

A S T H M A

Eine anwendungsorientierte

Benutzerschnittstelle

IB 515-83/10

Freigabe: Die Bearbeiter:
C. Buro
W. Schuster

Unterschriften:

Constance Buro
M. Schuster

Der Abteilungsleiter:

M. Kortüm

Dr. W. Kortüm

Der stellv. Institutsdirektor:

Der Institutsdirektor:

Ackermann

Dr. J. Ackermann

Dieser Bericht enthält:

68 Blatt davon

9 Bilder

 Diagramme

A S T H M A

Eine anwendungsorientierte Benutzerschnittstelle

Constanze Buro
Wolfgang Schuster

D F V L R
Institut Dynamik der Flugsysteme
Abteilung Mehrkoerperdynamik

515/83/10

24. Januar 1984

INHALTSVERZEICHNIS

Bildverzeichnis	v
Vorwort	vii
1. Einleitung: Wie kam es zu ASTHMA ?	1
2. Arbeitsweise der SPF-Dialog-Management-Services	2
2.1 SPF	2
2.2 Ablauf des Dialogs	2
2.3 Dateiumgebung fuer SPF	2
2.4 Vorhandene Serviceroutinen des Dialog-Managers	4
2.5 Von SPF zu ASTHMA	5
3. Aufgabenspektrum von ASTHMA	6
3.1 Anwendung 0 - SPF	6
3.2 Anwendung 1 - MEDUSA	7
3.3 Anwendung 2 - FADYNA	7
3.4 Anwendung 3 - DSNSICH	7
3.5 Anwendung 4 - MKD	7
3.6 Anwendung P - PARAM	8
4. ASTHMA - Terminologie und Konventionen	9
4.1 Das Panel	9
4.2 Namensgebung	10
4.3 Programmablauf	11
5. Wartbarkeit von ASTHMA	14
5.1 Vorbelegungen	14
5.2 Fehlersuche in Problemfaellen	14
5.3 Dokumentation	15
Anhang A. Glossar	16
Anhang B. Literaturverzeichnis	18
Anhang C. Kurze Beschreibung der Optionen von Asthma	19
C.1 Anwendung MEDUSA - ASTHMA-Option 1	20
C.2 Anwendung DSNSICH - ASTHMA-Option 3	21
C.3 Anwendung MKD - ASTHMA-Option 4	22
Anhang D. Struktur der Anwendungen	23
Anhang E. Der Dokumentationsteil PANELDOC	28
Anhang F. Die Initialisierungs- und Startprozedur von ASTHMA	34
Anhang G. Abbildung der Hauptmanuale von ASTHMA	36

Anhang H. Beispiele fuer jede Art des Programmablaufes	41
H.1 Panel - CLIST - Programm	41
H.2 CLIST - Panel - Skeleton	44
H.3 CLIST	49
H.4 Programm - Panel - Skeleton	50

BILDVERZEICHNIS

Bild 1.	Arbeitsweise von SPF	3
Bild 2.	Verkettung der Panels	4
Bild 3.	Beispiel fuer eine Table	5
Bild 4.	Anwendungen von ASTHMA	6
Bild 5.	Aufbau eines Panels	9
Bild 6.	Beispiele fuer die Namensgebung in der Panel-Library	11
Bild 7.	Beispiel fuer die Namensgebung in der Message-Library	11
Bild 8.	Uebersicht der Namenskonventionen der SPF-Datei Umgebung	12
Bild 9.	Moeglichkeiten des Programmablaufes	12

VORWORT

Der vorliegende Bericht wendet sich in erster Linie an einen Benutzer des DFVLR-Grossrechners, der sich und u.U. einer ganzen Arbeitsgruppe den Umgang mit dem Rechner vereinfachen will. Der Bericht soll zeigen, dass mit dem Hilfsmittel SPF auf einfache Weise eine Schnittstelle fuer Programmsteuerung und Dateneingabe zu erstellen ist. Die Betonung liegt hier bewusst auf Schnittstelle, da eventuell im Hintergrund stehende (Hilfs-)Programme vorhanden sein muessen. Die mit SPF vom Betriebssystem angebotenen Hilfsmittel eignen sich vorzueglich, immer wiederkehrende Ablaeufe zu standardisieren und damit fuer den taeglichen Gebrauch zu vereinfachen.

1. EINLEITUNG: WIE KAM ES ZU ASTHMA ?

Die Abteilung Mehrkoerperdynamik benutzt fuer ihre Problemlösungen seit Jahren in erheblichem Umfang den vorhandenen Grossrechner (IBM/MVS). Die Benutzer dieses Rechners aus der Abteilung sind vorwiegend wissenschaftliche Mitarbeiter, die den Rechner als Mittel zum Zwecke der Lösung bzw. Verifizierung ihrer Forschungsaufgaben sehen. Aus diesen Vorbedingungen heraus entstanden im Laufe der Zeit einige grosse Anwendungs-Softwarepakete mit diversen kleineren Hilfsprogrammen (z.B. erstellen von Lademodulen, Start von Programmen, Beschreiben von Magnetbaendern etc.). Diese wurden jeweils in der als geeignet erscheinenden Sprache geschrieben (FORTRAN, PL/1, TSO-Prozeduren). Die meisten Programme wurden mittels einer TSO-Prozedur aufgerufen, einige jedoch mit Hilfe einer Unzahl von vorgefertigten Job-Control-Saetzen. Die Schnittstelle fuer den Benutzer war also von Programm zu Programm unterschiedlich. Das Chaos wurde perfekt durch eine jeweils unterschiedliche Eingabeart (zeilenorientiert, seitenorientiert) und eine unterschiedliche Steuerung (Kommando, Manual). Insgesamt stand die Handhabung also im Gegensatz zu dem angebotenen Systemhilfsmittel SPF (System Productivity Facility) [1].

Mit Betrachtung der Situation lag es nahe, die vorhandene Programmsubstanz mit SPF zu einem Stueck zu verschmelzen. Mit der vom Rechenzentrum angebotenen [2] neuen Version von SPF (ISPF) wurde nun ein Hilfsmittel mit dem Namen ASTHMA (Anwendungen und **ST**rukturierte Programmier**H**ilfen der **M**ehrkoerperdynamik**A**bteilung) geschaffen. ASTHMA ist als Hilfe fuer den Benutzer gedacht, der in anwendungsnahen Formulierungen seine Anforderungen in leichter und einheitlicher Weise formulieren kann.

ASTHMA soll also auf der einen Seite dem wissenschaftlichen Benutzer helfen, mit moeglichst geringem DV-Wissen seine Probleme auf den Rechner zu bringen. Auf der anderen Seite soll es sich demjenigen, der die Programme verwaltet, als Einheit in der Programm-Struktur und den verwendeten Hilfsmitteln darstellen. Diese sind unter dem Namen "SPF-Dialog-Management-Services" bekannt und werden im Folgenden beschrieben.

2. ARBEITSWEISE DER SPF-DIALOG-MANAGEMENT-SERVICES

2.1 SPF

Das Programmpaket SPF (System Productivity Facility) [1] wird von IBM in einer Grundversion angeboten als eine Hilfe in der Programmentwicklung. Es unterstützt die interaktive Tätigkeit am Bildschirm und bildet somit die Schnittstelle zu den vorhandenen TSO-Kommandos (z.B. FORT, LINK, TESTFORT...) und den JCL-Prozeduren (z.B. FHGG, POCL).

SPF besteht aus zwei Teilen:

- *Program Developing Facility*: Darunter versteht man die Optionen, die der Benutzer unter dem Namen SPF zur Verfügung gestellt bekommt. (Z.B. Browse, Edit, Utilities)
- *Dialog-Manager*: Er besteht aus Control-Funktionen und den Dialog-Service-Routinen (siehe Bild 1 auf S.3). Durch seine Existenz wird die Neuentwicklung zusätzlicher Anwendungen möglich.

2.2 Ablauf des Dialogs

Durch das ISPF-Kommando wird der SPF-Controller aufgerufen, dessen Aufgabe es ist, die Voraussetzungen für SPF herzustellen, d.h. die Umgebung von SPF zu initialisieren, und danach den Select-Service aufzurufen. Mit diesem Aufruf wird die Wurzel der gesamten Struktur angesprungen (Master Application Menu siehe Bild 2 auf S.4).

Eine weitere Verzweigung erfolgt über eine hierarchische Folge von Select-Menus (eine Bildschirmmaske wird durch die vorhergehende übergeordnete aufgerufen). Am Ende dieser Kette von Select-(Anwendungs-) Menus erscheint dann in der Regel ein Dateneingabe-Panel, welches von einem Programm (z.B. PL/1) oder einer TSO-Prozedur, genannt die Dialogfunktion, ausgewertet wird. Um diesen Ablauf zu ermöglichen sind einige Datasets notwendig.

Weitere Control-Funktionen sind die physikalische Verwaltung von dem ganzen oder aufgeteilten Bildschirm und die Überwachung des Tutorials, in dem Hilfestellungen für den Benutzer stehen.

2.3 Dateiumgebung für SPF

Die Menge aller für den Dialog und dessen Abwicklung nötigen Informationen sind in einem Satz von Dateien, in der Regel Partitioned Datasets, entsprechend ihrer Bedeutung zusammengefasst abgespeichert. Im folgenden Absatz sind die Dateien und deren Inhalt kurz beschrieben.

Panel-Library Sie beinhaltet in jedem Member ein Panel. Dies besteht vereinfacht gesagt aus zwei Teilen. Der eine definiert die Bildschirmmaske, die dem Benutzer die Möglichkeit zur Parametereingabe bietet. Der zweite Abschnitt des Panels verarbeitet die Eingabedaten, belegt Eingabefelder mit Standardwerten vor und prüft die Angaben des Benutzers auf logische und syntaktische Richtigkeit.

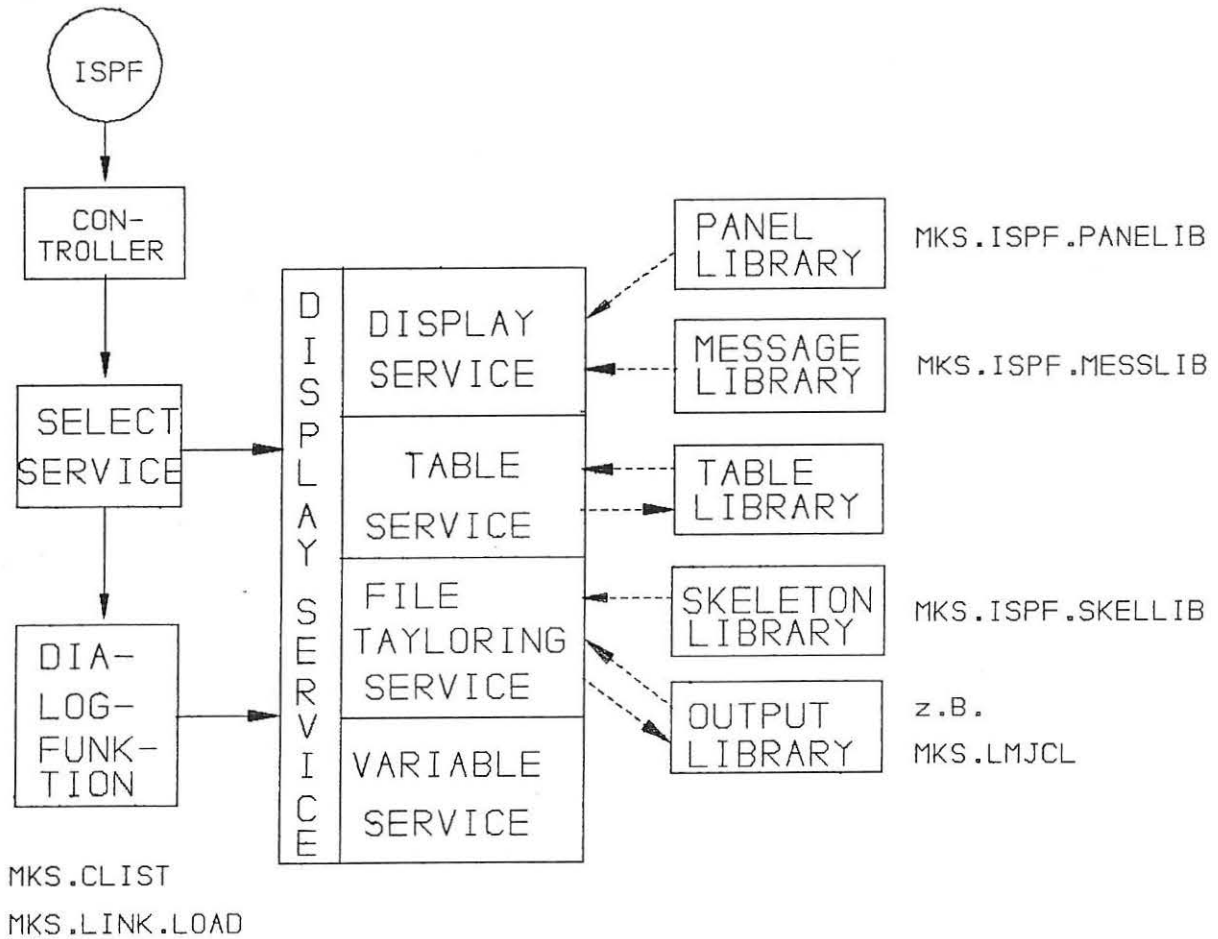


Bild 1. Arbeitsweise von SPFF

Message-Library Datei, in dessen Members jeweils zehn einzeilige Nachrichten in kurzer und langer Form abgespeichert sind, die Information fuer den Benutzer bzgl. der aktuellen Situation, in der er sich befindet, enthalten. z.B. kann bei fehlerhafter Parametereingabe eine Meldung an den Benutzer geschickt werden, die ihm seine Fehler aufzeigt.

Table-Library Datei, die die sog. Tables enthaelt. Das sind zwei-dimensionale Felder, die zur Aufbewahrung von Daten, die im Dialog verwendet werden, angelegt werden. Sie koennen temporaeer oder ueber eine Session hinaus gehalten werden, so dass die Information der Table fuer weitere Aufrufe von ASTHMA erhalten bleibt (siehe Bild 3 auf S.5). In ASTHMA wird bisher die Struktur der Table nicht benoetigt.

Skeleton-Library Datei, deren Member Gerueste von Job-Control-Statements enthalten, zur Erstellung einer neuen Datei fuer den Start von Programmen im Batch.

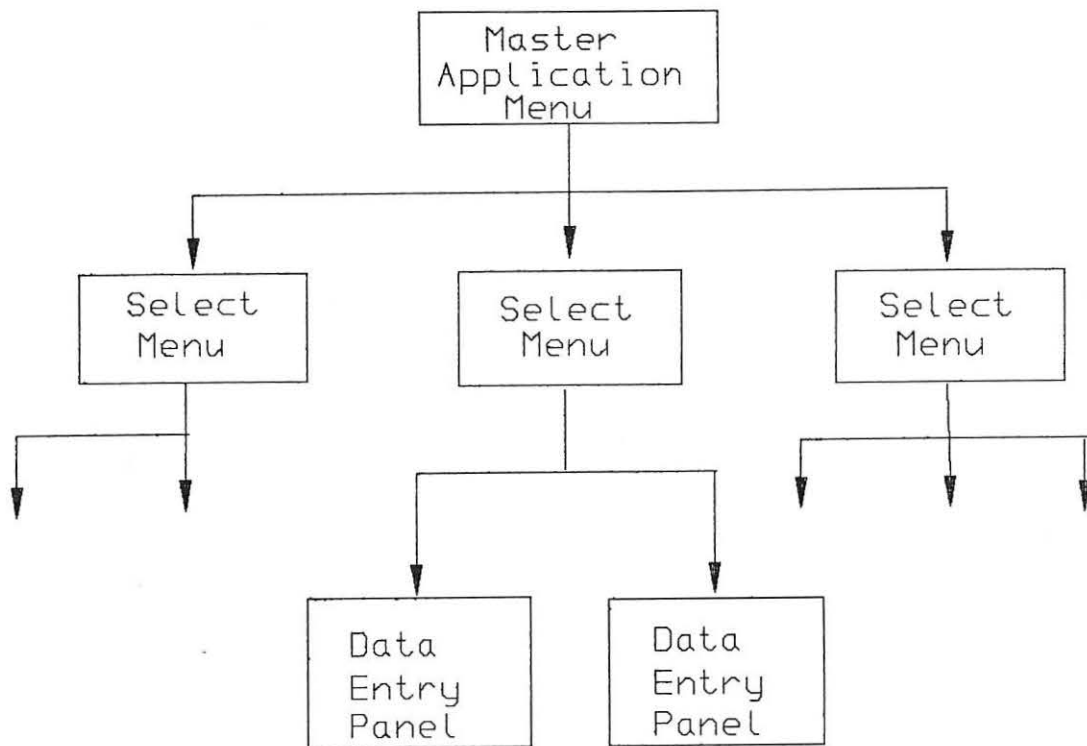


Bild 2. Verkettung der Panels

Output-Library Sie muss nur dann angegeben werden, falls die aktuelle Job-control, die aus dem Skeleton erstellt worden ist, nicht in dem vom System angelegten temporaeren File gespeichert werden, sondern noch nach der Laufzeit des Programms erhalten bleiben soll.

Dialog-Datei Je nach Programmiersprache der Funktion ist die Datei fuer die Dialogfunktion eine Datei fuer CLIST-Prozeduren und/oder eine fuer Programme. Die Programme muessen, da sie dynamisch gelinkt werden, als startfaehige Lademodule abgespeichert sein.

2.4 Vorhandene Serviceroutinen des Dialog-Managers

Fuer die Steuerung des Ablaufs, die Ausgabe der Panels und Informationen und den Datenaustausch werden eine Anzahl von Routinen durch den Dialog-Manager angeboten, die in gewissem Sinne mit der Dateioorganisation in Zusammenhang stehen. Im Folgenden soll nur ein grober Ueberblick ueber die einzelnen Gruppen von Routinen gegeben werden.

- Der *Display-Service* liest ein Panel aus der Bibliothek, initialisiert die noetigen Variablen und gibt es anschliessend am Bildschirm aus. Nach Eingabe durch den Benutzer werden die entsprechenden Variablen analysiert und unter Umstaenden eine Nachricht ausgegeben. Zuletzt werden die

Variablen in den internen Speicher fuer die weitere Verarbeitung durch die Dialogfunktion geschrieben. Diese Variablen werden auch Dialogvariablen genannt, weil sie den Informationstransfer im Dialog leisten.

- Der *File Tailoring-Service* liest ein Job-Control-Skelett aus der Skeleton-Library und fuehlt die Dialogvariablen mit den aktuellen Eingabewerten aus. Typische Operationen sind das Eroeffnen der temporaeeren Datei und das weitergeben als der Job-Control als Batch Job (submit).
- Der *Variable-Service* erlaubt einer Dialogfunktion die Definition und den Gebrauch von Dialogvariablen. Diese sind Zeichenketten, die als Informationstransporteure zwischen Dialogfunktion und Serviceroutinen dienen. Hierbei ist der Gueltigkeitsbereich der Dialogvariablen wichtig. Es gibt zwei Arten der Speicherung, entweder im sog. Shared-Pool, in dem die Variablen waehrend eines Aufrufs von SPF gespeichert sind, oder im Profile-Pool, in dem die Variablen ueber eine Sitzung hinaus aufgehoben werden. Typische Aufgabe des Variablen-Services ist es, die Variablen aus dem Shared- bzw. Profile-Pool herauszulesen und wieder zurueckzuschreiben. Er ermoeoglicht die Ansprechbarkeit der Dialogvariablen, welche Eingabewerte enthalten von der Dialogfunktion aus.
- Der *Table-Service* erstellt und verwaltet Saetze von Dialogvariablen, genannt Tables. Eine Table enthaelt in jeder Spalte verschiedene Werte fuer eine Dialogvariable, in jeder Zeile einen Satz von Werten zu diesen Dialogvariablen (siehe Bild 3). Typische Operationen sind das Anlegen und Eroeffnen von Tabellen, das Hinzufuegen oder Aendern einer Zeile in einer Table.

NAME	VORNAME	KENNUNG	WOHNORT
Meier	Hugo	345	Bonn
Brunner	Adele	7888	Koeln
Schmidt	Franz	91	Bruessel

Bild 3. Beispiel fuer eine Table

2.5 Von SPF zu ASTHMA

Geht man davon aus, dass SPF die Schnittstelle zu den Grundfunktionen eines Betriebssystems darstellt, so liegt es nahe, die Anforderungen an eine Schnittstelle bzgl. der einzelnen Anwendungen in gleicher Art anzubieten. ASTHMA nuetzt die Angebote des Dialog-Managers aus, mit Hilfe der Serviceroutinen eigene Optionen und Anwendungen zu schreiben. SPF wird als eine Anwendung neben anderen unter die Oberstruktur ASTHMA gestellt. Das folgende Kapitel gibt einen Ueberblick ueber das Aufgabenspektrum von ASTHMA.

3. AUFGABENSPEKTRUM VON ASTHMA

ASTHMA besteht aus sechs Anwendungen, die wiederum in mehrere Optionen unterteilt sind. Welche Optionen im einzelnen angeboten werden, kann man im Anhang G. auf S.36 und Anhang C. auf S.19 genau nachlesen. Die Optionen einer Anwendung fassen

- a) ein grosses Programmsystem nebst Umgebung oder
 - b) alle Programme mit einem gemeinsamen Nutzungsbereich
- zusammen. Ziel in den problemorientierten Anwendungen (1,2,3) ist eine Minimierung der noetigen Benutzereingaben. Dies setzt ein relativ "starres" Dateigeruest vorraus, auf welches immer automatisch zugegriffen wird.

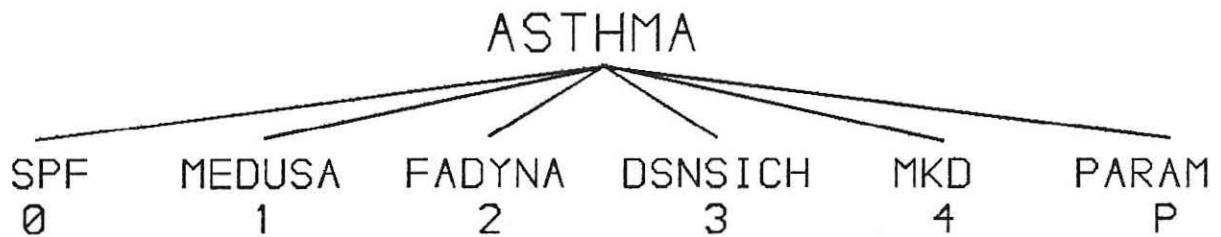


Bild 4. Anwendungen von ASTHMA

3.1 Anwendung 0 - SPF

Das von IBM angebotene und in der DFVLR installierte Programmprodukt SPF wurde in ASTHMA uebernommen. Die in ASTHMA enthaltene Option 0 wurde zu einer eigenstaendigen Anwendung (siehe Anwendung P) umgewandelt.

SPF beinhaltet Optionen, die problem- und aufgabenunabhaengig sind, d.h. sie sind nicht fuer einen speziellen Personenkreis geschrieben, sondern so allgemein, dass sie fuer eine grosse Anzahl von Benutzern interessant sind.

3.2 Anwendung 1 - MEDUSA

Hierunter findet man eine Anzahl von Prozeduren, die eng im Zusammenhang mit dem Programmpaket MEDUSA¹⁾ stehen [3]. Sie sollen der Entwicklergruppe von MEDUSA die Arbeit mit dem Rechner erleichtern, indem ihr DV-spezifische Probleme abgenommen werden.

Beispiele sind:

- Erstellen von Auslieferungsversionen
- Starten von Lademodulen
- Dokumentationshilfen
- Automatische Allocierung der Standarddateien
- Starten von Vorlaufprogrammen, die die MEDUSA-Umgebung zum Start vorbereiten.

3.3 Anwendung 2 - FADYNA

Wenn man diese Anwendung wählt, erscheint sofort die erste Bildschirmmaske von FADYNA²⁾ [4]. Es wird hier nur eine Schnittstelle zu einem komplexen Kommando-Prozeduren-System angeboten, weil sich auf Grund der Situation der Aufwand einer vollständigen Integration nicht lohnt. Ablauf und Struktur von FADYNA bleiben vollständig in ihrer alten Form erhalten.

3.4 Anwendung 3 - DSNSICH

Die beiden Optionen, die hier angeboten werden, sind wie in SPF programmübergreifend. Man kann hier entweder alle Dateien, die zu einem Programmkomplex gehören, auf dem MSS (Mass Storage System) oder einzelne Dateien unter Führung eines Katalogs auf Band sichern.

3.5 Anwendung 4 - MKD

Diese Anwendung könnte auch "SPF fuer die Abteilung MKD" genannt werden, denn unter MKD befinden sich zusätzliche Hilfsprogramme fuer die Rechnernutzung im Hinblick auf die Abteilung MKD. Entsprechend der Definition von SPF ist dieser Teil unabhängig von einem Programmsystem und von einer festen Dateistruktur. Letztgenante Freiheit muss natuerlich mit einer erhoekten Anzahl von Benutzereingaben erkaufte werden.

Beispiele sind:

- Erzeugen von Lademodulen (PL/1, Fortran, Linkage Editor)
- Ausgabe komprimierter Texte
- Aufruf des Watfiv-Compilers
- Sortieren von Labeln einer Fortran-Quelle

¹⁾ MEDUSA: MEhrkoerperDynamik Und SystemAnalyse

²⁾ FADYNA: Programmsystem zur Simulation der FahrzeugDYNAmik

3.6 Anwendung P - PARAM

Diese Anwendung besteht aus den drei Untergliederungen der Option 0 von SPF, das von IBM zur Verfügung gestellt wird, und einer vierten, in der der Drucker, auf dem der einzelne Benutzer seine Ausgabe haben will, voreingestellt werden kann. Diese Voreinstellung wird dann automatisch in den Optionen von ASTHMA verwendet. D.h. diese Anwendung beinhaltet allgemeine Angaben, die zum Teil für die Benutzung von ASTHMA notwendig sind, zum anderen Teil die Arbeit am Rechner erleichtern und beschleunigen (z.B. Belegung der PF-Keys).

4. ASTHMA - TERMINOLOGIE UND KONVENTIONEN

4.1 Das Panel

Wie schon kurz angedeutet besteht das Panel aus der Bildschirmmaske, die fuer den Benutzer sichtbar ist, und dem internen Verarbeitungsteil, in dem die Eingaben des Benutzers auf syntaktische und logische Richtigkeit und Vollstaendigkeit ueberprueft werden und in eine zur Weiterverarbeitung brauchbare Form umgesetzt werden. Das Panel ist Schnittstelle zwischen Dialogfunktion und Benutzer. Es gibt dem Benutzer in einheitlicher Form zu verstehen, welche Angaben von ihm gefordert werden, nimmt diese dann auf und gibt sie an die Dialogfunktion weiter.

In diesem Kapitel wird das Panel nur von der Benutzerseite betrachtet und die Form der Bildschirmmaske beschrieben (Bild 5). Fuer den Verarbeitungsteil gibt es im SPF-Handbuch [1] eine genaue Beschreibung der Control-Strukturen.

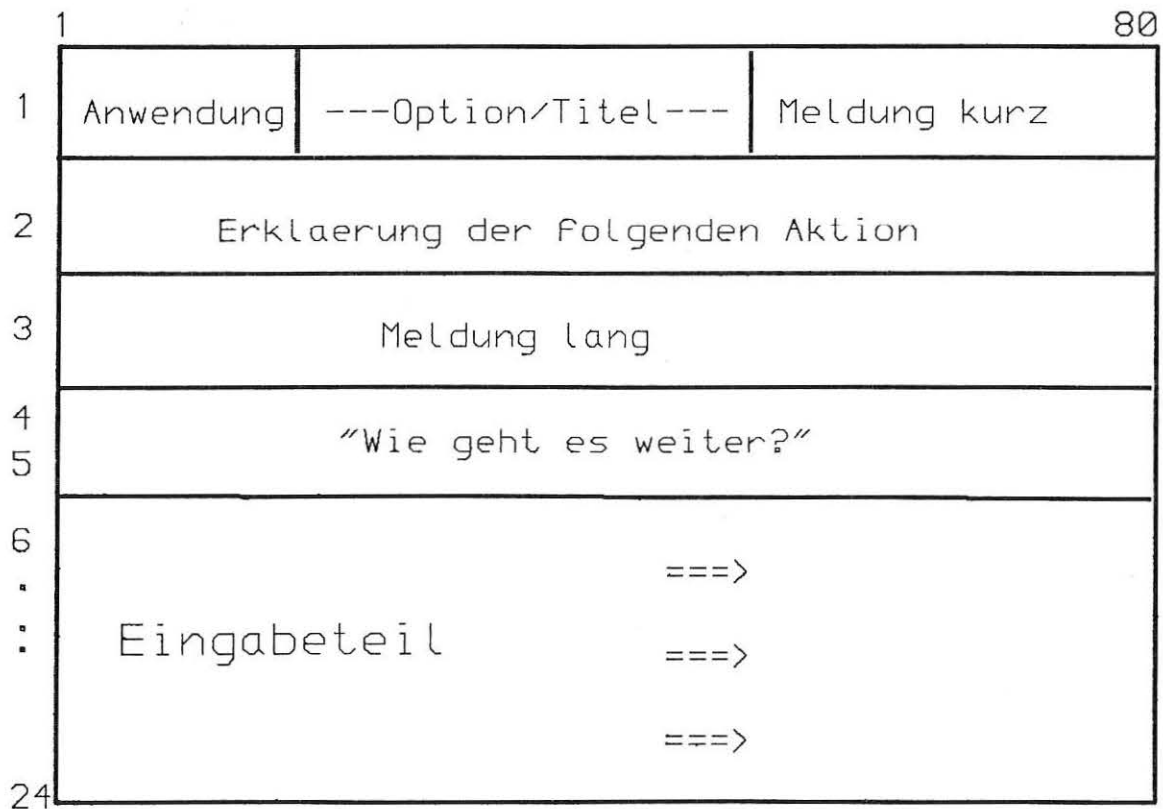


Bild 5. Aufbau eines Panels

Die erste Zeile der Bildschirmmaske enthaelt links die Anwendung (z.B. MEDUSA), in der Mitte die Kurzform der Option (z.B. IAINIT). Rechts ist Platz fuer eine kurze Meldung an den Benutzer. Dieser Text erscheint z.B., wenn eine Eingabe fehlerhaft ist.

Die zweite Zeile fordert den Benutzer zur Aktion auf. Im Allgemeinen soll er die Felder im Eingabeteil, die durch die Zeichen \implies gekennzeichnet sind, mit den gewünschten Daten belegen. In diesem Falle wuerde stehen:

EINGEBEN UND VERAENDERN DER UNTENSTEHENDEN PARAMETER

Die dritte Zeile zeigt sich vorerst einmal als frei, denn sie laesst Platz fuer einen groesseren Hilfstext. Diesen kann der Anwender sichtbar machen, indem er den Help-Key drueckt, nachdem im rechten, oberen Eck eine kurze Meldung erschienen ist. Falls also diese Meldung nicht fuer die Fehlerkorrektur ausreicht, kann man eine zusaetzliche ausfuehrliche Version der kleinen Meldung erhalten.

Auf den folgenden Zeilen wird erkluert, wie der Programmablauf fortgesetzt werden kann. z.B.

DRUECKEN SIE ENTER ZUM SUBMIT DES JOBS.

Darunter folgt der Eingabeteil. Die einzelnen Eingabefelder sind gekennzeichnet durch

\implies

4.2 Namensgebung

Sowohl fuer die Dateinamen als auch fuer die Membernamen soll gelten, dass man anhand von ihnen erkennen kann, um welche Art von Daten, welche Anwendung und Option es sich handelt. Die drei bzw. vier speziellen ASTHMA-Dateien erhielten gemaess SPF folgende Namen (siehe Bild 1 auf S.3).

```
MKS.ISPF.PANELIB (fuer Panels)
MKS.ISPF.SKELLIB ( "  Skeletons)
MKS.ISPF.MESSLIB ( "  Messages)
(MKS.ISPF.TABELIB) ( "  Tables)
```

An dem Namen PANELIB ist gleich zu erkennen, dass es sich um eine Library fuer Panels handelt. Wie findet man nun das gesuchte Member im Partitioned Data Set. Dies ist aehnlich einfach, wenn man die zugrunde liegenden Namenskonventionen kennt: (siehe hierzu Bild 8 auf S.12).

- Die ersten drei Buchstaben des Membernemens kennzeichnen die Anwendung z.B. MED, MKD.
- An der vierten Stelle steht ein Trennzeichen. Hierfuer wurde der Klammeraffe auch "at"-Sign(@) gewaehlt.
- Das fuenfte Zeichen gibt die Nummer der Option in einer Anwendung an. Falls das gesuchte Manual ein Hauptmanual ist, ist das fuenfte und sechste Zeichen ein "HM".
- Die folgenden Zeichen nummerieren die Panels in einer Option durch.

Fuer Members in den Dateien der Dialogfunktionen, in der

```
MKS.CLIST      (fuer TSO-Proceduren)
MKS.LINK.LOAD ( "  Programme)
```

M E D	@	8	1
Anwendung MEDUSA	Trenn- zeichen	Option 8	1. Panel in der Option
M K D	@	H M	
Anwendung MKD	Trenn- zeichen	Haupt- manual	

Bild 6. Beispiele fuer die Namensgebung in der Panel-Library

die ASTHMA betreffen, gilt dieselbe Regelung. Jedoch wird voran ein "Y" gesetzt. Dies geschieht, damit alle Programme von ASTHMA in der Memberliste zusammen stehen und durch das "Y" als zu ASTHMA gehoerig erkennbar sind. Nicht vollstaendig nach diesem Schema sind die Namen der Member in der Message-Library gebildet, denn nach den drei Buchstaben fuer die Anwendung und dem Trennzeichen folgt eine zweistellige Zahl (siehe Bild 7). Sie ist SPF-Vorschrift und bildet zusammen mit einer laufenden Nummer den vollstaendigen Namen der Message.

M e m b e r n a m e				

M E D	@	0	1	5

Anwendung Medusa	Trenn- Zeichen	zwei-stellige Zahl fuer den Membersnamen	laufende Nummer	

Name der Message Nummer 015 aus dem Member MED@01

Bild 7. Beispiel fuer die Namensgebung in der Message-Library

4.3 Programmablauf

In ASTHMA sind bisher vier verschiedene Strukturen einer Option verwendet worden, die hier , ausgehend von einem der Anwendungs-Hauptmanuale, beschrieben werden.

- *Panel - CLIST - Programm/Skeleton*
Vom Hauptmanual wird durch den Select-Service ein oder mehrere Select-Menus und zuletzt eine CLIST-Procedur aufgerufen , die dann ein Programm

Zeichen	Panel/Skeleton	Dialogfunktion	Messages
1	Anwendungskuerzel	y	Anwendungskuerzel
2	"	Anwendungskuerzel	"
3	"	"	"
4	@	"	@
5	Option / H	@	zwei-stellige
6	Option / M /	Option	Zahl
7	lfde. Nummer	lfde. Nummer	

Bild 8. Uebersicht der Namenskonventionen der SPF-Datei Umgebung

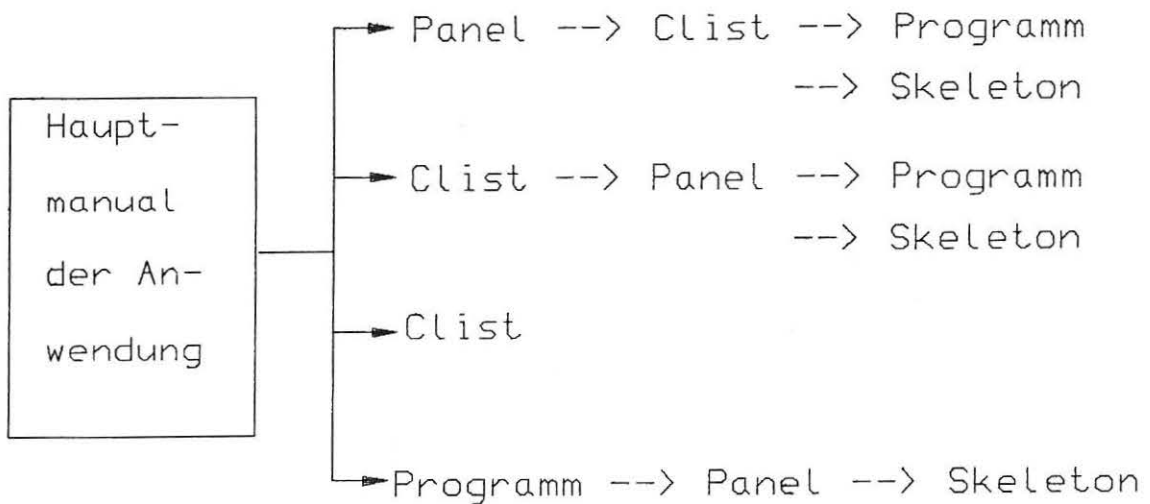


Bild 9. Moeglichkeiten des Programmablaufes

startet oder ein Skeleton mit den aktuellen Werten fuellt und abschickt (submit).

- *CLIST - Panel - Programm/Skeleton*
Hier wird vom Hauptmanual sofort eine CLIST-Prozedur aufgerufen, die den weiteren Ablauf der Option steuert. Von dieser CLIST-Prozedur werden durch den Display-Service die Panels am Bildschirm gezeigt. Die Prozedur wird solange aufgehalten, bis der Benutzer die Fortfuehrung erlaubt. Wenn alle benoetigten Informationen vom Benutzer gegeben worden sind, wird wieder wie im vorigen Fall ein Programm oder ein Batch-Job gestartet.
- *CLIST*

Diese Form ist die einfachste; denn sie benoetigt keine Information vom Benutzer. D.h. es gibt keine Panels, die vom Benutzer irgendwelche Eingaben verlangen. Vom Hauptmanual wird per Select-Service eine CLIST-Prozedur aufgerufen, die selbstaendig die gestellten Anforderungen an diese Option ausfuehren kann.

- *Programm - Panel - Skeleton*

In diesem Fall wird nun als Dialogfunktion ein Programm verwendet, das in PL/1 geschrieben ist und dieselbe Aufgabe hat, wie in den obigen Faellen die CLIST-Prozedur. Es wird mit Hilfe des Select-Services vom Hauptmanual aus aufgerufen und uebernimmt dann die Organisation. Von dem Programm aus wird das Erscheinen der Panels auf dem Bildschirm, das Erstellen der aktuellen Job-Control und deren Start gesteuert.

Um sich diese vier Methoden besser verdeutlichen zu koennen, ist im Anhang H. auf S.41 fuer jede der vier Formen ein Beispiel abgebildet. Jedoch kann man sie trotz ihrer Verschiedenheit auf zwei Grundprinzipien zurueckfuehren.

- Erstellen und Submit einer JCL (Skeleton)
- Ausfuehren eines Programmes im Foreground

5. WARTBARKEIT VON ASTHMA

ASTHMA muss sich leicht an Veraenderungen in Soft- und Hardware anpassen koennen. Es koennen sich oft verwendete Dateinamen aendern(z.B. Standarddateien von MEDUSA), neue Prozeduren fuer Compiler vom Rechenzentrum angeboten werden oder periphere Geraete bzw. deren Ansprechbarkeit geaendert werden; d.h. ASTHMA ist von den Aenderungen des Rechenzentrums und der Benutzer in gewissem Masse abhaengig und muss sich deshalb leicht in die neuen Voraussetzungen einfuegen.

5.1 Vorbelegungen

Deshalb ist in ASTHMA eine Initialisierungs- und Startprozedur definiert (siehe Anhang F. auf S.34). In ihr werden alle Variablen, die im Profile-Pool sind, in den Shared-Pool eingelesen und damit vor Ausgabe des ersten Panels die Variablen wieder mit den Werten des vorangegangenen Aufrufs belegt. Dies macht die Suche nach der Art einer Variable leichter, denn man muss nur noch in einem Programm nachsehen, ob die Variable in den Shared-Pool geschrieben wurde oder nicht. Im anderen Falle muesste man in allen Prozeduren nachsehen, ob und wie eine Variable dort behandelt wird, um Widersprueche zu vermeiden. Genauso ist es bei Vorbelegungen. Oft werden Eingabefelder in den Panels mit voreingestellten oder anderen haeufig gebrauchten Werten belegt, so dass sie der Benutzer nicht mehr eigenhaendig eintippen muss, falls er den vorbelegten Wert als Eingabe wuenscht. Die zweite Nutzung von Vorbelegungen geschieht in der Dialogfunktion. Dort werden oft am Anfang einer Prozedur bzw. eines Programms lange Dateinamen auf eine Variable gelegt, damit bei deren Aenderungen nur die Variable neu belegt werden muss. Kommt nun diese spezielle Variable in mehreren Programmen vor, wird sie in jedem Programm neu mit dem Namen dieser Datei belegt. Wird eine Aenderung notwendig, muessen doch wieder alle Programme nach dieser Variablen durchgesucht werden. Um diese unangenehme Suche zu umgehen, werden auch diese oft benutzten Variablen in der Hauptprozedur belegt und dann in den Shared-Pool geschrieben. Vergleichbar mit dem COMMON-Block in Fortran sind nun alle Variablen ueber den Shared-Pool ansprechbar.

5.2 Fehlersuche in Problemfaellen

Die zweite Anforderung an ASTHMA ist, den Programmablauf mitverfolgen zu koennen. Dies ist wichtig, um aufgetretene Fehler leichter finden, beheben und austesten zu koennen. Deshalb gibt es in ASTHMA eine sogenannte Testvariable, die die Werte "Ja" und "Nein" annehmen kann. Diese Variable ist ein unsichtbares Eingabefeld (vgl. Passwort) im Hauptmanual von ASTHMA. Je nach ihrem Wert wird die Ausgabe des Prozedurverlaufes auf das Terminal unterdrueckt (Test=Nein) oder ausgegeben (Test=Ja) und der gestarte Job geloescht

(Test=Nein) oder in der H-Klasse aufgehoben (Test=Ja).Voreingestellt ist fuer die Testvariable "Nein".

5.3 Dokumentation

- **Paneldoc**
In der MKS.CLIST gibt es ein Member, das alle Dialogvariablen beschreibt, die in den Panels verwendet werden. Fuer jedes Panel ist pro Variable der Name, eine kurze Beschreibung und eine Kennzeichnung, falls die Variable im Profile-Pool ist, im Paneldoc aufgelistet.
- **Art der Dokumentation der Dialogfunktionen**
Jede Dialogfunktion ist nach einem einheitlichen Schema aufgebaut. Nach der Beschreibung der Variablen aus den Panels der Option und aus der Haupt-Prozedur folgen die Deklarationen, das Einlesen der Variablen aus dem Sharedpool, Allocierungen von Dateien und/oder Abfragen auf Existenz der Dateien. Als Letztes folgt der Submit oder Programmaufruf. Jeder dieser Teile ist als Block vom anderen durch Kommentarzeilen abgegrenzt.
- **Dokumentation in der Message-Library**
In der Message-Library sind jeweils zwei Zeilen fuer die kurze und die lange Message vorgesehen. In der dritten Zeile, die als Kommentar gewertet wird, sind die Panels, Prozeduren und Programme aufgezählt, die die davorstehende Message verwenden. So ist die logische Verbindung zwischen aufrufender Einheit und Message dokumentiert.

ANHANG A. GLOSSAR

CLIST	Namensteil des IBM-Betriebssystems MVS, der als Inhalt TSO-Kommandoprozeduren ausweist.
Dialogfunktion	Programm oder TSO-Kommandoprozedur, die den Dialog mit dem Benutzer steuert.
Dialogvariable	Variable mit dem Inhalt der Benutzereingaben, die zum Datenaustausch von Programmen und Kommandoprozeduren dient.
FADYNA	Programm zur Simulation und Analyse der Fahrzeugdynamik bodengebundener Transportsysteme.
ISPF	Kommando zum Aufruf von SPF
Master Application Menu	Bildschirmmaske zur Auswahl der implementierten Anwendungen.
MEDUSA	Programm zur Simulation und Analyse von Mehrkoerpersystemen.
Menu	Bildschirmmaske mit genau einem Eingabefeld, das zur Angabe einer anzuwaehlenden Option dient.
Message	Meldung fuer den Benutzer in kurzer oder/und langer Form.
Panel	Bildschirmmaske mit definierten Ein- und Ausgabefeldern fuer den Dialog mit dem Benutzer. (Oberbegriff)
Partitioned Dataset	Dateiorganisationsform des IBM-Betriebssystems MVS. Bei ihr werden in einer sequentiellen Datei viele untergliederte Teile (Mmeber) verwaltet.
PF-Keys	Tasten auf der Tastatur des Terminals, die mit bestimmten Funktionen belegt werden koennen.

Profile-Pool	Member eines Partitioned Datasets welches den Inhalt aller wichtigen Variablen (Eingabewerte) beinhaltet.
Selection-Menu	Bildschirmmaske zur Auswahl einer Option. (auch Select-Menue genannt)
Shared-Pool	Bereich im virtuellen Speicher der zum Datenaustausch zwischen den einzelnen Dialogfunktionen (Prozeduren, Programme) dient.
Skeleton	Geruest aus Job-Control-Anweisungen, welches durch Benutzereingaben ausgefuellt und anschliessend abgeschickt wird.
Table	Zweidimensionales Datenfeld zur Speicherung und Verarbeitung von Listen.
Tutorial	Hilfstext von SPF in strukturierter Form.

ANHANG B. LITERATURVERZEICHNIS

- |1| IBM Program Product
System Productivity Facility: Dialog Management Services.
Order Nr. SC34-2036-0
- |2| "Freigabe von SPF"
Mitteilung des Rechenzentrums OP 09 82
- |3| W.Schuster, O.Wallrapp
Datenverarbeitungskonzept fuer das Programm MEDUSA zur Simulation me-
chanischer Systeme.
Interner Bericht 515-82/8
- |4| Programmsystem FADYNA: Benutzerhandbuch Version Maerz/81
DFVLR-A-552-78/3

ANHANG C. KURZE BESCHREIBUNG DER OPTIONEN VON ASTHMA

In diesem Anhang werden die einzelnen Optionen der verschiedenen Anwendungen in kurzer, stichpunktartiger Form aufgezählt. Er enthält keine ausführliche Beschreibung der Optionen, d.h. es soll hier nur der Eindruck der Möglichkeiten, die ASTHMA bietet, in uebersichtlicher Form dargestellt werden.

Die Nummern in den folgenden Auflistungen entsprechen der anzuwaehlenden Option im Panel und der daran anschliessende Name den Kurztiteln der Option.

C.1 Anwendung MEDUSA - ASTHMA-Option 1

- 1 - IAINIT Anlegen und Initialisieren einer neuen Interface- und Ausgabedatei fuer ein neues Simulationsmodell.
- 2 - IACOPY Kopieren einer Interface- und Ausgabedatei in eine andere bereits vorhandene.
- 3 - IAPUT Ausgabe einer beliebigen Matrix auf eine Interface- und Ausgabedatei nach Eingabe am Bildschirm.
- 4 - FORTEXT Bearbeitung von Dialog- oder Druckertexten in der Text- und Formatdatei von MEDUSA.
- 5 - FORQUEL Bearbeitung von MEDUSA-Programmquellen fuer eine Auslieferung.
- 6 - TUBVOR Vorlaufprogramme der TU-Berlin zur Erstellung und Aenderung von Rad/Schiene-Profildatensaetzen.
- 7 - SAURIER Start des Programmes SAURIER zur Unterstuetzung von Dokumentation und Wartung grosser Programme.
-- noch nicht implementiert --
- 8 - MEDUSA/D Programmstart mit WATFIV-Compiler mit beliebiger Quelle fuer das Entwicklerteam.
- 9 - MEDUSA/L Programmstart mit voruebersetztem Lademodul fuer Vorfuehrungen und Produktionslaeufe.
- 10- MEDUSA/A Programmstart mit einer Single- oder Double-Precision Auslieferungsversion zum abschliessendem Test.
- 11- ALLOC Allocierung aller MEDUSA-Standarddateien zum Test externer Programme, welche die Dateiumgebung benoetigen.
- 12- FREE Freigeben aller MEDUSA-Standarddateien.

C.2 Anwendung DSNSICH - ASTHMA-Option 3

- 1 - BACKMKD Sicherung aller wichtigen projektbezogenen Dateien auf das Massenspeichersystem.
- 2 - BANDKAT Aufruf des Bandkatalogsystems, d.h. sichern und verlagern von einzelnen Dateien auf und von Band unter Führung eines Kataloges.

ANHANG D. STRUKTUR DER ANWENDUNGEN

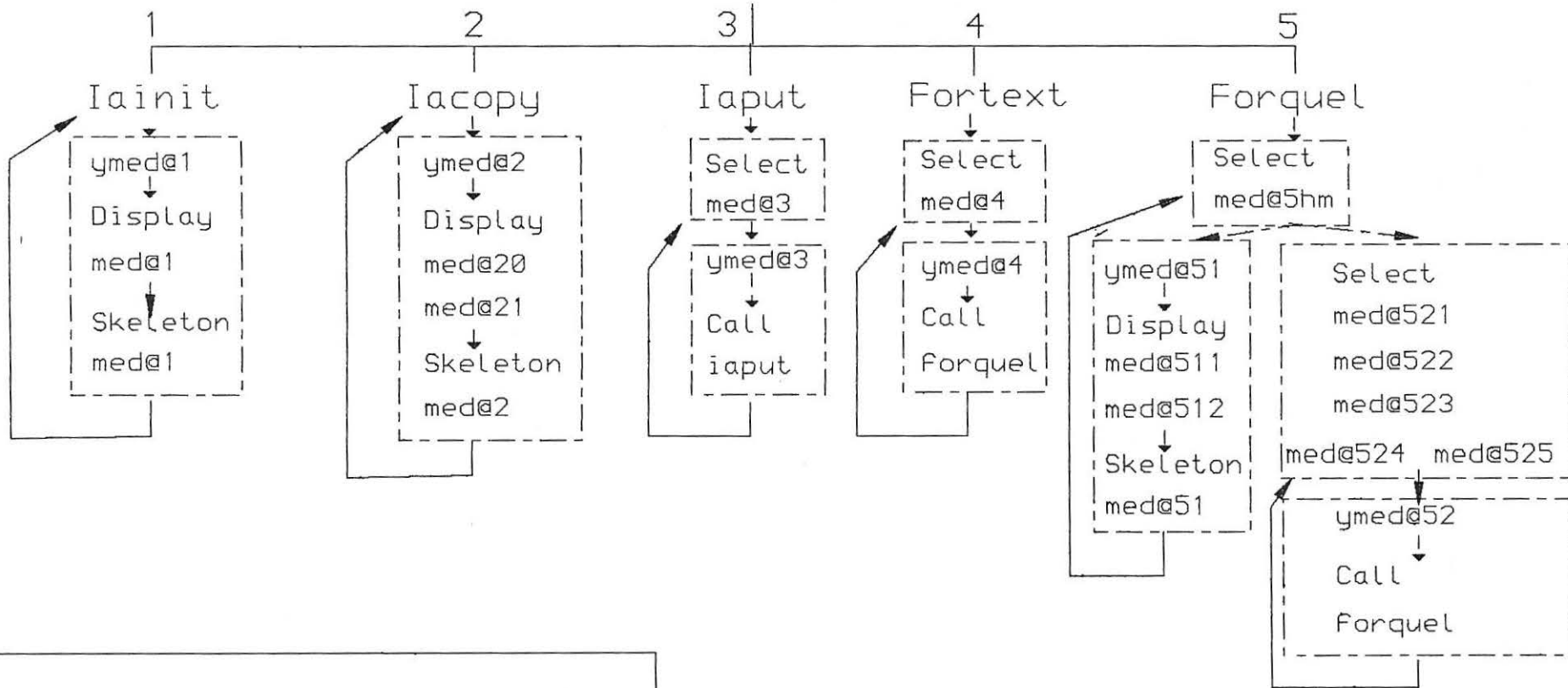
In Form eines Schemas soll in diesem Anhang der Zusammenhang (gestrichelte Kaesten) und der Ablauf (Pfeile) ueber die einzelnen Optionen dargestellt werden. Es wird ~~wird~~ gezeigt, was (CLIST,PANEL,PROGRAMM) bei druecken der Enter-Taste aufgerufen wird und an welchen Punkt nach korrekter Beendigung der Option zurueckgesprungen wird.

Ein Verstaendnis dieser Schemas ^{zu} setzt vorraus, dass der Leser die Namensgebung in ASTHMA gelesen und verstanden hat (siehe hierzu 4.2 auf S.10). Die Namen unter den Optionsnummern im Strukturbaum geben jeweils den Kurztitel der Option an.

MEDUSA

med@hm

Anwendung 1



Gibt einen zusammengehörigen Komplex an.

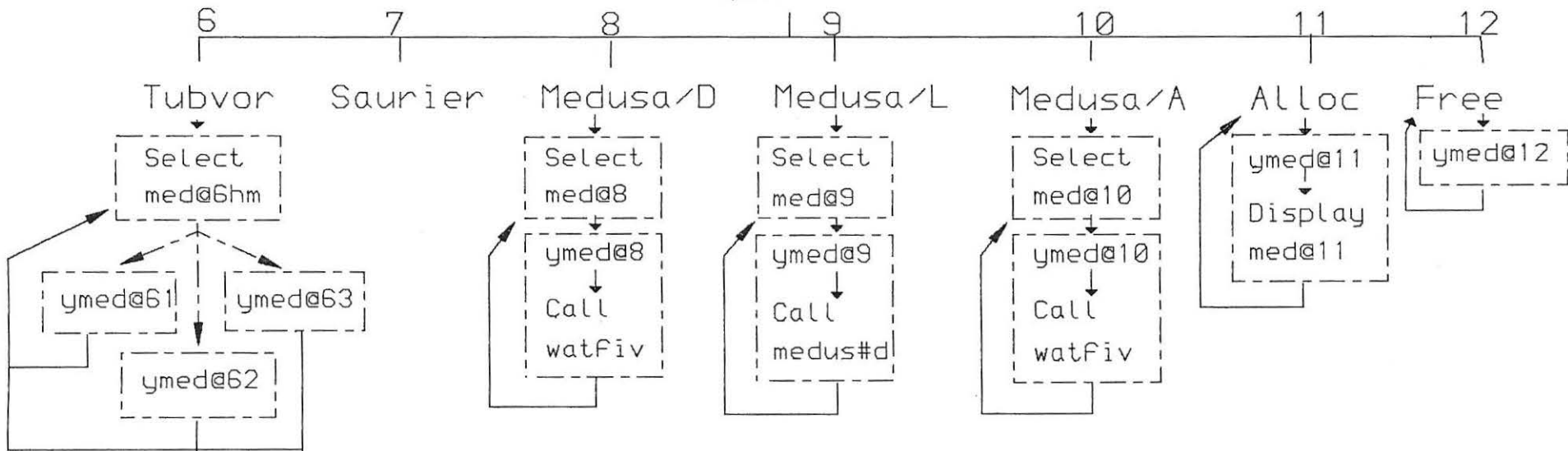


Verdeutlicht den Ablauf der Option und zeigt den Ruecksprung

MEDUSA

med@hm

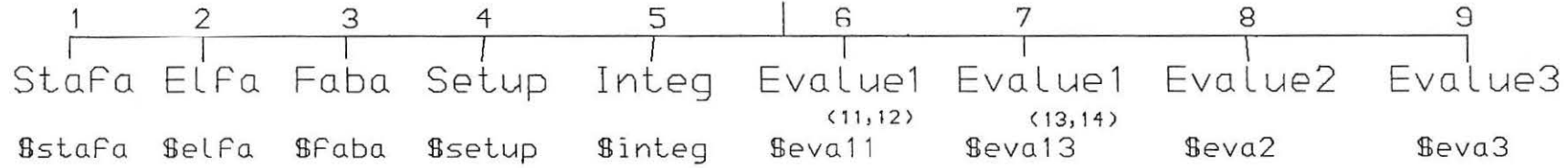
Anwendung 1



FADYNA

Anwendung 2

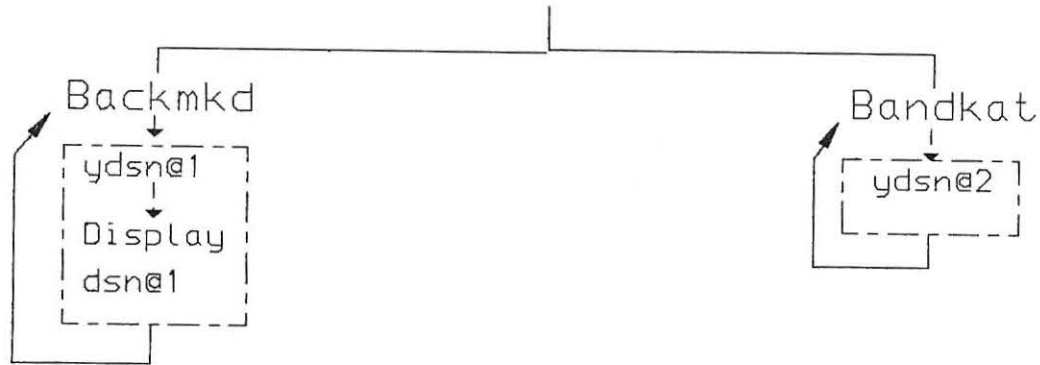
yFad@hm



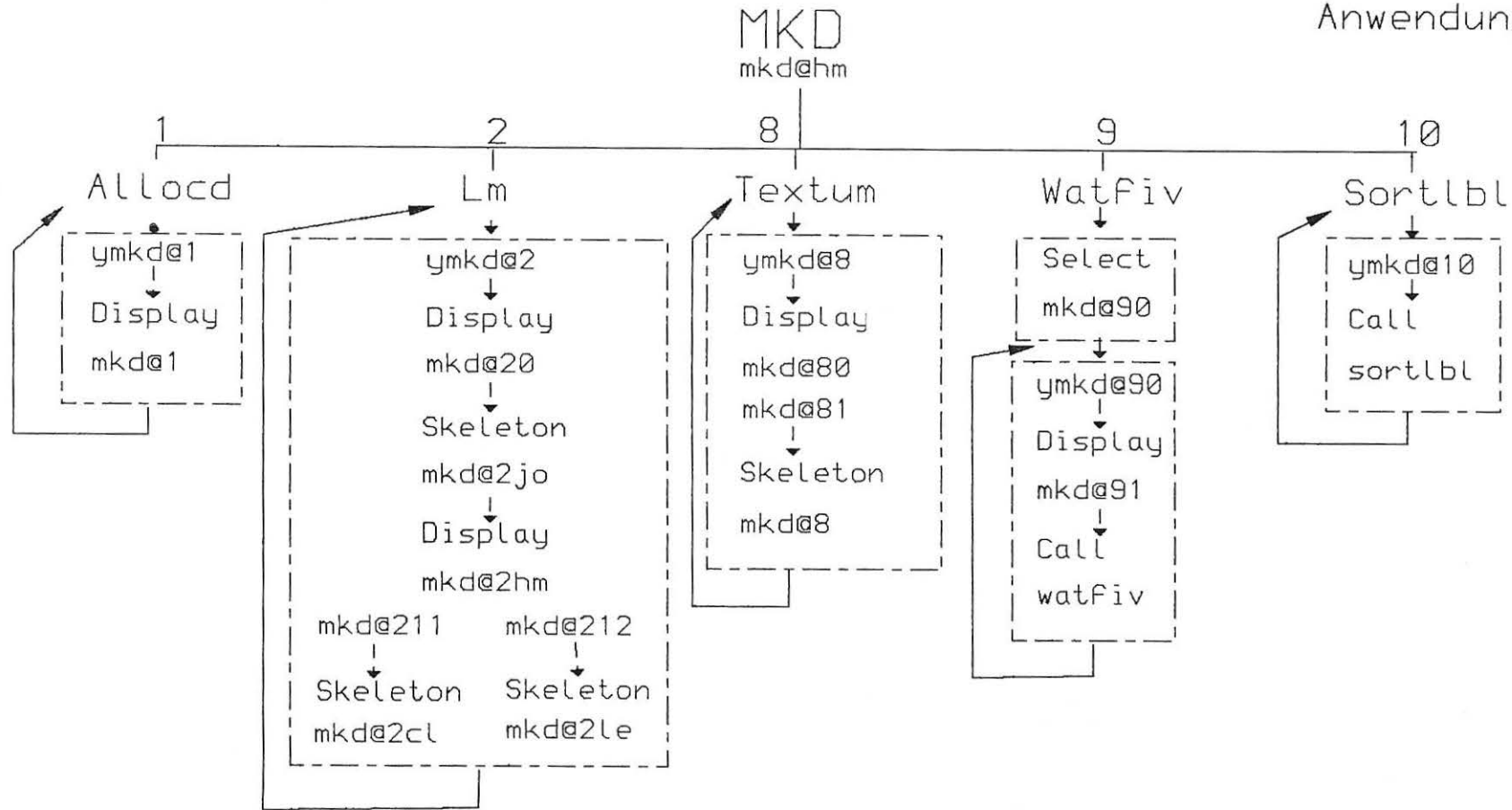
DSNSICH

Anwendung 3

dsn@hm



Anwendung 4



ANHANG E. DER DOKUMENTATIONSTEIL PANELDOC

Dieser Anhang enthaelt den Teil der internen Dokumentation von ASTHMA, der fuer jedes PANEL die verwendeten Dialogvariablen nebst deren Bedeutung auflistet. Desweiteren ist zu ersehen welche der Variablen sich im Profile-Pool befinden und somit ueber eine Session hinaus verfuegbar sind.

Wegen der Zweckmaessigkeit bezueglich einer Aenderung in ASTHMA befindet sich dieser Anhang als Dokumentations-Member im PDS der CLIST-Proceduren.

VERWENDETE DIALOGVARIABLE IN DEN ANWENDUNGSPANELS :

KENNZEICHNUNG DURCH * :

VARIABLE IST IM PROFILE-POOL

PANEL MED@1 - IAINIT

* USIA USERID DER I/A-DATEI F.D. START
* MOD MODELLNUMMER DER I/A-DATEI F.D. START
IAANZS .. GESAMTZAHL DER SAETZE DER I/A-DATEI
IAANZV .. ZAHL DER SAETZE DES VERZEICHNISSES DER I/A-DATEI
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@20 - IACOPY/QUELLE

* USIA USERID DER QUELLEN-I/A-DATEI
* MOD MODELLNUMMER DER QUELLEN-I/A-DATEI
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@21 - IACOPY/ZIEL

UIAZ USERID DER ZIEL-I/A-DATEI
MODZ MODELLNUMMER DER ZIEL-I/A-DATEI
IAKO STEUERGROESSE ZUR KOMPRIERUNG DER ZIEL-I/A-DATEI
IAVZ STEUERGROESSE ZUR AENDERUNG DES VERZEICHNISSES

PANEL MED@3 - IAPUT

* USIA USERID DER I/A-DATEI
* MOD MODELLNUMMER DER I/A-DATEI
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@4 - FORTEXT

MODUS ... BEHANDLUNG DER EINGABEFELDER VON FORTEXT
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@511/MED@521 - FORQUEL/STEUERGROESSEN

KOMM GRAD DER KOMPRIMIERUNG
KOM " " " (UMGESETZT)
GENA GENAUIGKEIT DER ZIELDATEI

```
GEN ..... " " " (UMGESETZT)
STRG .... STRINGS IM A10-FORMAT
STG ..... " " " " (UMGESETZT)
TESA .... TESTAUSGABEN IN ZIELQUELLE
TA ..... " " " (UMGESETZT)
PROT .... PROTOKOLL DER UMSETZUNG
PRO ..... " " " (UMGESETZT)
```

PANEL MED@512 - FORQUEL/DATEIVERSORGUNG

```
* LIBO .... ORGANISATIONSBEREICH MIT VERSIONSNUMMER
LIB ..... " OHNE VERSIONSNUMMER
* TYPO .... DATEITYP
DISP .... FORTSCHREIBUNG DER STATISTIKDATEI
PSWD .... PASSWORT FUER DEN GENERIERTEN JOB
```

PANEL MED@522 - FORQUEL/QUELLE

```
* PROJECT = PROJECT,USERID (S. SPF-EDIT)
* LIBRARY = PROGRAMM, QUELLE (S. SPF-EDIT)
* TYPE = DATEITYP (S. SPF-EDIT)
MEMQ1 = MEMBER 1
MEMQ2 = MEMBER 2
MEMQ3 = MEMBER 3
MEMQ4 = MEMBER 4
```

PANEL MED@523 - FORQUEL/ZIEL

```
* PRJZ .... PROJEKT,USERID
* LIBZ .... PROGRAMM, QUELLE
* TYPEZ ... DATEITYP
MEMZ .... MEMBER
NR ..... VERZWEIGUNGSNUMMER FUER DAS NAECHSTE PANEL
```

PANEL MED@524 - FORQUEL/INFO (MIT PROTOKOLL)

```
PRJST ... PROJEKT,USERID DER STATISTIKDATEI
LIBST ... PROGRAMM, QUELLE " "
TYPEST .. DATEITYP " "
TERM .... AUSGABE AUF TERMINAL
PRJLF ... PROJEKT,USERID DES LOGFILES
LIBLF ... PROGRAMM, QUELLE " "
TYPZELF . DATEITYP " "
```

PANEL MED@525 - FORQUEL/INFO (OHNE PROTOKOLL)

```
PRJST ... PROJEKT,USERID DER STATISTIKDATEI
LIBST ... PROGRAMM, QUELLE " "
```

TYPEST .. DATEITYP " "
TERM AUSGABE AUF TERMINAL

PANEL MED@6HM - TUBVOR

* RECHGEN . RECHGENGENAUIGKEIT

PANEL MED@8 - MEDUSA/D

BLKD MEMBERNAME VON BLOCKDATA
* HPQ VERTEILER
* HR1/2 ... ROUTINEN 1,2
* UR1/2 ... ROUTINEN 3,4
LIST ROUTINENLISTING
* USIA USERID DER I/A-DATEI
* MOD MODELLNUMMER DER I/A-DATEI
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@9 - MEDUSA/L

* USIA USERID DER I/A-DATEI
* MOD MODELLNUMMER DER I/A-DATEI
* VER PROGRAMMVERSION VON MEDUSA

PANEL MED@10 - MEDUSA/A

* USIA USERID DER I/A-DATEI
* MOD MODELLNUMMER DER I/A-DATEI
* VER VERSIONSNUMMER DES PROGRAMMES
* RECHGEN . RECHGENGENAUIGKEIT DES PROGRAMMS

PANEL MED@11 - ALLOC

* VER PROGRAMMVERSION VON MEDUSA

PANEL DSN@1 - BACKMKD

* VER PROGRAMMVERSION VON MEDUSA

PANEL MKD@1 - ALLOCD

FINR LOGISCHE FILENUMMER FUER FORTRAN

PANEL MKD@20 - LM/JOBPARAMETER

RTIME ... RECHENZEIT
LINES ... ZEILENZAHL IN TAUSEND
MSGL MESSAGELEVEL
MCLAS ... MESSAGECLASS
TEXT JOBTXT
ACNR ACCOUNT-NUMMER
PSWD PASSWORT
JCL1 ZUSAETZLICHE JOBKONTROL-ZEILEN (JCL2,JCL3,JCL4)

PANEL MKD@211- COMPILE AND LINK

* PROJECT . PROJECT,USERID (S. SPF-EDIT)
* LIBRARY . PROGRAMM, QUELLE (S. SPF-EDIT)
* TYPE DATEITYP (S. SPF-EDIT)
MEMQ MEMBER
PRJB PROJECT,USERID FUER DAS LADEMODUL
LIBB PROGRAMM, QUELLE
TYPEB ... DATEITYP
MEMB MEMBER
OPTCO ... OPTIONEN FUER DEN COMPILER

PANEL MKD@212- LM/LINKAGE EDITOR

* PRJHP ... PROJECT,USERID DES LADEMODUL-HAUPTPROGRAMMS
* LIBHP ... PROGRAMM, QUELLE
* TYPEHP .. DATEITYP
MEMHP ... MEMBER
PRJB PROJECT,USERID FUER DAS LADEMODUL
LIBB PROGRAMM, QUELLE
TYPEB ... DATEITYP
MEMBB ... MEMBER
UBIB1 ... UNTERPROGRAMMBIBLIOTHEK1
UBIBL1 .. " (UMGESETZT)
UBIB2 ... UNTERPROGRAMMBIBLIOTHEK2
UBIBL2 .. " (UMGESETZT)
UBIB3 ... UNTERPROGRAMMBIBLIOTHEK3
UBIBL3 .. " (UMGESETZT)
GBIB GRAPH.BIBLIOTHEK
MBIB MEDUSA STANDART-BIBLIOTHEK
VER VERSIONSNUMMER
OPTLE ... OPTIONEN FUER DEN LINKAGE EDITOR

PANEL MKD@80 - TEXTUM/QUELLE

* PROJECT . PROJECT,USERID (S. SPF-EDIT)
* LIBRARY . PROGRAMM, QUELLE (S. SPF-EDIT)
* TYPE DATEITYP (S. SPF-EDIT)
MEM TEXTMEMBER

PANEL MKD@81 - TEXTUM/DRUCKEROPTIONEN

ANZ ANZAHL DER KOPIEN
GK GROSS/KLEINSCHREIBUNG
PAP DRUCKPAPIER

PANEL MKD@90 - WATFIV/QUELLE

* PROJECT . PROJECT,USERID (S. SPF-EDIT)
* LIBRARY . PROGRAMM, QUELLE (S. SPF-EDIT)
* TYPE DATEITYP (S. SPF-EDIT)
MEM QUELLENMEMBER
OPTWAT .. OPTIONEN FUER DEN WATFIV-COMPILER
LDN6 STEUERGROESSE FUER AUSGABEMEDIUM DER FILENUMMER 6

PANEL MKD@91 - WATFIV/BIBLIOTHEKEN

* PRJBIB .. PROJECT,USERID
* LIBBIB .. PROGRAMM, QUELLE
* TYPBIB .. DATEITYP
IBIB STEUERGROESSE FUER IMSL-BIBLIOTHEK
* RECHGEN . RECHENGENAUIGKEIT
GBIB STEUERGROESSE FUER GSI-BIBLIOTHEK
MBIB STEUERGROESSE FUER MEDUSA-BIBLIOTHEKEN
* VER PROGRAMMVERSION VON MEDUSA

ANHANG F. DIE INITIALISIERUNGS- UND STARTPROZEDUR VON ASTHMA

```
/*
/*
/* Z W E C K : VORLAUF-PROCEDURE ZUM BEREITSTELLEN ALLER DIALOG-VAR=
/* ===== IABLEN FUER DIE ABTEILUNGS-PANELS UND AUFRUF VOM
/*           HAUPTMANUAL VON ASTHMA.
/*
/*           PROCEDURE FUER DAS PANEL ASTHMA
/*
/* #####
/*
/*
/* VORBELEGUNG VON STANDARDWERTEN :
/*
/* A) PROGRAMMVERSION VON MEDUSA
/*   ES KOENNEN MAXIMAL 2 VERSCHIEDENE PROGRAMMVERSIONEN VON MEDUSA
/*   BEDIENT WERDEN VERSION A (VERA) UND VERSION B (VERB).
/*   DIE VERSION VERO IST FUER EINE EVENTUELLE "ALTE" TESTVERSION, DIE
/*   VON BAND GEHOLT WIRD FREIGEHALTEN.
SET &VERA  EQ &STR(22)
SET &VERAP EQ &STR(2.2)
SET &VERB  EQ &STR(21)
SET &VERBP EQ &STR(2.1)
SET &VERO  EQ &STR(20)
SET &VEROP EQ &STR(2.0)
/*
/* B) VOREINSTELLUNG DER ANZAHL DER SAETZE BEI NEU-ALLOCIERUNG EINER
/*   MEDUSA-IA-DATEI
SET &IAANZSV EQ &STR(1500)
SET &IAANZVV EQ &STR(60)
/*
/* C) CLIST-EINHEITLICHE NAMEN
SET &CLIST EQ &STR(DF19.ISPF.CLIST)
SET &SKEL EQ &STR(DF19.ISPF.SKELLIB)
/*
/* D) DRUCKERVOREINSTELLUNG
ISPEXEC VGET (DRUCKER) PROFILE
ISPEXEC VPUT (VERA,VERAP,VERB,VERBP,VERO,VEROP,IAANZVV,IAANZSV,+
              CLIST,SKEL,DRUCKER +
              ) SHARED
/*
/*
/* EINLESEN ALLER DIALOG-VARIABLEN AUS DEM  P R O F I L E - P O O L
/*
ISPEXEC VGET (HPQ,HR1,HR2,UP1,UP2,VER,USIA,MOD,RECHGEN) PROFILE
ISPEXEC VGET (PRJZ,LIBZ,TYPEZ) PROFILE
ISPEXEC VGET ( LIBO ,TYPO ) PROFILE
ISPEXEC VGET (PRJBIB ,LIBBIB ,TYPBIB,OPTWAT) PROFILE
ISPEXEC VGET (PROJECT ,LIBRARY ,TYPE) PROFILE
ISPEXEC VGET (PRJHP ,LIBHP ,TYPEHP) PROFILE
/*
/*
/* AUSGABE IN DEN  S H A R E D - P O O L
/*
ISPEXEC VPUT (HPQ,HR1,HR2,UP1,UP2,VER,USIA,MOD,RECHGEN) SHARED
```



```
ISPEXEC VPUT (PRJZ,LIBZ,TYPEZ) SHARED
ISPEXEC VPUT ( LIBO ,TYPO ) SHARED
ISPEXEC VPUT (PRJBIB ,LIBBIB ,TYPBIB,OPTWAT) SHARED
ISPEXEC VPUT (PROJECT ,LIBRARY ,TYPE) SHARED
ISPEXEC VPUT (PRJHP ,LIBHP ,TYPEHP) SHARED
/*
/*
/* AUFBRUF VON A S T H M A
/*
ISPEXEC SELECT PANEL(ASTHMA)
/*
/*
/* EINLESEN AUS DEM S H A R E D - P O O L
/*
ISPEXEC VGET (HPQ,HR1,HR2,UP1,UP2,VER,USIA,MOD,RECHGEN) SHARED
ISPEXEC VGET (PRJZ,LIBZ,TYPEZ) SHARED
ISPEXEC VGET ( LIBO ,TYPO ) SHARED
ISPEXEC VGET (PRJBIB ,LIBBIB ,TYPBIB,OPTWAT) SHARED
ISPEXEC VGET (PROJECT ,LIBRARY ,TYPE) SHARED
ISPEXEC VGET (PRJHP ,LIBHP ,TYPEHP) SHARED
ISPEXEC VGET (DRUCKER) SHARED
/*
/*
/* AUSGABE IN DEN P R O F I L E - P O O L
/*
ISPEXEC VPUT (HPQ,HR1,HR2,UP1,UP2,VER,USIA,MOD,RECHGEN) PROFILE
ISPEXEC VPUT (PRJZ,LIBZ,TYPEZ) PROFILE
ISPEXEC VPUT ( LIBO ,TYPO ) PROFILE
ISPEXEC VPUT (PRJBIB ,LIBBIB ,TYPBIB,OPTWAT) PROFILE
ISPEXEC VPUT (PROJECT ,LIBRARY ,TYPE) PROFILE
ISPEXEC VPUT (PRJHP ,LIBHP ,TYPEHP) PROFILE
ISPEXEC VPUT (DRUCKER) PROFILE
```

ANHANG G. ABBILDUNG DER HAUPTMANUALE VON ASTHMA

```
%----- DSNSICH HAUPTMANUAL -----
%GEWAEHLTE OPTION ==>_OPT      +                +USERID :%&ZUSER
%                                +PREFIX :%&ZPREFIX
%
%  1 +BACKMKD      - BACKUP-LAUF      ZUR SICHERUNG AUF MSS
%  2 +BANDKAT      - BANDKATALOGSYSTEM, SICHERUNG AUF MAGNETBAND
%
%  B +BESCHREIBUNG
%
+DRUECKEN SIE DIE%END-TASTE+ZUM ABRUCH VON DSNSICH
%
%
)PROC
  &SEL = TRANS( TRUNC (&OPT, '.' )
                1, 'CMD(YDSN@1)'
                2, 'CMD(YDSN@2)'
                , ' '
                *, '?' )
)END
```

```
%----- SPF-MVS PRIMARY OPTION MENU -----
%SELECT OPTION ==>_OPT      +                +USERID :%&ZUSER
%                               +PREFIX :%&ZPREFIX
%  1 +BROWSE      - DISPLAY SOURCE DATA OR OUTPUT LISTINGS
%  2 +EDIT        - CREATE OR CHANGE SOURCE DATA
%  3 +UTILITIES   - PERFORM SPF UTILITY FUNCTIONS
%  4 +FOREGROUND  - COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG
%  5 +BACKGROUND  - COMPILE AND LINK, ASSEMBLE
%  6 +COMMAND     - ENTER TSO COMMAND OR CLIST
%  7 +SUPPORT     - TEST DIALOG OR CONVERT MENU/MESSAGE FORMATS
%  8 +DFVLR       - PERFORM DFVLR UTILITY FUNCTIONS
%  T +TUTORIAL    - DISPLAY INFORMATION ABOUT SPF
%
+  PRESS%END KEY+TO TERMINATE SPF+
%
%
)INIT
  .HELP = TTUTOR
  &ZHTOP = TTUTOR      /* TUTORIAL TABLE OF CONTENTS */
  &ZHINDEX = TINDEX   /* TUTORIAL INDEX - 1ST PAGE */
)PROC
  &SEL = TRANS( TRUNC (&OPT, '.' )
                1, 'PGM(ISPBRO)'
                2, 'PGM(ISPEDIT)'
                3, 'PANEL(ISPUTIL)'
                4, 'PANEL(ISPFORA)'
                5, 'PANEL(ISPJOB)'
                6, 'PGM(ISPTSO)'
                7, 'PANEL(ISPQTAC) NEWPOOL'
                8, 'PANEL($DFVLR)'
                T, 'PGM(ISPTUTOR) PARM(T)'
                , , ,
                *, '?' )
)END
```

```
%----- MEDUSA HAUPTMANUAL -----
%GEWAEHLTE OPTION ==>_OPT      +                +USERID :%&ZUSER
%                                +                +PREFIX  :%&ZPREFIX
%
%  1 +IAINIT          -  INITIALISIERUNG EINER INTERFACE/AUSGABEDATEI
%  2 +IACOPY          -  KOPIEREN          EINER INTERFACE/AUSGABEDATEI
%  3 +IAPUT           -  AUSGABE EINER MATRIX AUF EINE INTERFACE/AUSGABEDATEI
%  4 +FORTEXT         -  FORTRAN-TEXTBEARBEITUNGSPROGRAMM
%  5 +FORQUEL         -  FORTRAN-QUELLENBEARBEITUNGSPROGRAMM
%  6 +TUBVOR          -  VORLAUFPROGRAMME DER TU-BERLIN
%** 7 +SAURIER        -  PROGRAMMSTRUKTUR AUFBAUEN UND DARSTELLEN
%  8 +MEDUSA/D        -  MEDUSA-START MIT WATFIV-COMPILER (DP)
%  9 +MEDUSA/L        -  MEDUSA-START MIT LADEMODUL      (DP)
% 10 +MEDUSA/A        -  MEDUSA-TEST FUER AUSLIEFERUNG  (DP + SP)
% 11 +ALLOC           -  ALLOCIEREN DER MEDUSA-STANDARD-DATEIEN
% 12 +FREE            -  FREIGEBEN  DER MEDUSA-STANDARD-DATEIEN
%
%  B +BESCHREIBUNG
%
+DRUECKEN SIE DIE%END-TASTE+ZUM ABRUCH VON MEDUSA
%
%**+FUNKTION IST NOCH NICHT IMPLEMENTIERT
)PROC
  &SEL = TRANS( TRUNC (&OPT, '.' )
    1, 'CMD(YMED@1)'
    2, 'CMD(YMED@2)'
    3, 'PANEL(MED@3)'
    4, 'PANEL(MED@4)'
    5, 'PANEL(MED@5HM)'
    6, 'PANEL(MED@6HM)'
    8, 'PANEL(MED@8)'
    9, 'PANEL(MED@9)'
    10, 'PANEL(MED@10)'
    11, 'CMD(YMED@11)'
    12, 'CMD(YMED@12)'
    19, 'CMD(YMED@XX)'
    ' '
    *, '?' )
)END
```

```
%----- MKD HAUPTMANUAL -----
%GEWAEHLTE OPTION ==>_OPT + +USERID :%&ZUSER
% +PREFIX :%&ZPREFIX
%
% 1 +ALLOCD - ALLOCIEREN DES DRUCKERS
% 2 +LM - ERZEUGEN VON LAEMODULEN (PL/1 UND FORTRAN)
%*** 3 +MEMSICH - PDS-MEMBER ALS PS-DATEIEN AUF BAND
%*** 4 +PFORT - PORTABLE FORTRAN VERIFIER
%*** 5 +PICTPAR-P - AENDERN DER PICTPAR-PARAMETERDATEI
%*** 6 +SEQPDS - SEQ-FORTRAN-QUELLE IN PDS-FORTRAN-QUELLE UMWANDELN
%*** 7 +PDSSEQ - PDS-FORTRAN-QUELLE IN SEQ-FORTRAN-QUELLE UMWANDELN
% 8 +TEXTUM - AUSGABE KOMPRIMIERTER TEXTE
% 9 +WATFIV - AUFRUF DES WATFIV-COMPILERS
% 10 +SORTLBL - LABEL EINER FORTRAN-QUELLE SORTIEREN
%
% B +BESCHREIBUNG
%
+DRUECKEN SIE DIE%END-TASTE+ZUM ABRUCH VON MKD
%
%***+FUNKTION IST NOCH NICHT IMPLEMENTIERT
)PROC
&USID = &ZUSER
&PREFIX = &ZPREFIX
&SEL = TRANS( TRUNC (&OPT, '.' )
              1, 'CMD(YMKD@1)'
              2, 'PGM(YMKD@2)'
              8, 'CMD(YMKD@8)'
              9, 'PANEL(MKD@90)'
              10, 'CMD(YMKD@10)'
              ' '
              *, '?' )
)END
```

```
%----- ASTHMA PARAMETER OPTIONS -----  
%SELECT OPTION ==>_OPT      +  
%  
%  1 +TERMINAL  - SPECIFY TERMINAL CHARACTERISTICS  
%  2 +LOG/LIST  - SPECIFY SPF LOG AND LIST DEFAULTS  
%  3 +PF KEYS   - SPECIFY PF KEYS FOR &ZTERM TERMINAL WITH &ZKEYS PF KEYS  
%  4 +DRUCKER   - VOREINSTELLUNG DES DRUCKERS  
)INIT  
  .HELP = TOPT  
)PROC  
  &SEL = TRANS( &OPT,  
               1, 'PGM(ISPOPT) PARM(OPT01 )'  
               2, 'PGM(ISPOPT) PARM(OPT02 )'  
               3, 'PGM(ISPOPT) PARM(OPT03 )'  
               4, 'PANEL(PAR@4)'  
               * , '?' )  
)END
```

ANHANG H. BEISPIELE FUER JEDE ART DES PROGRAMMABLAUFES

H.1 Panel - CLIST - Programm

In diesem Beispiel wird vom Hauptmanual der Anwendung MEDUSA das **PANEL MED@3** mit Hilfe des SELECT-Service aufgerufen. Dieses ruft nun nach druecken von Enter seinerseits wieder mit Hilfe des SELECT-Service (&SEL = 'CMD(YMED@3)') die **CLIST-Prozedur** YMED@3 auf, in der das **Programm** IAPUT mit dem TSO-CALL-Statement aufgerufen wird.

Es folgen die entsprechenden Teile von ASTHMA.

-- M K S . I S P F . P A N E L I B (M E D @ 3) --

```
%MEDUSA ----- IAPUT -----
%EINGEBEN/VERAENDERN DER UNTENSTEHENDEN PARAMETER :
+
+
%MODELLDATENSPEICHER :
+  USERID DER INTERFACE- UND AUSGABEDATEI%=>_USIA+(BLANK=AKTUELLE USERID)
+  MODELLNUMMER                               %=>_MOD +
+
%VERSION+DES PROGRAMMES                       %=>_VER  +(Z.B. &VERA)
+
)PROC
  VER (&USIA,NAME                               MSG=MED@005)
  IF(&USIA = '      ')
    &USIA = &ZUSER
  VER (&MOD ,NONBLANK                           MSG=MED@007)
  VER (&MOD ,RANGE,0000,9999                   MSG=MED@003)
  VER (&MOD,PICT NNNN                          MSG=MED@021)
  VER (&MOD ,PICT ,NNNN                        MSG=MED@003)

  &VER = TRANS(&VER &VERA,&VERA &VERAP,&VERA &VERB,&VERB &VERBP,&VERB
              &VERO,&VERO &VEROP,&VERO
              MSG=MED@001)
&SEL = 'CMD(YMED@3)'
)END
```



```
-- M K S . C L I S T ( Y M E D @ 3 ) --

/*
/*
/* Z W E C K : PROCEDUR FUER DES ASTHMA - PANEL 'M E D @ 3 '
/*
/* =====
/*           AUSGABE EINER MATRIX AUF EINE INTERFACE/AUSGABEDATEI.
/*           OPTION 3 IN DER ANWENDUNG MEDUSA.
/*
/* VARIABLE AUS DEM PANEL :
/* =====
/*
/* USIA.... USERID DER IA-DATEI
/* MOD..... MODELLNUMMER DER IA-DATEI
/* VER..... VERSIONSNUMMER
/*
/* VARIABLE AUS YASTHMA :
/* =====
/* TEST.... TESTVARIABLE
/*
/* #####
/*
ISPEXEC CONTROL DISPLAY LINE
ISPEXEC VGET (VER,USIA,MOD)
/*
/* EINSCHALTEN VON TESTAUSGABEN
ISPEXEC VGET (TEST)
IF &TEST EQ J THEN +
  CONTROL CONLIST SYMLIST
/*
/*
/* A L L O C I E R U N G   D E R   D A T E I E N
/*
FREE ATTRLIST(INP)
ATTRIB INP INPUT
/*
/* I/A-DATEI
ALLOC DA('&USIA..IA&MOD') FI(FT99F001) REUSE
/*
/* MEDUSA-TEXT-DATEI FUER FEHLERMELDUNGEN
ALLOC DA('ZFF.MEDUSA&VER..TEXTFOR') FI(FT98F001) REUSE SHR USING(INP)
/*
/* P R O G R A M M S T A R T
CALL 'ZFF.VORLAUF.LOAD(IAPUT)'
FREE FI(FT99F001)
FREE FI(FT98F001)
FREE ATTRLIST(INP)
```

H.2 CLIST - Panel - Skeleton

In diesem Beispiel wird vom Hauptmanual der Anwendung MEDUSA die **CLIST-Prozedur** YMED@1 aufgerufen, welche mit Hilfe des DISPLAY-Service (ISPEXEC DISPLAY PANEL(MED@1)) das **Panel** MED@1 aufruft, anschliessend die JCL im **Skeleton-File** MED@1 ausfuellt und abschickt (submit).
Es folgen die entsprechenden Teile von ASTHMA.

-- M K S . C L I S T (Y M E D @ 1) --

```
/*
/*
/* Z W E C K : PROCEDURE FUER DAS ASTHMA - PANEL 'M E D @ 1 '
/* =====
/*             INITIALISIERUNG EINER INTERFACEDATEI.
/*             OPTION 1 IN DER ANWENDUNG MEDUSA.
/*
/* VARIABLE AUS DEM PANEL :
/* =====
/*
/* USIA.... USERID DER IA-DATEI
/* MOD..... MODELLNUMMER DER IA-DATEI
/* IAANZS.. ZAHL DER DATEISAETZE
/* IAANZV.. GROESSE DER VERZEICHNISSES
/* VER..... VERSIONSNUMMER
/*
/* VARIABLE AUS YASTHMA :
/* =====
/*
/* TEST.... TESTVARIABLE
/*
/* HINWEIS : DER JOBNAME IST &SYSUID.IAIN
/*
/* #####
/*
ISPEXEC DISPLAY PANEL(MED@1)
/* WENN &LASTCC GLEICH 8, DANN PF3 IM PANEL GEDRUECKT
IF &LASTCC EQ 8 THEN EXIT
/*
/* EINSCHALTEN DER TESTAUSGABE ==> MSGCLASS H (SONST T)
ISPEXEC VGET (TEST)
IF &TEST EQ J THEN DO
    SET &MCLASS EQ H
    CONTROL CONLIST SYMLIST
END
ELSE SET &MCLASS EQ T
/*
/*
/* GESAMTZAHL DER SAETZE GROESSER ALS ZAHL DER VERZEICHNISSAETZE
M: IF &IAANZS GT &IAANZV + 1 THEN +
DO
    /*
    /* OPTIMALE NUTZUNG DER SPUREN (65 SAETZE PRO SPUR)
    SET &DIV = &IAANZS /65
    SET &IAANZS = &DIV * 65 +65
    /*
    /* S U B M I T
    /*
    ISPEXEC VPUT (USIA,MOD,IAANZS,IAANZV,VER)
    ISPEXEC FTOPEN TEMP
    ISPEXEC FTINCL MED@1
    ISPEXEC FTCLOSE
    ISPEXEC VGET ZTEMPF SHARED
    SUB '&ZTEMPF' -
```

```
END
/*
/* FEHLERMELDUNG: VERZEICHNISSAETZE MEHR ALS GESAMTZAHL DER SAETZE
ELSE +
DO
  ISPEXEC DISPLAY MSG(MED@022) CURSOR(IAANZV)
  IF &LASTCC EQ 8 THEN EXIT
  GOTO M
END
```

-- M K S . I S P F . P A N E L I B (M E D @ 1) --

```
%MEDUSA ----- IAINIT -----
%EINGEBEN/VERAENDERN DER UNTENSTEHENDEN PARAMETER :
+
+       DRUECKEN SIE DIE%ENTERTASTE +ZUR GENERIERUNG DER JCL UND
+                               +ZUM SUBMIT DES JOBS
+
%NEUER+MODELLDATENSPEICHER :
+ USERID DER INTERFACE- UND AUSGABEDATEI %==>_USIA+(BLANK=AKTUELLE USERID)
+ MODELLNUMMER                               %==>_MOD +
+
%KENNDATEN+DER DATEI :
+ ZAHL DER SAETZE           %==>_IAANZS+ (MAX.  4000 FUER SPF-EDIT 1 MB.
+                               MAX. 18494 FUER USERPLATTEN  )
+ GROESSE DES VERZEICHNISSES%==>_IAANZV+
+
%VERSION+DES PROGRAMMES       %==>_VER +   (Z.B. &VERA)
+
+                               NAME DES JOBS :%&ZPREFIX.IAIN
+
)INIT
    &IAANZS = &IAANZSV
    &IAANZV = &IAANZVV
)PROC
    IF(&USIA = '      ')
        &USIA = &ZUSER
    VER(&USIA,NAME                               MSG=MED@005)

    VER(&MOD,NONBLANK                            MSG=MED@007)
    VER(&MOD,PICT,NNNN                          MSG=MED@021)
    VER(&MOD,RANGE,0000,9999                    MSG=MED@003)

    VER(&IAANZS,NUM                              MSG=MED@011)
    VER(&IAANZS,NONBLANK                        MSG=MED@012)
    VER(&IAANZS,RANGE,0003,999999              MSG=MED@014)

    VER(&IAANZV,NUM                              MSG=MED@011)
    VER(&IAANZV,NONBLANK                        MSG=MED@013)
    VER(&IAANZV,RANGE,0003,999999              MSG=MED@014)

    &VER = TRANS(&VER &VERA,&VERA &VERAP,&VERA &VERB,&VERB &VERBP,&VERB
                &VERO,&VERO &VEROP,&VERO
                MSG=MED@001)
)END
```

-- M K S . I S P F . S K E L L I B (M E D @ 1) --

```
)DEFAULT (&?!<c>
//&ZUSER.IAIN JOB ,NOTIFY=&ZUSER,MSGCLASS=&MCLASS,MSGLEVEL=(0,0)
/*JOBPARM L=3,T=9
//INITDA EXEC FHCG,OPTC='NOSOURCE,OPTIMIZE(0)',OPTL=MAP
//C.SYSIN DD *
```

```
    BLOCKDATA
    DOUBLE PRECISION R8HF
```

```
C
C   A C H T U N G :
C       DER COMMON-BLOCK PDIM UND DIE GROESSEN FUER DIE STATISTIK DER
C       DIMENSIONIERUNG SIND IN DIESEM BLOCK-DATA NICHT NOTWENDIG, DA
C       DIE ROUTINE EACOMP, IN DER DIE BLOECKE BENOETIGT WERDEN, NIE
C       AUFGERUFEN WIRD.
```

```
COMMON /DAHILF/ IHF(108) ,R4HF(54), R8HF(27), IDIRE(54)
COMMON /EAPARA/ IZEISA(2),IGRENV
DATA IZEISA/4H| ,4H====/, IGRENV/ &IAANZV/
* ,IHF/108*0/, R4HF/54*0.OE0/, R8HF/27*0.OD0/
* ,IDIRE/54*4H /
```

```
END
DEFINE FILE 99 (&IAANZS,216,L,IA)
COMMON /DADAT/ IDA,IA,NSAMAX,NBYTES,NDIR,IOPEN,NI2F,NI4F,
* NI4U,NI8F,NR4F,NR8F,NR8U,IERRH
INTEGER IBLANK(54)
DATA IBLANK/54*4H /, IDURCH/0/
WRITE (99'1,9000) IBLANK
NSANEU = &IAANZS
WRITE (99'2,9001) NSANEU
```

```
C BELEGUNG DER DATEIKONSTANTEN
```

```
IOPEN = 1
IDA = 99
NSAMAX = NSANEU - 2
NBYTES = 216
NDIR = 3
NI2F = 108
NI4F = 54
IERRH = 0
CALL EAPINF(8HIDURCH ,IDURCH,1,1,1,1,1,4HI2 ,4HF ,
* 32HDURCHLAUFKENNUNG FUER DAS MODELL,IERR)
```

```
STOP
9000 FORMAT(54A4)
9001 FORMAT(32X,I8)
END
```

```
/*
//G.SYSLIB DD DISP=SHR,DSN=SYS1.FORTLIB
// DD DISP=SHR,DSN=ZFF.DATORG&VER..LOAD
// DD DISP=SHR,DSN=ZFF.ALLROU&VER..LOAD
// DD DISP=SHR,DSN=ZFF.MEDUSA&VER..LOADRA
//G.FT99F001 DD DSN=&USIA..IA&MOD,
// DISP=(NEW,CATLG),UNIT=33XX,
// DCB=(DSORG=PS,RECFM=F,LRECL=216,BLKSIZE=216),
// SPACE=(216,(&IAANZS,0))
//
```

H.3 CLIST

-- M K S . C L I S T (Y M E D @ 1 2) --

```
/*
/*
/*  Z W E C K : PROCEDURE ZUM
/*  =====
/*                               FREIGEBEN DER MEDUSA-STANDARTDATEIEN
/*                               OPTION 12 IN DER ANWENDUNG MEDUSA
/*
/*  VARIABLE AUS YASTHMA :
/*  =====
/*  TEST.... TESTVARIABLE
/*
/*  #####
/*
/*
/*
ISPEXEC CONTROL DISPLAY LINE
/*
/*  EINSCHALTEN VON TESTAUSGABE
ISPEXEC VGET (TEST)
IF &TEST EQ J THEN +
  CONTROL CONLIST SYMLIST
/*
/*          D R U C K E R
FREE F(FT16F001)
/*          S T A R T S A T Z
FREE F(FT95F001)
/*          P A R A M E T E R
FREE F(FT96F001)
/*          T E X T D A T E I
FREE F(FT98F001)
ISPEXEC DISPLAY MSG(MED@017)
```

H.4 Programm - Panel - Skeleton

Im vorliegenden Beispiel wird vom Hauptmanual MKD direkt ein PL/1-Programm aufgerufen.

Dieses **Programm** bedient ein weiteres Select-Menue mit den dazugehoerigen **Panels**, fuehlt entsprechend der ausgewaehlten Option einen **Skeleton-File** aus und schickt ihn ab (submit).

Es folgen die entsprechenden Teile von ASTHMA.

-- M K S . I S P F . P L I (M K D @ 2) --

YMKD@2:PROC OPTIONS(MAIN);

```

/* */
/* */
/* Z W E C K: PROGRAMM FUER DAS ASTHMA - PANEL 'M K D @ 2HM/211/ */
/* ===== M K D @ 212/22 */
/* ERZEUGEN VON LADEMODULEN(PL/1 UND FORTRAN) */
/* OPTION 2 IN DER ANWENDUNG MKD */

```

/* VARIABLE AUS DEM PANEL: */

```

/* ===== */
/* PROC.... PROCEDURNAME */
/* PROJECT. PROJECT,USERID (QUELLE) */
/* LIBRARY. PROGRAMM,QUELLE */
/* TYPE.... DATEITYP */
/* MEMQ.... MEMBER */
/* PRJB.... PROJECT,USERID (LADEMODUL) */
/* LIBB.... PROGRAMM,QUELLE */
/* TYPEB... DATEITYP */
/* MEMB.... MEMBER (VORUEBERSETZTES LADEMODUL) */
/* MEMBB... MEMBER (STARTFAEHIGES LADEMODUL) */
/* PRJHP... PROJECT,USERID (VORUEBERSETZTES HAUPTPROGRAMM) */
/* LIBHP... PROGRAMM,QUELLE */
/* TYPEHP.. DATEITYP */
/* MEMIP... MEMBER */
/* UBIB1... UNTERPROGRAMMBIBLIOTHEK 1 (UBIBL1) */
/* UBIB2... UNTERPROGRAMMBIBLIOTHEK 2 (UBIBL2) */
/* UBIB3... UNTERPROGRAMMBIBLIOTHEK 3 (UBIBL3) */
/* GBIB.... GRAPHISCHE BIBLIOTHEK (GBIBL) */
/* MBIB.... MEDUSA-STANDART-BIBLIOTHEKEN */
/* VER..... VERSIONSNUMMER DER BLIOTHEKEN */
/* OPTCO... OPTIONEN FUER DEN COMPILER */
/* OPTLE... OPTIONEN FUER DEN LINKAGE EDITOR */
/* RTIME... RECHENZEIT IN SEKUNDEN */
/* LINES... ZEILEN IN TAUSEND */
/* MSGL.... MESSAGELEVEL */
/* MCLAS... MESSAGECLASS */
/* TEXT.... TEXT FUER JOBKARTE */
/* ACNR.... ACCOUNT-NUMMER */
/* JCL1.... ZUSAETZLICHE JCL-KARTEN (JCL2/JCL3/JCL4) */

```

/* VARIABLE AUS YASTHMA: */

```

/* ===== */
/* DRUCKER. VOREINGESTELLTE DRUCKAUSGABE */

```

/* D E K L A R A T I O N E N */

/* ===== */

```

DCL ISPLINK EXTERNAL ENTRY OPTIONS(ASM RETCODE);
DCL NAME EXTERNAL ENTRY (CHAR(4),CHAR(18));
DCL PLIRETV BUILTIN;
DCL LENGTH BUILTIN;
DCL SUBSTR BUILTIN;

```

```
DCL $DYNAL ENTRY (CHAR(255) VAR, FIXED BIN(31), FIXED BIN(31)),
IAO FIXED BIN (31) INIT(0),
IRC FIXED BIN (31) INIT(0),
COD1708 FIXED BIN (31) INIT(5896), /* HEX 1708 */
COD035C FIXED BIN (31) INIT(860), /* HEX 035C */
1 DALERR STATIC EXTERNAL, /* FEHLER-COMMONBLOCK */
2 ERRCOD FIXED BIN (31),
2 INFCOD FIXED BIN (31),
2 LERRT FIXED BIN (31),
2 ERRT FIXED BIN (31),
2 LINFT FIXED BIN (31),
2 INFT FIXED BIN (31);
DCL PDSMB EXTERNAL ENTRY (FILE, (*) CHAR(8) VAR,
FIXED BIN(31)),
CLDS FILE RECORD INPUT ENV(CONSECUTIVE, F(256, 256)),
MEMNAM(800) CHAR(8) VAR,
ANZMEM FIXED BIN(31),
DATASET CHAR(30) VAR;
DCL HILF FILE RECORD,
ISPFIL FILE RECORD,
ZEILE CHAR(80);
DCL I FIXED BIN(31),
CL FIXED BIN(31) INIT(0),
LE FIXED BIN(31) INIT(0);
DCL OPT CHAR(1),
OPTLM CHAR(1),
L_OPT FIXED BIN(31) INIT(1),
L_OPTLM FIXED BIN(31) INIT(1),
PROC CHAR(5) INIT((5) ' '),
L_PROC FIXED BIN(31) INIT(5),
TITEL CHAR(6) INIT((6) ' '),
L_TITEL FIXED BIN(31) INIT(6),
PREFIX CHAR(4) INIT((4) ' '),
L_PREFIX FIXED BIN(31) INIT(4),
PROJECT CHAR(8) INIT((8) ' '),
L_PROJECT FIXED BIN(31) INIT(8),
LIBRARY CHAR(8) INIT((8) ' '),
L_LIBRARY FIXED BIN(31) INIT(8),
TYPE CHAR(8) INIT((8) ' '),
L_TYPE FIXED BIN(31) INIT(8),
MEMQ CHAR(8) INIT((8) ' '),
L_MEMQ FIXED BIN(31) INIT(8),
PRJB CHAR(8) INIT((8) ' '),
L_PRJB FIXED BIN(31) INIT(8),
LIBB CHAR(8) INIT((8) ' '),
L_LIBB FIXED BIN(31) INIT(8),
TYPEB CHAR(8) INIT((8) ' '),
L_TYPEB FIXED BIN(31) INIT(8),
MEMB CHAR(8) INIT((8) ' '),
L_MEMB FIXED BIN(31) INIT(8),
MEMBB CHAR(8) INIT((8) ' '),
L_MEMBB FIXED BIN(31) INIT(8),
PRJHP CHAR(8) INIT((8) ' '),
L_PRJHP FIXED BIN(31) INIT(8),
LIBHP CHAR(8) INIT((8) ' '),
L_LIBHP FIXED BIN(31) INIT(8),
TYPEHP CHAR(8) INIT((8) ' '),
```

L_TYPEHP	FIXED BIN(31)	INIT(8),
MEMHP	CHAR(8)	INIT((8)' '),
L_MEMHP	FIXED BIN(31)	INIT(8),
UBIB1	CHAR(44)	INIT((44)' '),
L_UBIB1	FIXED BIN(31)	INIT(44),
UBIB2	CHAR(44)	INIT((44)' '),
L_UBIB2	FIXED BIN(31)	INIT(44),
UBIB3	CHAR(44)	INIT((44)' '),
L_UBIB3	FIXED BIN(31)	INIT(44),
UBIBL1	CHAR(1)	INIT((1)' '),
L_UBIBL1	FIXED BIN(31)	INIT(1),
UBIBL2	CHAR(1)	INIT((1)' '),
L_UBIBL2	FIXED BIN(31)	INIT(1),
UBIBL3	CHAR(1)	INIT((1)' '),
L_UBIBL3	FIXED BIN(31)	INIT(1),
BIB1	CHAR(44)	INIT((44)' '),
L_BIB1	FIXED BIN(31)	INIT(44),
BIB2	CHAR(44)	INIT((44)' '),
L_BIB2	FIXED BIN(31)	INIT(44),
BIB3	CHAR(44)	INIT((44)' '),
L_BIB3	FIXED BIN(31)	INIT(44),
GBIB	CHAR(5)	INIT((5)' '),
L_GBIB	FIXED BIN(31)	INIT(5),
GBIBL	CHAR(1)	INIT(' '),
L_GBIBL	FIXED BIN(31)	INIT(1),
MBIB	CHAR(1)	INIT(' '),
L_MBIB	FIXED BIN(31)	INIT(1),
VER	CHAR(2)	INIT(' '),
L_VER	FIXED BIN(31)	INIT(2),
OPTCO	CHAR(53)	INIT((53)' '),
L_OPTCO	FIXED BIN(31)	INIT(53),
OPTLE	CHAR(53)	INIT((53)' '),
L_OPTLE	FIXED BIN(31)	INIT(53),
RTIME	CHAR(2)	INIT(' '),
L_RTIME	FIXED BIN(31)	INIT(2),
LINES	CHAR(2)	INIT(' '),
L_LINES	FIXED BIN(31)	INIT(2),
MSGL	CHAR(3)	INIT(' '),
L_MSGL	FIXED BIN(31)	INIT(3),
MCLAS	CHAR(1)	INIT(' '),
L_MCLAS	FIXED BIN(31)	INIT(1),
TEXT	CHAR(18)	INIT((18)' '),
L_TEXT	FIXED BIN(31)	INIT(18),
PSWD	CHAR(8)	INIT((8)' '),
L_PSWD	FIXED BIN(31)	INIT(8),
ACNR	CHAR(8)	INIT((8)' '),
L_ACNR	FIXED BIN(31)	INIT(8),
JCL1	CHAR(72)	INIT((72)' '),
L_JCL1	FIXED BIN(31)	INIT(72),
JCL2	CHAR(72)	INIT((72)' '),
L_JCL2	FIXED BIN(31)	INIT(72),
JCL3	CHAR(72)	INIT((72)' '),
L_JCL3	FIXED BIN(31)	INIT(72),
JCL4	CHAR(72)	INIT((72)' '),
L_JCL4	FIXED BIN(31)	INIT(72),
DRUCKER	CHAR(10)	INIT((10)' '),
L_DRU	FIXED BIN(31)	INIT(10),

```
SEQ      CHAR(1)      INIT(' '),
L_SEQ    FIXED BIN(31) INIT(1);
DCL FLAG  BIT(1) INIT('1'B),
FLAG2    BIT(1) INIT('1'B),
A        FIXED BIN(31);
DCL (LP,LL,LT,LD) FIXED BIN(31);
DCL (IND1,IND2,IND3) FIXED BIN(31);
/*
/*****
/*
/*
/* DEFINIEREN DER VARIABLEN AUS DEN PANELS
/*
/*
CALL ISPLINK ('VDEFINE', '(OPT)', OPT, 'CHAR', L_OPT);
CALL ISPLINK ('VDEFINE', '(OPTLM)', OPTLM, 'CHAR', L_OPTLM);
CALL ISPLINK ('VDEFINE', '(PROC)', PROC, 'CHAR', L_PROC);
CALL ISPLINK ('VDEFINE', '(TITEL)', TITEL, 'CHAR', L_TITEL);
CALL ISPLINK ('VDEFINE', '(PREFIX)', PREFIX, 'CHAR', L_PREFIX);
CALL ISPLINK ('VDEFINE', '(PROJECT)', PROJECT, 'CHAR', L_PROJECT);
CALL ISPLINK ('VDEFINE', '(LIBRARY)', LIBRARY, 'CHAR', L_LIBRARY);
CALL ISPLINK ('VDEFINE', '(TYPE)', TYPE, 'CHAR', L_TYPE);
CALL ISPLINK ('VDEFINE', '(MEMQ)', MEMQ, 'CHAR', L_MEMQ);
CALL ISPLINK ('VDEFINE', '(PRJB)', PRJB, 'CHAR', L_PRJB);
CALL ISPLINK ('VDEFINE', '(LIBB)', LIBB, 'CHAR', L_LIBB);
CALL ISPLINK ('VDEFINE', '(TYPEB)', TYPEB, 'CHAR', L_TYPEB);
CALL ISPLINK ('VDEFINE', '(MEMB)', MEMB, 'CHAR', L_MEMB);
CALL ISPLINK ('VDEFINE', '(MEMBB)', MEMBB, 'CHAR', L_MEMBB);
CALL ISPLINK ('VDEFINE', '(PRJHP)', PRJHP, 'CHAR', L_PRJHP);
CALL ISPLINK ('VDEFINE', '(LIBHP)', LIBHP, 'CHAR', L_LIBHP);
CALL ISPLINK ('VDEFINE', '(TYPEHP)', TYPEHP, 'CHAR', L_TYPEHP);
CALL ISPLINK ('VDEFINE', '(MEMHP)', MEMHP, 'CHAR', L_MEMHP);
CALL ISPLINK ('VDEFINE', '(UBIB1)', UBIB1, 'CHAR', L_UBIB1);
CALL ISPLINK ('VDEFINE', '(UBIB2)', UBIB2, 'CHAR', L_UBIB2);
CALL ISPLINK ('VDEFINE', '(UBIB3)', UBIB3, 'CHAR', L_UBIB3);
CALL ISPLINK ('VDEFINE', '(UBIBL1)', UBIBL1, 'CHAR', L_UBIBL1);
CALL ISPLINK ('VDEFINE', '(UBIBL2)', UBIBL2, 'CHAR', L_UBIBL2);
CALL ISPLINK ('VDEFINE', '(UBIBL3)', UBIBL3, 'CHAR', L_UBIBL3);
CALL ISPLINK ('VDEFINE', '(BIB1)', BIB1, 'CHAR', L_BIB1);
CALL ISPLINK ('VDEFINE', '(BIB2)', BIB2, 'CHAR', L_BIB2);
CALL ISPLINK ('VDEFINE', '(BIB3)', BIB3, 'CHAR', L_BIB3);
CALL ISPLINK ('VDEFINE', '(GBIB)', GBIB, 'CHAR', L_GBIB);
CALL ISPLINK ('VDEFINE', '(GBIBL)', GBIBL, 'CHAR', L_GBIBL);
CALL ISPLINK ('VDEFINE', '(MBIB)', MBIB, 'CHAR', L_MBIB);
CALL ISPLINK ('VDEFINE', '(VER)', VER, 'CHAR', L_VER);
CALL ISPLINK ('VDEFINE', '(OPTCO)', OPTCO, 'CHAR', L_OPTCO);
CALL ISPLINK ('VDEFINE', '(OPTLE)', OPTLE, 'CHAR', L_OPTLE);
CALL ISPLINK ('VDEFINE', '(RTIME)', RTIME, 'CHAR', L_RTIME);
CALL ISPLINK ('VDEFINE', '(LINES)', LINES, 'CHAR', L_LINES);
CALL ISPLINK ('VDEFINE', '(MSG)', MSG, 'CHAR', L_MSG);
CALL ISPLINK ('VDEFINE', '(MCLAS)', MCLAS, 'CHAR', L_MCLAS);
CALL ISPLINK ('VDEFINE', '(TEXT)', TEXT, 'CHAR', L_TEXT);
CALL ISPLINK ('VDEFINE', '(PSWD)', PSWD, 'CHAR', L_PSWD);
CALL ISPLINK ('VDEFINE', '(ACNR)', ACNR, 'CHAR', L_ACNR);
CALL ISPLINK ('VDEFINE', '(JCL1)', JCL1, 'CHAR', L_JCL1);
CALL ISPLINK ('VDEFINE', '(JCL2)', JCL2, 'CHAR', L_JCL2);
CALL ISPLINK ('VDEFINE', '(JCL3)', JCL3, 'CHAR', L_JCL3);
CALL ISPLINK ('VDEFINE', '(JCL4)', JCL4, 'CHAR', L_JCL4);
```

```
CALL ISPLINK ('VDEFINE', '(DRUCKER)', DRUCKER, 'CHAR', L_DRU);
CALL ISPLINK ('VDEFINE', '(SEQ)', SEQ, 'CHAR', L_SEQ);
/* */
/* LESEN DER VARIABLEN AUS DEM SHARED-POOL */
CALL ISPLINK ('VGET', '(OPT,OPTLM,PREFIX,DRUCKER)', 'SHARED');
/* */
/* FEHLERBEHANDLUNG, FALLS MAN BEI EINEM READ AUF DIE EOF- */
/* MARKE TRIFFT. */
ON ENDFILE (ISPFIL)
  GOTO CLOS;
/* */
/* ALLOCIEREN DER M K S.L M J C L FUER DIE JCL */
/* ===== */
CALL $DYNAL ('ALLOC FI(ISPFIL) DA(''MKS.LMJCL'') OLD REUSE ;'
            , IAO, IRC);
/* */
/* UNTERPROGRAMM DAS DEN TEXT FUER DIE JOBKARTE MIT DEM */
/* NAMEN DES LM-BENUTZERS VORBELEGT. */
/* */
CALL NAME (PREFIX, TEXT);
CALL ISPLINK ('VPUT', '(TEXT)');
/* */
/* PANEL FUER JOBPARAMETER */
/* ===== */
D20: CALL ISPLINK ('DISPLAY', 'MKD@20 ');
IF PLIRETV() = 8 THEN GOTO FTCLOS;
/* */
/* ERSTELLEN DER AKTUELLEN JCL IN DER MKS.LMJCL */
CALL ISPLINK ('FTOPEN');
CALL ISPLINK ('VPUT', '(RTIME,LINES,MSG,L,MCLAS,TEXT,DRUCKER,
                    PREFIX,PSWD,ACNR,JCL1,JCL2,JCL3,JCL4)');
CALL ISPLINK ('FTINCL', 'MKD@2JO '); /* JCL FUER JOBKOPF */
/* */
/* PANEL ZUR AUSWAHL DER PROZEDUR */
/* ===== */
CALL ISPLINK ('DISPLAY', 'MKD@2HM ');
IF PLIRETV() = 8 THEN DO; /* END-TASTE GEDRUECKT */
  CALL ISPLINK ('FTCLOSE');
  GOTO D20;
END;
D1: DO WHILE (FLAG);
/* */
/* WELCHE OPTION WURDE GEWAHLT? */
/* ----- */
SEL1: SELECT (OPTLM);
      WHEN ('X') DO; /* RUECHKEHR NACH MKD-HAUPTPANEL */
        OPTLM = ' ';
        FLAG2 = '0'B;
        CALL ISPLINK ('FTCLOSE');
        GOTO FTCLOS;
      END;
/* */
/* AUFRUF DES LINKAGE EDITORS */
/* ===== */
```

```
WHEN ('5') DO;
  IF LE = 0
  THEN DO;
    CALL ISPLINK ('VGET', '(PRJHP,LIBHP,TYPEHP,MEMHP,PRJB,
                        LIBB,TYPEB,MEMBB,MEMHP,OPTLE,UBIB1,
                        UBIB2,UBIB3,GBIB,MBIB,VER)');
  /* PANEL FUER LINKAGE EDITOR */
  /* ===== */
  CALL ISPLINK ('DISPLAY', 'MKD@212 ');
  END;
  A = PLIRETV();
SEL2: SELECT (A);
  WHEN (8) FLAG = '0'B; /* END-TASTE GEDRUECKT
    --> RUECKKEHR INS HAUPTMANUAL */
  WHEN (0) DO; /* ENTER-TASTE "
    --> EINTRAG IN DIE MKS,LMJCL ERWUENSCHT */
  /* */
  /* UNTERPROG-BIBL.GLEICH HAUPTPROG-BIBLIOTHEK */
  /* ----- */
  IF UBIB1 = '*'
  THEN DO;
    LP = INDEX(PRJHP, ' ');
    IF (LP = 0) THEN LP = 9;
    LL = INDEX(LIBHP, ' ');
    IF (LL = 0) THEN LL = 9;
    LT = INDEX(TYPEHP, ' ');
    IF (LT = 0) THEN LT = 9;
    BIB1 = '' || SUBSTR(PRJHP,1,LP-1) || '.' ||
          SUBSTR(LIBHP,1,LL-1) || '.' ||
          SUBSTR(TYPEHP,1,LT-1) || '';
  END;
  ELSE BIB1 = UBIB1;
  /*
    ABFRAGE,OB UNTERPROG-BIBL. VOLLQUALIFIZIERTANGEGEBEN
    WURDEN */
  /*-----*/
  IND1 = INDEX (BIB1, '');
  IND2 = INDEX (UBIB2, '');
  IND3 = INDEX (UBIB3, '');
  IF UBIBL1 = 'J' THEN DO;
    IF (IND1 = 0) THEN
      BIB1 = PREFIX || '.' || BIB1;
    ELSE DO;
      BIB1 = SUBSTR(BIB1,IND1 + 1);
      IND1 = INDEX (BIB1, '');
      BIB1 = SUBSTR(BIB1,1,IND1 - 1);
    END;
  END;
  IF UBIBL2 = 'J' THEN DO;
    IF (IND2 = 0) THEN
      BIB2 = PREFIX || '.' || UBIB2;
    ELSE DO;
      BIB2 = SUBSTR(UBIB2,IND2 + 1);
      IND2 = INDEX (BIB2, '');
      BIB2 = SUBSTR(BIB2,1,IND2 - 1);
    END;
  END;
  END;
```

```
IF UBIBL3 = 'J' THEN DO;
  IF (IND3 = 0) & (UBIBL3 = 'J') THEN
    BIB3 = PREFIX || '.' || UBIB3;
  ELSE DO;
    BIB3 = SUBSTR(UBIB3,IND3 + 1);
    IND3 = INDEX (BIB3,'');
    BIB3 = SUBSTR(BIB3,1,IND3 - 1);
  END;
END;
END;
/*
/*
/* EINTRAG IN DIE M K S.L M J C L
/*
CALL ISPLINK ('VPUT','(PROC,PRJHP,LIBHP,TYPEHP,MEMHP,
  PRJB,LIBB,TYPEB,MEMBB,OPTLE,BIB1,BIB2,
  BIB3,UBIBL1,UBIBL2,UBIBL3,GBIB,GBIBL,
  MBIB,VER)');
CALL ISPLINK ('FTINCL','MKD@2LE ');
CALL ISPLINK ('VGET','(PROC,PRJHP,LIBHP,TYPEHP,MEMHP,
  PRJB,LIBB,TYPEB,MEMBB,OPTLE,BIB1,BIB2,
  BIB3,UBIBL1,UBIBL2,UBIBL3,GBIB,GBIBL,
  MBIB,VER)');
CALL ISPLINK ('DISPLAY','MKD@212 ','MKD@028 ');
IF PLIRETV() = 8 THEN FLAG = '0'B;
LE = LE + 1;
CL = 0;
END;
OTHERWISE ;
END SEL2;
END ;
/*
/*
/* FUER C O M P I L E AND L I N K
/*
OTHERWISE DO;
DISP: IF CL = 0
  THEN DO;
    CALL ISPLINK ('VGET','(PROJECT,LIBRARY,TYPE,PRJB,LIBB,
      TYPEB,MEMQ,MEMB,OPTCO)');
    /* PANEL FUER COMPILE AND LINK
    /*
    CALL ISPLINK ('DISPLAY','MKD@211 ');
  END;
  A = PLIRETV();
SEL3: SELECT(A);
  WHEN(8) FLAG = '0'B; /* END-TASTE GEDRUECKT
    --> RUECKKEHR INS HAUPTMANUAL
  WHEN(0) DO; /* ENTER-TASTE
    --> EINTRAG IN DIE MKS.LMJCL ERWUENSCHT
    IAO = 0;
    /*
    /* MEHRERE MEMBERS ERWUENSCHT
    /*
    IF (MEMQ = '*') | (MEMQ = '**')
    THEN DO;
      SEQ = 'N';
      DO WHILE (IAO = 0);
```

```

/*                                                    */
/* ZUSAMMENBAU DER EINGABEDATEI                        */
/*                                                    */
LP = INDEX(PROJECT,' ');
IF (LP = 0) THEN LP = 9;
LL = INDEX(LIBRARY,' ');
IF (LL = 0) THEN LL = 9;
LT = INDEX(TYPE,' ');
IF (LT = 0) THEN LT = 9;
DATASET = SUBSTR(PROJECT,1,LP-1) || '.' ||
          SUBSTR(LIBRARY,1,LL-1) || '.' ||
          SUBSTR(TYPE,1,LT-1);
LD = LP + LL + LT - 1;
CALL $DYNAL ('ALLOC FI(CLDS) DA(''
          || SUBSTR(DATASET,1,LD) ||
          ''') SHR REUSE ;',IAO,IRC);

/*                                                    */
/* FEHLERABFRAGE DES RETURN CODE                      */
/*                                                    */
IF IRC=0 THEN
DO;
  SELECT(ERRCOD);

  WHEN(COD1708) DO;
    PUT EDIT('***** FEHLER: DIE DATEI IST NICHT',
            ' KATALOGISIERT') (SKIP,A,A);
    GOTO DISP;
  END;
  WHEN(COD035C) DO;
    PUT EDIT('***** FEHLER: DATEINAME IST FALSCH')
            (SKIP,A);
    GOTO DISP;
  END;
  OTHERWISE DO;
    IAO = 1;
    /*                                                    */
    /* A L L O C I E R U N G DER DATEI FUER PDSMB      */
    /* ----- */
    CALL $DYNAL('ALLOC FILE(CLDS) DA('' ||
              SUBSTR(DATASET,1,LD) || ''')
              SHR REUSE ;',IAO,IRC);

    END;
    END; /* ENDE SELECT- ANWEISUNG */
  END;
  ELSE IAO = 1; /* RETURN CODE IST NULL */
END; /* ENDE DO -WHILE */
/* RETCODE GLEICH NULL --> AUFLISTUNG DER MEMBER */
/*                                                    */
CALL PDSMB (CLDS, MEMNAM, ANZMEM);
/*                                                    */
/* A L L E M E M B E R */
/* ----- */
IF MEMQ = '*'
THEN
D2: DO I = 1 TO ANZMEM;
      MEMQ = MEMNAM(I);
      MEMB = MEMNAM(I);
      /*-
*/

```



```
/* ERSTELLEN DER JCL IN DER MKS.LMJCL */
CALL ISPLINK ('VPUT', '(PROC,PROJECT,LIBRARY,TYPE,
                    SEQ,MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
CALL ISPLINK ('FTINCL', 'MKD@2CL ');
IF (I = ANZMEM )
THEN DO;
    MEMQ = '';
    MEMB = '';
    CALL ISPLINK ('VGET', '(PROC,PROJECT,LIBRARY,TYPE,
                        MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
    CALL ISPLINK ('DISPLAY', 'MKD@211 ', 'MKD@028 ');
    IF PLIRETV() = 8 THEN FLAG = '0'B;
END;
END D2;
/*
/* ALLE M E M B E R O H N E S O N D E R - Z E I C H E N
/* -----
/*
D3: DO I = 1 TO ANZMEM;
    IF (INDEX (MEMNAM(I), '@') = 0) &
        (INDEX (MEMNAM(I), '#') = 0) &
        (INDEX (MEMNAM(I), '$') = 0)
    THEN DO;
        MEMQ = MEMNAM(I);
        MEMB = MEMNAM(I);
        /*
        /* ERSTELLEN DER JCL IN DER MKS.LMJCL
        CALL ISPLINK ('VPUT', '(PROC,PROJECT,LIBRARY,TYPE,
                            SEQ,MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
        CALL ISPLINK ('FTINCL', 'MKD@2CL ');
        END; /* THEN-BLOCK */
        IF I = ANZMEM
        THEN DO;
            MEMQ = '';
            MEMB = '';
            CALL ISPLINK ('VGET', '(PROC,PROJECT,LIBRARY,TYPE,
                            MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
            CALL ISPLINK ('DISPLAY', 'MKD@211 ', 'MKD@028 ');
            IF PLIRETV() = 8 THEN FLAG = '0'B;
        END;
    END D3;
    CLOSE FILE (CLDS);
    CALL $DYNAL ('FREE FI(CLDS) ;', IAO, IRC);
END;
/*
/* E I N S P E Z I E L L E S M E M B E R
/* -----
/*
ELSE DO;
/*
/* ERSTELLEN DER JCL IN DER MKS.LMJCL
IF (MEMQ = '') THEN SEQ = 'J';
ELSE SEQ = 'N';
CALL ISPLINK ('VPUT', '(PROC,PROJECT,LIBRARY,TYPE,
                    SEQ,MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
CALL ISPLINK ('FTINCL', 'MKD@2CL ');
CALL ISPLINK ('VGET', '(PROC,PROJECT,LIBRARY,TYPE,
                    SEQ,MEMQ,PRJB,LIBB,TYPEB,MEMB,OPTCO)');
```

```
        CALL ISPLINK ('DISPLAY', 'MKD@211 ', 'MKD@028 ');
        IF PLIRETV() = 8 THEN FLAG = '0'B;
        END;
        CL = CL + 1;
        LE = 0;
        END;
        OTHERWISE ;
        END SEL3;
    END;
    END SEL1;
END D1;
/*
/* AUSWAHL-PANEL FUER DIE PROZEDUREN
/* =====
DISPL: CALL ISPLINK ('DISPLAY', 'MKD@2HM ');
A = PLIRETV();
SEL4: SELECT(A);
        WHEN (8)                                /* PF3-TASTE GEDRUECKT */
            GOTO FTCLOS;
        WHEN(0) DO;                               /* ENTER-TASTE GEDRUECKT */
            FLAG = '1'B;
            GOTO D1;                               /* NEUBEGINN */
        END;
        OTHERWISE ;
    END SEL4;
FTCLOS: CALL ISPLINK ('FTCLOSE');
/*
/* S U B M I T D E S J O B S
/* =====
IF FLAG2 = '1'B
THEN DO;
    CALL $DYNAL ('ALLOC FI(HILF) SYSOUT(C) SPGNM(INTRDR)'||
                ' RECFM(FB) BLKSIZE(6160) LRECL(80) ;',
                IAO,IRC);
/*
/* KOPIEREN DER JCL AUS DER MKS.LMJCL IN EINE DURCH DEN
/* INTERNAL READER ALLOCIERTE DATEI UND SUBMIT DES JOBS
/* DURCH DEN CLOSE AUF DEN HILFSFILE.
/*
/*
    OPEN FILE(HILF) OUTPUT;
    OPEN FILE(ISPFILE) INPUT;
D4: DO WHILE(1 = 1);
    READ FILE(ISPFILE) INTO (ZEILE);
    WRITE FILE(HILF) FROM (ZEILE);
    END D4;
CLOS: CLOSE FILE(HILF);
    CALL $DYNAL ('FREE FI(HILF) ;', IAO,IRC);
    END;
    CLOSE FILE(ISPFILE);
/* FREIGEBEN
/*
    CALL $DYNAL ('FREE FI(ISPFILE) ;', IAO,IRC);
    END YMKD@2;
```

-- M K S . I S P F . P A N E L I B (M K D @ 2 H M) --

```
%MKD ----- LM   HAUPTMANUAL -----
%GEWAEHLTE OPTION ==>_OPTLM   +                                +USERID :%&ZUSER
%                                                                    +PREFIX :%&ZPREFIX
%
%  1 +FTN H           - FORTRAN-H-COMPILER
%  2 +FTN G1         - FORTRAN-G1-COMPILER
%  3 +FTN 77         - FORTRAN-77-COMPILER
%  4 +PLI            - PLI-OPTIMIZER
%  5 +LE             - LINKAGE-EDITOR
%  X +ENDE          - RUECKKEHR IN'S HAUPTMANUAL OHNE SUBMIT
%
%  B +BESCHREIBUNG
%
+DRUECKEN SIE DIE%ENTER-TASTE+ZUR BEREITSTELLUNG DER QUELLDATEI
+
+NACH RUECKKEHR DRUECKEN SIE DIE%END-TASTE+ZUM SUBMIT DES JOBS.
%
)PROC
  VER(&OPTLM, NONBLANK                                MSG=MKD@020)
  &OPTLM = TRUNC(&OPTLM, '.')
  VER (&OPTLM, LIST,1,2,3,4,5,'X'                    MSG=MKD@026)
  &TITEL = TRANS(&OPTLM 1,'FTN H' 2,'FTN G1' 3,'FTN 77' 4,PLI 5,LE
                X,' ')
  &PROC  = TRANS(&OPTLM 1,FHCL 2,FGCL 3,FFCL 4,POCL 5,'
                X,' ')
)END
```

-- M K S . I S P F . P A N E L I B (M K D @ 2 0) --

```
)ATTR
  c TYPE(INPUT) INTENS(NON)
)BODY
%MKD ----- LM/JOBPARAMETER -----
%EINGEBEN/VERAENDERN DER UNTENSTEHENDEN PARAMETER :
+
+       DRUECKEN SIE DIE%ENTER-TASTE+ZUR AUSWAHL DER PROZEDUR
+
+RECHENZEIT (IN SEKUNDEN):           %=>_RTIME+
+ANZAHL DER ZEILEN (IN TAUSEND):     %=>_LINES+
+MESSAGELEVEL:                       %=>_MSGL +
+MESSAGECLASS:                       %=>_MCLAS+
+TEXT FUER JOB-KARTE:                %=>_TEXT           +
+ACCOUNTNUMMER:                      %=>_ACNR      + (8-STELLIG)
+
+PASSWORT:                            %=>CPSWD      +
+
+----- ZUSAETZLICHE JCL -----
_JCL1                                     +
_JCL2                                     +
_JCL3                                     +
_JCL4                                     +
+-----
+
+                               NAME DES JOBS :%&ZUSER.LM
)INIT
  .CURSOR = PSWD
    &RTIME = 2
    &LINES = 2
    &MSGL = '0,0'
    &MCLAS = H
    &JCL1 = '/*'
    &JCL2 = '/*EXEC COMPRESS,LIB=LINK,TYPE=LOAD,OUT='*'
    &JCL3 = '/*'
    &JCL4 = '/*'
)PROC
  VER(&RTIME ,NUM                      MSG=MKD@021)
  VER(&LINES ,NUM                      MSG=MKD@021)
  VER(&RTIME ,NONBLANK                 MSG=MKD@024)
  VER(&LINES ,NONBLANK                 MSG=MKD@024)
  VER(&MSGL ,NONBLANK                 MSG=MKD@024)
  VER(&MCLAS ,NONBLANK                 MSG=MKD@024)
  VER(&PSWD ,NONBLANK                 MSG=MKD@029)
  VER(&MCLAS ,LIST,H,A,T              MSG=MKD@023)
  VER(&MSGL ,LIST,'0,0','0,1','1,0','1,1','2,0','2,1' MSG=MKD@023)
  VER(&ACNR ,PICT,NNNNNNNN           MSG=MKD@021)
  IF (&ACNR = ' ')
    &ACNR = N
  VER(&JCL1 ,NONBLANK                 MSG=MKD@025)
  VER(&JCL2 ,NONBLANK                 MSG=MKD@025)
  VER(&JCL3 ,NONBLANK                 MSG=MKD@025)
  VER(&JCL4 ,NONBLANK                 MSG=MKD@025)
)END
```

-- M K S . I S P F . P A N E L I B (M K D @ 2 J O) --

```
//&ZUSER.LM JOB , '&TEXT' , MSGCLASS=&MCLAS , MSGLEVEL=(&MSGL) ,  
// NOTIFY=&ZUSER  
/*JOBPARM TIME=&RTIME , LINES=&LINES , W=&PSWD  
)SEL &ACNR ^= N  
/*JOBPARM ACCT=&ACNR  
)ENDSEL  
)SEL &MCLAS = A  
/*ROUTE PRINT &DRUCKER  
)ENDSEL  
&JCL1  
&JCL2  
&JCL3  
&JCL4
```

-- MKS . I S P F . P A N E L I B (M K D @ 2 1 1) --

```
%MKD ----- LM/&TITEL -----
%EINGEBEN/VERAENDERN DER UNTENSTEHENDEN PARAMETER :
+
+   DRUECKEN SIE DIE %ENTER-TASTE+ZUR ERSTELLUNG DER JCL,
+   DIE %END-TASTE+ZUR RUECKKEHR IN'S HAUPTMANUAL LM
+
+
%QUELLE+:
+   PROJEKT,USERID   %=>_PROJECT +
+   PROGRAMM,QUELLE %=>_LIBRARY +
+   DATEITYP        %=>_TYPE    +
+   MEMBER          %=>_MEMQ    +
+
+   %BLANK +DATEI SEQUENTIELL
+   %*      +ALLE MEMBER VORUEBERSETZEN
+   %**     +WIE *, ABER OHNE @,$ ODER # IM NAMEN
+
% LADEMODUL:
+   %=>_PRJB  +
+   %=>_LIBB  +
+   %=>_TYPEB +
+   %=>_MEMB  +
+
+   %=>GLEICHER MEMBERNAME
+
+COMPILER-%OPTIONEN+: %=>_OPTCO
+
)INIT
  .CURSOR = MEMQ
  &PRJB = &PROJECT
  &MEMB = '='
  &MEMQ = ''
  &OPTCO = NOSOURCE
)PROC
  IF (&MEMB = '=')
    &MEMB = &MEMQ
    VER(&PROJECT, NONBLANK      MSG=MKD@004)
    VER(&LIBRARY, NONBLANK      MSG=MKD@004)
    VER(&TYPE,    NONBLANK      MSG=MKD@004)
    VER(&PRJB,   NONBLANK      MSG=MKD@015)
    VER(&LIBB,   NONBLANK      MSG=MKD@015)
    VER(&TYPEB,  NONBLANK      MSG=MKD@015)
    VER(&MEMB,   NONBLANK      MSG=MKD@002)
    VER(&PROJECT, NAME          MSG=MKD@003)
    VER(&LIBRARY, NAME          MSG=MKD@003)
    VER(&TYPE,   NAME          MSG=MKD@003)
    VER(&PRJB,   NAME          MSG=MKD@003)
    VER(&LIBB,   NAME          MSG=MKD@003)
    VER(&TYPEB,  NAME          MSG=MKD@003)
)END
```

-- M K S . I S P F . S K E L L I B (M K D @ 2 C L) --

```
// EXEC &PROC,OPTC='&OPTCO'  
)SEL &SEQ = N  
//C.SYSIN      DD   DSN=&PROJECT..&LIBRARY..&TYPE(&MEMQ),DISP=SHR  
)ENDSEL  
)SEL &SEQ = J  
//C.SYSIN      DD   DSN=&PROJECT..&LIBRARY..&TYPE,DISP=SHR  
)ENDSEL  
//L.SYSLMOD    DD   DSN=&PRJB..&LIBB..&TYPEB(&MEMB),DISP=SHR
```

-- M K S . I S P F . P A N E L I B (M K D @ 2 1 2) --

```
%MKD ----- LM/&TITEL -----
%EINGEBEN/VERAENDERN DER UNTENSTEHENDEN PARAMETER :
+
+   DRUECKEN SIE DIE %ENTER-TASTE+ZUR ERSTELLUNG DER JCL,
+   DIE %END-TASTE+ ZUR RUECKKEHR IN'S HAUPTMANUAL LM
+
%HAUPTPROGRAMM                                %STARTFAEHIGES LADEMODUL
+   PROJEKT,USERID   %=>_PRJHP   +           %=>_PRJB   +   (*)
+   PROGRAMM,QUELLE %=>_LIBHP   +           %=>_LIBB   +
+   DATEITYP        %=>_TYPEHP  +           %=>_TYPEB  +
+   MEMBER          %=>_MEMHP   +           %=>_MEMBB  +
+
+                                           %*+GLEICHE BIBLIOTHEK WIE HP
%UNTERPROGRAMMBIBLIOTHEKEN :
+   UNTERPROGRAMMBIBLIOTHEK 1           ==>_UBIB1
+                                           %*+GLEICHE BIBLIOTHEK WIE HP
+   UNTERPROGRAMMBIBLIOTHEK 2           ==>_UBIB2
+   UNTERPROGRAMMBIBLIOTHEK 3           ==>_UBIB3
+   GRAPHISCHE BIBLIOTHEK                ==>_GBIB + (GINO/GRAPH/VTEC)
+   MEDUSA STANDART-BIBLIOTHEKEN        ==>_MBIB+ (J/N)
+   VERSIONSNUMMER                       ==>_VER+
+
%OPTIONEN+FUER DEN LINKAGE-EDITOR ==>_OPTLE
+
)INIT
  .CURSOR = MEMHP
  &PRJB   = '*'
  &LIBB   = '      '
  &TYPEB  = '      '
  &MEMHP  = ''
  &MEMBB  = ''
  &MBIB   = N
)PROC
  &GBIBL  = ''
  &UBIBL1 = ''
  &UBIBL2 = ''
  &UBIBL3 = ''
  VER(&PRJHP,   NONBLANK           MSG=MKD@015)
  VER(&LIBHP,   NONBLANK           MSG=MKD@015)
  VER(&TYPEHP,  NONBLANK           MSG=MKD@015)
  VER(&MEMHP,   NONBLANK           MSG=MKD@002)
  VER(&PRJB,    NONBLANK           MSG=MKD@015)
  IF (&PRJB = '*' )
    &PRJB = &PRJHP
    &LIBB = &LIBHP
    &TYPEB = &TYPEHP
  VER(&LIBB,    NONBLANK           MSG=MKD@015)
  VER(&TYPEB,   NONBLANK           MSG=MKD@015)
  VER(&MEMBB,   NONBLANK           MSG=MKD@002)
  VER(&PRJHP,   NAME                MSG=MKD@003)
  VER(&LIBHP,   NAME                MSG=MKD@003)
  VER(&TYPEHP,  NAME                MSG=MKD@003)
  VER(&MEMHP,   NAME                MSG=MKD@003)
  VER(&PRJB,    NAME                MSG=MKD@003)
```



```
VER(&LIBB,      NAME                MSG=MKD@003)
VER(&TYPEB,    NAME                MSG=MKD@003)
VER(&MEMBB,    NAME                MSG=MKD@003)
VER(&GBIB,     LIST,GINO,GRAPH,VTEC  MSG=MKD@027)
IF (&GBIB = ' ')
    &GBIBL = 'N'
&MBIB = TRANS(&MBIB JA,J NEIN,N J,J N,N MSG=MKD@007)
&UBIBL1 = TRUNC(&UBIB1,4)
IF (&UBIBL1 = ' ')
    &UBIBL1 = 'J'
&UBIBL2 = TRUNC(&UBIB2,4)
IF (&UBIBL2 = ' ')
    &UBIBL2 = 'J'
&UBIBL3 = TRUNC(&UBIB3,4)
IF (&UBIBL3 = ' ')
    &UBIBL3 = 'J'
IF (&MBIB = 'J')
    &VER = TRANS(&VER &VERA,&VERA &VERAP,&VERA &VERB,&VERB
                &VERBP,&VERB &VERO,&VERO &VEROP,&VERO
                MSG=MKD@009)
```

)END

-- MKS . I S P F . S K E L L I B (M K D @ 2 L E) --

```
//LINK0001 EXEC PGM=IEWL,PARM='&OPTLE'  
//SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR,DCB=BLKSIZE=8638  
// DD DSN=SYS1.PLIBASE,DISP=SHR  
)SEL &UBIBL1 = J  
// DD DISP=SHR,DSN=&BIB1  
)ENDSEL  
)SEL &UBIBL2 = J  
// DD DISP=SHR,DSN=&BIB2  
)ENDSEL  
)SEL &UBIBL3 = J  
// DD DISP=SHR,DSN=&BIB3  
)ENDSEL  
)SEL &MBIB = J  
// DD DISP=SHR,DSN=ZFF.MEDUSA&VER..LOADRA  
// DD DISP=SHR,DSN=ZFF.MEDUSA&VER..LOAD  
// DD DISP=SHR,DSN=ZFF.ALLROU&VER..LOAD  
// DD DISP=SHR,DSN=ZFF.DATORG&VER..LOAD  
)ENDSEL  
)SEL &GBIBL = N  
// DD DISP=SHR,DSN=SYS2.&GBIB.LIB  
)ENDSEL  
//LINHP DD DISP=SHR,DSN=&PRJHP..&LIBHP..&TYPEHP  
//SYSLIN DD *  
INCLUDE LINHP(&MEMHP)  
/*  
//SYSUT1 DD UNIT=VIO,SPACE=(4096,(30,30))  
//SYSLMOD DD DISP=OLD,DSN=&PRJB..&LIBB..&TYPEB(&MEMBB)  
//SYSPRINT DD SYSOUT=H  
/*
```