten JOBID's und LINK-Erweiterungen entladen.

Wird auf die Frage

Dokumentation mit entladen (J/N * Return=J) ?

ebenfalls ein **J** eingegeben so werden zusätzlich alle Dokumentationen, Dokumentationstexte, Modulreferenzen, Dateireferenzen, COMMON-Referenzen, Historien und Dokumenteinträge entladen.

Wird auf die Frage

COPYLIB's, Maskendateien und Schlagworte entladen (J/N * Return=J) ?

ein **J** eingegeben, so werden zusätzlich alle Stamminformationen und Referenzen zu den angesprochenen Daten mit entladen.

Wird auf die Frage

Versionshistorie entladen (J/N * Return=J) ?

mit **J** geantwortet, so werden ebenfalls die gespeicherten Sachgebietsversionen (Moduln mit Version pro Sachgebietsversion) entladen.

7.8 Das Projekt-Ladeprogramm SDLOAD

Zum Laden von Projekt(en) mit allen zugehörigen Einträgen aus einer Transferdatei steht das Modul SDLOAD zur Verfügung. In eine Transferdatei können mehrere Projekte entladen worden sein. Sie werden mit diesem Programm alle geladen. Das zu ladende Projekt darf in der SD/3000 Datenbank nicht vorhanden sein.

Das Programm arbeitet entweder mit direkter Eingabe eines Dateinamens für die Transferdatei oder mit einem interaktiven Dialog.

Zum einfachen Aufruf steht das folgende UDC / die XEQ-Datei zur Verfügung:

:SDLOAD [{entladedatei}]

Auf die Frage:

Name der Entladedatei ?

muß ein gültiger MPE-Dateiname einer vorhandenen Entladedatei eingegeben werden.

Die Anzahl der Datensätze die in das jeweilige SD-Dataset geladen wurden wird auf \$STDLIST angezeigt. Die in der Entladedatei vorhandenen JOBID's werden nur geladen, wenn sie noch nicht vorhanden sind.

7.9 Das FLABELINFO-Programm SDFLI

Für die einfachere Jobsteuerung unter MPE/V-Systemen diente das FLABELINFO-Utility. Dies Utility ermittelt die wesentlichen Parameter des LABELS einer MPE-Datei und stellt die Informationen in JCW's zur Verfügung. Es stellt damit auf MPE/V-Systemen einige der FINFO-Funktionen der MPE/iX-Systeme zur Verfügung. Es wird in der nächsten SD/3000-Version nicht mehr unterstützt.

Das Programm SDFLI ist wie folgt aufzurufen:

```
:FILE FILENAME={dateiname}  
:RUN SDFLI.PUB.GLOBAL
```

Nach dem Aufruf sind die folgenden JCW's gesetzt:

SDFSERR	Filesystem-Fehler beim Aufruf von FLABELINFO
SDPRIVFILE	0=Normale Datei; 1=Privilegierte Datei
SDFILECODE	Filecode, bei PRIV-Dateien der absolute Teil
SDNOLABELS	Max. Anzahl Userlabel
SDFLIMIT	Max. Dateigröße; wenn > 32767 = 32767
SDFLIMIT32767	0=FLIMIT < 32768; sonst 1
SDRECLN	Satzlänge in Byte
SDBLOCKLEN	Blocklänge in Byte
SDEOF	EOF-Pointer; wenn > 32767 = 32767
SDEOF32767	0: EOF < 32768; sonst 1

7.10 Das SPOOLFILE Namenprogramm SDODDM

Die \$STDLIST-Ausgaben im MPE/iX erhalten im Spooler einheitlich den Namen \$STDLIST und sind deshalb schwer voneinander zu unterscheiden. Um dies vor allem für die Compile/Link-Jobs zu ermöglichen wurde das Utility SDODDM geschaffen. Mit Hilfe dieses Utilities ist es möglich den Spoolfilenamen auf den Jobnamen zu ändern bzw. einen frei definierten Namen zu vergeben.

Um dem Spoolfile den Jobnamen zu geben, ist der folgende Aufruf erforderlich:

```
:RUN SDODDM.PUB.GLOBAL
```

Um dem Spoolfile einen sonstigen Namen zu geben, ist der folgende Aufruf erforderlich:

```
:RUN SDODDM.PUB.GLOBAL; INFO="{name}"
```

Als Name sind bis zu 8 Zeichen zulässig.

7.11 Das POWERHOUSE Dictionary Ladeprogramm SDQDD

Mit Hilfe dieses Utilities ist es möglich, die in bestehenden POWERHOUSE-Dictionaries definierten Dateien automatisch in SD/3000 zu laden. Bestehenden Einträge in der SD-Datenbank werden nicht überschrieben.

Das Utility wird aufgerufen über

:RUN SDQDD.PUB.GLOBAL

Für das Laden der Einträge müssen der Projektname und der Sachgebietsname eingegeben werden, in die die Dateibeschreibungen geladen werden sollen.

7.12 Der SEGMENTER-Preprozessor SDSEG

Mit Hilfe dieses Utilities ist es möglich, in allen SEGMENTER/ FASTSEG-Befehlen das Schlüsselwort **\$ALL** zu verwenden. Mit Hilfe dieser Möglichkeit wird es sehr einfach, komplexe Bibliotheken (z.B. RL's) mit nur einem Befehl aufzubauen. Bei allen Aktionen wird intern automatisch die richtige Kombination von Befehlen erzeugt. SDSEG wird beim Arbeiten mit dem SD-Generator immer benutzt.

Das Utility wird aufgerufen über

:SDSEG {parm} als UDC/XEQ-Datei
:RUN SDSEG.PUB.GLOBAL; PARM={parm}

{parm} 0 Es wird der MPE-Segmenter benutzt
 1 Es wird der FASTRAN-Segmenter benutzt

Beim Arbeiten mit SDSEG sind zunächst alle logisch richtigen SEGMENTER/FASTSEG Befehle zulässig. Beim Löschen von Einträgen in Zielbibliotheken muß jedoch abweichend von der normalen MPE-Regel vor dem Löschbefehl die Source-Bibliothek angegeben werden, damit die richtigen Segmente/Units für das Löschen gefunden werden können.

Beispiel:

zulässig mit SDSEG

```
:SDSEG 0
RL RL
USL $OLDPASS <- wichtig
PURGERL UNIT,$ALL
ADDRL $ALL
EXIT
```

nicht zulässig mit SDSEG

```
:SEGMENTER
RL RL
PURGERL UNIT,$ALL
USL $OLDPASS
ADDRL $ALL
EXIT
```

7.13 Das Passwort-Utility SDETPW

Mit Hilfe dieses Utilities ist es möglich, in einer Sitzung oder einem Job die für die Anmeldung erforderlichen Kennworte vom Betriebssystem in entsprechende JVAR's einstellen zu lassen. Dieses Utility wird z.B. bei den Generator-Batch-Befehlen benutzt. Es steht jedoch auch für weitere Anwendungen bereit.

Das Utility wird aufgerufen über

:RUN SDETPW.PUB.GLOBAL

Die folgenden Variablen werden erzeugt:

SDACCOUNTPASS
SDGROUPPASS
SDUSERPASS

7.14 Das Filecode Utility SDETFC

Mit Hilfe dieses Utilities ist es möglich den Filecode beliebiger Dateien zu verändern. Das Utility ist dazu gedacht um evtl. auftretende Störungen in der SD-Source-Bibliothek leichter beheben zu können. Es kann jedoch bei falschem Gebrauch katastrophale Auswirkungen haben. Es verändert sowohl positive, als auch negative Filecodes.

Das Utility wird aufgerufen über

:RUN SDETFC.PUB.GLOBAL; PARM={filecode1}; INFO="{dateiname}"

oder

:RUN SDETFC.PUB.GLOBAL; INFO="{dateiname};{filecode2}"

- {filecode1}* Der numerische Filecode, der gesetzt werden soll.
{filecode2} Der numerische oder Charakter-Filecode, der gesetzt werden soll.
{dateiname} Name der Datei, die den neuen Filecode haben soll.

A C H T U N G

Dies Utility sollte vom Systemmanager besonders geschützt bzw. vom System entfernt werden.

7.15 Der Source Scanner SDSCAN

SDSCAN ist ein Programm zum schnellen durchsuchen von Dateien. Wird eine String gefunden, so werden die über den Aroundbuffer definierte Anzahl Zeilen vor und nach dem gefundenen Datensatz angezeigt bzw. angedruckt.

Das Programm wird aufgerufen über

:SDSCAN {fileset} {string} {listfile} {aroundsize}

{fileset} Ein Dateinamen im LISTF-Format

{string} Die zu suchende String

{listfile} Ausgabeinheit, Voreinstellung ist \$STDLIST

{aroundsize} Anzahl Sätze vor und nach gefundenem Satz, Voreinstellung ist 10, Maximum ist 50.

Das Verhalten des Scanners kann über folgende Optionen (Jobvariablen) zusätzlich beeinflusst werden:

SDSCAN_NOKSAM	1:	Es werden keine KSAM/NMKSAM-Dateien verarbeitet.
SDSCAN_NOMSG	1:	Es werden keine MSG-Dateien verarbeitet.
SDSCAN_NOBINARY	1:	Es werden keine Binary-Dateien verarbeitet.
SDSCAN_NOVFORM	1:	Es werden keine VFORM/VFAST-Dateien verarbeitet.
SDSCAN_AUTOPASS	1:	Datei Kennworte werden automatisch ergänzt.
SDSCAN_UPSHIFT	1:	Vor dem durchsuchen wird ein UPSHIFT durchgeführt.
SDSCAN_DOWNSHIFT	1:	Vor dem durchsuchen wird ein DOWNSHIFT durchgeführt.

7.16 Das Band / DDS Reload Programm LOADTAPE

Wenn ein Band / DDS nach einer Aktion OFFLINE gegangen ist, ist es normalerweise erforderlich das Band zu entnehmen und neu einzulegen. Mit Hilfe dieses Programms kann dies jedoch online von jedem Bildschirm aus durchgeführt werden.

Das Programm wird aufgerufen über

:LOADTAPE {lu}

{lu} Die LU-Nummer des Bandlaufwerkes, Voreinstellung ist 7.

7.17 Das Projekt Rename Programm SDPRREN

Ein Projekt kann komplett umbenannt werden durch dies RENAME-Programm.

Das Programm wird aufgerufen über

:SDPRREN {pr_alt} {pr_neu}

{pr_alt} Der alte Projektname

{pr_neu} Der neue Projektname

8. COPYLIB Utilities

8.1 Das COPYLIB-Druckprogramm COPYLIST

Zum sauberen Ausdrucken der COPYLIB's mit einem zugehörigen Inhaltsverzeichnis dient das Utility COPYLIST.

Zum einfachen Aufruf steht das folgende UDC /XEQ-Datei zur Verfügung:

:COPYLIST {copylib} [, {ausgabeeinheit}]

{copylib} MPE-Dateiname der zu druckenden COPYLIB

{ausgabeeinheit} Filename der Ausgabedatei (Voreinstellung: \$STDLIST)

8.2 Der COPY-Member Generator COPYDBD

COPYDBD ist ein interaktives Programm zur Erzeugung der in COBOL Programmen notwendigen Feldbeschreibungen für Datenbank, Dataset-Namen und Keyitem-Namen.

Das Programm wird aufgerufen über

:COPYDBD als UDC/XEQ-Datei
:RUN COPYDBD.PUB.GLOBAL

COPYDBD benötigt keine externen Dateizuordnungen. Alle Dialoge werden über \$STDIN geführt.

Der Name der Datenbank und der Kennworte werden im Dialog eingegeben.

Die von COPYDBD generierten Definitionen werden im COPYLIB-Format in einer sequentiellen Datei gespeichert. Der Name dieser Datei sowie die Namen der COPYLIB-Member werden zu Beginn der Verarbeitung vom Bediener erfragt.

8.3 Der COPY-Member Generator COPYGEN

COPYGEN ist ein interaktives Programm zur Erzeugung der in COBOL Programmen notwendigen Datensatzbeschreibungen für V/3000-Masken und der IMAGE-Datensatzbeschreibungen.

Das Programm wird aufgerufen über

```
:COPYGEN                als UDC/XEQ-Datei
:RUN COPYGEN.PUB.GLOBAL
```

COPYGEN benötigt keine externen Dateizuordnungen. Alle Dialoge werden über \$STDIN geführt.

Die Namen der Eingabedateien werden im Dialog eingegeben. Eingaben sind die VIEW-Maskendatei und evtl. eine Datenbank.

Die von COPYGEN generierten Definitionen werden im COPYLIB-Format in einer sequentiellen Datei gespeichert. Der Name dieser Datei sowie die Namen der COPYLIB-Member werden zu Beginn der Verarbeitung vom Bediener erfragt.

Die Satzbeschreibungen für die Masken werden abhängig vom Datentyp generiert:

numerische Datentypen:

05 {feldname}	PIC 9(n).
05 {feldname}-X REDEFINES {feldname}	PIC X(n).
05 {feldname}-Z REDEFINES {feldname}	PIC Z(n-1)9.

alphanumerische Datentypen:

05 {feldname}	PIC X(n).
----------------------	------------------

Datumsfelder:

05 {feldname}.	
10 TT	PIC 99.
10 P1	PIC X.
10 MM	PIC 99.
10 P2	PIC X.
10 JJ	PIC 99.

In der generierten Satzbeschreibung werden die in der VIEW Datei gespeicherten Feldnamen zur Identifizierung der Einzelfelder benutzt.

Für die Ansteuerung von einzelnen Feldern auf der Maske mit den Routinen VSETERROR / VPLACE-CURSOR etc. wird für jedes auf der Maske definierte Feld eine Variable mit dem Namen **F-{feldname}** und der V/3000-Feldnummer in der VALUE-Angabe generiert.

Datenbank Satzbeschreibungen

Bei Masken für die Verwaltung von Datenbanksätzen (z.B. Stammdaten) ist COPYGEN in der Lage, auch für die Datenbank Datei eine komplette Satzbeschreibung zu generieren.

Hierzu müssen nur SET- und BASE-Name sowie ein MODE5 Kennwort (Read) eingegeben werden.

Datenbank - Satzbeschreibungen können nur nach der zugehörigen V/3000-Maskenbeschreibung generiert werden, da z.B. bei numerischen Feldern die Anzahl der Dezimalstellen hieraus übernommen werden.

8.4 Der COPY-Member Generator COPYPRN

COPYPRN ist ein Programm für den interaktiven Entwurf von Listen-Layouts. Es können Probedrucke erzeugt und letztlich auch die in COBOL-Programmen benötigten Datendefinitionen generiert werden. Alle Layouts werden in Bibliotheken gespeichert und können so immer wieder verwendet werden.

Das Programm wird aufgerufen über

```
:COPYPRN                als UDC/XEQ-Datei  
:RUN COPYPRN.PUB.GLOBAL
```

COPYPRN benötigt keine Dateizuordnungen per **:FILE** Kommando. Die eingegebenen Listbild-Entwürfe werden in einer **Layoutdatei** gespeichert. Bei Eingabe von **X** in das Feld Neu auf der Maske wird eine entsprechende Datei erzeugt. Eine Formsdatei kann bis zu 99 Listbild-Entwürfe speichern.

Die von COPYPRN generierten Definitionen werden in der mit **Genfile** bezeichneten Datei gespeichert. Wenn die **Genfile**-Datei bereits vorhanden ist, wird der Benutzer gefragt, ob die alte Datei gelöscht werden darf. Unter dem mit GENFILE angegebenen Namen wird eine Datei im EDITOR/COBOL-Format angelegt (Filecode= EDTCT).

Die Nummer des gewünschten Druckbildes muß unter **Liste** angegeben werden. Nach Drücken der ENTER-Taste sucht COPYPRN diesen Entwurf im Formsfile und zeigt ihn auf der Maske an. Falls nichts gespeichert ist, wird die Maske initialisiert.

Da auf dem Bildschirm nicht die komplette Zeilenlänge dargestellt werden kann, zeigt die Maske immer nur einen Ausschnitt (66 Stellen) des Entwurfs. Mit Hilfe von Funktionstasten läßt sie sich wie eine Schablone mit einem Fenster über die gesamte Formularbreite hin- und herschieben.

Alle Dateizuordnungen dürfen auch nachträglich geändert werden (z.B. Wechsel der Ausgabedatei Genfile nach einer Generierung).

Neben dem eigentlichen Listbildentwurf müssen für eine Zeile zuerst drei Angaben gemacht werden:

LINE

Hier muß eine zweistellige alphanumerische Identifikation für die Zeile eingegeben werden, welche für die Bezeichnung der generierten Datendefinitionen benutzt wird (z.B.: **K1** für Kopfzeile 1, **E1** für Einzelzeile 1).

REP

Hier wird der Wiederholungsfaktor für den Probeandruck eingegeben (01 - 99). Dies ist nötig, um ein realistisches Abbild der Liste zu erhalten. Kopf- und Summenzeilen erhalten in der Regel den Wert 1, während Einzelzeilen so oft wiederholt werden sollten, daß eine Seite komplett bedruckt wird.

LF

Hier wird die Anzahl der vor dem Druck vorzuschiebenden Leerzeilen angegeben. Die Eingabe eines **B** wird als Blattvorschub interpretiert.

Die eigentlichen Zeilenentwürfe werden in den rechten Teil der Maske eingegeben. Alle Konstanten werden in der Form dargestellt, wie sie später im Ausdruck erscheinen sollen.

Alle Ausgabefelder, die vor dem Druck vom Anwendungsprogramm gefüllt werden, müssen mit einer gültigen PICture Maske definiert werden.

Beispiel:

..... Seite: *ZZZ9* oder:..... Betrag: *ZZZ,ZZZ,ZZ9,99*

Die Druckzeilen auf der Maske müssen jeweils mit einer Leerzeile eingegeben werden.

Vor dem Drücken einer Funktionstaste, z.B. zum Verschieben des Bildausschnitts, muß nach einer Veränderung des Listbildentwurfs **immer** die ENTER-Taste gedrückt werden.

Die PIC-Angabe der Ausgabefelder muß mit einem - oder ^ unterstrichen werden.

Beispiel:

..... X99999

..... ^^^^^^

Durch den Wechsel des Unterstreichungszeichens werden zwei separate Felddefinitionen generiert.

Funktionstasten Belegung

Anzeige Spalte 1 bis 66
 Anzeige Spalte 67 bis 132
 Probedruck
 Refresh
 Anzeige 6 Spalten nach links
 Anzeige 6 Spalten nach rechts
 Generiere COBOL-Definitionen
 Programm Ende

A c h t u n g

Neue Layouts müssen vor ihrer Erfassung durch Eingabe der FORM Nummer und Drücken der ENTER-Taste angewählt werden.

9. SD/3000 Source Bibliothek

Die SD/3000 Source Bibliothek (im folgenden SDSLIB genannt) ist ein Werkzeug, um unterschiedliche Versionen von Moduln in einem zentralen System unter besonders geschützten Bedingungen zu speichern und verwalten. Bei der Entwicklung wurde besondere Rücksicht auf eine sichere Speicherung der Dateien und gleichzeitig eine einfache Handhabung für den Anwender gelegt.

Speicherung der Dateien

Die Speicherung der Dateien geschieht in speziellen Gruppen des jeweiligen Produktionsaccounts. Diese Gruppen haben den Namen **SDSLIB nn** wobei nn eine Zahl von 00 bis 99 ist. Die einzelnen Dateien sind durch negative Filecodes geschützt. Die Filecodes werden anhand der vorliegenden positiven Filecodes errechnet. Der Algorithmus hierfür wird auf Wunsch zur Verfügung gestellt. Normale Sourcedateien mit einem Filecode < 1000 oder 1052 (EDTCT) werden bei der Einspeicherung komprimiert. Dies geschieht über eine abgewandeltes COMPRESS.

Referenzierung der Dateien

Zu jeder Modulversion ist in einer Referenzierungsdatei der SD/3000-Datenbank ein Eintrag vorhanden. Über diesen Eintrag kann der echte Name der Sourcedatei in der SDSLIB nn -Gruppe vom Programm SDSLIB ermittelt werden.

Logfunktionen

Bei allen Transaktionen mit SDSLIB, mit Ausnahme der Funktion INFO, wird intern ein Logbuch geführt. Dieses Logbuch wird auch nach Löschen eines Moduls nicht gelöscht. Damit ist eine möglichst weitreichende Dokumentation möglich. Modulunabhängige Transaktionen (z.B. INIT) werden intern unter der Modulversion *BLANK* dokumentiert.

Aufruf SDSLIB

Der Aufruf der verschiedenen SDSLIB-Funktionen geschieht über eine XEQ-Datei und hat folgende Syntax:

:SDS {funktion} {name} [, {projekt}] [, {option1}] [, {option2}] [, {option3}]

{name} Dies ist abhängig von der Funktion ein SD/3000 Modulname, der Name einer XEQ-Datei mit vorangestelltem ^ , ein @ oder ein Volumeset Name.

{projekt} Wenn nicht das dem Account voreingestellt Projekt verwendet werden soll, so ist hier der Name des Projekts einzutragen.

Die Optionen werden bei der jeweiligen Funktion erklärt.

9.1 SDSLIB Funktion IN

Bei der Funktion IN wird ein neues Modul bzw. eine neue Modulversion in die SDSLIB eingestellt. Dabei wird die Sourcedatei aus der Installationsgruppe in die **SDSLIBnn**-Gruppe kopiert und nach erfolgreicher Speicherung in der Installationsgruppe gelöscht. Im Feld Name muß ein gültiger Modulname oder der Name einer Indirektdatei mit vorangestelltem ^ eingegeben werden.

Zur Ausführung des IN-Befehls ist es erforderlich, daß der Modulkopf geladen wurde. Wenn es sich nicht um den ersten IN-Befehl für das Modul handelt, muß die aktuell geladene Version größer der letzten in der SDSLIB gespeicherten Version sein.

Wenn das Modul bereits in einer früheren Version gespeichert wurde, muß der Anwender vor einem IN-Befehl das Modul mit einem **OUT-EXKLUSIV** für Änderungen angefordert haben. Nach erfolgreichem Einstellen der neuen Version wird diese **OUT-EXKLUSIV**-Anforderung gelöscht.

Der Benutzer **INSTALL**, Sachgebietsverantwortliche und deren Stellvertreter haben die Berechtigung, über die Option **HOLD** eine neue Modulversion ohne vorheriges **OUT-EXKLUSIV** einzustellen. Bei Anwendung der Option **HOLD** werden eventuelle OUT-Anforderungen nicht gelöscht.

9.2 SDSLIB Funktion OUT

Bei der Funktion OUT wird eine in der SDSLIB gespeicherte Modulversion in Form einer normalen MPE-Datei in die lokale Gruppe des Anwenders gestellt. Im Feld Name muß ein gültiger Modulname, der Name einer Indirektdatei mit vorangestelltem ^ bzw. ein Modulname im LISTF-Format eingegeben werden.

Die Anforderung eines Moduls ist pro Anwender nur einmal möglich. Wenn das Modul bereits an andere Anwender vergeben wurde, wird bei jeder weiteren OUT-Anforderung die Liste der bereits vergebenen Modulversionen mit angezeigt.

Über die Option **EXKLUSIV** (abgekürzt durch E) wird das Modul einem Anwender exklusiv für Änderungen bereitgestellt. Ohne Angabe dieser Option ist später ein Rückspeichern für normale Anwender nicht möglich. Nur der Sachgebietsverantwortliche und sein Stellvertreter können durch Angabe der IN-HOLD-Option diesen Schutz umgehen. Eine OUT-EXKLUSIV-Anforderung für ein Modul ist systemweit nur einmal möglich.

Soll eine frühere Version eines Moduls entnommen werden, so wird die Option **VERSION** (abgekürzt durch V) gefolgt von der Versionsnummer in Klammern angegeben. Die Versionsnummer muß exakt wie bei der Einspeicherung angegeben werden. Der Aufbau der Versionsnummer wird von SD/3000 nicht fest vorge-schrieben. Es steht hierfür ein 8-stelliges Feld zur Verfügung. Ohne Angabe dieser Option wird immer die zuletzt gespeicherte Version entnommen.

9.3 SDSLIB Funktion INFO

Die Funktion INFO informiert über in der SDSLIB gespeicherte Modulversionen und deren Geschichte. Im Feld Name muß ein gültiger Modulname, ein Modulname im LISTF-Format oder der Name einer Indirekt-Datei mit vorangestelltem ^ eingegeben werden, wenn nicht die Option **SYS** benutzt wird. Standardmäßig werden alle gespeicherten Modulversionen mit Datum/Zeit und Benutzer der Einspeicherung angezeigt.

Die Funktion INFO mit der Option **LOG** kann auch dann noch für Moduln genutzt werden, wenn diese bereits mit der Funktion DELETE in der SDSLIB gelöscht und aus dem COMPILE-Eintrag entfernt wurden.

Bei Eingabe der Option **LOG** (abgekürzt durch L) werden alle Transaktionen an dem Modul mit Datum/Zeit und Benutzer angezeigt.

Durch Eingabe der Option **OUT** (abgekürzt durch O) wird zusätzlich angezeigt, an welche Anwender eine Version vergeben ist und wann die Vergabe erfolgt ist.

Mit der Option **SYS** (abgekürzt durch S) ist es Benutzern mit der SM- oder AM-Berechtigung erlaubt, Systemtransaktionen anzuzeigen. Hierzu gehört z.B. die INIT-Funktion.

9.4 SDSLIB Funktion CANCEL

Bei der Funktion CANCEL wird eine OUT-Anforderung eines Anwenders für ein Modul gelöscht. Im Feld Name muß ein gültiger Modulname, ein @ oder der Name einer Indirektdatei mit vorangestelltem ^ eingegeben werden. Die Modul=@ - Option ist nur Benutzern mit der SM-Berechtigung bzw. dem besonderen Benutzer **INSTALL** vorbehalten.

Mit der Option **ALL** (abgekürzt durch A) ist es Sachgebietsverantwortliche und deren Stellvertretern erlaubt die OUT-Anforderungen aller Anwender zu einem Modul zu löschen.

9.5 SDSLIB Funktion DELETE

Bei der Funktion DELETE wird ein Modul bzw. alle Moduln eines Projekts komplett mit allen zugehörigen Versionen und Dateien gelöscht. Mit DELETE gelöschte Moduln sind unwiederbringlich verloren. Im Feld Name muß ein gültiger Modulname oder ein @ für alle Moduln eines Projekts eingegeben werden. Die Angabe von Indirektdateien ist bei dieser Funktion nicht möglich.

Vor dem Löschen wird bei Aufruf mit einem Modulnamen:

Die gespeicherte Inhaltsangabe des Moduls angezeigt.

bei Aufruf mit einem @:

Die Projektkurzbeschreibung angezeigt.

Danach muß zur Bestätigung ein J eingegeben werden.

9.6 SDSLIB Funktion INIT

Mit dieser Funktion wird die SDSLIB für einen Account eingerichtet. Auch wenn in einem Account mehrere Projekte bearbeitet werden, ist der Aufruf der INIT-Funktion nur einmal erforderlich. Für den Aufruf dieser Funktion ist es erforderlich, in dem entsprechenden Account als Benutzer mit der SM- oder AM-Berechtigung angemeldet zu sein. Wenn die **SDSLIBnn**-Gruppe nicht im System-Volumeset angelegt werden soll, so muß als Name ein gültiger Volumeset-Name angegeben werden.

Die INIT-Funktion legt zunächst eine **SDSLIB00**-Gruppe an. Danach erzeugt sie in der SD-Datenbank die erforderlichen Steuerdatensätze.

Theoretisch besteht die Möglichkeit, daß alle möglichen Dateinamen in einer **SDSLIBnn**-Gruppe verbraucht sind. In einer Gruppe können derzeit 259.999.974 Modulversionen gespeichert werden. Wenn dieser Fall eintritt, fordert die SDSLIB-IN-Funktion dann einen weiteren Aufruf der INIT-Funktion an.

9.7 SDSLIB Funktion HELP

Durch Eingabe von HELP ohne weitere Zusätze erhalten Sie eine Kurzzusammenfassung der möglichen SDSLIB-Funktionen. Bei Eingabe von HELP Funktionsname erhalten Sie eine Beschreibung der Funktion sowie deren Optionen.

9.8 SDSLIB Funktion LIST

Bei der Funktion LIST wird eine in der SDSLIB gespeicherte Modulversion am Bildschirm angezeigt. Im Feld Name muß ein gültiger Modulname, ein Modulname im LISTF-Format oder der Name einer Indirekt-Datei mit vorangestelltem ^ eingegeben werden.

Soll eine frühere Version eines Moduls angezeigt werden, so muß die Option **VERSION** (abgekürzt durch V) gefolgt von der Versionsnummer in Klammern angegeben. Die Versionsnummer muß exakt wie bei der Einspeicherung angegeben werden. Der Aufbau der Versionsnummer wird von SD/3000 nicht fest vorge-schrieben. Es steht hierfür ein 8-stelliges Feld zur Verfügung. Ohne Angabe dieser Option wird immer die zuletzt gespeicherte Version angezeigt.

9.9 SDSLIB Funktion PRINT

Mit der Funktion PRINT wird eine in der SDSLIB gespeicherte Modulversion ausgedruckt. Im Feld Name muß ein gültiger Modulname, ein Modulname im LISTF-Format oder der Name einer Indirektdatei mit vorangestelltem ^ eingegeben werden.

Soll eine frühere Version eines Moduls angedruckt werden, so muß die Option **VERSION** (abgekürzt durch V) gefolgt von der Versionsnummer in Klammern angegeben. Die Versionsnummer muß exakt wie bei der Einspeicherung angegeben werden. Der Aufbau der Versionsnummer wird von SD/3000 nicht fest vorge-schrieben. Es steht hierfür ein 8-stelliges Feld zur Verfügung. Ohne Angabe dieser Option wird immer die zuletzt gespeicherte Version angedruckt.

Inhaltsverzeichnis

1.	Aufgaben von SD/3000.....	1
2.	Das Verwaltungsmodul SDMAINT.....	2
2.0.1	Das Verwaltungsmenue.....	3
2.0.2	Standard - Dialogablauf.....	3
2.1	Übersetzungsjobs definieren.....	4
2.1.1	Schlüsselbegriffe.....	4
2.1.2	Folgejob.....	4
2.1.3	Herkunft (Source).....	5
2.1.4	Ziel.....	6
2.1.5	Job Parameter.....	8
2.1.6	Änderungshistorie.....	8
2.1.7	Jobtexte.....	8
2.2	Bindejobs definieren.....	9
2.2.1	Schlüsselbegriffe.....	9
2.2.2	Folgejob.....	9
2.2.3	Herkunft (Source).....	10
2.2.4	Ziel.....	10
2.2.5	Job Parameter.....	11
2.2.6	Änderungshistorie.....	11
2.2.7	Jobtexte.....	11
2.3	Projekte.....	12
2.4	Sachgebiete.....	12
2.5	JOBID (Job-Identifikation).....	13
2.6	Schlagworte.....	14
2.7	INTRINSIC's.....	14
2.8	Voreinstellungen.....	15
2.9	Dateinamen.....	16
2.10	Common.....	16
2.11	Bildschirm Definition.....	17
2.12	Drucker Definition.....	18
2.13	Install-Account Verwaltung.....	19
2.14	Dokumentations Phasen.....	20
2.15	Dokumentations Typen.....	20
2.16	Dokument Stati.....	20
2.17	Dokumentation Dateitypen.....	21
2.18	Dokument Einträge.....	21
2.19	COPYLIB Definition.....	22
2.20	Datei Zugriffsroutinen.....	22
3.	Der Jobgenerator SDGEN.....	23
	:GC (Übersetzung).....	23
	:GL (Binden).....	23
	:GCL (Sachgebiet komplett übersetzen).....	23
	:GPR (Projekt komplett übersetzen).....	23
	:GCF (Übersetzung aus Execute-Datei).....	24
	:GLF (Binden aus Execute-Datei).....	24
	Aufruf von SDGEN aus anderen Subsystemen.....	24
3.1	Optionen für die Generator-Aufrufe.....	25
3.2	Sonderoptionen SDGEN.....	26

Inhaltsverzeichnis

3.3	Generator im Background.....	28
	:GB {generatorbefehl} [{optionen}].....	28
	:GBE {generatorbefehl}.....	28
3.4	Parameter Modifikation beim Generator	28
4.	Der Jobgenerator SDSTR	29
5.	Das Kopf-Ladeprogramm SDKOPF	30
	:KM (einzelnes Modul)	30
	:KCL (Sachgebiet komplett laden).....	30
	:KPR (Projekt komplett laden)	30
	:KF (Moduln über Execute-Datei)	30
	:KC (COPYLIB / INCLUDE-Gruppe laden)	30
	:KCF (COPYLIB / INCLUDE-Gruppe über Execute-Datei)	31
5.1	Optionen für den Aufruf von SDKOPF	31
5.2	Ladelogik	32
5.2.1	Allgemeines Vorgehen.....	32
5.2.2	Kopfanfang / Kopfende.....	32
5.2.3	Modulname	32
5.2.4	Inhaltsangabe	32
5.2.5	Versionen-Zeile.....	33
5.2.6	Maschinentyp.....	33
5.2.7	Übersetzer	33
5.2.8	Programmaufruf	34
5.2.9	Eingabe-Parameter/Ausgabe-Parameter.....	34
5.2.10	Eingabe-Dateien/Ausgabe-Dateien (altes Format)	34
5.2.11	Dateireferenzen (neues Format)	35
5.2.12	Externe Programme	35
5.2.13	COMMON-Angaben	35
5.2.14	Schlagworte	36
5.2.15	Masken Referenzen.....	36
5.2.16	COPY Moduln / INCLUDE-Dateien	36
5.2.17	Bemerkungen	36
5.3	Der automatische Kopflader (SDKALL)	37
6.	Das Druckprogramm SDPRINT	38
6.0.1	Aufruf.....	38
6.0.2	Befehlseingabe.....	38
6.0.3	Listenformat	38
6.0.4	Schlüsselbegriffe.....	38
6.0.5	LISTF-Format (generische Eingabe)	39
6.0.6	Ja/Nein-Fragen.....	39
6.0.7	Voreinstellungen	39
6.0.8	Abbruch eines Befehls	39
6.1	Allgemeine Befehle.....	40
6.1.1	Programm beenden	40
6.1.2	Hilfe	40
6.1.3	Ausgabeeinheit definieren.....	40
6.1.4	Vorlaufseite unterdrücken.....	41
6.1.5	Vorlaufseite aktivieren.....	41
6.1.6	Anzahl Zeilen pro Seite einstellen	41

Inhaltsverzeichnis

6.2	Andrucke der SD/3000 internen Stammdaten	42
6.2.1	Übersicht der Ausgabeinheiten	42
6.2.2	Übersicht der Bildschirmeinheiten	42
6.2.3	ACCOUNT-Definitionen	42
6.2.4	Datei Zugriffsroutinen	43
6.3	Andrucke der Stammdaten	43
6.3.1	Projektübersicht	43
6.3.2	Sachgebietsübersicht	43
6.3.3	JOBID Verzeichnis	44
6.3.4	Dateiverzeichnis sortiert nach Namen	44
6.3.5	Dateiverzeichnis sortiert nach Projekten	45
6.3.6	COMMON Verzeichnis	45
6.3.7	Maskenverzeichnis	46
6.3.8	COPYLIB / INCLUDE-Gruppen Verzeichnis	46
6.3.9	Schlagwort Verzeichnis	47
6.3.10	Intrinsic Verzeichnis	47
6.4	Andrucke der Generierungsinformationen für Jobs	48
6.4.1	Übersetzungsjob Verzeichnis	48
6.4.2	Bindejob Verzeichnis	49
6.5	Dokumentationen	50
6.5.1	Modul Dokumentation	50
6.5.2	Bibliotheksübersicht sortiert nach Modulnamen	52
6.5.3	Bibliotheksübersicht sortiert nach Segmentnamen	52
6.5.4	Datei Referenzliste	53
6.5.5	COMMON Referenzliste	53
6.5.6	Masken Referenzliste	54
6.5.7	COPYLIB-Member / INCLUDE-Datei Referenzliste	54
6.5.8	Schlagwort Referenzliste	55
6.5.9	Versionsverzeichnis	55
6.5.10	Summary	56
6.5.11	Change Log	56
6.6	Dokumenten Verwaltung	57
6.6.1	Dokumentationsstruktur	57
6.6.2	Dokument Stati Verzeichnis	57
6.6.3	Dokumenttypen Verzeichnis	57
6.6.4	Dokument Einträge Verzeichnis	58
7.	Utilities	59
7.1	Das SPERR-System	59
7.1.1	SPERR (Sperren einer Datei)	59
7.1.2	ENTSPERR (Freigeben einer Datei)	59
7.1.3	CSPERR (Freigeben einer Datei mit allen wartenden)	60
7.1.4	LSPERR (Andruck Warteliste)	60
7.2	Der INCLUDE-Prozessor SDINCL	61
7.3	Das LIST-Programm SDLIST	62
7.4	Sourcekomprimierung mit COMPRESS/UNPRESS und SQUISHER	64
7.5	Der Software Transfer Generator SDTRANS	65
7.6	Das Projekt-Löschprogramm SDPURGE	66
7.7	Das Projekt-Entladeprogramm SDUNLOAD	67
7.8	Das Projekt-Ladeprogramm SDLOAD	68
7.9	Das FLABELINFO-Programm SDFLI	69
7.10	Das SPOOLFILE Namenprogramm SDODDM	69
7.11	Das POWERHOUSE Dictionary Ladeprogramm SDQDD	70

Inhaltsverzeichnis

7.12	Der SEGMENTER-Preprozessor SDSEG	70
7.13	Das Passwort-Utility SDSETPW	71
7.14	Das Filecode Utility SDSETFC	71
7.15	Der Source Scanner SDSCAN	72
7.16	Das Band / DDS Reload Programm LOADTAPE	72
7.17	Das Projekt Rename Programm SDPRREN	72
8.	COPYLIB Utilities	73
8.1	Das COPYLIB-Druckprogramm COPYLIST	73
8.2	Der COPY-Member Generator COPYDBD	73
8.3	Der COPY-Member Generator COPYGEN	74
8.4	Der COPY-Member Generator COPYPRN	75
9.	SD/3000 Source Bibliothek	77
	Speicherung der Dateien	77
	Referenzierung der Dateien	77
	Logfunktionen	77
	Aufruf SDSLIB	77
9.1	SDSLIB Funktion IN	78
9.2	SDSLIB Funktion OUT	78
9.3	SDSLIB Funktion INFO	79
9.4	SDSLIB Funktion CANCEL	79
9.5	SDSLIB Funktion DELETE	80
9.6	SDSLIB Funktion INIT	80
9.7	SDSLIB Funktion HELP	80
9.8	SDSLIB Funktion LIST	81
9.9	SDSLIB Funktion PRINT	81

Index

#	52
\$STDLIST Druckername.....	24
:COMPRESS	80
:COPYDBD	91
:COPYGEN	92
:COPYLIST	91
:COPYPRN.....	93
:CSPERR	76
:ENTSPERR	75
:GC	30
:GCF	32
:GCFX	32
:GCL	30
:GCLX	32
:GCX.....	32
:GL.....	30
:GLF	32
:GLFX.....	32
:GLX	32
:GPR	30
:GPRX	32
:KC	41
:KCF	42
:KCL	40
:KF	40
:KM.....	40
:KMOD.....	49
:KPR	40
:L	78
:LIST.....	78
:LOADTAPE.....	89
:LSPERR	76
:M	2
:P	50
:S	39
:SDLOAD.....	85
:SDPRREN	90
:SDPURGE.....	82
:SDS XEQ-Datei.....	96
:SDSCAN	89
:SDSEG	87
:SDTRANS.....	81
:SDUNLOAD	83
:SPERR.....	75
:SQ.....	80

Index

:UNPRESS.....	80
:UNSQ.....	80
?	52
@	52
^ bei COPYPRN.....	94
Abbruch eines Befehls in SDPRINT	52
ACCEss	57
Account.....	56
Account Option SDGEN	34
ACCOUNT's drucken.....	55
ALL-Option bei SDSLIB CANCEL Funktion.....	98
Änderungshistorie.....	11, 15, 20, 23, 24, 25
Aufruf SDPRINT.....	50
Ausgabeeinheit in SDPRINT	53
Ausgabeeinheiten drucken.....	55
Band / DDS Reload	89
Befehlseingabe SDPRINT	50
Benutzer INSTALL	97, 98
Bibliotheksübersicht nach Modulnamen drucken	68
Bibliotheksübersicht nach Segmentnamen drucken.....	68
Bildschirm LU	23
Bildschirm LU's drucken	55
Bildschirm Typ	23
Bindejob	30
Bindejob Definition	12
Bindejob Definitionen drucken.....	65
Binden mit Executedatei.....	32
BOTTOM-UP Referenzen.....	67
CANCEL SDSLIB Funktion	98
CHANGE.....	72
Change Log drucken.....	72
CLass	57
CModul.....	64
COmmon.....	60
COMMON.....	46
COMMON.....	21
COMMON Referenzen drucken	69
COMMON Verzeichnis drucken	60
CommonRef.....	69
COMMON-Typ	21
Compile-only Option SDGEN.....	34
Compiler Optionen	6
COMPRESS	80, 96
CONTROL-Y in SDPRINT.....	52
COPYDBD.....	91
COPYGEN	92
COPYLib.....	61
COPYLIB.....	29
COPYLIB / INCLUDE-Gruppe laden.....	41
COPYLIB /INCLUDE-Gruppe über Executedatei laden	42
COPYLIB Member.....	48
COPYLIB Typ.....	29

Index

COPYLIB Verzeichnis drucken	61
COPYLIB-Member Referenzen drucken.....	70
COPYLIST	91
COPYPRN.....	93
COPYRef.....	70
DATA-DICTIONARY Gruppe	25
Datei Referenzen drucken.....	69
Dateinamen	21
Dateireferenzen im Modulkopf (Eingabe/Ausgabe)	45
Dateireferenzen im Modulkopf (neues Format).....	46
Dateischutz	9, 15
Dateistatus	21
Dateityp	21
Dateiverzeichnis nach Namen drucken	59
Dateiverzeichnis nach Projekten drucken	60
Dateizugriff Kennbuchstabe	29
DELETE SDSLIB Funktion	99
Doc	74
Dokument Stati Verzeichnis drucken.....	73
Dokumentation Dateityp	28
Dokumentationsphase	26
Dokumentationsphase Beschreibung	26
Dokumentationsphase Priorität	26
Dokumentationsstatus	26
Dokumentationsstruktur andrucken	73
Dokumentationsstruktur Beschreibung.....	26
Dokumentationstyp	26
Dokumentationstyp Level	26, 28
Dokumentationstyp Priorität	26
Dokumenteinträge.....	28
Dokumenteinträge drucken	74
Dokumenttyp	28
Dokumenttyp Server	28
Dokumenttypen Verzeichnis drucken	73
Drucker	20, 23, 24
Drucker Filestatement.....	24
Drucker Klasse.....	24
Drucker LU.....	24
Drucker Name.....	24
Drucker Standort.....	24
Drucker Typ.....	24
Einheit.....	28
Einheitentyp	28
End SDPRINT	53
ENV/TT-Dateien Druckausgaben.....	24
EXKLUSIV-Option bei SDSLIB OUT Funktion	97
Externe Programme im Modulkopf	46
FILE SDLISTE.....	2, 20, 24, 35, 42
FILE SDTERM.....	3
FileName	59
FileProject.....	60
FileRef	69
FileType	73
Folgebefehl	5, 12
Folgemodul.....	5, 12

Index

Folgeprojekt.....	5, 12
FORMFile.....	61
FORMRef.....	70
Funktionstastenbelegung COPYPRN	94
GB Background Generator starten.....	37
GBE Background Generator beenden.....	37
Generator im Background.....	37
Generator Parameter Modifikation	37
Genfile-Feld bei COPYPRN.....	93
Group Option SDGEN.....	34
Help <befehl> in SDPRINT	53
Help All in SDPRINT.....	53
Help in SDPRINT.....	53
HELP SDSLIB Funktion	99
Hilfe SDPRINT	53
HOLD-Option bei SDSLIB IN Funktion.....	97
IN SDSLIB Funktion.....	97
INCLUDE-Bibliothek Gruppe.....	25
INCLUDE-Datei Referenzen drucken	70
INCLUDE-Dateien	48
INCLUDE-Gruppe	29
INCLUDE-Gruppen Verzeichnis drucken	61
Include-Prozessor	77
INFO SDSLIB Funktion	98
INIT SDSLIB Funktion	99
INPRI.....	11, 15, 18, 20
Install Option SDGEN	34
INSTALL-Account.....	17, 25, 55
INSTALL-Benutzer	97, 98
INTR Projekt	46
Intrinsic	19, 46, 63
Intrinsic Typ	19
Intrinsic Verzeichnis drucken	63
Intrinsicname	19
Intrinsicyp.....	46
Ja/Nein-Fragen SDPRINT	52
JCW SDCLEAN.....	35
JCW SDINCLUDE.....	6, 35, 77
JCW SDLISTE	35
JCW SDLU.....	23
JCW SDMPEINCLUDE	77
JCW SDXL.....	33, 35
JCW STDLISTDELETE	35
Jobid	59
JOBID.....	11, 15, 18
JOBID Verzeichnis drucken	59
Jobtexte.....	11, 16
JVAR SD_ADD_COPT	37
JVAR SD_ADD_LOPT.....	37
JVAR SD_ADD_POPT.....	37
JVAR SD_JOBQ	36
JVAR SD_JQM.....	36
JVAR SD_PURGELINK.....	36
JVAR SD_REP_COPT.....	38

Index

JVAR SD_REP_LOPT	38
JVAR SD_REP_POPT	38
JVAR SDLOCALCOPYLIB	25
JVAR SDSCAN_AUTOPASS	89
JVAR SDSCAN_DOWNSHIFT	89
JVAR SDSCAN_NOBINARY	89
JVAR SDSCAN_NOKSAM	89
JVAR SDSCAN_NOMSG	89
JVAR SDSCAN_NOVFORM	89
JVAR SDSCAN_UPSHIFT	89
JVAR SDSTR_FORCE_ACCOUNT	39
JVAR SDSTR_FORCE_GROUP	39
JVAR SDSTR_FORCE_USER	39
Keyword	63
KeywordRef	71
Kopf Option SDKOPF	42
Kopienanzahl Druckausgaben	24
Layoutdatei-Feld bei COPYPRN	93
LF-Feld bei COPYPRN	94
LIBM	68
LIBS	68
LINE-Feld bei COPYPRN	94
LINK/PREP-Optionen	9, 15
LINK-Job	12
LIST Funktion SDSLIB	100
List Option SDGEN	34
List Option SDKOPF	42
Liste-Feld bei COPYPRN	93
Listenformat SDPRINT	50
LISTF-Format SDPRINT	52
LISTF-Platzhalterzeichen	52
LModul	65
LOADTAPE	89
Logfunktionen SDSLIB	96
LOG-Option bei SDSLIB INFO Funktion	98
Maschinentyp im Modulkopf	44
Masken	48
Masken Referenzen drucken	70
Maskenverzeichnis drucken	61
Modul	5, 12, 30, 40, 66
Modul Dokumentation drucken	66
Modul laden	40
Modul über Executedatei laden	41
Modulkopf Anfang/Ende	43
Modulköpfe	43
Modulname im Modulkopf	43
MPE/iX	32, 35

Index

Nachricht	18, 20
Nachricht Option SDGEN	34
Name der Zugriffsroutine.....	29
Neu-Feld bei COPYPRN	93
NOPAGE0	54
OCTCOMP	35
Optionen	30, 40
Out	53
OUT SDSLIB Funktion	97
OUT-Option bei SDSLIB INFO Funktion.....	98
PAGE0.....	54
PAGELEN	54
Parameter im Modulkopf	45
Parameternummer der Zugriffsroutine	29
PICture Maske bei COPYPRN	94
Platteneinheit	13, 20
Position	28
Preprozessor Optionen.....	6
Print Funktion SDSLIB	100
Print Option SDGEN	34
PRInter.....	55
Produktionsaccount.....	25, 55
Programmaufruf im Modulkopf.....	45
PRoject	57
Projekt	5, 12, 17, 20, 30, 40
Projekt laden	40
Projekt Rename.....	89
Projekt übersetzen.....	30
Projektübersicht drucken	57
Projektvoreinstellung	25
Referenzierung SDSSLIB-Dateien	96
REP-Feld bei COPYPRN	94
RL	9, 15
RUNPRI	11, 15, 18, 20
Sachgebiet.....	5, 12, 17
Sachgebiet laden	40
Sachgebiet übersetzen.....	30
Sachgebiets Verantwortung	25
Sachgebietssperre	17
Sachgebietssperren	55
Sachgebietsübersicht drucken	57
Save Option SDGEN	34
Schlagwort	48
Schlagwort	19
Schlagwort Referenzen drucken	71
Schlagwort Verzeichnis drucken.....	63
Schlüsselbegriffe SDPRINT	50
Schlüsselworte im Modulkopf	43
SD_ADD_COPT JVAR	37
SD_ADD_LOPT JVAR.....	37
SD_ADD_POPT JVAR.....	37
SD_JOBQ JVAR	36
SD_JQM JVAR	36
SD_PURGELINK JVAR.....	36

Index

SD_REP_COPT JVAR.....	37
SD_REP_LOPT JVAR.....	37
SD_REP_POPT JVAR.....	37
SD3000.DATABASE.GLOBAL.....	2
SDCLEAN JCW.....	35
SDFLI.....	86
SDG.....	32
SDG{generatorbefehl}IN-Dateien.....	37
SDGCIN-Datei.....	37
SDGCXIN-Datei.....	37
SDGEN.....	30
SDGEN Optionen.....	34
SDGEN TDP/EDITOR/QEDIT Interface.....	32
SDGLIN-Datei.....	37
SDGLXIN-Datei.....	37
SDINCL.....	77
SDINCLUDE JCW.....	6, 35, 77
SDK.....	42
SDKALL.....	49
SDKOPF.....	40
SDKOPF Ladelogik.....	43
SDKOPF Optionen.....	42
SDKOPF TDP/QEDIT/EDITOR Interface.....	42
SDLIST.....	78
SDLIST Befehle.....	79
SDLISTE FILE.....	2, 20, 24, 35, 42
SDLISTE JCW.....	35
SDLOAD.....	85
SDLOCALCOPYLIB JVAR.....	25
SDLU JCW.....	23
SDM.....	2
SDMAINT.....	2
SDMAINT Dialogablauf.....	4
SDMAINT Funktionstasten.....	2
SDMAINT TDP/EDITOR/QEDIT Interface.....	2
SDMAINT Verwaltungsmenue.....	4
SDMAINT Zugriffsberechtigungen.....	4
SDMPEINCLUDE JCW.....	77
SDODDM.....	86
SDP.....	50
SD'P'ASCII.....	28
SDPRINT.....	50
SDPRREN.....	89
SDPURGE.....	82
SDQDD.....	87
SDSCAN.....	89
SDSCAN_AUTOPASS.....	89
SDSCAN_DOWNSHIFT.....	89
SDSCAN_NOBINARY.....	89
SDSCAN_NOKSAM.....	89
SDSCAN_NOMSG.....	89
SDSCAN_NOVFORM.....	89
SDSCAN_UPSHIFT.....	89
SDSEG.....	87
SDSETFC.....	88
SDSETPW.....	88
SDSLIB.....	96
SDSLIB Funktion CANCEL.....	98

Index

SDSLIB Funktion DELETE	99
SDSLIB Funktion HELP	99
SDSLIB Funktion IN	97
SDSLIB Funktion INFO	98
SDSLIB Funktion INIT	99
SDSLIB Funktion LIST	100
SDSLIB Funktion OUT	97
SDSLIB Funktion PRINT	100
SDSLIB Option ALL bei Funktion CANCEL	98
SDSLIB Option EXKLUSIV bei Funktion OUT	97
SDSLIB Option HOLD bei Funktion IN	97
SDSLIB Option LOG bei Funktion INFO	98
SDSLIB Option OUT bei Funktion INFO	98
SDSLIB Option SYS bei Funktion INFO	98
SDSLIB Option VERSION bei Funktion LIST	100
SDSLIB Option VERSION bei Funktion OUT	97
SDSLIB Option VERSION bei Funktion PRINT	100
SDSLIBnn-Gruppen	96
SDSPERR	75
SDSTR	39
SDSTR_FORCE_ACCOUNT JVAR	39
SDSTR_FORCE_GROUP JVAR	39
SDSTR_FORCE_USER JVAR	39
SDTERM FILE	3
SDTRANS	81
SDUNLOAD	83
SDXL JCW	33, 35
Segment	9, 15
Source Bibliothek	96
Source Scanner	89
Sourcedatei	6, 13
Sourcetyp	6, 13
Speicherung von SDSLIB-Dateien	96
SPERR System	20, 75
Spoolpriorität Druckausgaben	24
SQUISHER	80
Standortbeschreibung	23
STAt	73
STDLISTDELETE JCW	35
STRucture	73
Summary	72
Summary drucken	72
SYS-Option bei SDSLIB INFO Funktion	98
SYS-Projekt	45
System-Utility Dateiname	45, 46
SYS-UT Sachgebiet	45

Index

Terminal.....	55
Test Option SDGEN.....	34
Testaccount.....	25, 55
TOP-DOWN Referenzen.....	67
Übersetzen mit Executedatei.....	32
Übersetzer im Modulkopf.....	44
Übersetzungsjob.....	30
Übersetzungsjob Definition.....	5
Übersetzungsjob Definitionen drucken.....	64
UNPRESS.....	80
Version.....	71
VERSION-Option bei SDSLIB LIST Funktion.....	100
VERSION-Option bei SDSLIB OUT Funktion.....	97
VERSION-Option bei SDSLIB PRINT Funktion.....	100
Versionsverzeichnis drucken.....	71
Versionszeilen im Modulkopf.....	44
Voreinstellungen.....	20
Voreinstellungen SDPRINT.....	52
Vorlaufseite in SDPRINT aktivieren.....	54
Vorlaufseite in SDPRINT unterdrücken.....	54
Zeilen pro Seite in SDPRINT einstellen.....	54
Zieldatei.....	8, 13
Zieleinheit.....	9
Zieltyp.....	8, 13
Zugriffskode.....	29, 46
Zugriffsroutinen.....	29
Zugriffsroutinen Definition drucken.....	57