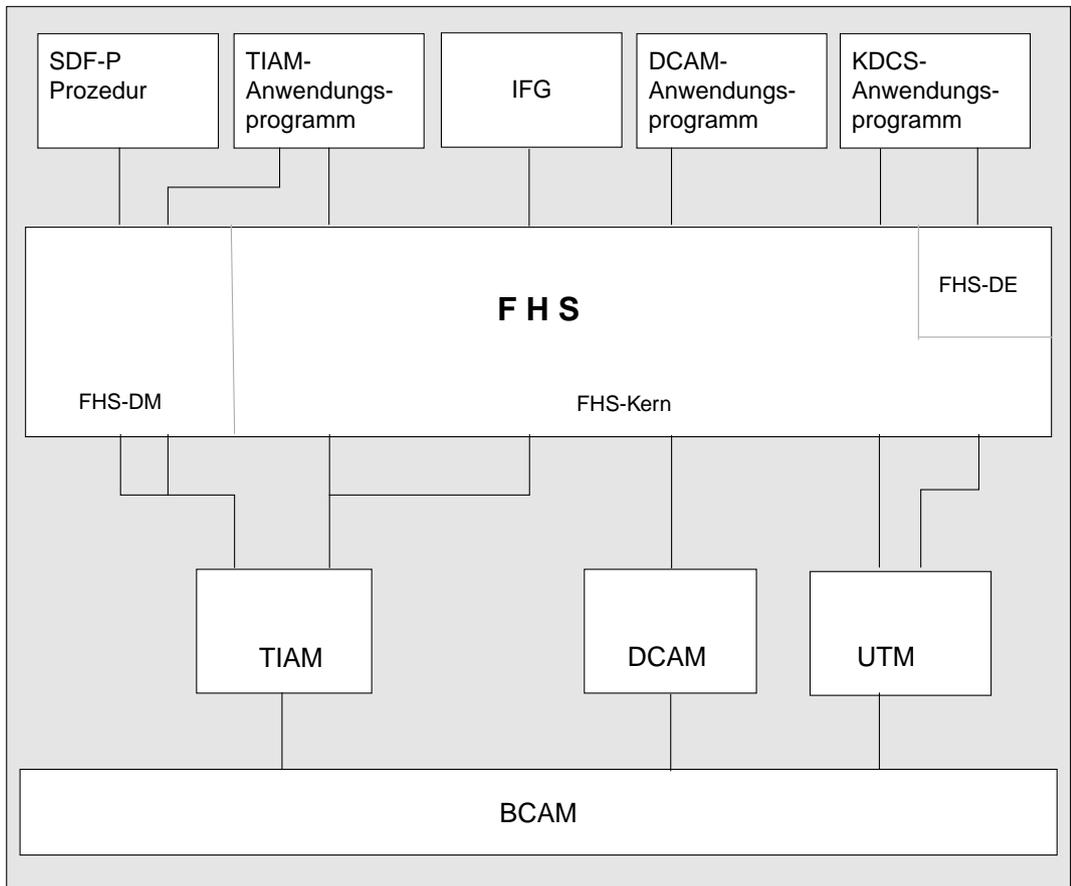


1 Einleitung

Das **Format Handling System (FHS)** ist ein Programm zur Formatsteuerung, das den Austausch formatierter Nachrichten zwischen Anwendungsprogrammen und Datenstationen unterstützt. Durch den Einsatz von FHS ist das Anwendungsprogramm weitgehend unabhängig von den physischen Eigenschaften der Datenstationen. FHS kann für Anwendungsprogramme im Transaktionsbetrieb, im Teilhaber- und Teilnehmerbetrieb eingesetzt werden. Die Einbettung von FHS in die Systemumgebung zeigt das folgende Bild.



FHS-DM in der Systemumgebung

Bestandteile von FHS

FHS besteht aus drei Teilen:

- FHS-Kern:
In ihm wird eine statische Formatierung durchgeführt, d.h. es werden Programmdateien mit einem Format verbunden. Es wird eine Nachricht erzeugt, die an einem Bildschirm oder an einem Drucker ausgedruckt werden kann.
Daten die von einer Datensichtstation kommen, werden dem Format entsprechend aufbereitet und der Anwendung zur Verfügung gestellt. Der Kern enthält auch einige Servicemodule.
- Verbindungsmodule zu UTM-Anwendungsprogrammen, einschließlich der Dialogerweiterung für UTM.
- FHS-Dialogmanager:
FHS-DM realisiert eine entkoppelte Verbindung zwischen einem TIAM-Anwendungsprogramm bzw. einer SDF-P-Prozedur und einer Datensichtstation

Anwendungsprogramme, die den FHS-Kern und die Dialogerweiterung für UTM für die Formatierung verwenden (TIAM-, DCAM-, UTM-Programme), nutzen für den Datenaustausch mit FHS einen Datenübergabebereich. Die Struktur dieses Übergabebereiches wird bei der Formaterstellung mit dem Interaktiven Formatgenerator (IFG) festgelegt. Wird das Format später geändert, kann sich auch der Datenübergabebereich ändern. Aus diesem Grund ist meist eine Neuübersetzung des Anwendungsprogramms notwendig.

Erfolgt der Datenaustausch über den FHS-Dialogmanager (TIAM-Anwendungsprogramme oder SDF-P-Prozedur), dann werden Datenbereiche verwendet, die durch die Dialogvariablen definiert sind. Der Dialogmanager verbindet die im Format definierten Felder, denen bei der Formaterstellung ein Name zugeordnet wurde, mit Dialogvariablen. Ein Variablenhandler stellt die Daten-Verbindung zwischen dem Anwendungsprogramm bzw. der SDF-P-Prozedur und dem FHS-Dialogmanager her.

Eine Änderung des Formates erfordert nicht unbedingt eine Neuübersetzung.

Bei der Erstellung der Formate und für die Bedienung von FHS-DE und FHS-DM wurden die Empfehlungen des "Alpha-Style-Guides" der Siemens-Nixdorf-Informationssysteme AG berücksichtigt.

Hilfen und Prüfungen müssen nicht im Anwendungsprogramm programmiert werden, sondern werden im Format beschrieben und durch FHS-DE oder FHS-DM ausgeführt.

FHS arbeitet mit Formaten, die Sie vorher mit dem IFG erstellt haben.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Dieses Benutzerhandbuch beschreibt den Dialogmanager von FHS (im folgenden wird die Bezeichnung FHS-DM verwendet) und richtet sich an Datenstationsbenutzer und Programmierer, die die TIAM-Schnittstellen für Datenfernverarbeitung im BS2000 verwenden. Die bisherige FHS-Schnittstelle ist im Handbuch "FHS - Formatierungssystem für UTM, TIAM, DCAM" beschrieben.

Zum Verständnis des Handbuches sind Grundkenntnisse des BS2000 sowie Kenntnisse der verwendeten Programmiersprache nötig.

1.2 Wegweiser durch das Handbuch

Die vorliegende Beschreibung ist wie folgt gegliedert:

- Einführung in FHS
- Einführung in die Dialogelemente
- Datentransport, Prüfungen und Aufbereitung
- Bedienung an der Datenstation
- Schnittstelle zu TIAM-Anwendungsprogrammen
- Schnittstelle zu SDF-P
- Beispiel-Programme

Readme-Datei

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

`SYSDOC.produkt.version.READ-ME.D.`

Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrem zuständigen Systemverwalter. Die Readme-Datei können Sie mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
PRINT-FILE FILE-NAME=dateiname,LAYOUT-CONTROL=PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

2 Einführung in FHS

Was ist ein Format?

Ein Format oder auch Maske genannt, entspricht einem Formular, das am Bildschirm einer Datensichtstation angezeigt wird. Genau wie die Formulare, die Ihnen täglich begegnen (Auftragsformular, Bestellformular), besteht ein Format aus Feldern, die Sie ausfüllen können und aus festgelegten Texten, die zum Formular gehören. Hinter diesem "Formular" verbirgt sich eine logische Datenstruktur. Diese besteht aus:

- Feldern mit festen Texten (Textfelder)
- Feldern, in die der Datenstationsbenutzer und/oder das Anwendungsprogramm etwas eintragen können (variable Felder)
- Angaben über die Position dieser Felder am Bildschirm
- Angaben über die Eigenschaften des Formates, z.B. an welcher Datenstation das Format ausgegeben werden kann
- Angaben über Eigenschaften (Attribute) der Felder des Formates, z.B. unterstrichen
- Angaben zu den Aufbereitungseigenschaften der Feldinhalte
- Namen der Felder (Dialogvariablen-Namen)
- Meldungskennzeichen für Nuttermeldungen
- Verbindungen zu Hilfeformaten
- Verbindungen zu Funktionstastenbelegung

Das folgende Bild zeigt ein Beispiel für ein Format wie es am Bildschirm angezeigt wird.

```
-----
                          Pers. Datei
-----
BITTE ANSCHRIFT EINGEBEN

Name: _____
Vorname: _____
Strasse: _____
PLZ:      00000
Ort:     _____
Telefon Vorwahl: 00000   Nummer: 0000000

Kunden-Nr.: 0000000000
-----
Kommando:
F1 = Help   F3 = Exit   F12 = Cancel
```

Beispiel für ein Format

Da FHS die Anwendungsprogramme unabhängig macht von den physikalischen Eigenschaften der Datenstationen, kann der Anwender mit verschiedenen Datenstationen arbeiten, ohne deren unterschiedlichen Eigenschaften kennen zu müssen. Er arbeitet mit logischen Datenstationen, FHS bedient die Schnittstelle zu den tatsächlichen Datenstationen.

Mit welchen Datenstationen arbeitet FHS-DM?

FHS-DM V8.1 unterstützt die Arbeit mit Datensichtstationen 8160, 9750, 9755, 9763 und 3270, sowie äquivalente Geräte und Emulationen

Voraussetzung für die Unterstützung des IBM-Systems 3270 durch FHS ist die Installation des Produktes TRANSIT-CD im Vorrechner und die Datenstationen müssen als Systemgerätetyp 3270 generiert sein.

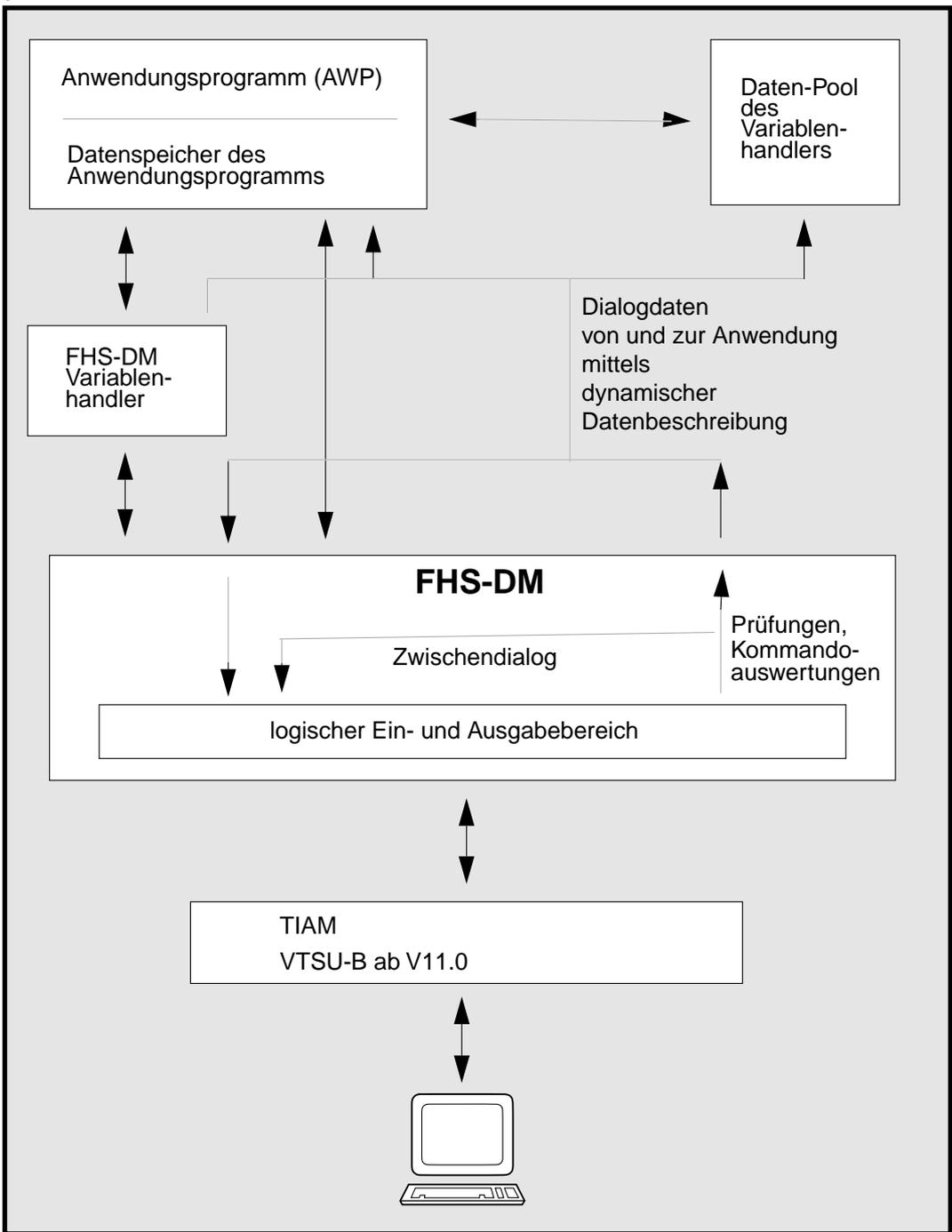
Die Druckerstationen können sowohl lokal an eine Datensichtstation angeschlossen sein wie auch zentral über eine Druckerstationssteuerung.

Bei falscher Angabe des Datenstationstyps im PDN können bei der Formatierung Fehler auftreten. Der tatsächliche Datenstationstyp und der in PDN generierte Datenstationstyp müssen übereinstimmen.

Wirkungsweise

FHS unterstützt die Ein- und Ausgabe formatierter Nachrichten, die im Dialog zwischen Anwendungsprogramm und Datenstation ausgetauscht werden. Außerdem kann es selbst Zwischendialoge führen. Zum Beispiel können ohne Aufruf des Anwendungsprogramms Fehlermeldungen ausgegeben werden, die den Terminal-Benutzer zur Korrektur fehlerhaft eingegebener Daten auffordern. Auch kann FHS-DM Hilfeformate anzeigen, ohne daß eine Aktion des Anwendungsprogramms stattfindet.

Das folgende Bild zeigt die Abläufe bei der Arbeit mit FHS-DM:



Im Anwendungsprogramm werden mit den Variablendiensten Dialogvariablen definiert. Um eine Maske auszugeben, wird der Display-Dienst des FHS-DM aufgerufen. Dieser überträgt die Daten eines Formates (konstante Texte und Inhalt der variablen Felder) und logischen Ausgabesteuerzeichen in einen Ausgabebereich. Die dabei entstandene Ausgabemessage im "extended line mode" wird mit VTSU-B am entsprechenden Gerät ausgegeben.

Eingabemessages , die bei FHS-DM eintreffen sind immer geräteunabhängig. Die Daten werden den entsprechenden Eingabefeldern zugeordnet. Es erfolgt die Auswertung des eingegebenen Kommandos und die Prüfung der Eingabedaten. Wurde eine Hilfe angefordert oder wurden fehlerhafte Daten eingegeben, so wird eine erneute Ausgabe gestartet. Sind Kommando und die Eingabedaten fehlerfrei, werden die Daten dem Variablenhandler übergeben und stehen damit der Anwendung zur Verfügung.

3 Einführung in die Dialogerweiterung

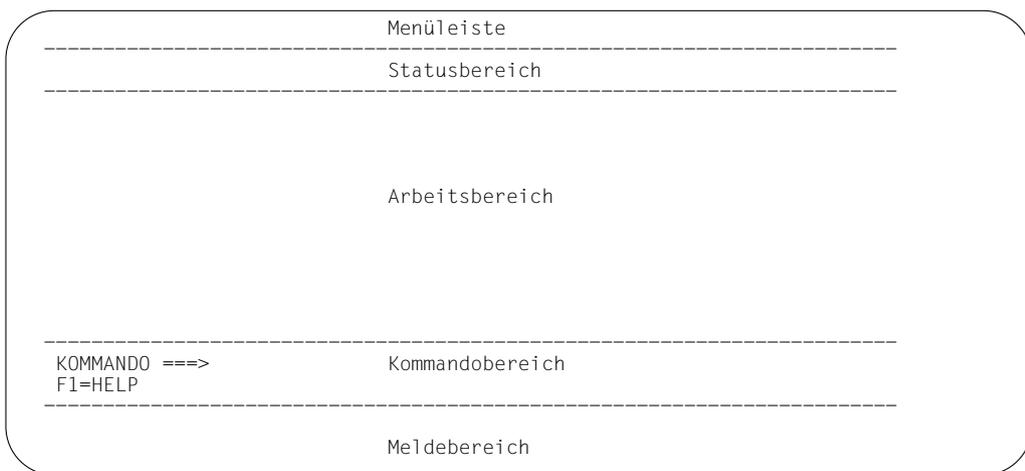
3.1 Aufbau von DE-Formaten für den FHS-Dialogmanager

DE-Formate müssen Sie mit dem IFG ab V8.1 erstellen. Um ein DE-Format zu erhalten, geben Sie im IFG ("Benutzerprofile/Aufbereitungseigenschaften der Formate") explizit den Schalter "Dialogerweiterung: Ja" an. Diese DE- Formate können Sie mit FHS-DE und mit FHS-DM für TIAM und für SDF-P nutzen

Für ein DE-Format gibt es zwei Darstellungsformen:

- als Vollformat, d.h. das Format nimmt die ganze Breite des Bildschirms ein.
- in einer Box, d.h. das Format nimmt sowohl in der Höhe als auch in der Breite nur einen Teil des Bildschirms ein, siehe Abschnitt "Dialog-Boxen" auf Seite 13ff .

Ein DE-Format besitzt zusätzliche Bereiche, die ausschließlich der Dialogführung dienen. Es besteht in der Regel aus fünf Bereichen und ist wie folgt aufgebaut:



Menüleiste

Die Menüleiste ist ein einzeiliger Bereich am oberen Bildschirmrand und wird durch eine Trennzeile begrenzt. Die Menüleiste enthält Menütitel. Jeder Menütitel steht für eine Gruppe von Auswahlmöglichkeiten, die als Pull-Down-Menü unterhalb des Menütitels angezeigt werden (siehe auch Seite 29). Eine Menüleiste ist nur im Vollbild (nicht in einer Dialogbox) erlaubt.

Statusbereich

Im Statusbereich steht (zentriert) der Titel des Formats. Am Zeilenanfang wird zusätzlich der Formatname (PANELID) ausgegeben, wenn dies mit dem FHS-Kommando "PANELID ON" verlangt wird, siehe Abschnitt "FHS-Kommandos". Der Titel ist wahlfrei und wird mit dem IFG festgelegt. Bei Formaten ohne Titel entfällt der Statusbereich.

Arbeitsbereich

Der Arbeitsbereich ist der eigentliche Aktionsbereich des Terminal-Benutzers (= Datenstationsbenutzer) und enthält Textfelder und variable Felder (wie bei den bisherigen FHS-Formaten). Zusätzlich sind bei FHS-DM Auswahlfelder und Listenausgaben möglich.

Kommandobereich

Der Kommandobereich enthält die Kommandozeile sowie die (ein oder zwei Zeilen lang) Anzeige der Funktionstasten-Belegung. Die Kommandozeile besteht aus einem Textfeld (im Beispiel "KOMMANDO ==>") und einem Eingabefeld für Kommandos, welches als *Kommandofeld* bezeichnet wird.

Meldebereich

In diesem Bereich werden Meldungen ausgegeben. Der Meldebereich darf nur für Vollformate definiert werden. Bei Boxen gibt es keinen Meldebereich. Umfangreiche Meldungen können auch in speziellen Boxen, genannt Meldungsboxen, ausgegeben werden, siehe Seite 18.

Von den genannten Bereichen ist nur der Arbeitsbereich zwingend vorgeschrieben, aus Kompatibilität zu Vorgängerversionen. Es wird jedoch dringend empfohlen, ein DE-Format immer mit allen zulässigen Bereichen zu definieren, damit der Terminal-Benutzer eine einheitliche und komfortable Bildschirmoberfläche vorfindet.

3.2 Datenaustausch mit Dialogvariablen

Dialogvariable dienen dem Datenaustausch zwischen Anwendungsprogramm, Dialogmanager und Dialogelementen. Dialogvariable übernehmen die Funktion eines Datenübergebereichs.

Von besonderer Bedeutung ist der Datenaustausch zwischen Maskenfeldern und Daten des Anwendungsprogramms. Bei der Formatdefinition mittels IFG werden den Ein- und Ausgabefeldern einer Maske Namen von Dialogvariablen zugeordnet. Im IFG können Sie im Bild "Angaben zur Anwendungsbibliothek" den entsprechenden Schalter auf "Y" setzen. In der selben Maske können Sie auch einen Syntax-Check für Dialogvariablen oder für SDF-P-Syntax einstellen.

Bei der Anzeige einer Maske durch den Dialogmanager werden in den Maskenfeldern die Werte der zugeordneten Dialogvariablen angezeigt. Bei der Eingabe von Daten in ein Eingabefeld der Maske werden diese als Wert der entsprechenden Dialogvariable gespeichert, siehe auch Abschnitt "Dialogvariable" ab Seite 98.

3.3 Dialog-Boxen

Mit FHS-DM können Sie Zwischendialoge realisieren, indem Sie das zugrundeliegende Format durch Dialog-Boxen überlagern, siehe auch Handbuch "Alpha-Styleguide". Die Zwischendialoge können mehrstufig sein, d.h. es können sich mehrere Boxen auf dem Bildschirm überlagern.

Eine Dialog-Box ist ein Rahmen in Form eines "Bildschirms im Bildschirm". Dieser Rahmen enthält wiederum ein Format, das nun nicht mehr den ganzen Bildschirm einnimmt. Die Formate, die in Boxen angezeigt werden, erzeugen Sie mit dem Formatgenerator IFG.

Eine Box kann folgendes enthalten:

- Ein Format mit Eingabe-, Ausgabe- und Auswahlfeldern
- Meldungen, siehe Abschnitt "Meldungsboxen"
- Hilfetexte, siehe Abschnitt "Hilfeboxen"

Es gibt Boxen, die nur zur Informationsausgabe dienen und keine Eingaben verlangen, z.B. bestimmte Hilfe- oder Meldungsboxen. Solche Boxen werden als *nichtmodale* Boxen bezeichnet. In die darunterliegenden Bereiche des Bildschirms sind weiterhin Eingaben möglich.

Modale Boxen sind Boxen, die eine Eingabe erwarten. Nach Ausgabe einer modalen Box sind alle anderen Teile des Bildschirms gegen Eingaben geschützt.

Mit FHS-DM können Boxen durch das Programm oder durch FHS ausgegeben werden. Bei Ausgabe durch das Programm spricht man von *expliziten* Boxen. Boxen, die durch FHS ausgegeben werden (ohne Beteiligung des Programms), bezeichnet man als *implizite* Boxen. Hilfetexte werden typischerweise als implizite Box ausgegeben, d.h. Sie können ein komplettes Hilfesystem einrichten, ohne die Anwendung zu belasten.

Nachstehend finden Sie ein einfaches Beispiel für ein Vollformat mit Dialogbox.

```

                                Adressverwaltung - Neuaufnahme
-----
Bitte tragen Sie die Adressdaten in die entsprechenden Felder ein.

Name: Mueller...----- Vorname: Hans.....
      : Berufsstatus      :
Strasse: Schille:      :
      : Bitte waehlen Sie: :
Postleitzahl: 79:  _  1. Angestellter : Ort: Stuttgart.....
      : 2. Arbeiter      :
Telefon: 0711 : 3. Selbstaendig :
      : 4. Ausbildung :
Berufsstatus: * : 5. Sonstiges : Familienstand: vh
      :
Kinderzahl: 01 : F1=HELP F12=ENDE : Konfession: ev
-----

Kommando:
F1=HELP F3=Beenden F12=Abbrechen
-----

```

In das Feld "Berufsstatus", das mit der Voreinstellung "*" am Bildschirm ausgegeben worden war, hatte der Terminal-Benutzer im ersten Dialogschritt nichts eingetragen. Das Programm des zweiten Dialogschritts interpretiert das Zeichen "*" entsprechend und gibt die Box mit dem Auswahlfeld "Berufsstatus" aus.

Eingaben in Dialog-Boxen

Befinden sich (wie im Beispiel) eine oder mehrere Boxen auf dem Bildschirm, dann sind Eingaben nur in die oberste modale Box möglich. Alle darunter liegenden modalen Boxen/Formate sind inaktiv, d.h. die Eingabefelder werden zu geschützten Feldern und dementsprechend am Bildschirm dargestellt. Eine darunterliegende Box wird erst wieder aktiviert, nachdem alle darüberliegenden Boxen entfernt wurden.

Ist die oberste Box nichtmodal, z.B. eine Hilfebox für ein bestimmtes Feld, dann sind Eingaben in die darunterliegende Box/Format möglich, sofern die gewünschten Felder nicht ganz oder teilweise verdeckt sind. Für die Eingabe in verdeckte Felder muß die Box zuvor entfernt werden.

Entfernen von Dialog-Boxen

Implizite Boxen werden mit den FHS-Kommandos CANCEL und EXIT entfernt. CANCEL entfernt die oberste implizite Box, EXIT entfernt alle impliziten Boxen, siehe auch Abschnitt "FHS-Kommandos" ab Seite 63.

Explizite Boxen können nur durch die Programme der Anwendung entfernt werden.

3.3.1 Explizite Boxen

Ein Anwendungsprogramm hat mit dem Dialogmanager die Möglichkeit, Masken in Dialogboxen auszugeben. Diese Boxen werden als explizite Boxen bezeichnet.

Die Position der Dialogbox auf dem Bildschirm legt das Anwendungsprogramm mit dem Dialogdienst ADDPOP fest. Die Position kann bezüglich der linken oberen Ecke der unterliegenden Maske oder bezüglich eines Feldnamens der unterliegenden Maske angegeben werden.

Die Größe der Box wird durch die Größe der anzuzeigenden Maske bestimmt. Dabei wird in einer Box um die Maske noch ein Rahmen gelegt. Die Breite und die maximale Höhe der Maske sind in der Formatbeschreibung enthalten. Sie ergeben sich aus der Zeilenanzahl und der maximalen Breite der festen Bereiche (Status-, Kommandobereich...) der Maske.

In einer expliziten Box muß eine von einem Anwender selbst erstellte Maske vollständig anzeigbar sein, da ein Verschieben des Arbeitsbereiches bei expliziten Boxen nicht unterstützt wird.

Wird die Position einer Box explizit durch Zeilen- und Spaltenangaben festgelegt, muß sie entsprechend ihrer Größe an der gewünschten Position anzeigbar sein.

Wird als Position einer Box ein Feldname angegeben, wird zunächst versucht die Box unterhalb des Feldes auszugeben. Reicht der Platz unter dem Feld nicht aus, wird die Box oberhalb und, falls dort nicht möglich, rechts vom Feld positioniert. Dabei wird versucht, das Bezugsfeld möglichst nicht durch die Box abzudecken.

Wenn eine Box durch das Anwendungsprogramm ausgegeben wird, werden die Felder der unterliegenden Maske inaktiv, d.h. die Eingabefelder dieser Maske werden zu geschützten Feldern. Farbliche oder andere Hervorhebungen werden zurückgestellt.

Eine durch das Anwendungsprogramm ausgegebene Box ist immer eine modale Box.

Erzeugen und Entfernen von expliziten Dialogboxen

Durch das Anwendungsprogramm erzeugte Dialogboxen dienen allgemein dazu, den Dialog im Hauptfenster (Vollbild) zu erweitern. Es gelten folgende allgemeine Regeln:

- Die Box ist immer modal, d.h. die unterliegende Maske ist für Eingaben gesperrt.
- Die Position der Box kann feldbezogen oder zum Beginn der unterliegenden Maske festgelegt werden (Offset Positionierung).

Das Anwendungsprogramm muß zunächst den ADDPOP-Dienst aufrufen, um eine Ausgabe in eine Box zu initialisieren. Danach ist der DISPLAY-Dienst aufzurufen, um eine Maske in der Box anzuzeigen. Die Größe der Box wird durch die Größe der anzuzeigenden Maske bestimmt (mittels IFG festgelegt). Die durch den ADDPOP-Aufruf initialisierte Maskenausgabe in einer Dialogbox bleibt für alle nachfolgenden DISPLAY-Aufrufe wirksam bis ein REMPOP-Aufruf erfolgt oder ein weiterer ADDPOP-Dienst aufgerufen wird.

Nachfolgende DISPLAY-Aufrufe zeigen Masken in einer Box an, für die die gleiche Startposition gilt. Die Ausgabe in eine Box wird durch den REMPOP-Dienst beendet. Das Entfernen der Box erfolgt jedoch erst mit dem nächsten DISPLAY-Aufruf. Dabei kann bei Angabe eines Formatnamens eine andere Maske oder durch Weglassen des Formatnamens die unterliegende Maske mit den ursprünglichen Feldinhalten wieder angezeigt werden. Mehrfache ADDPOP-Aufrufe mit jeweils nachfolgenden DISPLAY-Aufrufen erzeugen mehrere Boxen-Ebenen (Kaskaden).

Der wahlweise Operand ALL für einen REMPOP-Aufruf entfernt alle Boxen bis zum Hauptfenster. Ohne den Operanden ALL wird nur die jeweils oberste Box entfernt. Wird der Operand ALL nicht verwendet, müssen ADDPOP- und REMPOP-Aufrufe paarig sein.

Das folgende Beispiel demonstriert in vereinfachter Codierung die Anzeige einer Maske im Hauptfenster mit anschließender Anzeige von zwei Masken in einer Boxkaskade, das Entfernen der Boxen und die Wiederanzeige der Anfangsmaske im Hauptfenster.

```

DISPLAY PANEL(PRIM)
.
.
ADDPOP
    DISPLAY PANEL(BOX1)
    ADDPOP
        DISPLAY PANEL(BOX2)
REMPop ALL
DISPLAY

```

3.3.2 Implizite Boxen

Implizite Boxen werden durch FHS gesteuert und dienen zur Ausgabe von Meldungen und Hilfeformaten. Bei impliziten Boxen kennt das Programm weder den Namen noch die Felder des Formats.

Implizite Boxen werden von FHS als modale oder nichtmodale Boxen ausgegeben. Bei modalen Boxen wird das darunterliegende Format inaktiv, bei nichtmodalen Boxen sind Eingaben möglich, falls das betreffende Feld nicht überdeckt ist.

Position impliziter Boxen

Die Position impliziter Boxen wird durch FHS bestimmt. Boxen ohne Bezugspunkt gibt FHS in der Mitte des Bildschirms aus. Boxen mit Bezugspunkt (z.B. feldbezogene Hilfen oder Meldungen) versucht FHS zunächst mit der Standardverschiebung auszugeben, d.h. zwei Zeilen unterhalb und zwei Spalten rechts vom Bezugspunkt. Reicht der Platz nicht aus, dann versucht FHS die Ausgabe so, daß das Bezugsfeld vollständig sichtbar bleibt. Wird es jedoch ganz oder teilweise verdeckt, dann wird das Bezugsfeld auch bei nichtmodalen Boxen zum geschützten Feld.

3.3.3 Meldungsboxen

Eine Meldungsbox ist eine implizite Box und wird durch FHS erzeugt. Die Höhe der Meldungsbox hängt von der Länge des Meldungstextes ab. Wird durch die Aufbereitung die vorgegebene Maximalgröße der Box (sechs Zeilen mit je 56 Zeichen) überschritten, so wird der Meldungstext unaufbereitet in die Meldungsbox übernommen.

```

.....
: Meldungsschlüssel           : Meldungskennzeichen
: Meldungstext                :
:      .                      :
:      .                      : Meldungstext
:      .                      :
: Meldungstext                :
: ==>                        : Kommandozeile
: F1=Hilfe  F3=...           : Tastenbelegungsanzeige
.....
    
```

Der Text der Meldung ist hervorgehoben, z.B. hell, invers oder farbig, je nach Datenstation und Meldungstyp. Die anderen Bereiche der Box sind halbhell dargestellt. Die Kommandozeile wird nur für modale Meldungen erzeugt.

Eine Meldung wird entweder durch FHS (implizit) oder durch das Programm (explizit) ausgegeben, siehe Abschnitt "Meldungen ausgeben" ab Seite 84.

Bei der Erzeugung des Meldungstextes können Kennzeichnungen zur Aufbereitung der Meldung angegeben werden. Die Position der Hilfe- oder Meldungsbox wird durch FHS bestimmt. Bei einer feldbezogenen Hilfe- oder Meldungsbox wird die Box bei dem Feld positioniert, für das sie aktiviert wird. Die Ausgabe der Box erfolgt, abhängig vom verfügbaren Platz unterhalb, oberhalb oder neben dem Feld. Dabei wird versucht, das Feld nicht zu überdecken.

3.3.4 Hilfe-Boxen

Eine Hilfe-Box ist eine nichtmodale implizite Box. Die Größe anwendungseigener Hilfe-Boxen bestimmen Sie beim Erstellen mit dem IFG; sie können den ganzen Bildschirm überdecken.

Feldbezogene Hilfe-Boxen sollten kleiner sein. FHS gibt sie möglichst unterhalb des Feldes aus, siehe Abschnitt "Position impliziter Boxen". Bleibt das Feld dabei sichtbar, können Sie die Hilfe lesen und gleichzeitig das Feld ausfüllen.

Die Größe einer Hilfebox wird durch das Hilfeformat bestimmt. Bei der Definition des Hilfeformates können Sie festlegen, ob eine Hilfebox eine bestimmte Höhe haben soll, oder ob FHS-DM die Höhe der Box abhängig von deren Position vergrößern darf. Die definierte Höhe ist dann die Mindesthöhe der Hilfebox. Beim Vergrößern wird der Meldebereich des Bildes nicht überschrieben. Die Boxbreite wird nicht verändert.

Paßt ein Hilfetext nicht in eine Box, dann enthält die Box eine Blätterinformation, z.B. "Mehr: +". Mit dem Kommando "+" können Sie weiteren Text anfordern.

Eine Hilfe-Box erzeugen Sie mit dem IFG als DE-Format . Dort müssen Sie explizit angeben, daß das Format ein Hilfe-Format ist.

Weitere Informationen über Hilfen finden Sie ab Seite 88.

Das Makro kann mehrfach codiert werden, so daß für verschiedene Typen von Datensichtstationen verschiedene Rahmen definiert werden können. Die Angaben zu den Operanden DEV, DIM, CCSNAME und COLORED dienen zur Auswahl einer Rahmenbeschreibung. Beim Aufbau einer Box vergleicht FHS-DM die Eigenschaften der aktuellen Datensichtstation mit den Angaben zur Auswahl der Rahmenbeschreibung. Wird eine entsprechende Rahmenbeschreibung gefunden, wird diese zur Darstellung des Boxrahmens verwendet. Ansonsten wird das Standardlayout erzeugt.

Wenn eine Datensichtstation vom Typ 9763 mit Farbbildschirm für die Arbeit mit FHS-DM verwendet werden soll, ist bei der Definition der Rahmenbeschreibung folgendes zu beachten:

1. Bei Angabe eines ICE-Farb-Zeichensatzes muß im Operanden COLOR die Farbe weiß angegeben werden.
2. Die Farbdarstellung ist von der Geräteeinstellung abhängig (durch SIDATA). Es kann der Bildschirmhintergrund schwarz und die Zeichen in der angegebenen Farbe ausgegeben werden oder die Zeichen schwarz auf farbigen Hintergrund dargestellt werden. Im zweiten Fall wird der Boxrahmen in inversen Farben dargestellt (z.B. gelb wird zu blau, cyan wird rot). Aus diesem Grund sind im Lieferumfang zwei ICE-Farbzeichensätze für die Rahmendefinition enthalten. IDHTSD1C sollte bei schwarzen Hintergrund, IDHTSD1D bei farbigen Hintergrund verwendet werden.

Hinweis:

Es wird empfohlen halbhelle Zeichen in der Farbe cyan (SIDATA-Einstellung) darzustellen. Bei der Formatdefinition sollte für "HOLE COLOR" nicht die Farbe weiß angegeben werden.

3. Wenn ein monochromer ICE-Zeichensatz vereinbart wird, so wird der Boxrahmen schwarz-weiß angezeigt.

Für die Definition eines Zeichensatzes steht das SNI-Produkt "Interaktiver Zeichensatzgenerator" (ICE) zur Verfügung.

Format und Operandenbeschreibung des Makros IDHMBDR

Operation	Operanden
IDHMBDR	[BORDER=rahmenzeichen] [,NEXT=name [,DEV=geräteangabe] [,CCSNAME=ccsname] [,ICENAME=icename] nur für DSS 9763 1) [,DIM=24X80/27X132/32X80] nur für DSS 9763 [,COLORED=YES/NO/BOTH] nur für DSS 9763 [,COLOR=farbe] nur für DSS 9763

1) gilt auch für andere DSS die sich analog 9763 verhalten.

BORDER=rahmenzeichen

rahmenzeichen: 14 Zeichen in folgender Reihenfolge:

- 1 linke obere Ecke
- 2 rechte obere Ecke
- 3 linke untere Ecke
- 4 rechte untere Ecke
- 5 obere wagerechte Linie
- 6 untere wagerechte Linie
- 7 linke senkrechte Linie
- 8 rechte senkrechte Linie
- A Randzeichen linke obere Ecke
- B Randzeichen rechte obere Ecke
- C Randzeichen linke untere Ecke
- D Randzeichen rechte untere Ecke
- E Randzeichen linke senkrechte Linie
- F Randzeichen rechte senkrechte Linie

Standardmäßig wird hierfür C'.....: ' angenommen.

Wenn ein ICE-Zeichensatz angegeben wird, sind diese Zeichen der interne Code.

NEXT=name

Anzugeben, wenn weitere Makrodefinitionen folgen sollen. name ist der Name, der bei der nächsten Makrodefinition angegeben ist.

DEV=geräteangabe

Anzugeben, wenn die Rahmenzeichen an einen bestimmten Typ des Bildschirms gebunden werden sollen. Diese Angabe ist notwendig, wenn mehrere Rahmendefinitionen gewünscht werden. Bei der letzten Makrodefinition kann der Parameter fehlen.

9750 8160 9751 9752 9753 9754 9755 9763 3270	Typ der Station / des Geräts
XHCS	Standarddefinition für 8-Bit-Terminal (letzte 8-Bit-Definition)
DEF	Standard für alle Geräte (nur in letztem Makroaufruf zugelassen)

CCSNAME=ccsname

Nur für 8-Bit Terminal zu verwenden. Wenn ein Format mit angegebenen CCSNAMEN verarbeitet wird, ist der mit dieser Makrodefinition definierte Rahmen zu verwenden.

ICENAME=icename

Name eines ICE-Zeichensatzes (nur für 7-Bit-Formate)

DIM=24X80/27X132/32X80

Dimension des Bildschirmes (nur für DSS 9763)

COLORED=YES/NO/BOTH

Gilt für Farbschirm, Monochromschirm oder für beide Arten (nur für DSS 9763)

COLOR=farbe

farbe: BLUE, RED, MAGENTA, GREEN, CYAN, YELLOW, WHITE

Farbe der Rahmenzeichen wenn kein ICE-Zeichensatz angegeben ist.

Wenn ein ICE-Zeichensatz angegeben wird, muß COLOR=WHITE codiert werden (nur für DDS 9763)

Das unten angegebene Beispiel ist der Standard-Modul für die Rahmendefinition und befindet sich in der Standard-Formatbibliothek des Lieferumfanges.

Folgende ICE-Zeichensätze werden bereitgestellt:

- IDHTSD1B für DSS 9763 monochrom, Bildschirmdimension 24x80
- IDHTSD1C für DSS 9763 Farbschirm schwarzer Hintergrund, 24x80
- IDHTSD1D für DSS 9763 Farbschirm farbiger Hintergrund (cyan), 24x80
- IDHTSD4B für DSS 9763 monochrom, Bildschirmdimension 27x132

```

IDHBORD  START
IDHBORD  RMODE ANY
IDHBORD  AMODE ANY
                ENTRY IDHBORD

IDHBORD  CSECT
*
* Device 9763; color screen; screen dimension 24 x 80;
*      ICE-Format IDHTSD1C.
* T9763BC  IDHMBDR  NEXT=T9763B7,DEV=9763,ICENAME=IDHTSD1C,
*      BORDER=ABCDEFGHabcdef,COLOR=WHITE,COLORED=YES
*
* Device 9763; monochrome screen; screen dimension 24 x 80;
*      ICE-Format IDHTSD1B.
*
T9763B7  IDHMBDR  NEXT=T9763D4,DEV=9763,ICENAME=IDHTSD1B,
                BORDER=ABCDEFGHabcdef
*
* Device 9763; monochrome screen; screen dimension 27 x 132;
*      ICE-Format IDHTSD4B.
*
T9763D4  IDHMBDR  NEXT=T9758B8,DEV=9763,ICENAME=IDHTSD4B,
                BORDER=ABCDEFGHabcdef,DIM=27X132
*
* Device 9758; with CCS name
*      any CCS-name
*
T9758B8  IDHMBDR  NEXT=T9763B8,DEV=9755,CCSNAME=*ANY
*
* Device 9763; with CCS name; screen dimension 24 x 80;
*      any CCS-name      (no ICE)
*
T9763B8  IDHMBDR  NEXT=DEF,DEV=9763,CCSNAME=*ANY
*
* for all other devices
*      default: no ICE, no color, border signs '.....:      '
*
DEF      IDHMBDR  DEV=DEFAULT
END

```

3.4 Formate mit CCS-Namen

Bei der Erstellung der Formate mit IFG können Sie einem Format einen CCS-Namen (coded character set) zuordnen. Damit haben Sie die Möglichkeit, Ihren Zeichensatz zu ändern oder zu erweitern (siehe auch Abschnitt "Codetabellen" ab Seite 44). Die Nutzung setzt das SNI-Produkt XHCS (Extended Host Code Support) und 8-bit-Datensichtstationen voraus.

Wenn Formate als Box ausgegeben werden sollen, so erfolgt eine Prüfung auf Verträglichkeit der CCS-Namen. Dabei gilt für explizite und implizite Boxen folgender Grundsatz:

Enthält ein Format, das als Box angezeigt werden soll, einen CCS-Namen, so muß dieser CCS-Name bereits vom Vollbild genutzt werden. Ist in dem Format, das als Box angezeigt werden soll, kein CCS-Name angegeben, so kann es auf ein Bild mit beliebigen CCS-Namen ausgegeben werden. Der CCS-Name vererbt sich auf die Box.

3.5 Auswahlfelder

Mit einem Auswahlfeld kann der Terminal-Benutzer auf einfache Weise unter mehreren Möglichkeiten wählen. Es gibt zwei Arten von Auswahlfeldern:

- Einfachauswahlfeld: Unter mehreren Möglichkeiten wählt der Terminal-Benutzer *eine* Möglichkeit aus.
- Mehrfachauswahlfeld: Der Terminal-Benutzer kann *mehrere* Einträge auswählen.

Ein Auswahlfeld ist Teil des Arbeitsbereichs eines Formats und ist in der Regel mehrzeilig. Es besteht aus einer Überschrift, einer Reihe von Einträgen und einem oder mehreren Auswahl-Eingabefeldern.

Auswahlfelder erstellen Sie mit dem IFG. Sie können sowohl einem Auswahlfeld als auch jedem Eintrag ein Hilfeformat zuordnen (wird beim Erstellen mit dem IFG festgelegt).

3.5.1 Einfachauswahlfeld

Ein Einfachauswahlfeld besitzt immer *ein* Eingabefeld, in das der Terminal-Benutzer die Zeichen für die Auswahl einträgt.

In das Auswahl-eingabefeld ist der Wert der gewünschten Auswahl einzugeben. Bei Betätigen der ENTER-Taste wird die Auswahl aktiviert. Das Auswahl-eingabefeld ist ein oder zwei Zeichen lang, abhängig davon, wie das Format durch IFG erstellt wurde. Gemäß Alpha-Styleguide sollten für die Auswahl Ziffern verwendet werden: Bis zu 9 Auswahlmöglichkeiten eine Ziffer (siehe unten), ab 10 Auswahlmöglichkeiten zwei Ziffern. Auswahlmöglichkeiten, die in der aktuellen Dialogsituation nicht zur Verfügung stehen, können durch das

Anwendungsprogramm als für die Auswahl gesperrt gekennzeichnet werden (in der Maske dargestellt als Ausschlußzeichen). Bleibt das Auswahleingabefeld leer, erfolgt keine Auswahl.

Das Auswahleingabefeld kann durch einen Wert voreingestellt sein, wenn ein bestimmter Eintrag häufiger benötigt wird. Durch Eingabe einer Nummer oder eines Leerzeichens kann dieser Wert geändert werden.

Der Dialogmanager prüft, ob der eingegebene Wert eine zulässige Auswahl ist. Die eingegebene Wert wird als Dialogvariable an das Anwendungsprogramm zur Auswertung übergeben.

Beispiel für ein Einfachauswahlfeld:

```

          Berufsstatus
    Bitte waehlen Sie:
    - 1. Angestellter
      2. Arbeiter
      3. Selbststaendig
      *. Ausbildung
      5. Sonstiges

    Kommando:
    F1=HILFE F3=Beenden F12=Abbrechen
  
```

Die Auswahlmöglichkeiten sind in diesem Beispiel von 1 bis 5 numeriert. Der Benutzer trägt die entsprechende Ziffer in das Eingabefeld ein. Das Eingabefeld ist hier mit einem "_" (Unterstrich) gekennzeichnet. Um z.B. "Arbeiter" auszuwählen, ist in das Auswahleingabefeld eine 2 einzutragen. Die Auswahl 4 "Ausbildung" steht nicht zur Verfügung. Es kann eine Hilfe zu den Auswahlmöglichkeiten angefordert werden. Die Zeichen für die Auswahl legen Sie beim Erstellen mit dem IFG fest.

Bei der Definition eines Einfachauswahlfeldes mittels IFG muß jedem Auswahleingabefeld eine Dialogvariable zugeordnet werden. Dem Text für die Auswahlmöglichkeiten kann eine Dialogvariable zugeordnet werden. Die einer Auswahlmöglichkeit zugeordnete Dialogvariable dient zum Sperren des Auswahleintrages bei der Anzeige. Im folgenden wird die Dialogvariable für das Auswahleingabefeld SELVAR-Variable genannt und die für das Sperren als LOCK-Variable bezeichnet. Die LOCK-Variable ist eine Steuervariable, deren Name mittels IFG als LOCK-Steuervariable festgelegt wird. Der Name der SELVAR-Variablen wird bei der Formatdefinition als Feldname definiert.

Die LOCK-Variable sollte eine ein Byte lange einfache Dialogvariable vom Typ CHAR sein.

Wenn eine Auswahlmöglichkeit für den Terminal-Benutzer augenblicklich nicht zur Verfügung stehen soll, ist der Wert der entsprechenden Dialogvariablen auf X'FO' zu setzen. Eine solche Auswahlmöglichkeit wird bei Anzeige durch das Ausschlußzeichen "*" gekennzeichnet und ist für die Auswahl gesperrt. Eine Veränderung des Wertes der Dialogvariablen wird erst beim nächsten DISPLAY-Aufruf mit PANEL-Operand wirksam (siehe auch Beschreibung des DISPLAY-Dienstes).

Ist für eine Auswahlmöglichkeit keine LOCK-Variable beim Definieren des Formates zugeordnet oder existiert die entsprechende Dialogvariable nicht, ist kein Ausschluß möglich.

Externer und interner Wert:

Mittels IFG kann für das Auswahleingabefeld ein externer und wahlweise zusätzlich ein interner Wert vereinbart werden.

Der vom Terminal-Benutzer im Auswahleingabefeld eingegebene Wert wird dem Anwendungsprogramm in der dem Maskenfeld zugeordneten Dialogvariablen zur Verfügung gestellt. Wenn für den eingegebenen Wert (externer Wert) ein interner Wert existiert, wird statt des eingegebenen Wertes der interne Wert in die Dialogvariable übertragen.

Beim Aufruf des DISPLAY-Dienstes wird der Wert der SELVAR-Variablen zur Voreinstellung verwendet. Wenn ein interner Wert vereinbart wurde, ist der interne Wert anzugeben, angezeigt wird der externe Wert. Wenn der Wert der Dialogvariablen Null ist und Null als Vergleichswert nicht vorkommt, wird das Auswahleingabefeld mit Füllzeichen belegt. Besteht der Wert der Dialogvariablen aus Leerzeichen oder hat dieser die Länge Null, werden ebenfalls Füllzeichen ausgegeben.

3.5.2 Mehrfachauswahlfeld

Ein Mehrfachauswahlfeld besteht aus einer Überschrift (wahlweise) und mehreren Auswahleingabefeldern mit entsprechenden nicht numerierten Einträgen für die Auswahlmöglichkeiten. Vor jedem Eintrag steht ein Auswahleingabefeld. Der Benutzer wählt eine Möglichkeit aus, indem er das zugehörige Eingabefeld markiert.

Bei einem Mehrfachauswahlfeld wird durch den DISPLAY-Dienst geprüft, ob in den Eingabefeldern ein zulässiger Wert angegeben ist. Das kann eines der Auswahlzeichen oder ein Leerzeichen sein.

Beispiel für ein Mehrfachauswahlfeld:

```
Ueberstandene Kinderkrankheiten

Bitte tragen sie ein:
- Mumps
- Keuchhusten
7 Roeteln
- Scharlach
7 Masern
- TBC
* sonstige Krankheiten

Kommando:
F1=HILFE F3= Beenden F12=Abbrechen
```

Röteln und Masern sind überstandene Kinderkrankheiten. Sie wurden durch den Schrägstrich ausgewählt. "Sonstige Krankheiten" sind nicht auswählbar. Es kann eine Hilfe zu den Auswahlmöglichkeiten angefordert werden.

Bei einem Mehrfachauswahlfeld ist bei der Formatdefinition für jeden Auswahleintrag zwingend eine SELVAR-Variable mit unterschiedlichen Namen anzugeben (siehe auch Einfachauswahlfeld).

Die entsprechende einfache Dialogvariable sollte vom Typ CHAR und 1 Byte lang sein, deren Name mittels IFG als Feldname für das jeweilige Eingabefeld festgelegt wird. Sie kann vor der Anzeige der Maske folgende Werte enthalten:

binär null / Leerzeichen	auswählbar, nicht vorbelegt
"1"	Vorauswahl
"0"	Auswahl gesperrt

Für einen gesperrten Eintrag wird bei der Anzeige der Maske im Auswahleingabefeld das Ausschlußzeichen "*" ausgegeben. Dieses Eingabefeld wird dann zum geschützten Feld.

Für einen voreingestellten Eintrag wird bei der Anzeige der Maske im Auswahleingabefeld das Auswahlzeichen "/" ausgegeben.

Bei der Eingabe wird als Auswahlzeichen "/", "X" oder "x" durch den Dialogmanager akzeptiert (gemäß SNI-Alpha-Styleguide).

Bei der Rückkehr zum Anwendungsprogramm enthalten die SELVAR-Variablen der ausgewählten Einträge den Wert 1. Das Anwendungsprogramm muß darauf entsprechend reagieren.

3.6 Menüleiste und Pull-Down-Menüs

Formate können eine einzeilige Menüleiste enthalten. In dieser werden Menütitel angezeigt. Um einen Menütitel auszuwählen, positionieren Sie die Schreibmarke auf diesen Menütitel und drücken die Enter-Taste.

Datei	Projekt	Hilfe

:	_	1. Neuaufnahme : waltung - Anmeldung
:	:	2. Loeschen :
:	:	3. Drucken... :
Bitte t :	*	Ansehen : in die entsprechenden Felder ein.
:	::
Datum	:	02.12.1994
Projektname	:
Autor	:
Version	:
Programmiersystem	:
KOMMANDO ==>		
F1=Hilfe F3=Beenden F12=Abbrechen		

Positionieren des Cursors in die Menüleiste

Eine Möglichkeit den Cursor in die Menüleiste zu bringen, ist das Drücken der F-Taste, der das Systemkommando ACTIONS zugeordnet wurde (Standard: F10).

Die ACTIONS-Taste bewirkt eine Positionierung des Cursors durch FHS-DM. Dabei wird die augenblickliche Position des Cursors gespeichert und der Cursor wird auf das erste Zeichen des ersten Menütitels der Menüleiste positioniert. Bei der nächsten Betätigung der ACTIONS-Taste wird der Cursor auf die gespeicherte Ausgangsposition zurückgestellt.

Eine andere Möglichkeit ist die Benutzung der Cursortasten. Da diese Funktion durch die Hardware realisiert wird, ist in diesem Fall ein Speichern der Cursorposition durch FHS-DM nicht möglich.

Auswählen eines Menütitels

Innerhalb der Menüleiste wählen Sie einen Menütitel aus, indem Sie den Cursor mit der Tabulatortaste oder den Cursortasten auf ein Zeichen des gewünschten Menütitels positionieren und die Eingabetaste drücken. Dann wird für diesen Menütitel ein Pull-Down-Menü sichtbar. In diesem wird die vorhandene Auswahlmöglichkeit angezeigt. Während das Pull-

Down-Menü angezeigt wird, sind alle Felder der unterliegenden Maske für Eingaben gesperrt und der Cursor wird in das Auswahleingabefeld des Menüs gestellt. Markieren oder Eingaben in einen Menütitel werden ignoriert, sie führen nicht zur Auswahl eines Pull-Down-Menüs.

Anzeige eines anderen Pull-Down-Menüs

Sie können in ein anderes Pull-Down-Menü wechseln, indem Sie mit den Cursortasten einen anderen Menütitel auswählen.

Eine Rückkehr zur Menüleiste bei angezeigtem Pull-Down-Menü ist auch durch Drücken der CANCEL-Taste möglich. Dabei wird das Pull-Down-Menü gelöscht. Eine im Pull-Down-Menü eingegebene Auswahl wird ignoriert. Der Cursor wird auf den ersten Menütitel positioniert. Danach können Sie einen anderen Menütitel auswählen.

Abbrechen einer Pull-Down-Anzeige und Verlassen der Menüleiste

Wenn mit der ACTIONS-Taste in die Menüleiste positioniert wurde, können Sie durch erneutes Drücken der ACTIONS-Taste die Anzeige eines Pull-Down-Menüs abbrechen. Das Pull-Down-Menü wird gelöscht und der Cursor wird auf die gespeicherte Position im Arbeitsbereich zurückgestellt. Eine im Pull-Down-Menü eingegebene Auswahl wird ignoriert. Mit der CANCEL-Taste kann die Anzeige eines Pull-Down-Menüs ebenfalls abgebrochen werden. Der Cursor wird in diesem Fall auf den ersten Menütitel der Menüleiste positioniert. Die Menüleiste verlassen Sie dann mit den Cursortasten.

Achtung:

Falls Sie sich eine eigene Tastenbelegungstabelle erzeugen, müssen Sie für ein Format mit Menüleiste eine ACTIONS- oder CANCEL-Taste definieren!

Begründung: Einem Pull-Down-Menü ist keine eigene Tastenbelegung zugeordnet und bei angezeigtem Pull-Down-Menü kann kein Kommando im Kommandoeingabefeld eingegeben werden. Das heißt ohne eine ACTIONS- oder CANCEL-Taste ist ein Beenden des Pull-Down-Menüs nicht möglich.

Arbeiten mit Pull-Down-Menüs

Pull-Down-Menüs sind nur im Vollbild (gesamter Bildschirm) möglich. Die Menüleiste und die Pull-Down-Menüs werden mit dem IFG erstellt. Menütitel in der Menüleiste sind markierbare Eingabefelder. Durch Eingabe einer Auswahl in das Auswahleingabefeld oder Verwendung des vorbelegten Wertes und Drücken der Eingabetaste, während sich der Cursor im Pull-Down-Menü befindet, wird die Auswahl aktiviert. Eingaben in und Markierungen von Menütiteln werden ignoriert, die Auswahlen erfolgen nur aufgrund der Cursorposition.

In die Eingabefelder werden die Eingaben aus den Pull-Down-Menüs eingetragen. Diese Eintragungen erfolgen nicht, wenn nach der Eingabe in einem Pull-Down-Menü das ACTIONS-Kommando gegeben wird oder ein anderes Pull-Down-Menü ausgewählt wird.

In Pull-Down-Menüs kann je ein Einfachauswahlfeld definiert werden. Dieses kann vorbelegt werden und Auswahleinträge können gesperrt werden. Es können interne Werte vereinbart werden (siehe auch Seite 27).

Verarbeitung einer Menüleiste

Bei der Definition einer Menüleiste mit Hilfe des IFG muß für jeden Menütitel ein Name einer einfachen Dialogvariablen zugeordnet werden, im folgenden CHECK-Variable genannt. Außerdem ist jedem Menütitel ein Einfachauswahlfeld zuzuordnen, das den Inhalt des Pull-down-Menüs bildet. Für die Definition des Auswahlfeldes eines Menütitels gelten die gleichen Bedingungen wie für ein Auswahlfeld.

Für die Verarbeitung der Menüleiste durch das Anwendungsprogramm kann man sich ein aktivierte Pulldown-Box wie ein zusätzliches Auswahlfeld des Formates vorstellen. Zur Erkennung welcher Menütitel ausgewählt wurde, dient die CHECK-Variable. Die CHECK-Variable sollte ein Bytes lang und vom Typ CHAR sein. Wenn die CHECK-Variable nicht existiert, wird sie implizit mit dem Typ CHAR angelegt.

Vor dem Aufruf des DISPLAY-Dienstes sind alle CHECK-Variablen durch das Anwendungsprogramm auf den Wert 0 (C"0") zu setzen.

Nach der Rückkehr vom DISPLAY-Aufruf enthält die CHECK-Variable des ausgewählten Titels den Wert 1 (C"1").

Die Verarbeitung des in der Pulldown-Box ausgewählten Eintrages erfolgt in gleicher Weise wie für ein übliches Einfachauswahlfeld (siehe Einfachauswahlfeld).

3.7 Listenverarbeitung

Definition eines Listenbereiches

Die Listenanzeige gibt ein oder mehrere Arrays von Dialogvariablen als Liste in einem Listenbereich aus. Ein Listenbereich ist ein blätterbarer Teil des Arbeitsbereiches eines Formates.

Die Realisierung der Listenanzeige erfolgt mit dem DISPLAY-Dienst. Das anzuzeigende Format muß einen Listenbereich enthalten.

Ein Format darf nur einen Listenbereich enthalten. Links und rechts des Listenbereiches dürfen im Arbeitsbereich nur Leerzeichen stehen.

Ein Listenbereich besteht aus folgenden Elementen, die mit dem IFG definiert werden:

- Listentitel mit Listenposition und Verschiebeinformation
- Spaltentitel
- Listenzeilen, Listenfelder
- Endezeile

Listenzeile

Eine Listenzeile wird beim Definieren eines Listenbereiches als Modellzeile vereinbart. Die Modellzeile kann aus Ein-, Ausgabe- und Textfeldern bestehen. Jeweils ein Eingabe-, ein Ausgabe- oder ein Textfeld der Modellzeile bilden ein Listenfeld. Alle Listenzeilen werden durch den DISPLAY-Dienst entsprechend der Modellzeile angezeigt.

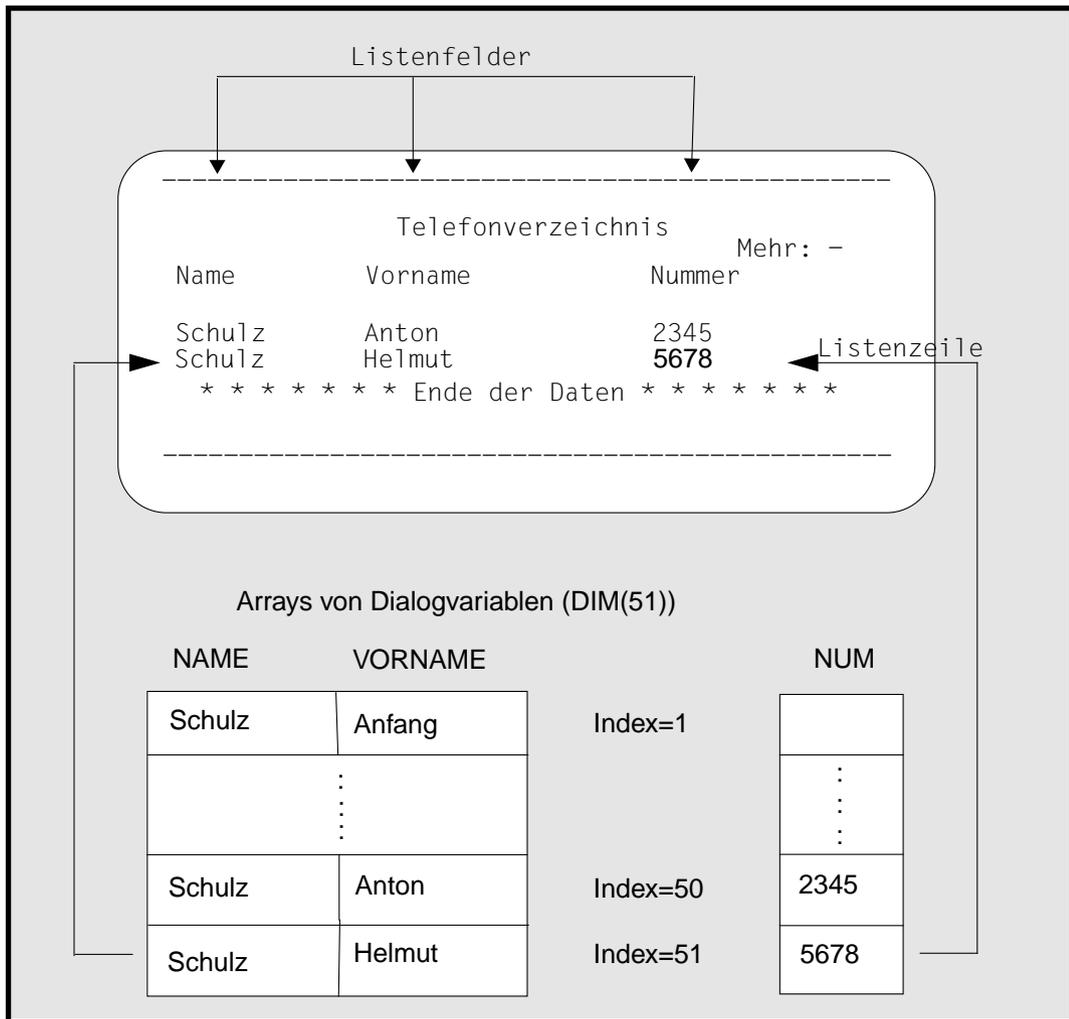
Wird einem Eingabe- oder Ausgabe-Listenfeld der Name einer Dialogvariablen, die zu einem Array von Dialogvariablen gehört, zugeordnet, bilden alle Werte einer solchen Dialogvariablen ein Listenfeld für alle Listenzeilen (siehe auch Bild nächste Seite). Die Dialogvariablen mit dem gleichen Index liefern die Daten für eine Listenzeile (siehe auch Abschnitt "Arten von Dialogvariablen"). In der Modellzeile werden die Namen der Dialogvariablen ohne Indexangabe (z.B. #3) vereinbart.

In der Modellzeile gelten für die Dialogvariablen eines TIAM- bzw. SDF-P-Anwendungsprogramms bestimmte Konventionen die auf den Seiten 108 (TIAM) und 187 (SDF-P) beschrieben sind.

Wird einem Ausgabefeld in der Modellzeile eine einfache Dialogvariable zugeordnet, hat dieses Feld in jeder Listenzeile den gleichen Wert.

Textfelder haben in jeder Listenzeile denselben Wert wie in der Modellzeile. Eingabefelder können vom Terminal-Benutzer der Anwendung mittels Tastatureingaben geändert werden. Die Werte der geänderten Eingabefelder stehen beim Beenden des DISPLAY-Dienstes in den entsprechenden Dialogvariablen.

Die folgende Darstellung zeigt die Beziehung zwischen einem Array von Dialogvariablen und einem Listenfeld einer Listenzeile. NAME, VORNAME und NUM sind die Namen der Dialogvariablen, die in der Modellzeile für die Listenfelder vereinbart wurden. Die Indexangabe bezieht sich immer auf das Arrayelement, obwohl dafür im folgenden auch die Bezeichnung Zeilennummer verwendet wird.



Jede Listenzeile kann eine Zeile des Listenbereiches umfassen oder in mehrere Zeilen des Listenbereiches aufgeteilt sein (mehrzeilige Listenzeile). Zwischen den Listenzeilen kann bei der Definition der Modellzeile eine Trennzeile vereinbart werden. Die Trennzeile wird durch die Listenanzeige jeweils nach den zusammengehörenden Angaben angezeigt.

Folgende Beispiele zeigen die Möglichkeiten der Darstellung von Listenzeilen.

Beispiel für einfache Listenzeilen:

Name	Vorname	Telefon	PLZ	Ort
Mustermann	Hans	4567	01234	Musterdorf
Musterfrau	Frieda	5678	01234	Musterdorf

Beispiel für mehrzeilige Listenzeilen:

Name	Mustermann	Vorname	: Hans
Telefon	: 4567		
PLZ	: 01234	Ort	: Musterdorf

Name	: Musterfrau	Vorname	: Frieda
Telefon	: 5678		
PLZ	: 01234	Ort	: Musterdorf

Blättern in einem Listenbereich

Wenn die Liste soviel Zeilen hat, daß diese nicht alle gleichzeitig im Listenbereich darstellbar sind, können durch Verschiebekommandos die nicht sichtbaren Teile angezeigt werden.

Ein Verschieben kann durch Eingabe eines Kommandos im Kommandoingabebereich oder durch das Betätigen einer entsprechenden Funktionstaste veranlaßt werden. Der DISPLAY-Dienst unterstützt ein Verschieben nach oben und unten. Folgende Kommandos stehen dafür zur Verfügung:

FORWARD [zahl] / +[zahl]	Verschieben in Richtung Listenende
BACKWARD [zahl] / -[zahl]	Verschieben in Richtung Listenanfang
++	Verschieben an das Listenende
--	Verschieben an den Listenanfang

Für die Nutzung dieser Kommandos gibt es folgende Möglichkeiten:

- der Cursor befindet sich nicht in einer Listenzeile
 - FORWARD/+ unterste Listenzeile wird oberste
 - FORWARD *zahl* bezogen auf die oberste Listenzeile wird *zahl* Zeilen in Richtung Listenende verschoben
 - +*zahl* bezogen auf die oberste Listenzeile wird *zahl* Zeilen in Richtung Listenende verschoben
 - BACKWARD oberste Listenzeile wird unterste
 - BACKWARD *zahl* bezogen auf die oberste Listenzeile wird *zahl* Zeilen in Richtung Listenanfang verschoben
 - zahl* bezogen auf die oberste Listenzeile wird *zahl* Zeilen in Richtung Listenanfang verschoben

- der Cursor befindet sich in einer Listenzeile
 - FORWARD/+ die durch den Cursor markierte Listenzeile wird oberste Zeile
 - BACKWARD/- die durch den Cursor markierte Listenzeile wird unterste Zeile

Hinweis: Wenn sich der Cursor in der ersten oder letzten Listenzeile befindet, erfolgt das Verschieben so, als ob sich der Cursor nicht in einer Listenzeile befindet.

Die Werte der modifizierten Listenfelder werden vor Ausführung eines Verschiebekommandos geprüft und in die entsprechenden Dialog-variablen gespeichert.

Maskenfelder, die nicht zum Listenbereich gehören, werden erst geprüft, wenn keine Verschiebekommandos mehr anliegen. Außerdem werden diese Feldinhalte erst dann in die Dialogvariablen gespeichert, wenn zur Anwendung zurückgegangen wird. Das Kommando CANCEL bewirkt, daß die Inhalte der Nicht-Listenfelder auch nicht in die Dialogvariablen gespeichert werden.

Sonstige Definitionen für eine Liste

Zur Steuerung der Listenanzeige dienen spezielle Dialogvariable (Steuervariable), die mittels IFG (Feld Steuervariable) bei der Formaterstellung mit der Liste verbunden werden. Die Namen dieser Dialogvariablen sind wahlfrei und werden zum Zeitpunkt der Formatdefinition festgelegt.

Die im folgenden verwendete Bezeichnung einer Steuervariablen dient nur der Kennzeichnung, die Namen der Variablen sind ansonsten wahlfrei. Diese Dialogvariablen dienen folgenden Zwecken:

- NUMROW-Variable

Eine einfache explizite oder implizite Dialogvariable vorzugsweise vom Typ FIXEDS(TIAM) / INTEGER(SDF-P). Der Wert legt den aktuellen maximalen Index für ein Array von Dialogvariablen fest. Der Wert muß kleiner oder gleich dem Vielfachfaktor (DIM-Operand) der einem Listenfeld zugeordneten Dialogvariablen sein. Ist ihr Wert größer als der Vielfachfaktor, wird der Wert des kleinsten Vielfachfaktors verwendet. Durch diese Angabe kann das Anwendungsprogramm den für die Dialogvariablen der Listenfelder durch VDEFINE vereinbarten Platz nur teilweise füllen bzw. eine SDF-P-Liste kann nur teilweise bearbeitet werden.

Ist bei der Formatdefinition keine NUMROW-Variable angegeben oder existiert die vereinbarte Dialogvariable nicht, wird hierfür der kleinste Wert des Vielfachfaktors der den Listenfeldern zugeordneten Dialogvariablen verwendet.

- TOPINDEX-Variable

Gibt den Index (Zeilennummer) der Dialogvariablen an, die im Listenbereich als erste Zeile angezeigt werden soll.

Die TOPINDEX-Variable kann eine einfache implizite oder explizite Dialogvariable sein (eine explizite Dialogvariable sollte vorzugsweise vom Typ FIXEDS / INTEGER sein). Ihr Wert muß eine positive ganze Zahl sein und darf nicht größer als 32767 sein. Ein ungültiger Wert führt zu einem Fehler. Wenn der Wert für TOPINDEX größer ist als der Wert für NUMROW oder als der Vielfachfaktor (bei mehreren Arrays von Listendaten der kleinste Vielfachfaktor), wird als TOPINDEX der kleinere Wert dieser Angaben verwendet.

Regeln für TOPINDEX:

- Der DISPLAY-Dienst bzw. der ADDPOP-Dienst wird mit einem Fehlercode beendet, wenn die TOPINDEX-Variable existiert und ein Index angegeben wird, der sich nicht im Listenbereich anzeigen läßt. unter den nachfolgend beschriebenen Bedingungen unter Beachtung des Wertes der TOPINDEX-Variablen

Wenn z.B. TOPINDEX den Wert 11 hat und der DISPLAY-Dienst im MSGLOC-Operanden mit einer Indexangabe 15 aufgerufen wird (z.B. MSGLOC(FELD3#15)), wird die Meldungsbox an die 5. Listenzeile im Listenbereich plaziert. Können maximal 10 Listenzeilen im Listenbereich angezeigt werden, darf im obigen Beispiel die Indexangabe für den MSGLOC-Operanden die Werte 11 bis 21 haben.

- Ist der TOPINDEX Null, wird bei einem DISPLAY-Aufruf mit PANEL-Operand der Index 1 für die erste Zeile angenommen. Ohne PANEL-Operand wird TOPINDEX auf den Wert der ersten Zeile im aktuellen Listenbereich als TOPINDEX gesetzt.
- Ist bei der Formatdefinition keine TOPINDEX-Variable angegeben oder existiert die vereinbarte Dialogvariable nicht, wird hierfür bei Beginn der Listenanzeige der Wert 1 angenommen. Wenn keine TOPINDEX-Variable bei der Formatdefinition zugeordnet wurde, kann der Dialogmanager den Index der ersten Zeile im Listenbereich nicht an das Anwendungsprogramm zurückliefern. (Sie wird nicht automatisch angelegt).

Wenn in diesem Fall durch den CURSOR- oder MSGLOC-Operanden beim DISPLAY-Aufruf mit PANEL-Operand oder durch den POPLOC-Operanden beim ADDPOP-Aufruf ein Index für eine Listenzeile angegeben wird, wird ein automatisches Blättern durchgeführt,

- Bei einem DISPLAY-Aufruf ohne PANEL-Operand für eine Liste, wird der Wert der TOPINDEX-Variablen nicht berücksichtigt. Es wird der Listenbereich so dargestellt, wie er beim vorherigen DISPLAY-Aufruf verlassen wurde. Widersprüche mit einer Indexangabe im CURSOR- oder MSGLOC-Operanden führen nicht zum Fehler. Durch den Dialogmanager wird in diesem Fall eine darstellbare Form gewählt.

- MODINDEX-Variable

Ein Array von Dialogvariablen vom Typ FIXEDS (2 oder 4 Bytes), das die Indizes der Dialogvariablen derjenigen Listenzeilen enthält, die modifiziert wurden.

Normalerweise wird vor dem Aufruf des DISPLAY-Dienstes das erste Element den Wert 0 enthalten. Nach der Rückkehr vom DISPLAY-Dienst enthalten die Elemente Indizes der geänderten Listenzeilen (die erste Zeile der Liste hat den Index 1). Die Reihenfolge der Indizes endet mit dem ersten Element, das eine 0 enthält oder am Ende des Arrays.

Wenn zum Beispiel bei der Listenanzeige die Zeilen mit den Indizes 1, 3 und 51 durch den Anwender modifiziert wurden, enthält bei der Rückkehr vom DISPLAY-Dienst das erste Element der MODINDEX-Variablen den Wert 1, das zweite Element den Wert 3 das dritte Element den Wert 51 und das vierte Element den Wert 0 (zur Endekennzeichnung). Die Dimension der MODINDEX-Variable bestimmt die maximale Zeilenanzahl, die als modifiziert vom Dialogmanager gemeldet werden kann. Das Anwendungsprogramm kann die MODINDEX-Variable mit Indizes vorbelegen und dann den DISPLAY-Dienst aufrufen. In diesem Fall werden die entsprechenden Listenzeilen als modifiziert voreingestellt. In Verbindung mit der Angabe "Eingabepflicht" für ein Listenfeld (mittels IFG vereinbart) kann durch das Anwendungsprogramm eine Eingabe für ein Listenfeld erzwungen werden (siehe auch Prüfung auf Eingabepflicht für Listenfelder).

Die Werte der MODINDEX-Variablen müssen beim Aufruf des DISPLAY-Dienstes nicht in sortierter Reihenfolge gespeichert sein. Bei der Rückkehr vom DISPLAY-Aufruf sind die Indizes immer aufsteigend sortiert.

Ist bei der Formatdefinition keine MODINDEX-Variable angegeben oder existiert die vereinbarte Dialogvariable nicht, kann die damit verbundene Funktion nicht verwendet werden.

Aufruf der Listenanzeige

Die Listenanzeige erfolgt durch einen Aufruf des DISPLAY-Dienstes. In der Wirkungsweise unterscheiden sich ein DISPLAY-Aufruf mit PANEL-Operand und ein DISPLAY-Aufruf ohne PANEL-Operand.

DISPLAY-Aufruf mit PANEL-Operand:

Vom DISPLAY-Dienst werden folgende Werte bei der Anzeige einer Liste im Listenbereich verwendet:

- Die TOPINDEX-Variable bestimmt den Index, den die Elemente der Listenfelder haben, deren Werte in der ersten Zeile des Listenbereiches stehen.
Ist der Wert des Indexes ungültig (zu groß oder negativ) wird der DISPLAY-Dienst mit einem Fehlercode beendet.
- Die weiteren Zeilen des Listenbereiches enthalten die Werte der Elemente mit dem um jeweils 1 erhöhten Wert des Top-Indexes. Existieren keine weiteren Elemente, wird, falls angegeben, die Endezeile angezeigt. Der eventuell noch vorhandene Platz im Listenbereich bleibt leer.
- Die Operanden CURSOR und CSRPOS des DISPLAY-Dienstes bestimmen die Position des Cursors in der angezeigten Maske.
Gibt der CURSOR-Operand den Namen einer Dialogvariablen an, die mit einem Listenfeld verbunden ist, wird der Cursor in dieses Feld positioniert. In welche Zeile des Listenbereiches er dabei positioniert wird, kann durch eine Indexangabe zum Feldnamen festgelegt werden (z.B CURSOR(FELD#3)). Ist keine Indexangabe vorhanden, wird die oberste Listenzeile im Listenbereich angenommen. Der Operand CSRPOS bestimmt die Position des Cursors innerhalb des Listenfeldes.

DISPLAY-Aufruf ohne PANEL-Operand:

Bei einem DISPLAY-Aufruf ohne PANEL-Operand wird der Wert der TOPINDEX-Variablen nicht berücksichtigt. Es wird der Listenbereich so dargestellt, wie er beim vorherigen DISPLAY-Aufruf verlassen wurde. Widersprüche mit einer Indexangabe im CURSOR- oder MSGLOC-Operanden führen nicht zum Fehler. Durch den Dialogmanager wird in diesem Fall eine darstellbare Form gewählt.

Prüfen von Listenfeldern

Für Eingabe-Listenfelder kann bei der Formaterstellung mittels IFG eine Prüfung der Eingabedaten vorgesehen werden. Wird eine Prüfung für ein Listenfeld vereinbart, erfolgt die Prüfung für jedes modifizierte Eingabefeld in jeder Listenzeile.

Die Prüfungen werden durch den Dialogmanager bei der Beendigung der Verarbeitung des Formates (nicht beim Kommando CANCEL) und bei Vorliegen eines Verschiebekommandos ausgeführt (Blättern des Listenbereiches).

Bei einem erkannten Fehler wird eine Meldung ausgegeben. Nach der Ausgabe der Meldung kann der Terminal-Benutzer seine Eingaben korrigieren. Das Verschiebekommando oder das Kommando, das zur Beendigung der Verarbeitung der Maske führen sollte, muß noch einmal gegeben werden. Eine Besonderheit gilt für die Prüfung "Eingabepflicht" für ein Listenfeld.

Prüfung auf Eingabepflicht für Listenfelder

Eine Prüfung auf Eingabepflicht wird nur für modifizierte Listenzeilen durchgeführt. Eine Listenzeile gilt als modifiziert, wenn mindestens ein Eingabefeld der Listenzeile durch den Terminal-Benutzer mittels Tastatureingabe modifiziert wurde oder wenn diese Zeile mit Hilfe der MODINDEX-Variablen als modifiziert voreingestellt wurde.

Wenn z.B. eine Liste ausgegeben werden soll, in der der Terminal-Benutzer ein bestimmtes Listenfeld zwingend mit Daten versorgen soll, kann der Anwendungsprogrammierer wie folgt verfahren:

Bei der Formatdefinition wird für dieses Feld in der Modellzeile die Prüfung "Eingabepflicht" vereinbart. Beim Aufruf des DISPLAY-Dienstes wird die MODINDEX-Variable mit allen Indizes des Dialogvariablenarrays (von 1 bis max. Index) versorgt. Die Listenanzeige realisiert dies wie folgt: Wenn die Liste angezeigt wird, kann der Terminal-Benutzer beginnen, die fehlenden Daten in den Listenzeilen zu ergänzen. Er kann dabei blättern (vorwärts/rückwärts), ohne daß die Eingabe von noch nicht ausgefüllten Feldern in andern Zeilen angefordert wird. Die im Listenbereich modifizierten Daten werden vor dem Blättern in die entsprechenden Dialogvariablen geschrieben. Erst bei Beenden der Anzeige (ENTER-Taste betätigt), wird durch den Dialogmanager überprüft, ob in allen als modifiziert voreingestellten Zeilen die Felder mit Eingabepflicht modifiziert wurden. Existieren noch nicht modifizierte Felder, wird durch den Dialogmanager eine Meldung ausgegeben, die den Terminal-Benutzer anweist, fehlende Daten einzugeben. Die erste Zeile, für die noch Daten erforderlich sind, wird als erste Zeile im Listenbereich angezeigt. Ein Blättern ist möglich. Die Prüfung auf Erfüllung der Eingabepflicht erfolgt, wie bereits beschrieben, beim Beenden der Anzeige (jedoch nicht beim Kommando CANCEL).

Beenden einer Listenanzeige

Eine Listenanzeige wird dann beendet, wenn die Verarbeitung der Maske beendet wird, das den Listenbereich enthält.

Die Verarbeitung der Maske wird bei Ausführung eines Kommandos zum Verschieben der Listenzeilen nicht beendet.

Rückkehr von der Listenanzeige:

Vom DISPLAY-Dienst werden folgende Werte beim Beenden der Verarbeitung einer Maske mit einem Listenbereich bereitgestellt:

- Die TOPINDEX-Variable liefert den Index, den die Dialogvariablen haben, die in der obersten Zeile des Listenbereiches stehen. Der Wert kann gegenüber dem beim DISPLAY-Aufruf übergebenen Wert durch Verschiebekommandos verändert worden sein.
- Die Systemvariablen SYS-CURSOR-FIELD, SYS-CURSOR-INDEX und SYS-CURSOR-POS enthalten die Position des Cursors. Der Wert von SYS-CURSOR-INDEX ist nur dann ungleich Null, wenn SYS-CURSOR-FIELD den Namen eines Listenfeldes enthält. Der Wert ist der Index der Dialogvariablen, die mit diesem Feld verbunden ist.
- In den Elementen der MODINDEX-Variablen stehen die Indizes der Listenzeilen, in denen eine Eingabe vorgenommen wurde oder die als voreingestellt eingetragen wurden. Die Folge dieser Indizes endet mit dem Wert 0 oder beim letzten Element der MODINDEX-Variablen. Sie sind aufsteigend sortiert. Wenn keine Eingaben erfolgt sind, hat bereits das erste Element den Wert 0.
- In den Dialogvariablen sind die Werte der modifizierten Listenfelder eingetragen.

3.8 Datenaufbereitung und Prüfung

Der Dialogmanager kann die Feldinhalte einer Maske nach bestimmten Vorgaben aufbereiten und prüfen. Wie die Datenaufbereitung und Prüfung erfolgen soll, wird bei der Formatdefinition mit dem IFG festgelegt (siehe IFG-Handbuch).

Bei der Ausgabe des Formates (Aufruf des DISPLAY-Dienstes) werden die Daten der zum jeweiligen Maskenfeld gehörenden Dialogvariablen gelesen und in die externe Darstellung umgewandelt. Dabei wird die für das jeweilige Maskenfeld vereinbarte Datenaufbereitung berücksichtigt. Außerdem erfolgt eine Prüfung der Ausgabedaten entsprechend den für das Maskenfeld festgelegten Eigenschaften (z.B. arithmetisches Feld). Wird hierbei ein Fehler festgestellt (z.B. Unverträglichkeit des Datentypes des Maskenfeldes mit dem Datentyp der Dialogvariablen), ist es sinnvoll eine Fehlermaske auszugeben, bevor die Rückkehr zum Anwendungsprogramm mit einem entsprechenden Fehlercode erfolgt.

Für die Umwandlung von der internen Form in die externe Darstellung (und auch umgekehrt) ist durch den Anwendungsprogrammierer zu berücksichtigen, daß jede Dialogvariable einen Datentyp besitzt (siehe auch VDEFINE-Dienst). Andererseits werden auch für ein Maskenfeld Eigenschaften bei der Formatdefinition festgelegt. Es ist auf Verträglichkeit des Datentypes der Dialogvariablen und der Eigenschaften des Maskenfeldes zu achten.

Bei der Eingabe der Maske (der Terminal-Benutzer betätigt die Eingabe-Taste) oder bei Vorliegen eines Verschiebekommandos für einen Listenbereich wird der Inhalt der Maskenfelder entsprechend den für das jeweilige Maskenfeld vereinbarten Eigenschaften geprüft. Wird dabei ein Fehler festgestellt, gibt der Dialogmanager eine Meldebox mit einer entsprechenden Meldung aus. Der Bediener muß danach den betroffenen Feldinhalt korrigieren. Die Rückkehr zum Anwendungsprogramm erfolgt erst, wenn korrekte Daten eingegeben wurden oder wenn die Anzeige abgebrochen wird. Die Eingabedaten werden von der externen Darstellung in die interne Form umgewandelt. Bei der Rückkehr zum Anwendungsprogramm stehen die Eingabedaten in der zum Maskenfeld gehörenden Dialogvariablen.

Wird die Anzeige mit dem Kommando CANCEL abgebrochen, erfolgt keine Prüfung der Eingabedaten.

Im folgenden wird dargestellt, welche Eigenschaften für Maskenfelder durch den Dialogmanager unterstützt werden. Die Bedeutung der einzelnen Eigenschaften wird ausführlich im IFG-Handbuch erläutert.

Aufbereitungseigenschaften

Für die Aufbereitung von Maskenfeldern stehen folgende Möglichkeiten zur Verfügung:

- Dezimaltrenner
- Zifferntrenner
- Darstellung des Datums
- Darstellung der Uhrzeit
- Ausrichtung des Inhaltes eines Maskenfeldes
- Ausgabefüllzeichen
- Nullenunterdrückung
- Nullwertdarstellung als Leerfeld
- gleitendes Vorzeichen
- Umwandlung in Großbuchstaben

Eingabe- und Prüfeigenschaften

Dazu zählen folgende Eingabeeigenschaften:

- Eingabepflicht
- minimale Eingabelänge
- automatische Eingabe
- markierbares/nicht markierbares Feld
- geschütztes/nicht geschütztes Feld
- hardwareunterstützte NUM-Sperre

Für den Feldinhalt stehen folgende Prüfungen zur Verfügung:

- beliebige Zeichenkette
- alphabetische Zeichenkette
- arithmetische Zeichenkette
 - mit oder ohne Dezimalstellen
 - mit oder ohne Vorzeichen
 - mit oder ohne Zifferngruppierung
- Datum
- Uhrzeit
- Werteliste
- Wertebereich
- Zeichenliste

3.8.1 Codetabellen

Bei den Datenaufbereitungen und Prüfungen im FHS-DM werden Codetabellen genutzt. Die Codetabellen werden aus Tabellen des XHCS-Tabellensatzes generiert.

Folgende Tabellen werden generiert:

- Tabelle zur Umwandlung von Klein- in Grossbuchstaben
- Tabelle für darstellbare Zeichen
- Tabelle für alphabethische Zeichen
- Tabelle für numerische Zeichen

Beim Erstellen eines Formates mit IFG können Sie eine Zuordnung mittels CCS-Name zu einem Tabellensatz treffen. Erstellung und Modifikation dieser Tabellen sind im Handbuch "XHCS - Extended Host Code Support" erklärt.

Ein CCS-Name kann auch durch den CONTROL-Dienst des FHS-DM angegeben werden oder das Terminal gibt ihn vor.

Wurde zu einem Format kein CCS-Name angegeben oder ist am Host XHCS nicht vorhanden, werden Tabellen verwendet, die dem CCS-Namen EDF03IRV entsprechen, bzw. dem CCS-Namen EDF03DRV, wenn erkannt wird, dass eine deutsche Tastatur genutzt wird.

3.8.2 Datenaufbereitungen

Folgende Angaben werden formatglobal vereinbart:

Dezimaltrenner

Erlaubte Zeichen sind "," und ".". Dezimaltrenner und Ziffertrenner müssen verschieden sein. Der Dezimaltrenner dient der Trennung von Vor- und Nachkommastellen in arithmetischen Maskenfeldern (z.B. 123,45). Der Dezimaltrenner wird nicht in die Dialogvariable übernommen.

Zifferntrenner

Erlaubte Zeichen sind ".", ",", " " und Leerzeichen. Zifferntrenner und Dezimaltrenner müssen verschieden sein. Der Zifferntrenner dient zur Gruppierung von Vorkommastellen in arithmetischen Maskenfeldern (z.B. 123.456,78). Der Zifferntrenner wird nicht in die Dialogvariable übernommen.

Darstellung des Datums

Für die Darstellung des Datums in einem Maskenfeld sind die folgenden Angaben zu vereinbaren:

- Reihenfolge von Tag, Monat und Jahr
- Trennzeichen der Datumsangaben
- Jahresangabe zweistellig oder vierstellig

Für die Reihenfolge von Tag, Monat und Jahr sind alle Kombinationen zugelassen.

Als Trennzeichen sind alle Zeichen außer die Ziffern 0-9 erlaubt. Das Trennzeichen trennt die Angaben Tag, Monat und Jahr (z.B. 10.12.92).

Darstellung der Uhrzeit

Für die Darstellung der Uhrzeit sind folgende Aufbereitungsangaben notwendig:

- Zeittrennzeichen
- mit oder ohne Sekundenangabe

Als Trennzeichen sind alle Zeichen außer die Ziffern 0-9 erlaubt.

Das Zeittrennzeichen dient der Trennung von Stunden, Minuten und Sekunden bei der Darstellung der Zeitangabe (z.B. 12:34:30).

Spezifisch für jedes Maskenfeld sind folgende Angaben möglich:

Ausrichtung und Füllzeichen bei der Ausgabe

Bei der Formatdefinition kann festgelegt werden, ob bei der Ausgabe einer Dialogvariablen in ein Maskenfeld eine Ausrichtung der Daten im Maskenfeld erfolgt und mit welchem Zeichen nicht belegte Stellen im Maskenfeld aufgefüllt werden.

Für den Inhalt einer Dialogvariablen wird eine Angabe einer Ausrichtung und eines Füllzeichens nicht unterstützt, da hierfür der Dialogmanager in Abhängigkeit vom Datentyp der Dialogvariablen feste Werte verwendet. Das mit dem IFG festgelegte Füllzeichen wird von FHS-DM nicht genutzt, im folgenden Text ist mit Füllzeichen immer das IFG-Ausgabefüllzeichen gemeint.

Ausrichtung für ein Maskenfeld:

Die Art der Ausrichtung gibt an, wie bei der Anzeige (Ausgabe) der Wert der Dialogvariablen im Maskenfeld ausgerichtet werden soll und wie bei einer Eingabe in ein Maskenfeld Füllzeichen zu behandeln sind.

Es können folgende Arten der Ausrichtung für beliebige und alphabetische Zeichenketten bei der Formatdefinition vereinbart werden:

- Ausrichtung links
- Ausrichtung rechts
- keine Ausrichtung

Arithmetische Zeichenketten werden immer rechts ausgerichtet.

Eine Ausrichtung des Maskenfeldes erfolgt, wenn die Länge der relevanten Daten einer Dialogvariablen kürzer ist als die Länge des Maskenfeldes. Sind die relevanten Daten der Dialogvariablen länger als das Maskenfeld, so ist die Reaktion des DISPLAY-Dienstes abhängig davon, ob das Maskenfeld ein Ein- oder Ausgabefeld ist.

Bei einem Ausgabefeld werden die Daten abgeschnitten und die Verarbeitung mit einer Warnung fortgesetzt. Ist ein Eingabefeld zu kurz, wird der DISPLAY-Dienst mit einem Fehlercode beendet. Ausrichtung links oder rechts beschreibt, wie der Inhalt des Maskenfeldes bei der Anzeige auf dem Bildschirm durch den Dialogmanager ausgerichtet wird. Bei einer Ausrichtung links oder rechts werden die Daten einer Dialogvariablen vom Typ CHAR oder BINSTR ohne führende Leerzeichen angezeigt. Wenn als Ausrichtung "keine Ausrichtung" angegeben ist, werden führende Leerzeichen angezeigt und die Daten der Dialogvariablen links ausgerichtet.

Ausgabefüllzeichen:

Wenn die Daten einer Dialogvariablen nicht das gesamte Maskenfeld füllen, werden an den verbleibenden Positionen im Maskenfeld die mit dem IFG vereinbarten Ausgabefüllzeichen eingetragen. Wenn die Dialogvariable leer ist, so enthält das ganze Maskenfeld bei der Anzeige nur Füllzeichen.

Bei der Ausgabe in ein Maskenfeld wird das Maskenfeld zunächst mit den ausgerichteten Daten belegt und dann aufgefüllt.

Füllzeichenbehandlung bei der Eingabe

Der Terminal-Benutzer kann die Daten in ein Maskenfeld ohne Berücksichtigung der für das Maskenfeld vereinbarten Ausrichtung eingeben. Bei der Verarbeitung der Eingabedaten durch den Dialogmanager werden die relevanten Daten und deren Länge ermittelt. Dabei werden für beliebige oder alphabetische Zeichenketten führende oder nachfolgende Füllzeichen wie folgt behandelt:

- Für ein links oder rechts ausgerichtetes Maskenfeld werden führende und nachfolgende Füllzeichen, Leerzeichen und NIL-Zeichen entfernt.
- Für ein nicht ausgerichtetes Maskenfeld werden nachfolgende Füllzeichen, Leerzeichen und NIL-Zeichen entfernt und führende Füllzeichen, Leerzeichen oder NIL-Zeichen in Leerzeichen umgewandelt und gespeichert.
- Enthält das Eingabefeld nur Leer-, Füll- oder NIL-Zeichen, ergibt dies relevante Daten der Länge Null (Leerwert).
- In keinem Fall werden in der Zeichenkette eingeschlossene Füllzeichen, Leerzeichen oder NIL-Zeichen entfernt.

Für ein arithmetisches Maskenfeld werden zur Ermittlung der relevanten Daten führende oder nachfolgende Füllzeichen wie folgt behandelt:

- Das Maskenfeld gilt immer als rechts ausgerichtet. Wenn das Füllzeichen ungleich Null ist, werden führende und nachfolgende Füllzeichen und NIL-Zeichen entfernt.
- Führende Nullen werden immer entfernt, auch wenn das Füllzeichen Null ist.
- Enthält das Eingabefeld nur Leer-, Füll- oder NIL-Zeichen, ergibt dies relevante Daten der Länge Null.

Unabhängig von der Art der Ausrichtung werden für ein Kommandoingabefeld führende und nachfolgende Füllzeichen entfernt.

Nach der Eingabebehandlung werden die relevanten Daten in die entsprechenden Dialogvariablen gespeichert. Dabei werden die Daten entsprechend der für den jeweiligen Datentyp der Dialogvariablen geltenden Ausrichtung und mit der ermittelten Länge der relevanten Daten gespeichert. Für explizite Dialogvariable erfolgt anschließend ein Auffüllen mit den Standardfüllzeichen, wenn die Datenlänge kleiner ist als die definierte Länge der Dialogvariablen. Implizite Dialogvariable werden immer in der relevanten Länge gespeichert (außer wenn die Länge Null ist).

Möglichkeiten für Füllzeichen und Ausrichtung:

Maskenfeld	Ausgabe-Füllzeichen	Ausrichtung
beliebige Zeichenkette	beliebiges Zeichen	rechts,links,ohne
alphabetische Zeichenkette	beliebiges Zeichen	rechts,links,ohne
arithmetische Zeichenkette	keine Ziffer 1 bis 9, kein Vorzeichen	immer rechts
Datum	keine Ziffer 1 bis 9	nicht möglich
Uhrzeit	keine Ziffer 1 bis 9	nicht möglich

Dialogvariable	Eingabe-Füllzeichen	Ausrichtung
CHAR / STRING	Leerzeichen	links
BINSTR	X'00'	links
NUMS	X'F0'	rechts
NUMU	X'F0'	rechts
FIXEDS / INTEGER	X'00'	rechts
FIXEDU	X'00'	rechts
PACK	X'00'	rechts

Die Datentypen sind beim VDEFINE-Dienst beschrieben.

3.8.3 Eingabeeigenschaften

Minimale Eingabelänge

Bei der Eingabe in ein Feld müssen mindestens so viele Zeichen eingegeben werden, wie es die Mindesteingabelänge festlegt.

Eingabepflicht

Eingabepflicht für ein Maskenfeld bedeutet, daß der Terminal-Benutzer dieses Feld mittels Tastatureingabe modifizieren muß. Erfüllt er diese Forderung nicht, wird er durch den Dialogmanager durch eine Meldung zur Eingabe aufgefordert. Wurde für ein Feld die Eingabepflicht durch den Terminal-Benutzer erfüllt, wird dies durch den Dialogmanager intern registriert. Bei einem nachfolgenden DISPLAY-Aufruf ohne den PANEL-Operand oder bei Angabe des gleichen Formatnamens wird nicht nochmals zur Eingabe für dieses Feld aufgefordert.

Soll auch bei einer nachfolgenden Ausgabe für das gleiche Format eine Eingabepflicht wirksam werden, muß vor dem DISPLAY-Aufruf ein ATTR-Dienst mit TYPE(MANDATORY) für dieses Feld aufgerufen werden oder es ist der Operand MANDATORY beim DISPLAY-Aufruf anzugeben. Für Listenfelder erfolgt eine spezielle Behandlung der Eingabepflicht (siehe Listenverarbeitung).

3.8.4 Prüfungen für den Inhalt von Maskenfeldern

Es erfolgt bei der Ein- und Ausgabe eine Prüfung und Aufbereitung der Daten wie sie im folgenden beschrieben wird.

Beliebige Zeichenkette

Das Maskenfeld darf alle Zeichen enthalten. Es erfolgt keine Prüfung und Aufbereitung durch den Dialogmanager. Nichtdruckbare Zeichen (außer NIL) werden durch SUB (X'3F') dargestellt.

Alphabetische Zeichenkette

Das Maskenfeld darf nur Buchstaben der Codetabellen und Leerzeichen enthalten.

Arithmetische Zeichenkette

Um die Aufbereitungseigenschaften eines arithmetischen Maskenfeldes sinnvoll zu nutzen, muß die entsprechende Dialogvariable vom Typ NUMS, NUMU, FIXEDS, FIXEDU oder PACK sein. Ein arithmetisches Maskenfeld darf nicht mehr als 15 Ziffernstellen enthalten. Die Anzahl der Ziffernstellen ergibt sich aus der Feldlänge abzüglich der Anzahl der Stellen für Vorzeichen, Dezimaltrenner und Zifferntrenner, falls die entsprechenden Eigenschaften vereinbart wurden.

Allgemeine Eingabepfung:

Das Feld darf nur Ziffern (0- 9) und, abhängig von den Eigenschaften Dezimalstellen, Zifferngruppierung und Vorzeichen, an bestimmten Positionen die Zeichen ",", ".", "+" und "-" enthalten. Leerzeichen gefolgt von Ausgabefüllzeichen vor der Zahl bzw. Ausgabefüllzeichen gefolgt von Leerzeichen nach der Zahl führen zu einem Fehler, falls das Ausgabefüllzeichen ungleich Leerzeichen ist.

Ausgabepfung:

Die Dialogvariable muß vom Typ FIXEDS, FIXEDU, PACK, NUMS oder NUMU sein und tygerechte Daten enthalten.

Arithmetische Zeichenkette mit Dezimalstellen

Für ein Maskenfeld kann eine Anzahl von Dezimalstellen vereinbart werden.

Eingabeprüfung:

Es dürfen maximal i Ziffern (mittels IFG festgelegt) nach dem Dezimaltrenner folgen.
(Rechts stehende Nullen werden ignoriert.)

Eingabeaufbereitung :

Eingegebene Nachkommastellen werden in die am weitesten links befindlichen Ziffernstellen für die Nachkommastellen übertragen und rechts mit "0" aufgefüllt.

Ausgabeaufbereitung:

Die am weitesten rechts liegenden i Ziffern werden als Nachkommastellen interpretiert und unter Beachtung des Dezimaltrenners rechts ausgegeben.

Arithmetische Zeichenkette mit Vorzeichen

Die Angabe eines Vorzeichens ist für dieses Feld erlaubt. Das Vorzeichen kann als gleitend vereinbart werden.

Eingabepfung:

Falls kein Vorzeichen erlaubt ist, darf weder "+" noch "-" im Maskenfeld vorhanden sein. Falls ein Vorzeichen erlaubt ist, darf es bei der Eingabe links oder rechts stehen. Zwischen Vorzeichen und Zahl dürfen Leerzeichen stehen. Nur ein Vorzeichen im Maskenfeld (ohne Ziffern) ist nicht erlaubt.

Ausgabepfung:

Die Dialogvariable muß vom Typ FIXEDS, NUMS oder PACK sein.

Eingabeaufbereitung:

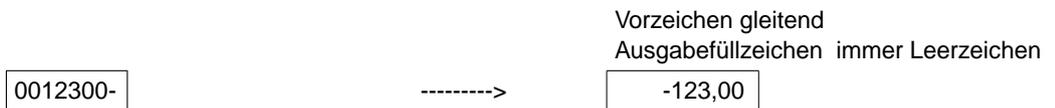
Das Vorzeichen "-" wird an die letzte Stelle einer NUMS-Dialogvariablen gesetzt; "+" oder kein Vorzeichen ergibt "+" an der letzten Position.



Ausgabeaufbereitung:

Ein positives Vorzeichen wird als Leerzeichen, ein negatives als "-" dargestellt.

In Abhängigkeit von der Angabe "Vorzeichen gleitend" wird das Vorzeichen entweder an die letzte Stelle des Maskenfeldes oder vor die Zahl gesetzt. Die links übrig bleibenden Stellen werden entweder mit dem Ausgabefüllzeichen oder mit Leerzeichen (bei "Vorzeichen gleitend") gefüllt.



Arithmetische Zeichenkette mit Zifferngruppierung

Vorkommastellen werden durch Zifferntrenner in Gruppen zu drei Ziffern getrennt.

Eingabepfung:

Die Vorkommastellen dürfen, fortschreitend von rechts nach links, in 3-er Gruppen durch den Zifferntrenner geteilt sein. Führende und zusammenhängende Nullen werden ignoriert. Falls ein Zifferntrenner im Feld vorkommt, müssen alle Vorkommastellen richtig gruppiert sein.

Eingabeaufbereitung:

Zifferntrenner werden nicht übernommen.



Ausgabeaufbereitung:

Zifferntrenner werden an der richtigen Stelle eingesetzt. Dabei werden die Vorkommastellen von rechts beginnend übertragen und nach jeder dritten Ziffer ein Zifferntrenner eingesetzt, falls noch eine weitere Ziffer folgt. (Falls Nullenunterdrückung nicht erlaubt ist, kann auch an der ersten Stelle des Feldes ein Zifferntrenner stehen)



Arithmetische Zeichenkette mit Nullenunterdrückung

Führende Nullen werden weggelassen.

Eingabepfung:

Es dürfen beliebig viele Füllzeichen vor der Zahl stehen.

Vor dem Dezimaltrenner darf eine führende Null weggelassen werden; eine Eingabe wie z.B. ",5" statt "0,5" ist korrekt.

Eingabeaufbereitung:

Füllzeichen werden nicht übernommen.

Ausgabeaufbereitung:

Falls Nullenunterdrückung erlaubt ist, werden alle führenden Nullen bis auf die erste Vorkommastelle durch Ausgabefüllzeichen ersetzt, im anderen Fall bleiben alle führenden Nullen erhalten.

Ausnahme:

Falls vor dem Dezimaltrenner kein Platz mehr bleibt, steht vor dem Dezimaltrenner keine führende Null.

Dialogvariable

Typ NUMS

0000012-

----->

0000012-

----->

Maskenfeld

Vorzeichen nicht gleitend
Ausgabefüllzeichen "\$"
zwei Dezimalstellen
Nullenunterdrückung

\$\$\$\$\$0,12-

keine Nullenunterdrückung

000000,12-

Nullwert als Leerwert:

Der Anwendungsprogrammierer kann mittels IFG festlegen, ob der Nullwert einer numerischen Dialogvariablen als Leerwert in einem arithmetischen Maskenfeld dargestellt werden soll. In diesem Fall werden nur Leerzeichen angezeigt.

Aufbereitung von arithmetischen Maskenfeldern

Die Ausgabeaufbereitung erfolgt in folgender Form (Ausrichtung immer rechts):

Zifferngruppierung nicht erlaubt:	
Vorzeichen nicht gleitend	[F...][9...][9][D9[9...]][-]
Vorzeichen gleitend	[b...][-][9...][9][D9[9...]]

Zifferngruppierung erlaubt:	
Vorzeichen nicht gleitend	[F...][9[9]][9][Z999...][D9[9...]][-]
Vorzeichen gleitend	[b...][-][9[9]][9][Z999...][D9[9...]]

Beispiele für die Anzeige in einem arithmetischen Maskenfeld:

123,5
 1.234,5
 -123
 123,4-
 000123
 000,12
 0,12

Eingaben in ein arithmetisches Maskenfeld sind in folgender Form möglich:

Zifferngruppierung nicht erlaubt:	
[F...][b/n...][+/-][b...][9...][D[9...]][b/n...][F...] oder [F...][b/n...][9...][D[9...]][b...][+/-][b/n...][F...]	

Zifferngruppierung erlaubt:	
[F...][b/n...][+/-][b/n...][9[9]][Z999...][D[9...]][b/n...][F...] oder [F...][b/n...][9[9]][Z999...][D[9...]][b...][+/-][b/n...][F...]	

Ist eine Zifferngruppierung erlaubt, kann auch ohne Zifferngruppierung eingegeben werden. Es bedeuten:

- n NIL (X'00')
- b Leerzeichen
- 9 Ziffer 0-9
- Z Zifferntrenner
- D Dezimaltrenner
- F Ausgabefüllzeichen

Beispiele für gültige Eingabedaten (keine Ausrichtung notwendig):

,5
1.234,5
1.234,
.234
+ 1234
123-

Beispiele für ungültige Eingabedaten:

1.2345
0,123
2,123.
123.

Zifferngruppierung muß in 3-er Gruppen erfolgen zuviele Nachkommastellen (definiert wurden zwei) Trennzeichen nicht korrekt (entspr. Definition)

Datum

Das Maskenfeld enthält eine Datumsangabe. Die Dialogvariable muß vom Typ CHAR sein und 12 oder 14 Bytes lang sein.

Eingabeprüfung:

Das Feld darf nur die Ziffern 0 bis 9, Leerzeichen und zweimal das Trennzeichen für Datumsangabe enthalten. Bei Tag, Monat und zweistelliger Jahresangabe können führende Nullen weggelassen werden. Vor und nach dem Trennzeichen dürfen Leerzeichen stehen. Leerzeichen gefolgt von Ausgabefüllzeichen vor dem Datum bzw. Ausgabefüllzeichen gefolgt von Leerzeichen nach dem Datum führen zu einem Fehler, falls das Ausgabefüllzeichen nicht das Leerzeichen ist. Am Anfang bzw. am Ende des Datums dürfen Leerzeichen stehen (unabhängig vom Ausgabefüllzeichen).

Bei zweistelliger Jahresangabe wird bei der Prüfung auf ein gültiges Datum der Zeitraum von 1901 bis 2099 angenommen. Bei vierstelliger Jahresangabe darf das Datum nur im Zeitraum vom 15-10-1582 (Beginn des Gregorianischen Kalenders) bis zum 31-12-2099 liegen.

Ausgabeprüfung:

Die Dialogvariable muß in ISO- bzw. ISO4-Form ein gültiges Datum enthalten. Die Richtigkeit des laufenden Tages und die letzte Stelle (Leerzeichen) werden nicht überprüft. Die Darstellung des Datums in der Dialogvariablen (interne Darstellung) entspricht der des GDATE/GTIME-Makros (siehe Handbuch "BS2000 Makroaufrufe an den Ablaufteil").

ISO-Format:

0		2		3		5		6		8		11
j j	X'60'	m m	X'60'	t t	i i i	X'40'						
Jahr	-	Monat	-	Tag	laufender Tag des Jahres							

ISO4-Format:

0		4		5		7		8		10		13
j j j j	X'60'	m m	X'60'	t t	i i i	X'40'						
Jahr	-	Monat	-	Tag	laufender Tag des Jahres							

Eingabeaufbereitung:

Der Dialogmanager berechnet den laufenden Tag des Jahres und liefert das Datum in ISO- bzw. ISO4-Form in der entsprechenden Dialogvariablen zurück.

Maskenfeld		Dialogvariable
Datum mit Kalenderprüfung		Typ CHAR
Jahr zweistellig		Länge 12 BYTES
Trennzeichen "/"		

<input type="text" value="1 / 4/88"/>	----->	<input type="text" value="88-04-01092"/>
---------------------------------------	--------	--

ohne Kalenderprüfung

<input type="text" value="33/0/88"/>	----->	<input type="text" value="88-00-33000"/>
--------------------------------------	--------	--

Ausgabeaufbereitung:

Es werden Tag, Monat und Jahr und die Trennzeichen an die entsprechende Position des Maskenfeldes übertragen. Der laufende Tag wird nicht ausgegeben.

Die Stellen für Tag, Monat und Jahr werden, falls notwendig, mit führenden Nullen ergänzt, so daß die Ausgabe immer 8 bzw. 10 Stellen lang ist.

Dialogvariable		Maskenfeld
----------------	--	------------

<input type="text" value="88-04-01...."/>	----->	<input type="text" value="01/04/88"/>
---	--------	---------------------------------------

Ohne Kalenderprüfung

<input type="text" value="88-00-33...."/>	----->	<input type="text" value="33/00/88"/>
---	--------	---------------------------------------

Uhrzeit

Das Maskenfeld enthält eine Zeitangabe. Die Dialogvariable muß vom Typ CHAR sein.

Eingabepfung:

Als Zeittrennzeichen darf nur das vereinbarte Zeichen an bestimmten Positionen im Maskenfeld auftreten. Bei Stunden, Minuten und Sekunden können führende Nullen weggelassen werden. Vor und nach dem Trennzeichen dürfen Leerzeichen stehen. Leerzeichen gefolgt von Ausgabefüllzeichen vor der Uhrzeit bzw. Ausgabefüllzeichen gefolgt von Leerzeichen nach der Uhrzeit führen zu einem Fehler, falls das Ausgabefüllzeichen nicht das Leerzeichen ist. Am Anfang bzw. am Ende der Uhrzeit dürfen Leerzeichen stehen (unabhängig vom Ausgabefüllzeichen).

Ausgabepfung:

Die Dialogvariable muß in Abhängigkeit von der Sekundenangabe den Wert HH:MM:SS oder HH:MM enthalten. Dabei gilt: HH 00-23 Stunden, MM 00-59 Minuten und SS 00-59 Sekunden

Eingabeaufbereitung:

Als Zeittrennzeichen wird immer ":" verwendet.

Maskenfeld		Dialogvariable
Uhrzeit mit Sekunden		Typ CHAR
Zeittrennzeichen ":"		
<input type="text" value="12:08:36"/>	----->	<input type="text" value="12:08:36"/>

Ausgabeaufbereitung:

Die Zeittrennzeichen werden an der richtigen Stelle eingefügt.

Dialogvariable		Maskenfeld
Typ CHAR		Uhrzeit mit Sekunden
Länge 8		Zeittrennzeichen ":"
<input type="text" value="12:08:36"/>	----->	<input type="text" value="12.08.36"/>

Werteliste

Diese Angabe ist für jedes Eingabefeld der Maske möglich.

Eingabeprüfung:

Entweder muß der eingegebene Wert mit einem Wert einer Werteliste übereinstimmen (Prüfung auf gleich) oder der Wert darf nicht in der Werteliste stehen (Prüfung auf ungleich). Die Prüfbedingungen und die Werteliste werden bei der Formatdefinition angegeben.

Beispiel:

Eingegebener Wert: HUND

Vorgegebene Werteliste: KATZE HUND HUHN SCHAF

Prüfung auf gleich: ohne Fehler

Wertebereich

Diese Angabe ist nur für arithmetische Maskenfelder erlaubt.

Eingabeprüfung:

Der eingegebene Wert muß in einem vorgegebenen numerischen Wertebereich enthalten sein (einschließlich der Grenzen). Vorzeichen sind möglich. Bei negativen Werten muß das Minuszeichen angegeben werden.

Beispiel:

Eingegebener Wert: 123

Vorgegebene Grenzen: 222- 222+

Ergebnis der Prüfung: ohne Fehler

Zeichenmenge

Diese Angabe ist für alle Eingabefelder einer Maske möglich.

Eingabeprüfung:

Entweder müssen alle eingegebenen Zeichen in einer mittels IFG definierten Zeichenmenge enthalten sein (Prüfung auf gleich) oder die Zeichen dürfen nicht in der Zeichenmenge vorkommen (Prüfung auf ungleich). Die Prüfbedingungen und die Zeichenliste werden bei der Formatdefinition angegeben.

Eingegebener Wert: AB

Vorgegebene Zeichenmenge: ABCDEF

Prüfung auf ungleich: Fehlermeldung

Datenumwandlung beim DISPLAY-Dienst

Die folgende Tabelle enthält die mögliche Datenumwandlung zwischen Dialogvariable und Maskenfeld für TIAM.

Dialog- variable	Maskenfeld				
	Zeichenkette	arithm. Zeichenkette		Datum	Uhrzeit
mit Vorz.		ohne Vorz.			
CHAR	+	+ 1)	+ 1)	spez. Daten	spez. Daten
BINSTR	+	+ 1)	+ 1)	spez. Daten	spez. Daten
NUMS	+ 2)	+	+	-	-
NUMU	+ 2)	+	+	-	-
FIXEDS	+ 2)	+	+	-	-
FIXEDU	+ 2)	+	+	-	-
PACK	+ 2)	+	+	-	-

Die folgende Tabelle enthält die mögliche Datenumwandlung zwischen Dialogvariable und Maskenfeld für SDF-P.

SDF-P- Variable	Maskenfeld				
	Zeichenkette	arithm. Zeichenkette		Datum	Uhrzeit
mit Vorz.		ohne Vorz.			
STRING	+	+ 1)	+ 1)	spez. Daten	spez. Daten
INTEGER	+ 2)	+	+	-	-

Dabei gilt:

- + Datenumwandlung möglich (evtl. Vorzeichenfehler)
- Datenumwandlung nicht unterstützt (Konvertierungsfehler)
- 1) Der Wert der anzuzeigenden Dialogvariablen muß numerisch sein
- 2) Der Inhalt des Maskenfeldes bei der Eingabe muß numerisch sein.

Einschränkungen und Unverträglichkeit

Vorzeichen gleitend: Nur in Kombination mit "Vorzeichen erlaubt" möglich.

Eingabepflicht: Das Attribut "Eingabepflicht" darf nicht mit "geschützt, markierbar und automatische Eingabe" kombiniert werden:

Geschützt Für ein geschütztes Feld darf nicht "Mindesteingabelänge > 0" verlangt werden.

Automatische Eingabe: wird für Menütitel und Auswahleingabefelder ignoriert.

Nullenunterdrückung: Die Kombination von "Nullenunterdrückung" und "Vorzeichen gleitend" bei Ausgabefüllzeichen ungleich Leerzeichen ist nicht sinnvoll, da "Vorzeichen gleitend" immer mit Leerzeichen auffüllt. "Nullenunterdrückung" und Ausgabefüllzeichen "0" schließen einander aus.

3.9 Kommandos

Der Dialogmanager unterstützt die Verwendung von Kommandos. Dabei gibt es in Abhängigkeit von der Art der Verarbeitung folgende Möglichkeiten:

- Systemkommandos; sie werden vom Dialogmanager bearbeitet und nicht dem Anwendungsprogramm übergeben,
- Anwendungskommandos; diese werden dem Anwendungsprogramm übergeben und müssen von diesem verarbeitet werden.

Entsprechend der Art der Kommandoingabe kann unterschieden werden in:

- Eingabe eines Kommandos im Kommandoingabefeld
- Auslösen eines Kommandos durch Betätigen einer Funktionstaste. Dazu muß in einer Tastenbelegungstabelle die Zuordnung von Funktionstaste und Kommando beschrieben werden.
- Kommandoingabe durch Zusammensetzen eines Kommandos durch eine Funktionstaste und der Eingabe im Kommandoingabefeld.

Kommandobereich

Der Kommandobereich in einer Maske wird beim Definieren des Formates mit dem IFG festgelegt. Ein Kommandobereich ist nicht zwingend notwendig.

Wird ein Systemkommando im Kommandoingabefeld eingegeben, erfolgt eine Verarbeitung des Kommandos durch den Dialogmanager. Anschließend wird das Kommandoingabefeld gelöscht, falls nicht der Kommandonamen mit einem vorangestellten "&" eingegeben wurde.

Ist die eingegebene Zeichenkette kein Systemkommando, wird diese als Anwendungskommando betrachtet. Die Eingabedaten im Kommandoingabefeld werden in die diesem Maskenfeld zugeordnete Dialogvariable geschrieben. Wenn der Name dieser Dialogvariablen nicht SYS-COMMAND ist, werden die Eingabedaten außerdem in der Dialogvariablen SYS-COMMAND im Funktionspool bereitgestellt. Sie wird, falls sie nicht existiert, implizit angelegt. SYS-COMMAND wird auch geschrieben, wenn kein Kommandoingabefeld existiert und mittels Funktionstaste ein Anwendungskommando eingegeben wird.

Ist beim DISPLAY-Aufruf die dem Kommandoingabefeld der Maske zugeordnete Dialogvariable nicht leer, wird ihr Wert im Kommandobereich angezeigt. Damit ist eine Vorbelegung durch das Anwendungsprogramm realisierbar.

Hinweis:

Durch Zuordnung der Dialogvariablen SYS-COMMAND zu allen Kommandoingabefeldern kann man erreichen, das in der gesamten Anwendung Anwendungskommandos in der Dialogvariablen SYS-COMMAND geschrieben und gelesen werden.

Kommandos über Funktionstasten eingeben

F-Tasten können System- und Anwendungskommandos zugeordnet werden. Stehen nicht ausreichend F-Tasten zur Verfügung, können durch das Systemkommando SETP F-Tasten durch P-Tasten simuliert werden. Das Drücken einer F-Taste wirkt so wie Eintragen des Kommandos in das Kommandoeingabefeld und Betätigen der ENTER-Taste. Ein auf diese Art eingegebenes Anwendungskommando steht dem Anwendungsprogramm in der Systemvariablen SYS-COMMAND zur Verfügung.

Zusammengesetzte Kommandos eingeben

Ist einer F-Taste per Tastenbelegungstabelle ein Kommando zugeordnet und existiert in der Maske ein Kommandoingabefeld, so kann in dieses eine Zeichenkette eingetragen werden und anschließend die F-Taste gedrückt werden. Der Dialogmanager verbindet die Daten der F-Taste, mit einem Leerzeichen getrennt, mit den Daten aus dem Kommandoingabefeld. FHS kombiniert beide Zeichenketten wie folgt miteinander:



Beispiel:

1. Belegung der Funktionstaste F20 mit "KEYAREA"
2. - "OFF" in das Kommandoingabefeld eintragen
3. Betätigen der Funktionstaste F20 - es wird das Kommando "KEYAREA OFF" ausgeführt.

FHS-Kommando von einer Anwendung auslösen

Möchten Sie ein FHS-Kommando von einer Anwendung auslösen, so definieren Sie ein Format mit Kommandobereich. Die Dialogvariable, die mit dem Kommando-Eingabefeld verbunden ist, belegen Sie mit dem gewünschten Kommando (Zeichenkettenzuweisung). Anschliessend rufen Sie den Display-Dienst mit PANEL(name) und dem Operanden NODISPLAY auf. FHS-DM verarbeitet das Kommando, ohne daß das Format angezeigt wird.

Beispiel 1:

Sie möchten die P-Tasten 1,3 und 12 zur Simulation von F-Tasten benutzen. Das zugehörige Kommando dazu heißt:

```
SETP (P1,P3,P12) ON.
```

Belegen Sie die zum Kommandoingabebereich zugeordnete Dialogvariable mit dieser Zeichenkette und rufen den FHS-DM-Dienst mit DISPLAY PANEL(name) NODISPLAY.

Anschließend stehen die Tasten P1, P3 und P12 für FHS zur Verfügung.

Beispiel 2:

Kommando "SYS"

Belegen Sie die Dialogvariable mit:

```
"SYS FILE-STATUS A*"
```

und rufen den Display-Dienst wie oben, so werden alle Files der angegebenen Schablone angezeigt. Vor der nächsten Anzeige des Display-Dienstes werden die Promptzeichen "ACK" ausgegeben und eine Eingabe wird erwartet.

3.9.1 Systemkommandos des Dialogmanagers

Operation	Bedeutung
ABORT	Anwendung beenden
ACTIONS	Cursor in Menüleiste/Maskenfelder positionieren
CANCEL	Anzeige abbrechen
EXIT	Anwendungsabschnitt beenden
EXTHELP	Erweiterte Hilfe anfordern
HARDCOPY	Bildschirminhalt auf Drucker ausgeben
HELP	Hilfe anfordern
HELPHelp	Hilfe zum Hilfesystem
INDEX	Übersicht zu Hilfen
KEYAREA	Anzeige Tastenbelegung ein/ausschalten
KEYSHELP	Hilfe zur Tastenbelegung
PANELID	Anzeige von Formatnamen und Meldungsschlüssel ein/ausschalten
RMSG	Wiederanzeige einer Meldung
SETP	P-Tasten zuordnen
SYS	BS2000-Kommandos ausführen
RESHOW (K-Taste)	Wiederanzeige der Maske
FORWARD BACKWARD + - ++ --	Blätter- und Verschiebekommandos

Systemkommandos können in Klein- oder in Großbuchstaben eingegeben werden.

Die Kommandos KEYAREA, PANELID und SETP gelten für alle weiteren Ausgaben für den gesamten Dialog-Komplex unabhängig vom Verständigungsbereich (siehe Seite 118). Diese Angaben werden beim DMCLOSE für den Dialog-Komplex in das Profil geschrieben und werden beim nächsten DMOPEN mit diesem Profil wieder bereitgestellt.

ABORT - Anwendung beenden

Operation	Operanden
ABORT	

Wenn FHS-DM einen Fehler erkannt hat, der die normale Fortsetzung der Verarbeitung eines FHS-Dienstes verbietet, kann eine Fehlermaske ausgegeben werden. Danach wird der zum Fehler gehörende Rückkehrcode an die Anwendung geliefert.

Wird bei der Anzeige der Fehlermaske das Kommando ABORT gegeben, wird bei Rückkehr zur Anwendung der MAINCODE auf 399999 gesetzt. Die Anwendung sollte bei diesem Rückkehrcode beendet werden. (FHS-DM bricht normalerweise nicht selbst die Verarbeitung ab!)

Wird ABORT in einer Fehlermaske eingegeben, die auf Grund impliziter Aktionen ausgegeben wird und normalerweise die Fortsetzung der Anwendung ermöglicht, wird auch hier der Display-Dienst mit dem MC=399999 beendet.

ACTIONS - Cursor in Menüleiste positionieren

Operation	Operanden
ACTIONS	

ACTIONS positioniert den Cursor in die Menüleiste. Wenn sich der Cursor bereits in der Menüleiste oder in einer Pull-Down-Box befindet, wird er in das Feld gesetzt, in dem er sich beim vorherigen ACTIONS-Kommando befunden hat. Gab es kein vorheriges ACTIONS-Kommando, wird der Cursor in das erste Eingabefeld des Arbeitsbereiches gestellt. Dieses Kommando ist nur als F-Taste (Standard F10) sinnvoll zu benutzen. Es ist wirkungslos, wenn keine Menüleiste existiert.

Angegebene Operanden werden ignoriert.

CANCEL - Anzeige beenden/abbrechen

Operation	Operanden
CANCEL	

CANCEL für eine implizite Box (der Cursor befindet sich in einer Box) bewirkt das Entfernen der obersten impliziten Box und zeigt, falls eine unterliegende implizite Box existiert, diese an. Ansonsten wird das Grundbild angezeigt. Das Anwendungsprogramm erhält davon keine Kenntnis.

Zum Schutz gegen versehentlich falsches Positionieren des Cursors werden alle impliziten Boxen auch dann entfernt, wenn sich der Cursor im Grundbild oder in einer expliziten Box befindet. Das Kommando wirkt in diesem Fall nicht für das Grundbild oder für eine explizite Box. Erst nach dem Entfernen aller impliziten Boxen ist ein erneutes CANCEL für das Grundbild wirksam.

CANCEL für eine Pulldown-Box bewirkt das Entfernen der Pulldown-Box und das Positionieren des Cursors auf den ersten Menütitel. Eine eingegebene Auswahl in der Pulldown-Box wird ignoriert.

CANCEL für eine explizite Box oder das Grundbild bewirkt das Beenden des aktuellen DISPLAY-Aufrufs mit einem entsprechenden Rückkehrcode. Alle Eingabedaten gehen verloren. Eine Prüfung der Eingabefelder durch den Dialogmanager erfolgt nicht. Die Dialogvariablen haben nach der Rückkehr zum Anwendungsprogramm den gleichen Inhalt wie vor dem Aufruf des DISPLAY-Dienstes, außer für eine Liste. Bei der Anzeige einer Liste werden die geänderten Listenfelder bereits beim Blättern in die Dialogvariablen geschrieben. Durch ein CANCEL-Kommando werden nur die augenblicklichen Daten des Listenbereiches nicht verarbeitet. Das Anwendungsprogramm muß auf den Rückkehrcode entsprechend reagieren (z.B. einen Anzeigeschritt zurück). Die Rückkehrcodes sind im Anhang ab Seite 231 aufgelistet.

Angegebene Operanden werden ignoriert.

EXIT - Anwendungsabschnitt beenden

Operation	Operanden
EXIT	

EXIT für eine implizite Box (der Cursor befindet sich in der Box) bewirkt das Entfernen aller impliziten Boxen und führt zur Anzeige des Grundbildes. Das Anwendungsprogramm erhält davon keine Information.

Zum Schutz gegen versehentlich falsches Positionieren des Cursors werden implizite Boxen auch dann entfernt, wenn sich der Cursor im Grundbild oder in einer expliziten Box befindet. Das Kommando wirkt in diesem Fall nicht für das Grundbild oder für eine explizite Box. Erst nach dem Entfernen aller impliziten Boxen ist ein erneutes EXIT für das Grundbild wirksam.

EXIT für eine explizite Box oder das Grundbild soll einen Anwendungsabschnitt oder eine Anwendung beenden. Der DISPLAY-Aufruf wird beendet und liefert einen entsprechenden Rückkehrcode an das Anwendungsprogramm (siehe Seite 231ff). Das Anwendungsprogramm muß die Beendigung eines Anwendungsabschnittes realisieren. Wird das EXIT-Kommando in Verbindung mit geänderten Eingabedaten gegeben, so werden die Eingabefelder entsprechend den dafür vereinbarten Prüfbedingungen geprüft. Wird dabei ein Fehler erkannt, erfolgt eine Fehlermeldung durch den Dialogmanager und das EXIT-Kommando wird ignoriert. Es kann nach der Korrektur der fehlerhaften Daten nochmals eingegeben werden. Sind die Eingabedaten ohne Fehler, werden die entsprechenden Dialogvariablen aktualisiert.

Angegebene Operanden werden ignoriert.

EXTHELP - Erweiterte Hilfe anfordern

Operation	Operanden
EXTHELP	

Die erweiterte Hilfe liefert Informationen zur Maske, aus der heraus die Hilfe angefordert wurde. Alle eventuell auf dem Bildschirm vorhandenen Hilfen werden zuvor entfernt.

Angegebene Operanden werden ignoriert.

EXTHELP in der KEYSHELP-Maske fordert die Hilfe zur Tastenbelegung an.

HARDCOPY - Bildschirminhalt auf Drucker ausgeben

Operation	Operanden
HARDCOPY	

Der aktuelle Bildschirminhalt wird auf einen lokalen Hardcopy-Drucker ausgegeben. Die Zeichenfolge "HARDCOPY" wird bei der Kopie aus dem Kommandobereich entfernt.

Angegebene Operanden werden ignoriert.

HELP - Hilfe anfordern

Operation	Operanden
HELP	kommando

HELP liefert feldbezogene Hilfen zu einem Eingabefeld, einem Auswahlfeld, einem Menütitel, einem markierbaren Feld oder einem Kommandoingabefeld. Für geschützte Ausgabefelder und Textfelder kann auch eine Hilfe definiert werden. Voraussetzung für diese Anzeige ist, daß der Cursor auf geschützte Felder positioniert werden kann (NOAUTOTAB).

Je nach Art des Feldes gilt folgendes:

- Steht der Cursor auf einem Einfachauswahlfeld mit einem gültigen Eintrag, dann wird die Hilfe zu diesem Eintrag geliefert.
- Steht der Cursor auf einem leeren Einfachauswahlfeld, dann wird die Hilfe zum ganzen Einfachauswahlfeld (= globale Hilfe) ausgegeben.
- Steht der Cursor in einem Eingabefeld eines Mehrfachauswahlfeldes, dann erscheint die Hilfe zu diesem Eingabefeld. Gibt man dann erneut das Kommando HELP, dann wird die globale Hilfe zum Mehrfachauswahlfeld ausgegeben (falls vorhanden), andernfalls die erweiterte Hilfe.
- Steht der Cursor im Kommandoingabefeld und ist dieses leer oder enthält ein Anwendungskommando, dann wird die globale Hilfe zum Kommandofeld angezeigt. Ist diese nicht vorhanden, dann gibt FHS die Standardhilfe zum Kommandobereich aus. Steht im Kommandofeld ein FHS-Kommando, dann wird die Hilfe zu diesem Kommando ausgegeben.
- Existiert zu einem normalen Eingabefeld keine Hilfe, dann wird die erweiterte Hilfe zum Format ausgegeben. Ist auch diese nicht vorhanden, dann gibt FHS eine entsprechende Meldung aus.

HELP "kommando" liefert Hilfen zum Systemkommando "kommando". Ist "kommando" kein Systemkommando, so wird die Hilfe zur Kommandozeile angezeigt. Der Cursor muß sich im Kommandoingabefeld befinden.

HELPHelp - Übersicht über das Hilfesystem anfordern

Operation	Operanden
HELPHelp	

Die Hilfe zum Hilfesystem liefert Informationen zur Nutzung der Hilfefunktionen HELPHelp, EXTHelp und KEYSHELP sowie zum Hilfesystem.

Angegebene Operanden werden ignoriert.

INDEX - Anzeigen von Begriffen zu FHS-DM

Operation	Operanden
INDEX	

INDEX bewirkt die Ausgabe eines Hilfeformates (IDHAIDX). In diesem stehen FHS-DM-Begriffe, die als Querverweise vereinbart sind. Positionieren Sie den Cursor auf einen dieser Begriffe und drücken die ENTER- oder die HELPH-Taste, so wird die Hilfe zu diesem Begriff angezeigt. Diese Hilfe (IDHAIDX) können Sie um eigene Begriffe erweitern.

Angegebene Operanden werden ignoriert.

KEYAREA - Tastenbelegung ein/ausschalten

Mit KEYAREA können Sie die Anzeige der Tastenbelegung ein- oder ausschalten. Eine Einstellung bleibt bis zur nächsten Änderung wirksam und wird als Systemvariable SYS-KEY-AREA im Profilpool gespeichert.

Operation	Operanden
KEYAREA	[ON / OFF]

ON Anzeige für Tastenbelegung einschalten

OFF Anzeige für Tastenbelegung ausschalten. Dadurch steht mehr Platz für die Ausgabe von Hilfetexten zur Verfügung.

Wird kein Operand angegeben, wird die Anzeige umgeschaltet.

KEYSHELP - Hilfe zur Tastenbelegung

Tastenbelegung wird in Form einer Tabelle ausgegeben, die den Namen der Taste, das zugeordnete Kommando und eine Kurzbezeichnung der Taste enthält.

Operation	Operanden
KEYSHELP	[SHORT]

SHORT es wird eine Kurzform der Tastenbelegung angezeigt

PANELID - Formatnamen und Meldungskennzeichen anzeigen

Mit dem Kommando PANELID schalten Sie die Anzeige von Formatnamen und Meldungskennzeichen ein oder aus. Eine Einstellung bleibt bis zur nächsten Änderung wirksam und wird als Systemvariable SYS-PANEL-ID im Profilpool gespeichert.

Operation	Operanden
PANELID	[ON / OFF]

ON Anzeige einschalten

OFF Anzeige ausschalten

Wird kein Operand angegeben, wird die Anzeige umgeschaltet.

RESHOW - Wiederanzeige einer Maske

Dieses Kommando ist immer an eine K-Taste gekoppelt, es kann nicht direkt in ein Format eingegeben werden. Standardmäßig wird diese Funktion durch das Betätigen der Taste K3 ausgeführt, siehe auch Abschnitt "Tastenbelegungstabellen" auf Seite 77ff.

Die Funktion Wiederanzeige veranlaßt die Wiederholung der letzten Ausgabe der Maske. Eingaben, die bereits am Bildschirm gemacht wurden, gehen dabei verloren.

Operation	Operanden
K-Taste	

Die Funktion Wiederanzeige ist kein eigentliches Systemkommando, da es für sie keine Zeichenkette gibt, die in das Kommandoeingabefeld eingetragen werden kann. Die Funktion Wiederanzeige ist immer an eine K-Taste (i.a. K3) gekoppelt und kann nur durch Betätigen dieser K-Taste ausgeführt werden. In der Tastenbelegungstabelle wird die Funktion durch die Zeichenfolge "RESHOW" der K-Taste zugeordnet.

RMSG - Wiederanzeige einer Meldung

Die Funktion RMSG veranlaßt die nochmalige Anzeige einer Meldung, die von der Anwendung im DISPLAY-Kommando angefordert wurde. Wird eine Fehlermaske angezeigt, wird die Meldung zum Fehlercode ausgegeben.

Operation	Operanden
RMSG	

Angegebene Operanden werden ignoriert.

SETP - P-Tasten zuordnen

SETP ordnet P-Tasten die Funktion der Funktionstasten mit der gleichen Kennnummer zu bzw. löst diese Zuordnung. Eine Zuordnung bleibt bis zur nächsten Änderung wirksam und wird als Systemvariable SYS-P-KEYS-SETTING im Profilpool gespeichert

Operation	Operanden
SETP	Pn / (Pn, ... ,Pm) / (Pn-Pm) ON / OFF [, ...]

Pn	Die P-Taste mit der Nummer n.
(Pn,...,Pm)	Alle aufgezählten P-Tasten Pn, ..., Pm
(Pn-Pm)	Alle P-Tasten von Pn bis Pm
ON	Die Zuordnung für die angegebenen Tasten wird wirksam.
OFF	Die Zuordnung für die angegebenen Tasten wird gelöst

Die Operanden von SETP können mehrmals angegeben werden. Sie müssen dann jeweils durch ein Komma getrennt sein, siehe folgendes Beispiel. Jedes Komma kann von beliebig vielen Leerzeichen umgeben sein.

Beispiel

```
SETP P1 ON,      P2 OFF,  (P3,P7,P8) ON,  (P4-P6) OFF
```

Nach diesem SETP-Kommando gilt folgende Zuordnung:

```
P1 - F1      P3 - F3      P7 - F7      P8 - F8
```

Die Zuordnung von P2 sowie P4, P5 und P6 wurde gelöst.

Blätterkommandos

Diese Kommandos bewirken ein Verschieben eines Listenbereiches einer Maske, oder eines Hilfetextes.

Die Zeichen "++" und "--" bewirken ein Positionieren auf das Ende in der angegebenen Richtung.

Format und Operandenbeschreibung

Operation	Operanden
FORWARD	[zahl]
BACKWARD	[zahl]
+	[zahl]
++	
-	[zahl]
--	

Der Operand zahl gibt die Zeilenanzahl an, um die der Listenbereich verschoben wird. Wird zahl nicht angegeben, wird bei FORWARD die unterste Listenzeile die oberste und bei BACKWARD die oberste zur untersten Listenzeile.

Außerdem ermöglicht der Dialogmanager ein cursorabhängiges Verschieben. Befindet sich der Cursor in einer Zeile einer Liste oder im Hilfetext, so wird bei "+" oder FORWARD diese Zeile erste Zeile im Listenbereich oder im Arbeitsbereich der Hilfemaske, bei "-" oder BACKWARD letzte Zeile.

SYS - Ausführen von BS2000-Kommandos

Das als Operand angegebene BS2000-Kommando wird ausgeführt. Anschließend wird das Bildschirmbild so wiederhergestellt wie es vor der Kommandoausführung war.

Operation	Operanden
SYS	BS2000-kommando

Als Operand kann ein BS2000-Kommando angegeben werden, das vom BS2000-Makro CMD unterstützt wird. Es sollte kein START-Kommando verwendet werden, da dadurch das Anwendungsprogramm abgebrochen wird.

3.10 Tastenbelegungstabellen

Tastenbelegungstabellen werden mit dem IFG erzeugt und in der Formatbibliothek als spezielle Formate (KEY-Formate) gespeichert. Bei der Definition eines Formates kann jedem Format der Name eines KEY-Formates zugeordnet werden. Wenn kein Name eines KEY-Formates angegeben wird, erfolgt die Zuordnung eines Standard-KEY-Formates. Damit ist jedem Format eine Tastenbelegungstabelle zugeordnet.

Eine Tastenbelegungstabelle enthält für jede Funktionstaste folgende Angaben:

Fxx [kommando] [bezeichnung]

"Fxx" bezeichnet die Funktionstaste, "kommando" ist das dieser Taste zugeordnete Kommando und "bezeichnung" ist eine frei wählbare Kurzbezeichnung (maximal 12 Zeichen) für diese Taste.

Durch den Dialogmanager wird die einem Format zugeordnete Tastenbelegungstabelle im Tastenbelegungsbereich der Maske angezeigt. Der Tastenbelegungsbereich ist wahlweise und besteht aus maximal zwei Zeilen. Die Anzeige erfolgt in der Form "Fxx=bezeichnung".

Wenn in der Tastenbelegungstabelle keine Kurzbezeichnung für eine Taste angegeben wurde, wird diese Taste nicht angezeigt. Damit hat der Anwendungsprogrammierer die Möglichkeit, nur wichtige Tasten im Tastenbelegungsbereich anzuzeigen.

Auch wenn nicht alle Tastenbelegungen im Tastenbelegungsbereich angezeigt werden können, sind alle Tastenzuordnungen wirksam. Dies gilt auch, wenn kein Tastenbereich für ein Format definiert wurde. Die wirksame Tastenbelegung kann man sich durch das Systemkommando KEYSHELP als Tabelle anzeigen lassen.

Paßt die Anzeige aller Tasten der Tastenbelegungstabelle nicht in den Tastenbelegungsbe- reich, werden durch den Dialogmanager nach der letzten möglichen vollständigen Eintra- gung maximal drei Punkte gesetzt.

Standard-Tastenbelegungstabellen

Zusammen mit dem Dialogmanager werden Standard-Tastenbelegungstabellen ausgeliefert. Folgende KEY-Formate werden in englischer und deutscher Sprache bereitgestellt:

IDHKEYS	Standard-KEY-Format
IDHKDER	KEY-Format für Fehlerbehandlung im Dialog
IDHKEYA	KEY-Format für Vollbild mit Menüleiste
IDHKEYF	KEY-Format für feldbezogene Hilfe
IDHKEYE	KEY-Format für erweiterte Hilfe
IDHKEYH	KEY-Format für Hilfe zur Hilfe
IDHKEYK	KEY-Format für Hilfe zur Tastenbelegung
IDHKEYM	KEY-Format für Meldungen in einer Box
IDHKEYN	KEY-Format für Meldeboxen ohne Hilfe
IDHKEYI	KEY-Format für Indexformat
IDHKEYU	KEY-Format für Boxen

Die Tabellen der deutschen Version haben folgenden Inhalt:

IDHKEYS - allgemeine Formate

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	Beenden
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKDER - Key-Format für Fehlerbehandlung im Dialog

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	Beenden
F4	HARDCOPY	Drucken
F5	RMSG	Hole Meldung
F6	ABORT	
F9	KEYSHELP	Tasten
F11	INDEX	
F12	CANCEL	Abbrechen
K1	ABORT	
K3	RESHOW	

IDHKEYA - Key-Format für Vollbild mit Menüleiste

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	Beenden
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F10	ACTIONS	Menue
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYF - Feldbezogene Hilfe

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F2	EXTHELP	erweiterte Hilfe
F3	EXIT	
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F9	KEYSHELP	Tasten
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYE - Erweiterte Hilfe

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F9	KEYSHELP	Tasten
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYH - Hilfe zur Hilfe

Taste	Kommando	Kurzbezeichnung
F3	EXIT	
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F9	KEYSHELP	Tasten
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYK - Hilfe zur Tastenbelegung

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F2	EXHELP	erweiterte Hilfe
F3	EXIT	
F4	HARDCOPY	
F7	BACKWARD	-
F8	FORWARD	+
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	Wiederanzeige

IDHKEYM - Meldebox mit Hilfe

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	
F4	HARDCOPY	
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYN - Meldebox ohne Hilfe

Taste	Kommando	Kurzbezeichnung
F1	HELP	
F3	EXIT	
F4	HARDCOPY	
F11	INDEX	
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYI - Indexformat

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	
F4	HARDCOPY	
F7	BACKWARD	
F8	FORWARD	
F9	KEYSHELP	Tasten
F12	CANCEL	Abbrechen
K3	RESHOW	

IDHKEYU - Format für Boxen

Taste	Kommando	Kurzbezeichnung
F1	HELP	Hilfe
F3	EXIT	
F12	CANCEL	Abbrechen
K3	RESHOW	

Tastenbelegung anzeigen

Standardmäßig wird die Belegung der Tasten F1, F3 und F12 im Kommandobereich angezeigt; bei Listenausgabe zusätzlich die Belegung von F7 und F8. Diese Anzeige können Sie mit dem FHS-Kommando KEYAREA aus- und einschalten.

Für jedes DE-Format können Sie sich die Tastenbelegung mit dem FHS-Kommando KEYSHELP ausgeben lassen.

Hinweis:

Reservieren Sie eine K-Taste für RESHOW!

Begründung: Wird eine Maske mehrmals nacheinander ausgegeben, werden von FHS-DM nur die variablen Felder neu ausgegeben (Update-Ausgabe). Überschreibt z.B. eine Operator-Nachricht die Maske, so wird die Maske bei der nächsten Ausgabe nicht neu aufgebaut. Erst durch RESHOW wird die komplette Maske neu ausgegeben (siehe auch Seite 94).

3.11 Meldungen ausgeben

Mit FHS-DM kann die Arbeit des Terminal-Benutzers erleichtert werden, indem er mit DM-Meldungen über bestimmte Ereignisse informiert wird. Diese DM-Meldungen, im folgenden Meldungen genannt, werden entweder im Meldebereich des Formats oder in eigenen Meldungsboxen ausgegeben. FHS unterscheidet implizite und explizite Meldungen.

Implizite Meldungen werden von FHS ausgegeben, ohne daß die Anwendung beteiligt ist. Implizite Meldungen sind z.B. die Meldungen, die FHS beim Prüfen von Eingabefeldern ausgibt.

Explizite Meldungen werden durch das Programm initiiert, indem beim DISPLAY-Dienst ein Meldungsschlüssel und im MSGLOC-Operand ein Feldname angegeben wird.

Meldungen erstellen

Meldungen müssen Sie mit dem IFG erstellen. Dazu rufen Sie im IFG das Bild "Meldungen bearbeiten" auf. In diesem Bild legen Sie den Text der Meldung fest. Der Meldungstext darf maximal 256 Zeichen lang sein. Neben dem Meldungstext definieren Sie noch weitere Eigenschaften:

- Den Meldungsschlüssel in der Form AAAAnnn oder AAAAnnnn. Dabei sind AAAA alphabetische Zeichen (A-Z) und nnn sind Ziffern (0-9). Die Bezeichnungen IDHx sollten Sie nicht als Meldungsschlüssel verwenden, da diese von FHS für eigene Meldungen verwendet werden.
- Den Ausgabeort der Meldung. Dieser kann der Meldebereich eines Formats, eine modale Meldungsbox oder eine nichtmodale Meldungsbox sein. Näheres siehe Abschnitt "Implizite Boxen" auf Seite 17ff.

Falls "Ausgabeort = Meldebereich" definiert wurde, aber die Ausgabe in den Meldebereich nicht möglich ist, dann gibt FHS die Meldung in Form einer nichtmodalen Box aus. Dies kann z.B. vorkommen, wenn der Meldebereich zu klein ist oder kein Meldebereich im Format definiert wurde.

- Den Typ der Meldung: Information/Warnung/Fehler/Gefahr
- Wahlweise den Namen eines Hilfeformats für die Meldung

Aufbereitung des Meldungstextes

Wenn im Meldungstext die Zeichenfolge "%" auftritt, wird der nachfolgende Text ab Spalte 2 der nächsten Zeile in der Meldungsbox fortgesetzt. Durch diese Aufbereitung kann die Breite einer Meldungsbox verringert werden. Bei der Ausgabe der Meldung in den Meldbereich, wird "%" durch ein Leerzeichen ersetzt. Die Zeichenfolge "%" selbst wird entfernt.

"%%%%" bewirkt eine Leerzeile.

Das Zeichen "&" muß durch "&&" im Meldungstext angegeben werden.

Durch FHS-DM wird die Breite einer Meldungsbox dem Meldungstext angepaßt. Die Maskenbreite beträgt minimal 20 Zeichen und maximal 56 Zeichen. Eine Meldungsbox kann maximal 6 Textzeilen besitzen. Eine Maskenbreite von weniger als 56 Zeichen ergibt sich, wenn der Text entsprechend kurz ist, oder wenn durch Aufbereitungszeichen eine Zeilenlänge vorgegeben wird. Wenn der Meldungstext entsprechend lang ist wird die maximal Breite gewählt und der Meldungstext der Boxgröße angepaßt. Eine Trennung erfolgt bei einem Leerzeichen.

Überschreitet der aufbereitete Meldungstext die maximale Größe einer Meldungsbox, so werden Aufbereitungszeichen durch ein Leerzeichen ersetzt und der Text nach je 56 Zeichen umgebrochen und, falls immer noch nicht alles anzeigbar ist, am Ende der letzten Zeile mit "..." beendet.

Wenn FHD-DM im Meldungstext "&name" findet, so wird der aktuelle Inhalt dieser Variablen in den Meldungstext eingefügt.

Implizite Meldungen

Implizite Meldungen werden durch den Dialogmanager ausgegeben, ohne daß das Anwendungsprogramm beteiligt ist. Solche Meldungen werden z.B. beim Prüfen von Eingabedaten ausgegeben. Für implizite Meldungen werden Standard-Meldungsformate ausgeliefert. Diese Formate müssen in die Formatbibliothek der Anwendung kopiert werden oder als alternative Formatbibliothek zur Verfügung gestellt werden. Die Standardmeldungen haben den Meldungsschlüssel IDHxnnn, wobei nnn eine dreistellige Zahl ist, für "x" können die Buchstaben "Z", "A", "D", "U", "P", "V" oder "O" stehen. Der Meldungstext kann durch den Anwender mit dem IFG geändert werden.

Wird ein fehlerhaftes Feld durch den Dialogmanager ermittelt, erfolgt die Ausgabe einer Meldebox bezüglich des ersten fehlerhaften Feldes, falls nicht der Meldebereich als Ausgabeort vereinbart wurde. Sind mehrere Felder fehlerhaft, erfolgt ein Hervorheben (z.B. Unterstreichen) aller dieser Felder. Der Cursor wird in das erste fehlerhafte Feld gestellt. Ein anliegendes Verschiebekommando wird ignoriert. Es kann nach der Fehlerkorrektur wiederholt werden.

Im IFG kann die Nachricht für das Feld wie folgt angegeben werden:

- *NONE oder Leerzeichen: Standardmeldung von FHS
- Meldungskennzeichen: eigene implizite Meldung

Eigene implizite Meldungen können zusätzlich mit folgenden Angaben versehen werden:

&SYS-PAR0	Name des Vollformats, bzw. Name der aktiven expliziten Box.
&SYS-PAR1	Name des aktuellen Feldes
&SYS-PAR2	Inhalt des aktuellen Feldes
&SYS-PAR3	Name des aktuellen Formats

Wird im Meldungstext die Kennzeichnung &SYS-PAR0, &SYS-PAR1, &SYS-PAR2 oder &SYS-PAR3 gefunden, so wird diese Kennzeichnung durch den entsprechenden Wert ersetzt und in den Meldungstext eingefügt. Diese Angabe ist für Meldungen möglich, deren Meldungskennzeichen durch IFG für Prüfungen festgelegt wurden.

Diese Meldungen werden standardmäßig in einer Meldungsbox ausgegeben. Sie können mit dem IFG jedoch den Ausgabeort ändern, so daß die Meldung im Meldebereich des Formats ausgegeben wird.

Explizite Meldungen

Soll durch das Anwendungsprogramm eine Meldung ausgegeben werden, die sich auf den Inhalt eines bestimmten Feldes der Maske (auch für ein Listenfeld einer bestimmten Listenzeile) bezieht, so kann eine Meldebox an dieses Feld plaziert werden. Dazu ist im MSG-Operanden des DISPLAY-Dienstes ein Meldungsschlüssel und im MSGLOC-Operanden der Name des Feldes anzugeben, das für die Positionierung einer Meldebox verwendet werden soll.

Falls bei der Definition der Meldung als Ausgabeort eine Meldebox vereinbart wurde, ergeben sich nachfolgende Möglichkeiten:

- Ist MSGLOC nicht angegeben, erfolgt die Ausgabe der Meldebox bezüglich des durch den Operanden CURSOR angegebenen Feldes. Wenn es sich um ein Listenfeld handelt, kann die Feldangabe mit Index erfolgen. Der Cursor wird entsprechend den Angaben für CURSOR und CSRPOS gesetzt.
- Ist für MSGLOC ein Feldname, der kein Listenfeld beschreibt, angegeben, wird die Meldebox bezüglich dieses Feldes ausgegeben. Die Cursorpositionierung erfolgt dabei entsprechend den Angaben für CURSOR und CSRPOS.
- Ist für MSGLOC der Name eines Listenfeldes angegeben, kann ein Index zur Auswahl der Zeile angegeben werden. Die Meldebox wird dann bezüglich dieses Feldes und der durch den Index beschriebenen Listenzeile ausgegeben. Wird kein Index angegeben, erfolgt die Ausgabe der Meldebox bezüglich des Feldes der obersten Zeile im Listenbereich (siehe auch Listenverarbeitung, TOPINDEX- Variable).
- Als Wert für den MSGLOC-Operanden kann auch eine absolute Position (\$zzz#sss) angegeben werden. Der Wert *CENTRAL bewirkt die Ausgabe der Meldebox in der Mitte des Bildschirms.
- Wird durch Angabe von MSGLOC die angeforderte Cursorposition durch die Meldebox verdeckt, wird MSGLOC ignoriert und die Meldebox entsprechend den Cursorangaben positioniert.
- Ist weder MSGLOC noch die Cursorposition angegeben, wird die Meldung in der Mitte des Bildschirms ausgegeben.

Ob die Meldebox modal oder nichtmodal ausgegeben wird, hängt von der Angabe bei der Meldungsdefinition ab.

Wurde für die auszugebende Meldung als Ausgabeort der Meldebereich vereinbart, wird der MSGLOC-Operand ignoriert.

3.12 Hilfesystem

Mit FHS-DM kann der Anwendungsentwickler für den Terminal-Benutzer ein umfangreiches Hilfesystem erstellen. Dieses Hilfesystem kann speziell auf die jeweilige Anwendung zugeschnitten werden.

Die Hilfeinformationen werden in Form von impliziten Boxen angezeigt und sind selber Formate. Anwendungsspezifische Hilfe-Formate erstellen Sie mit dem IFG, bestimmte Standard-Hilfeformate werden mit FHS ausgeliefert. Die Eigenschaften impliziter Boxen sind auf Seite 17 beschrieben.

Bei Hilfeformaten sind die Blätterkommandos FORWARD und BACKWARD erlaubt, da der Arbeitsbereich so groß sein kann, daß die Information nicht komplett in eine Hilfe-Box paßt. Die Möglichkeiten für den Anwendungsentwickler sind im folgende Abschnitt aufgeführt.

3.12.1 Hilfen, die der Anwendungsentwickler erstellen kann

Der Anwendungsentwickler kann folgende Hilfen erstellen:

- erweiterte Hilfe zum Format
- feldbezogene Hilfe zu Eingabefeldern, Ausgabefeldern und Auswahlfeldern
- globale Hilfe zu Auswahlfeldern und zum Kommandobereich
- Hilfe zu Meldungen
- Querverweise

Erweiterte Hilfe zum Format

Beim Erstellen eines Formats mit dem IFG können Sie dem Format ein Hilfeformat zuordnen, das sie ebenfalls mit dem IFG erstellen. Diese Hilfe wird ausgegeben, wenn der Anwender

- das Kommando EXTHELP eingibt (auch durch eine Funktionstaste ausgelöst), siehe Abschnitt "FHS-Kommandos" auf Seite 64.
- den Cursor in einer Hilfebox positioniert, die für eine feldbezogene Hilfe angefordert wurde, und die Taste "EXTHELP" betätigt
- den Cursor in der Maske oder Dialogbox auf einem Feld positioniert, für das keine feldbezogene Hilfe existiert und "HELP" aktiviert. Bei AUTOTAB (Standardwert) ist nur eine Positionierung in ungeschützten Eingabefeldern oder in geschützten markierten Feldern möglich. Eventuell muß die erweiterte Hilfe wie im ersten Fall angefordert werden.
- den Cursor in ein Feld positioniert, für das bereits eine feldbezogene oder eine globale Hilfe angezeigt wird und dann nochmals HELP drückt.

Ist keine erweiterte Hilfe vereinbart, so wird bei einer Anforderung eine entsprechende Meldung ausgegeben.

Feldbezogene Hilfe

Im IFG können Sie jedem Feld (auch einem geschützten Ausgabefeld) ein Hilfeformat zuordnen.

Wenn Sie eine Hilfe für ein Ausgabefeld vereinbaren, dann muß der Cursor auf dieses Feld positionierbar sein, d.h. das Ausgabefeld muß entweder markierbar sein oder das Format muß das Tabulator-Attribut NO AUTOTAB besitzen.

Der Terminal-Benutzer erhält das Hilfeformat, wenn er den Cursor auf das betreffende Feld positioniert und das Kommando HELP eingibt bzw. die zugeordnete F-Taste drückt. Gehört das Feld zu einem Einfachauswahlfeld, dann muß der Benutzer in das zugehörige Eingabefeld die Nummer der gewünschten Auswahl eintragen und das Kommando HELP geben (trägt er keine Nummer ein, dann erhält er die Hilfe des Einfachauswahlfeldes falls vorhanden).

Globale Hilfe

Eine globale Hilfe steht im Informationsgehalt zwischen der erweiterten Hilfe, die für ein ganzes Format gilt und der feldbezogenen Hilfe, die nur ein einzelnes Feld betrifft. Ein globale Hilfe erzeugen Sie als Hilfeformat mit dem IFG, sie kann folgenden Objekten zugeordnet werden:

- einem Einfachauswahlfeld. Der Benutzer erhält diese globale Hilfe, indem er den Cursor in das *leere* Eingabefeld des Einfachauswahlfeldes positioniert und das Kommando HELP gibt.
Existiert zu einer Auswahl keine feldbezogene Hilfe, so wird die globale Hilfe sofort ausgegeben, andernfalls erst nach dem zweiten HELP.
- einem Mehrfachauswahlfeld. Der Benutzer erhält diese globale Hilfe, wenn er den Cursor in ein beliebiges Eingabefeld des Mehrfachauswahlfeldes positioniert und ein oder zweimal das Kommando HELP gibt. Existiert keine feldbezogene Hilfe, dann erhält er die globale Hilfe sofort, andernfalls erst nach dem zweiten HELP.
- dem Kommandobereich eines Formats. Ausgelöst wird diese Hilfe durch das Kommando HELP, wenn der Cursor im Kommandofeld steht und das Kommandofeld entweder leer ist oder ein Anwendungskommando enthält.
- dem Listenbereich eines Formats. Der Benutzer erhält diese Hilfe, wenn er den Cursor in ein Listenfeld positioniert und ein- oder zweimal das Kommando HELP gibt. Existiert keine feldbezogene Hilfe, dann erhält er die globale Hilfe sofort, andernfalls erst nach dem zweiten HELP.
Die globale Hilfe wird auch dann ausgegeben, wenn der Cursor im Listenbereich außerhalb eines Feldes positioniert ist und HELP gedrückt wird.

Die Hilfe wird in einer Box ausgegeben. Diese Box liegt bei Auswahlfeldern möglichst unterhalb des betreffenden Eingabefeldes, beim Kommandobereich immer oberhalb des Kommandofeldes.

Hilfe zu Meldungen

Wollen Sie für eine Meldung eine Hilfe definieren, dann geben Sie beim Erstellen des Meldungsformats mit dem IFG das gewünschte Hilfeformat explizit an.

Dieses Hilfeformat wird ausgegeben, wenn der Terminal-Benutzer nach der Meldungsausgabe die Taste HELP drückt. Bei Meldungsausgabe in einer Box muß sich der Cursor innerhalb der Meldungsbox befinden.

Querverweise

Querverweise sind Hilfen zu Textfeldern von Hilfeformaten. Sie entsprechen feldbezogenen Hilfen in Aktionsformaten. Der Anwender erhält die Informationen zu einem Querverweis, indem er den Cursor auf ein Textfeld, zu dem eine Hilfe festgelegt wurde, positioniert und die ENTER-Taste drückt oder ein HELP-Kommando auslöst. Dann wird eine neue Hilfebox ausgegeben, in der die Hilfe-Informationen angezeigt werden. In dieser Box können wieder neue Querverweise enthalten sein.

Wird eine Hierarchie von Querverweisen angefordert, erfolgt die Ausgabe der entsprechenden Hilfeboxen immer mit der gleichen Boxposition. Damit wird erreicht, daß nur die oberste Box sichtbar ist, falls die Querverweise alle die gleiche Größe haben. Dies sollten Sie beim Entwerfen der Hilfeformate beachten.

INDEX

Mit dem Kommando INDEX können Sie die mit der FHS-Dialogerweiterung ausgelieferten Hilfen aufsuchen. Nach Eingabe des Kommandos INDEX erhalten Sie eine Hilfe-Maske mit Begriffen zur FHS-Dialogerweiterung angezeigt. Wenn Sie den Cursor auf einen Begriff positionieren und die ENTER-Taste drücken, wird eine neue Hilfebox ausgegeben, in der die Hilfe-Informationen angezeigt werden. In dieser Box können wieder neue Querverweise enthalten sein.

3.12.2 Hilfen, die FHS selbst anbietet

Mit FHS werden eine Reihe von Standard-Hilfeformaten ausgeliefert. Die Namen dieser Hilfeformate beginnen mit IDHx. Diese Standard-Hilfeformate befinden sich in der Standard-Bibliothek SYSFHS.FHS.081.FHS-DM.D (deutsche Formate). Ein LINK mit BLSLIBxx genügt, um die Formate einzubinden (xx = 00 - 99). Dieser LINK kann z.B. folgendermaßen aussehen:

```
SET-FILE-LINK FILE-NAME=$.SYSFHS.FHS.081.FHS-DM.D, LINK-NAME=BLSLIB00.
```

FHS bietet dem Terminal-Benutzer folgende Hilfen:

- Hilfe zu Meldungen von FHS
- Hilfen zu FHS-Kommandos
- Hilfe zur Tastenbelegung
- Hilfe zum Hilfesystem

Hilfe zu Meldungen

Die Hilfe zu Meldungen von FHS wird behandelt wie die Hilfe zu anwendungsspezifischen Meldungen, siehe oben.

Hilfe zu FHS-Kommandos

Hilfe zu einem bestimmten FHS-Kommando kann der Benutzer anfordern, indem er in das Kommandofeld entweder HELP "fhs-kommando" einträgt und die ENTER-Taste drückt oder indem er nur das Kommando einträgt und die Taste für HELP drückt. Die Hilfe zu FHS-Kommandos erscheint als globale Hilfe. Existiert eine anwendungseigene Hilfe für ein FHS-Kommando, dann erhält der Benutzer beim ersten HELP die anwendungseigene und beim zweiten HELP die FHS-Hilfe.

Hilfe zur Tastenbelegung

Die Hilfe zur Tastenbelegung kann auf zwei Arten aktiviert werden:

- der Terminal-Benutzer gibt das FHS-Kommando KEYSHELP oder drückt die zugeordnete F-Taste
- der Cursor wird in die Tastenbelegungsanzeige oder auf den unteren Rahmen der Box positioniert und das HELP-Kommando wird gegeben

FHS gibt die Belegung aller F- und K-Tasten in einer Tabelle aus, siehe Beschreibung des Kommandos KEYSHELP. Es werden dabei nur die Tasten angezeigt, für die Angaben bei der Herstellung des KEY-Formates eingetragen wurden. Sind nicht alle Tastenbelegungen sichtbar, kann vorwärts und rückwärts geblättert werden.

Wenn die Tastenbelegung angezeigt wird, kann weitere Hilfe angefordert werden:

- Wurde bei der Definition der Tastenbelegung ein Hilfeformat vereinbart, so liefert das Kommando EXTHELP die Anzeige des Hilfeformats. Befindet sich der Cursor außerhalb des Kommandobereichs, so wird durch das HELP-Kommando ebenfalls diese Hilfe angezeigt.
- Wurde kein Hilfeformat vereinbart, so liefert das Kommando EXTHELP die Meldung "Es existiert keine Hilfe". Befindet sich der Cursor außerhalb des Kommandobereichs, so wird durch das HELP-Kommando die Hilfe zum Kommando KEYSHELP angezeigt.

Das zugehörigen Standardhilfeformat (IDHTKHP) können Sie mit dem IFG in begrenztem Umfang modifizieren, z.B. indem Sie die Textfelder in Überschriften, die Anzeigeeigenschaften oder den in < > eingeschlossene Text für die Erläuterung von "*" ändern. Dieser Text steht im Kommandofeld der ersten Datenzeile von IDHTKHP.

Die Struktur dieses Formates dürfen Sie nicht ändern. Wird die Struktur des Formates IDHTKHP durch weitergehende Änderungen zerstört, kann die Funktion KEYSHELP nicht mehr realisiert werden.

Hilfe zum Hilfesystem

Die Hilfe zum Hilfesystem liefert Informationen darüber, welche Hilfen vorhanden sind und wie sie benutzt werden können. Aktiviert wird diese Hilfe mit dem FHS-Kommando HELPHelp bzw. der entsprechenden F-Taste.

3.13 Positionierung des Cursors

Die Positionierung des Cursors erfolgt bei Eingaben und Ausgaben unterschiedlich.

Cursorpositionierung bei Ausgabe

Die Cursorposition kann durch die Operanden CURSOR und CSRPOS des DISPLAY-Dienstes gesteuert werden. Ist der Operand CURSOR nicht angegeben, wird der Cursor in das erste Eingabefeld gestellt. Enthält das Format kein Eingabefeld, wird der Cursor auf die erste Zeile, erste Spalte des Bildschirms gestellt.

Es gelten folgende Regeln:

1. CURSOR enthält eine absolute Position (\$zzz#sss). In diesem Fall werden die Angaben für CSRPOS ignoriert.
2. CURSOR enthält einen Feldnamen. Ist der Feldname nicht gültig, das heißt er existiert nicht im aktuellen Format, endet der DISPLAY-Dienst mit einem Fehler. Bezieht sich der Feldname auf eine Listenzeile, kann ein Index angegeben werden. Wird in diesem Fall kein Index angegeben, wird die oberste im Listenbereich angezeigte Zeile angenommen (siehe auch Listenverarbeitung, TOPINDEX-Variable). Mit CSRPOS wird die Verschiebung im Feld angegeben. Ist der Wert von CSRPOS ungültig, wird er auf 1 gesetzt.

Cursorangaben bei der Eingabe

Die Cursorposition wird in den Systemvariablen SYS-CURSOR-FIELD, SYS-CURSOR-POS und SYS-CURSOR-INDEX zurückgemeldet. Steht der Cursor auf einem unbenannten Feld (Textfeld) oder im Zwischenraum zwischen den Feldern, enthält SYS-CURSOR-FIELD die absolute Cursorposition (\$zzz#sss). SYS-CURSOR-POS und SYS-CURSOR-INDEX enthalten in diesem Fall undefinierte Werte

3.14 Differenzausgabe

Wenn nacheinander dasselbe Format ausgegeben wird und keine impliziten Aktionen des Dialogmanagers (z.B. Hilfeanforderungen, Meldungen) durchgeführt worden sind, wird zum Bildschirm eine Differenzausgabe ("Update-Ausgabe") gesendet, um die zu übertragene Nachricht zu verkürzen. Es werden nur die veränderbaren Felder neu ausgegeben (Ein- und Ausgabefelder). Textfelder werden nicht übertragen.

Normalerweise erkennen die Anzeigendienste, ob eine Differenzausgabe möglich ist. Wenn jedoch Fremdausgaben (z.B. Operatornachrichten) eingeschoben werden, kann die Kontrolle nicht durch die Anzeigendienste erfolgen.

Wenn die Zwischenausgaben durch das Anwendungsprogramm erzeugt werden (direkt oder indirekt durch den Aufruf von Systemprogrammen), kann mittels CONTROL-Dienst mit den Operanden REFRESH oder ACK eine Differenzausgabe verhindert werden.

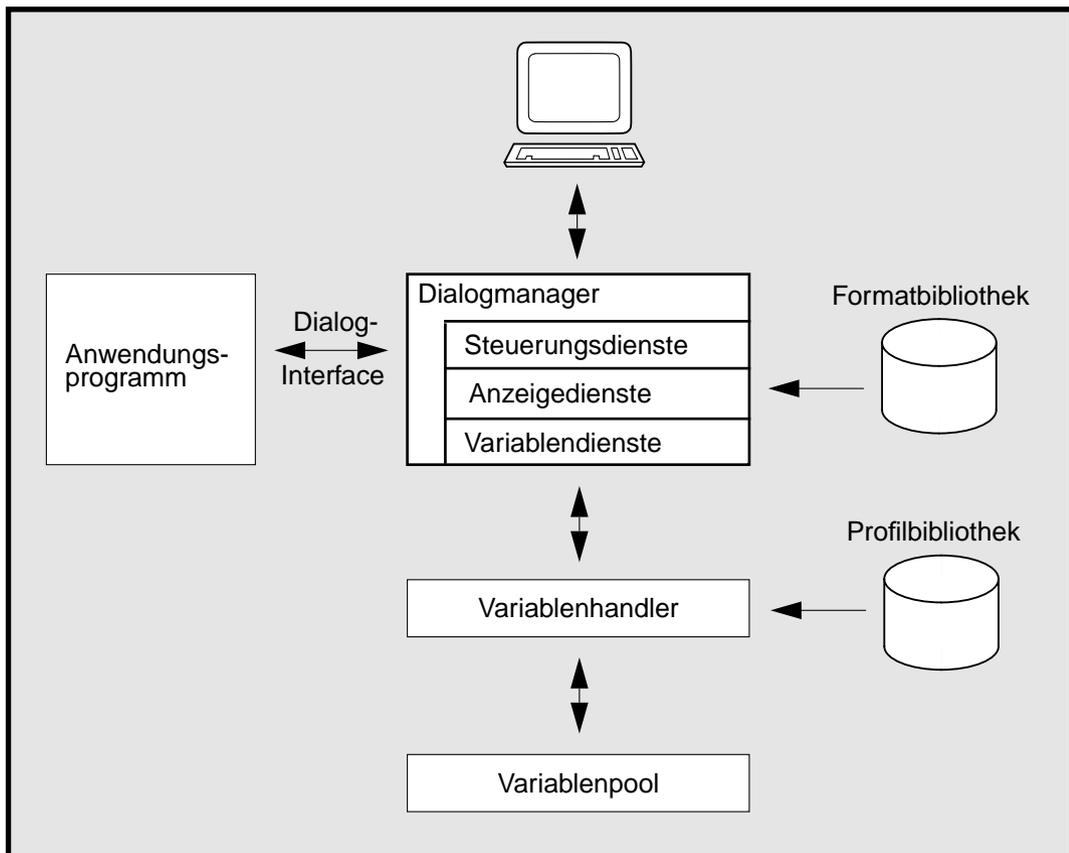
Falls nicht erwartete Nachrichten dazwischen geschoben werden, z.B. Nachrichten des Operators, muß in dem Tastenbelegungsformat eine K-Taste (Empfehlung:K3) mit RESHOW belegt werden. Der Terminal-Benutzer kann dann durch das Drücken dieser K-Taste erreichen, daß das Bild vollständig neu aufgebaut wird.

Der DISPLAY-Dienst erkennt, wenn Ausgaben zu unterschiedlichen DMCOMM eines Dialog-Komplexes angefordert werden. Nicht erkannt werden kann eine Ausgabe in einem zweiten Dialog-Komplex. In diesem Fall ist prinzipiell der CONTROL-Dienst mit dem Operanden REFRESH aufzurufen.

4 Programm-Schnittstelle für TIAM-Anwendungsprogramme

Der Dialogmanager bietet verschiedene Dialogelemente und Dialogdienste, die die Erstellung von Dialoganwendungen erheblich vereinfachen und beschleunigen. Die eigentliche Verarbeitung führt das Anwendungsprogramm aus. Es nutzt dabei die Dialogdienste des Dialogmanagers um Masken, Meldungen und Hilfetexte an der Datensichtstation anzuzeigen.

Im folgenden Bild werden die Komponenten des Dialogmanagers im Überblick dargestellt.



Es gibt folgende Dialogelemente:

- Formate
- Meldungen
- Hilfetexte
- Tastenbelegungstabellen

Diese Dialogelemente werden mit Hilfe des interaktiven Formatgenerators (IFG) erzeugt und in einer Formatbibliothek gespeichert. Um Dialogelemente mit dem Dialogmanager verarbeiten zu können, müssen diese mit IFG ab Version 8.1 erstellt sein.

Es gibt folgende Dialogdienste:

Anzeigedienste

Gestaltung der Bildschirmoberfläche gemäß dem SNI-Alpha-Styleguide

Name	Zweck
DISPLAY	Anzeigen eines Formates und/oder einer Meldung
ATTR	Definieren dynamischer Feldattribute
ADDDPOP	Initialisieren einer Dialogbox
REMPPOP	Entfernen einer Dialogbox

Allgemeine Variablendienste

Definieren, Verändern und Löschen von Dialogvariablen durch das Anwendungsprogramm.

Name	Zweck
VCOPY	Lesen von Dialogvariablen
VDEFINE	Definieren von expliziten Dialogvariablen
VDELETE	Löschen von Dialogvariablen im Funktionspool
VREPLACE	Ersetzen von Dialogvariablen

Variablendienste

Steuerung eines Anwendungsprofils

Name	Zweck
VPUT	Schreiben der Dialogvariablen in den Profilpool / SDF-P-Pool
VGET	Lesen der Dialogvariablen aus dem Profilpool / SDF-P-Pool
VERASE	Löschen von Dialogvariablen im Profilpool

Steuerungsdienste

für das Initialisieren und Beenden der Dienstbereitschaft des Dialogmanagers für ein Anwendungsprogramm

Name	Zweck
DMOPEN	Beginn der Nutzung von Dialogdiensten
DMCLOSE	Ende der Nutzung von Dialogdiensten
CONTROL	Einstellen von Arbeitsmodi

Sonstige Dienste

Unterstützung von Funktionstasten	
Systemkommandos des Dialogmanagers	Abbruch der Formatanzeige Anzeige des Formatnamens und des Meldungsschlüssels Bildschirmkopie Ausführen von BS2000-Kommandos Anfordern von Hilfe Anzeige der Tastenbelegung
Bereitstellung von Systemvariablen	
Unterstützung für Anwendungskommandos	

4.1 Variablen des Dialogmanagers

In diesem Handbuch wird zwischen folgenden Variablentypen unterschieden:

- Dialogvariable (siehe folgende Abschnitte)
- Programmvariable (Variablen, die in einem Anwendungsprogramm verwendet werden)
- Prozedurvariable / S-Variable (Variablen, die in einer SDF-P-Prozedur verwendet werden)

4.1.1 Dialogvariable

Dialogvariable dienen dem Datenaustausch zwischen Anwendungsprogramm, Dialogmanager und Dialogelementen. Durch den Einsatz der Variablendienste des Dialogmanagers können Dialogvariablen durch das Anwendungsprogramm definiert, mit Werten belegt, verändert und gelöscht werden. Es können auch Verbindungen von Dialogvariablen zu Programmvariablen definiert werden.

Von besonderer Bedeutung ist der Datenaustausch zwischen Maskenfeldern und Daten des Anwendungsprogramms. Bei der Formatdefinition mittels IFG werden den Ein- und Ausgabefeldern einer Maske Namen von Dialogvariablen zugeordnet. Bei der Anzeige einer Maske durch den Dialogmanager werden in den Maskenfeldern die Werte der zugeordneten Dialogvariablen angezeigt. Bei der Eingabe von Daten in ein Eingabefeld der Maske werden diese als Wert der entsprechenden Dialogvariable gespeichert.

Außer in Masken können Dialogvariable in den Operanden beim Aufruf eines Dialogmanagerdienstes und in Meldetexten auftreten. Dabei wird eine Dialogvariable durch ihren Wert ersetzt und danach der Dienst ausgeführt oder die Meldung ausgegeben.

Die Dialogvariablen des Anwendungsprofils werden über die gegenwärtige Nutzung hinaus aufbewahrt und stehen beim nächsten Start des Dialogprogramms (durch denselben Terminal-Benutzer) sofort wieder zur Verfügung.

Dialogvariable können auch zur Kommunikation verschiedener Anwendungsprogrammteile untereinander benutzt werden.

Vorteile von Dialogvariablen

Bis FHS V8.0 benötigten FHS-Anwendungen zum Austausch von Daten zwischen Anwendungsprogramm und Maskenfeldern vordefinierte Datenbereiche (Adressierungshilfen). Die Verwendung von Dialogvariablen für den Datentransport hat folgende Vorteile:

- Entkopplung der im Anwendungsprogramm bereitgestellten Daten von der Anordnung und Gestaltung bei der Präsentation. Die Definition der Programmdaten kann ohne Zwang durch die Formatbeschreibung (Reihenfolge der Maskenfelder) optimal für die jeweilige Programmiersprache erfolgen.
- Es ist nicht nötig für jede Formatanzeige die Daten in einem Datenbereich bereitzustellen. Der Variablenpool kann zu unterschiedlichen Zeiten gefüllt werden und steht jeder Formatanzeige zur Verfügung. Damit entfällt die zeitliche Abhängigkeit der Datenbereitstellung.
- Eine komfortable Listenanzeige ist mit Dialogvariablen leicht realisierbar. Die Listenanzeige ist ein wichtiges Element der Präsentation.
- Mit Dialogvariablen können variable Texte in Meldungen und in Hilfetexten ersetzt werden. Mit Adressierungshilfen ist dies nicht möglich, da der Zeitpunkt des Ersetzens dem Anwendungsprogramm nicht bekannt ist.
- Dialogvariable unterstützen eine automatische Typumwandlung der anzuzeigenden Daten (z.B. kann eine Charakter-Zeichenkette im Programm in einem Maskenfeld ausgegeben werden). Mit Adressierungshilfen muß das Anwendungsprogramm diese Umwandlung selbst durchführen.
- Das permanente Speichern von Dialogvariablen (Gedächtnis) ist auf einfache Weise möglich.
- Implizite Dialogvariable (z.B. nur im Format angegeben) und deren Werteübergabe z.B. an ein anderes Format, kann ohne Code im Anwendungsprogramm realisiert werden.
- Durch FHS-Systemvariable sind Systemdienste (z.B. Datum, Uhrzeit) auf einfache Weise dem Anwendungsprogramm oder der Maske zugänglich.
- Kommunikation einzelner Programmteile untereinander.

Arten von Dialogvariablen

Entsprechend ihrer Speicherstruktur können Dialogvariable unterschieden werden in:

- einfache Dialogvariable
- Struktur von einfachen Dialogvariablen
- Array von Dialogvariablen

Einfache Dialogvariable können nicht weiter unterteilt werden, sie bestehen aus einem Element.

Struktur von einfachen Dialogvariablen:

Eine Struktur besteht aus einfachen Dialogvariablen, die unterschiedlich im Datentyp und/oder in der Länge sind. Die Elemente einer Struktur müssen aufeinanderfolgenden Speicherplatz belegen. Die Dialogvariablen einer Struktur werden durch ihren Namen angesprochen. Strukturen von Dialogvariablen müssen explizite Dialogvariable sein.

Beispiel zum Aufbau einer Struktur:

NAME, VORNAME und ORT sind die Namen der Dialogvariablen

NAME	VORNAME	ORT
------	---------	-----

Array von Dialogvariablen

Ein Array besteht aus einem Array-Element, das sich mehrfach wiederholt. Die Array-Elemente müssen aufeinanderfolgenden Speicherplatz belegen. Ein Array-Element kann entweder eine einfache Dialogvariable oder eine Struktur von einfachen Dialogvariablen sein. Die Dialogvariablen eines Array-Elementes werden über ihren Namen und einen ganzzahligen positiven Index angesprochen.

Arrays haben eine besondere Bedeutung für die Listenanzeige des Dialogmanagers. Arrays von Dialogvariablen können nur explizite Dialogvariable sein.

Beispiele zum Aufbau von Arrays:

Array von einfachen Dialogvariablen

NAME	Index=1
NAME	Index=2
NAME	Index=3

Array von Strukturen

NAME	VORNAME	ORT	Index=1
NAME	VORNAME	ORT	Index=2
NAME	VORNAME	ORT	Index=3

Eine Dialogvariable hat folgende Eigenschaften:

- einen Namen
- einen Wert in einer bestimmten Länge
- einen Datentyp
- eventuell einen Wiederholungsfaktor

Namensbildung von Dialogvariablen

Der Zugriff zu Dialogvariablen erfolgt nur über deren Namen, wodurch eine große Unabhängigkeit zwischen dem Anwendungsprogramm und den Dialogelementen (z.B. Maske) gewährleistet ist.

Für die Namensbildung von Dialogvariablen gelten folgende Regeln:

- maximal 255 Zeichen
- erstes Zeichen muß ein Buchstabe (A- Z) sein.
- als nachfolgende Zeichen können verwendet werden:
A-Z, 0- 9, - (Bindestrich),.(Punkt)

Nach Bindestrich und Punkt darf nicht wieder Bindestrich oder Punkt stehen. Das letzte Zeichen des Namens darf nicht der Bindestrich oder der Punkt sein.

- Klein- und Großbuchstaben werden nicht unterschieden

Die Namensbildung von Dialogvariablen entspricht im wesentlichen den Syntaxregeln für Variablennamen in SDF-P-Prozeduren. Dadurch soll erreicht werden, daß die gleiche Maske sowohl in einer S-Prozedur als auch in einem Programm genutzt werden kann.

Ein Unterschied besteht in der Bedeutung des Punktes. Ein Punkt im Namen einer Dialogvariablen hat keine spezielle Bedeutung im Gegensatz zu SDF-P-Variablen. Dort dient er zur Kennzeichnung von Strukturen.

Einem derartigen Namen kann eine Indexangabe folgen.

Indexangabe

Eine Indexangabe ist durch ein "#" gekennzeichnet dem ein ganzzahliger Wert (1 bis 32767, ohne Vorzeichen, ohne Zifferntrenner) oder der Name einer einfachen Dialogvariablen folgt, dessen Wert in eine 4-Byte positive binäre Ganzzahl umwandelbar ist (Typ FIXEDS).

Beim Lesen oder Schreiben einer Dialogvariablen wird die Zeichenfolge analysiert, die dem Indexsymbol # folgt: Ist das erste Zeichen eine Ziffer wird die Zeichenfolge als direkte Indexangabe interpretiert.

Ist das erste Zeichen ein Buchstabe, wird die Zeichenfolge als Variablenname interpretiert. Es wird dann eine Variable mit dem angegebenen Namen gesucht. Dieser Name muß eine einfache Variable bezeichnen, die einen gültigen Wert enthält.

Eine Indexangabe ist zusätzlich zum Variablennamen möglich, wenn die Dialogvariable mittels des VDEFINE-Dienstes als ein Array von Dialogvariablen definiert worden ist.

Systemvariable des Dialogmanagers (spezielle Dialogvariable) beginnen mit "SYS-". Ein Anwender sollte keine eigenen Dialogvariablen mit "SYS-" beginnen lassen.

Beispiele für gültige Namen von Dialogvariablen:

```
TELEFON-NR  
A  
A.123#5  
ORT#INDEX
```

Wert einer Dialogvariablen

Der Wert einer Dialogvariablen hat eine gewisse Länge. Diese Länge kann ein feststehender Wert, kann durch eine Längenangabe oder durch ein Endekennzeichen festgelegt sein. Die möglichen Werte für die Länge einer Dialogvariablen sind durch den Datentyp bestimmt.

Datentyp einer Dialogvariablen

Der Datentyp legt fest, ob der Wert der Dialogvariablen z.B. eine Zeichenkette, Binärzahl oder Dezimalzahl ist.

Die einzelnen Datentypen sind beim VDEFINE-Dienst beschrieben.

4.1.1.1 Explizite Dialogvariablen

Eine explizite Dialogvariable wird mittels des VDEFINE-Dienstes definiert. Mit dem VDEFINE-Dienst wird interner Speicherplatz des Anwendungsprogrammes einer Dialogvariablen zugeordnet. Jede Änderung des Inhaltes dieses Platzes ändert sofort den Wert der Dialogvariablen. Die Dienste des Dialogmanagers greifen zum Lesen und Schreiben einer expliziten Dialogvariablen direkt auf den Speicherplatz des Programmes zu.

Dies erlaubt einem Anwendungsprogramm, eine Programmvariable als Dialogvariable zu erklären, den DISPLAY-Dienst des Dialogmanagers zum Anzeigen einer Maske, in der diese Variable enthalten ist, aufzurufen und anschließend im Programm diese Variable auf normale Weise, z.B. IF NAME = "OTTO" THEN..., zu verarbeiten.

Der Variablenpool enthält den Namen und die Adresse des Wertes einer expliziten Dialogvariablen. Dieser Eintrag im Variablenpool kann durch den VDELETE-Dienst gelöscht werden.

Beschreibung einer expliziten Dialogvariablen:

Eine oder mehrere explizite Dialogvariable werden bei einem Aufruf des VDEFINE-Dienstes durch einen der folgenden Punkte beschrieben:

- einen Namen oder eine Namensliste,
- einen Datentyp oder eine Liste von Datentypen,
- eine Länge oder eine Liste von Längewerten,
- eine Adresse (die Anfangsadresse der Werte) und
- einen Vielfachfaktor

Namen, Datentypen und Längewerte an der gleichen Position in den Listen korrespondieren miteinander.

Die Adresse einer Variablen der Namensliste ergibt sich aus der Anfangsadresse und den Längen aller vorher in diesem VDEFINE-Dienst definierten Variablen.

Durch einen einzigen Aufruf des VDEFINE-Dienstes können somit einzelne Teile eines zusammenhängenden Speicherbereiches als explizite Dialogvariable definiert werden. Andere Teile dieses Speicherbereiches können als Füllbereiche übergangen werden, indem in den Listen nur die Länge des Füllbereiches und für Namen und Datentyp ein "*" eingetragen wird.

Solche festgelegten Folgen von Programmvariablen sind im Sinne von Programmiersprachen Strukturen.

Das folgende Bild veranschaulicht den Sachverhalt:

var_1	var_2	füllbereich	var_3	...
länge_1	länge_2	länge_fb	länge_3	
Anfangsadresse				

Das durch die Listen geprägte Layout des Speicherplatzes wiederholt sich mehrfach, wenn ein Wert für die Vielfachfaktor (größer als 1) angegeben wurde. Jede in der Namensliste definierte Dialogvariable ist dann eine indizierte Dialogvariable.

Mittels des VDEFINE-Dienstes lassen sich einfache explizite Dialogvariable, Strukturen von expliziten Dialogvariablen und Arrays von expliziten Dialogvariablen definieren.

Eine Dialogvariable eines Arrays von Dialogvariablen wird durch `name_der_variable#index` (z.B. `VLEN(10)`, siehe auch nächste Seite) angesprochen. Mittels der Indexangabe wird ein bestimmtes Element eines Arrays von Dialogvariablen bezeichnet. Fehlt die Indexangabe, wird das erste Element angenommen.

Beispiel für die Definition von mehreren expliziten Dialogvariablen:

Mittels folgender C-Anweisungen werden die Dialogvariablen VI und VC definiert.

```
DMCOMM dmcomm;
char buffer[255];
long buflen;
long lfeld;

long vi;
char vc [50];

strcpy (buffer "VDEFINE (VI) FORMAT(FIXEDS)");
lfeld = 4;
buflen = strlen(buffer);
ispci2(&dmcomm,&buflen,buffer,&lfeld,&vi);

strcpy (buffer "VDEFINE (VC) FORMAT(BINSTR)");
lfeld = 50;
buflen = strlen(buffer);
ispci2(&dmcomm,&buflen,buffer,&lfeld,&vc);
```

Die Variablen "vi" und "vc" sind nicht innerhalb einer C-Struktur definiert, wodurch die Lage des ihnen zugeordneten Speicherplatzes für den Programmierer unbekannt ist. Deshalb werden zwei Aufrufe des VDEFINE-Dienstes benötigt, um sie als Dialogvariable zu deklarieren.

Beispiel für die Definition eines Arrays von Strukturen:

Mittels folgender COBOL-Anweisungen werden die Programmvariablen A, B und C der Struktur S als explizite Dialogvariable X, Y, Z erklärt.

```

01 LBUF          PIC S9(7)  COMP VALUE 512.
01 BUF          PIC X(512) VALUE SPACE.
01 LENFELD.
   02 VLEN      PIC S9(7)  COMP OCCURS 3 TIMES.
01 S.
   02 SE       OCCURS 3 TIMES.
       03 A     PIC X(10).
       03 B     PIC S9(7)  COMP.
       03 C     PIC X(2).

MOVE 10 TO VLEN(1).
MOVE  4 TO VLEN(2).
MOVE  2 TO VLEN(3).
MOVE "VDEFINE (X Y Z) FORMAT(CHAR FIXEDS CHAR) DIM(3)
      OPTION(LIST)" TO BUF.
CALL "ISPC12" USING DMCOMM LBUF BUF LENFELD S.

```

Es wird also folgendes Speicherlayout (mit den angedeuteten Offsets) beschrieben:

	0	10	14	16
0	A(1)	B(1)	C(1)	
16	A(2)	B(2)	C(2)	
32	A(3)	B(3)	C(3)	

Die Dialogvariablen X, Y und Z bilden ein Array von Strukturen und müssen zusammenhängenden Speicherplatz belegen.

Datentypen von expliziten Dialogvariablen

Mittels des VDEFINE-Dienstes erhält jede explizite Dialogvariable einen Datentyp zugeordnet. Dieser Datentyp gibt darüber Auskunft, wie der Speicherplatz der Dialogvariablen vom Anwendungsprogramm interpretiert wird, beschreibt somit die interne Form der Dialogvariablen.

Explizite Dialogvariable können mit folgenden Datentypen definiert werden:

- CHAR, BINSTR, NUMS, NUMU zur Speicherung von Zeichenketten,
- FIXEDS, FIXEDU zur Speicherung von Binärzahlen,
- PACK zur Speicherung von gepackten Dezimalzahlen.

Beim Zugriff auf eine Dialogvariable kann durch den Dialogmanager eine Datenkonvertierung erfolgen, um unterschiedliche Datentypen ineinander umzuwandeln. Die Datentypen sind beim VDEFINE-Dienst ab Seite 156 beschrieben.

Dies kann in folgenden Fällen notwendig sein:

- Wenn ein VCOPY-, VREPLACE-, VPUT- oder VGET-Dienst aufgerufen wird.
- Wenn eine Dialogvariable in einer Maske oder einer Meldung angezeigt wird.

4.1.1.2 Implizite Dialogvariable

Wenn der Dialogmanager eine Dialogvariable schreiben soll, die nicht explizit durch das Anwendungsprogramm erklärt ist, definiert er implizit eine Dialogvariable. D.h. es wird im Funktionspool (oder im Profilpool) der Name und der Wert eingetragen. Bei einer Leseoperation für eine nicht definierte Dialogvariable wird die Länge 0 geliefert und der Bereich wird gelöscht.

Eine implizite Dialogvariable ist eine Zeichenkette vom Typ CHAR oder eine Zahl vom Typ FIXEDS (4 Bytes) und ist kompatibel zu den SDF-P-Variablen vom Typ STRING bzw. INTEGER. Die Länge einer impliziten Dialogvariablen vom Typ CHAR ist auf 16383 Zeichen beschränkt.

Implizite Dialogvariable können nur mittels der Variablen- und Anzeigedienste verarbeitet werden.

Ein Programm kann mit dem VCOPY-Dienst den Wert einer impliziten Dialogvariablen in einen Programmbereich kopieren, mit dem VREPLACE-Dienst den Wert verändern oder die Dialogvariable implizit anlegen.

Ist ein Eingabefeld einer Maske mit einer impliziten Dialogvariablen verbunden, kann ihr Wert durch die Eingabe über die Tastatur geändert werden.

Eine implizite Dialogvariable ist immer eine einfache Dialogvariable. Eine implizite Dialogvariable vom Typ CHAR wird immer in der Länge der bedeutsamen Zeichen (ohne nachfolgende Leerzeichen) gespeichert.

Namensgleichheit von Dialogvariablen

Die Namen von Dialogvariablen in einem Variablenpool müssen eindeutig sein. Wenn diese Eindeutigkeit verletzt wird, können folgende Fälle auftreten:

- Eine explizite Dialogvariable und eine implizite Dialogvariable sollen den gleichen Namen erhalten. Dies tritt dann auf, wenn durch den VDEFINE-Dienst eine Dialogvariable mit einem Namen definiert wird, der bereits für eine implizite Dialogvariable existiert. Wenn VDEFINE mit dem Operanden COPY aufgerufen wird, wird der Wert der impliziten Dialogvariablen kopiert und die implizite Dialogvariable gelöscht. Ohne den COPY-Operanden wird die explizite Dialogvariable angelegt und die implizite Dialogvariable wird gelöscht.
- Wenn zwei explizite Dialogvariable den gleichen Namen erhalten sollen, wird der VDE- FINE-Aufruf mit einem Fehlercode beendet.
- Zwei implizite Dialogvariable mit gleichen Namen gibt es nicht. Es wird nur der Wert geändert.

4.1.1.3 Regeln für Dialogvariable

Es gelten folgende Regeln in Verbindung mit Dialogvariablen:

- Eine implizite FIXEDS-Dialogvariable hat immer eine Länge von 4 Bytes.
- Eine FIXEDS-Dialogvariable wird auch dann implizit angelegt, wenn sie den Wert 0 hat.
- Ein CHAR/BINSTR-Wert wird nur dann als eine impliziten Dialogvariablen angelegt, wenn der Quellwert nicht die relevante Länge 0 hat.

Relevante Länge: Länge des Wertes einer Dialogvariablen vom Typ CHAR/BINSTR ohne nachfolgende nicht bedeutsame Leerzeichen. Ob nachfolgende Leerzeichen bedeutsam sind, wird bei der expliziten Definition einer Dialogvariablen (VDEFINE) durch den Operanden NOBSCAN festgelegt. Für implizite Dialogvariable gibt es keine nachfolgenden bedeutsamen Leerzeichen (nachfolgende Leerzeichen werden beim Speichern entfernt).

- Die Länge einer bestehenden impliziten Dialogvariablen vom Typ CHAR wird beim Aktualisieren (VREPLACE) entsprechend der relevanten Länge verändert (verkürzt/verlängert).
- Eine implizite FIXEDS-Dialogvariable kann nur durch VREPLACE erzeugt werden (nicht durch Maskenfelder).
- VCOPY für CHAR/BINSTR als Zieltyp liefert die relevante Länge und verändert damit die durch den Anwender vorgegebene Zielfeldlänge. Für andere Typen bleibt die vorgegebene Zielfeldlänge unverändert.
- Profil-Dialogvariable können vom Typ CHAR oder FIXEDS sein.

- Implizite Dialogvariablen können nur explizit gelöscht werden.
- Keine häufigen Typumwandlungen vornehmen, diese erschweren die Übersicht. Wenn numerische Werte vorliegen und der Wertebereich ausreicht, sollte FIXEDS bevorzugt werden.

4.1.1.4 Systemvariable des Dialogmanagers

Systemvariable enthalten allgemeingültige Daten des Dialogmanagers oder des Betriebssystems (z.B. das aktuelle Datum) und sind im Funktionspool oder in einem speziellen Systempool gespeichert.

Die Namen von Systemvariablen des Dialogmanagers beginnen mit der Zeichenfolge "SYS-".

Einige Systemvariable (z.B. Uhrzeit) werden aktuell ermittelt. Systemvariable, die im Funktionspool enthalten sind, werden wie normale Dialogvariable behandelt, d.h. wenn es noch keine Variable mit diesem Namen gibt, wird sie definiert, gibt es bereits eine Variable mit diesem Namen, wird diese unabhängig davon verwendet, ob es eine implizite oder explizite Dialogvariable ist. Eine Übersicht über die Systemvariablen finden Sie im Anhang ab Seite 259.

4.1.1.5 Konventionen für Dialogvariablen in der Modellzeile (Listenverarbeitung)

Name ohne Indexangabe z.B. ABC oder A.B-C

Variablendefinition im TIAM-Anwendungsprogramm:

```
VDEFINE (ABC,A.B-C) FORMAT(CHAR) DIM(100)
```

Der Einfachheit wegen wird von Dialogvariablen gesprochen.

4.2 Variablenpool

Dialogvariable werden in Variablenpools als (Name,Wert)- oder (Name, Adresse des Wertes)-Kombinationen gespeichert. Spezielle Variablenpools werden in einer Profilibibliothek abgelegt, und enthalten die Dialogvariablen des Anwendungsprofils

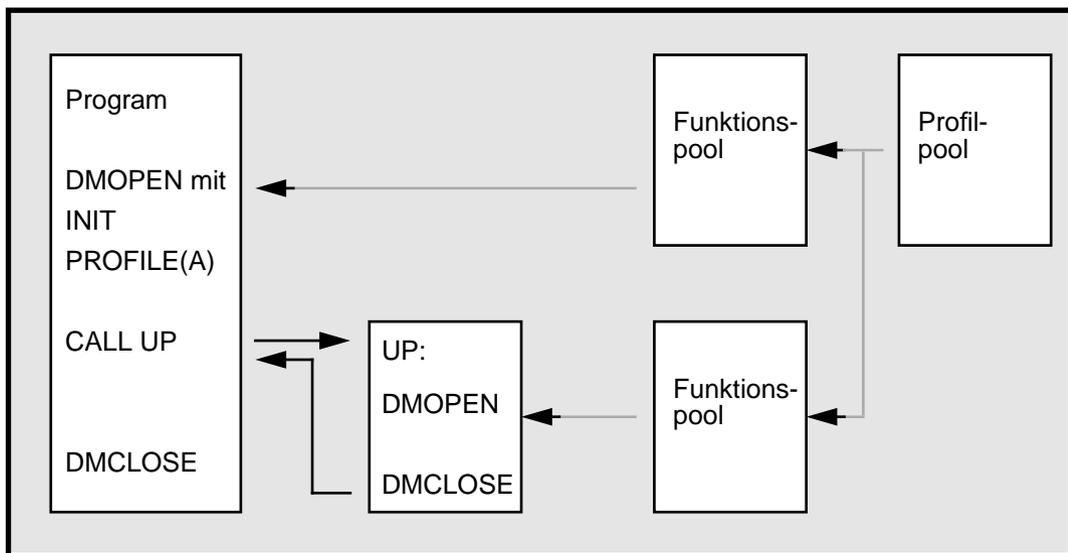
Dialogvariable für ein Anwendungsprogramm werden durch die Variablen-Dienste des Dialogmanagers definiert oder erzeugt. Sie werden in Variablenpools aufbewahrt.

Funktionspool und Profilpool

Der Dialogmanager unterscheidet zwei Arten von Variablenpools:

- Funktionspool
- Profilpool

Das folgende Bild stellt die Zusammenhänge dar. Es zeigt den Ablauf eines Anwendungsprogrammes, in das ein Unterprogramm (UP) eingeschachtelt ist, und zeigt, wann welcher Variablenpool zur Verfügung gestellt wird (gestrichelte Linie).



Funktionspool und Profilpool bilden eine zweistufige Poolhierarchie. Beim Lesen einer Dialogvariablen wird zunächst der Funktionspool durchsucht. Wird die Dialogvariable dort nicht gefunden, wird der Profilpool durchsucht.

Die Existenz des Funktionspools beginnt mit der Ausführung eines DMOPEN-Dienstes und endet mit der Ausführung des zugehörigen DMCLOSE-Dienstes.

Bei der Ausführung eines DMOPEN-Dienstes mit dem Operanden INIT kann zusätzlich zum Funktionspool mit Hilfe des PROFILE-Operanden ein Profilpool erzeugt werden.

In der benutzerspezifischen Profilbibliothek sind in einem Element Namen und Werte von Profil-Dialogvariablen gespeichert. Falls dieses Element existiert, wird beim Eröffnen des Profilpools dieser mit den in diesem Element enthaltenen einfachen Dialogvariablen initialisiert. Andernfalls beginnt die Anwendung mit einem leeren Profilpool.

Die Werte und die Namen der Dialogvariablen des Profilpools werden beim Abschließen des Pools, d.h. beim Beenden des Zugriffes zum Dialogmanager (DMCLOSE-Dienst), in das Bibliothekselement ausgegeben. Ein Profilpool enthält immer implizite Dialogvariable (d.h. der Profilpool enthält Name und Wert der Dialogvariablen), deren Werte den Typ CHAR oder FIXEDS haben.

Wird der Operand NOSAVE beim DMCLOSE angegeben, wird der Profilpool nicht geschrieben.

Definitionen von Dialogvariablen im Funktionspool können mit dem VDELETE-Dienst gelöscht werden. Eine explizite Dialogvariable muß dann gelöscht werden, wenn der adressierte Speicherplatz nicht mehr zur Verfügung steht.

Dialogvariablen können mittels des VREPLACE-Dienstes Werte zugewiesen werden. Existiert dabei eine Variable im Funktionspool noch nicht, wird sie als implizite Dialogvariable definiert. Auch der DISPLAY-Dienst kann implizite Dialogvariable im Funktionspool definieren.

Der Wert von Dialogvariablen kann mittels des VCOPY-Dienstes gelesen werden (z.B. von impliziten Variablen).

Die in einem Element abgespeicherten Profilvariablen des Dialogmanagers sind kompatibel mit einem von SDF-P (ab Version 2.0) erzeugten Profilelement. Es ist deshalb möglich, SDF-P-Kommandos (oder eine S-Prozedur) zur Bearbeitung eines FHS-Profilelements zu verwenden. Im folgenden Beispiel wird ein Profil-Element mit dem Namen PROF1 in der Bibliothek PROFLIB bearbeitet:

```
/OPEN-VARIABLE-CONTAINER PROF1,*LIB(PROFLIB,PROF1),  
  AUTOMATIC-DECLARE=*
```

Mit diesem SDF-P-Kommando wird ein SDF-P-Variablenbehälter mit dem Namen PROF1 eröffnet und alle Variablen "importiert".

```
/SHOW-VARIABLE *ALL  
  Die Variablen werden angezeigt.
```

```
/variablenname = "neuer_wert"  
  Eine Variable wird geändert.
```

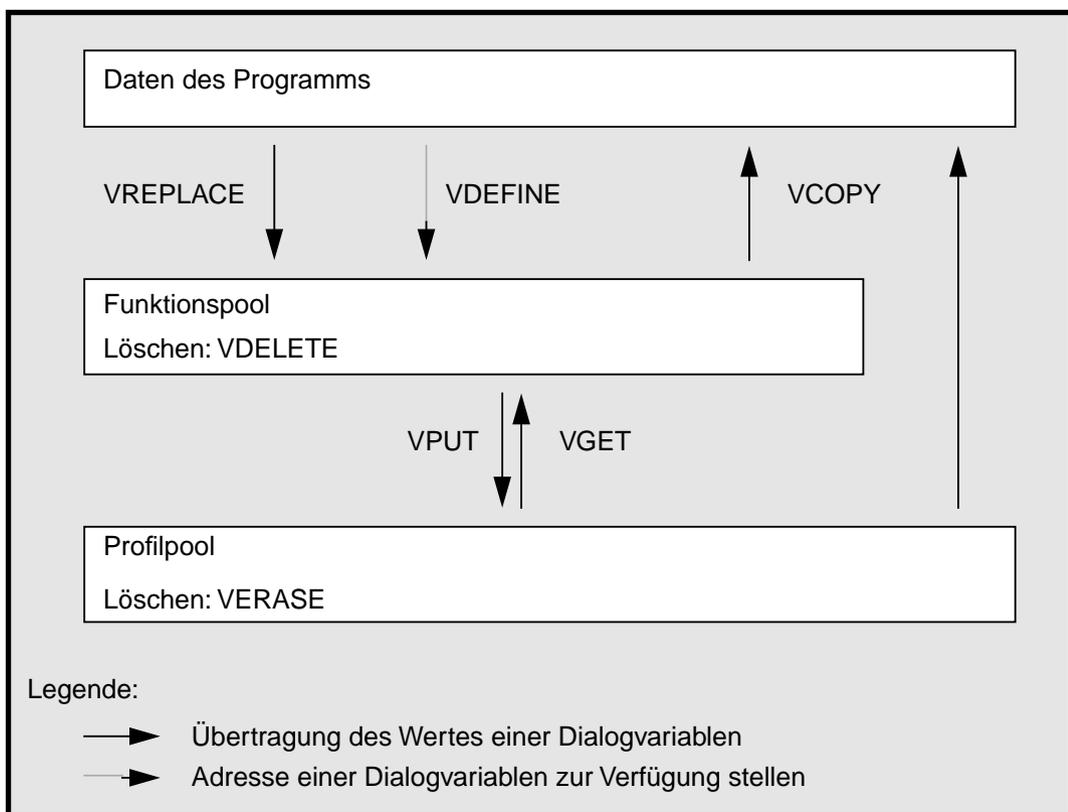
/SAVE-VARIABLE-CONTAINER PROF1

Das geänderte Profil-Element wird geschrieben.

Hinweis:

Die Bearbeitung eines FHS-Profiles mit SDF-P setzt voraus, daß die Syntax der Variablennamen der des SDF-P entspricht.

Das folgende Bild demonstriert, welche Dienste Daten zwischen den Variablenpools oder Programmbereichen vermitteln oder in welchem Variablenpool ein Löschen von Dialogvariablen erfolgt.



Im Profilpool werden Dialogvariable mittels des VPUT-Dienstes erzeugt oder deren Wert geändert. Der aktuelle Wert der Dialogvariablen wird dabei aus dem Funktionspool in den Profilpool kopiert, ohne das die Dialogvariable im Funktionspool verändert wird. Mittels des VGET-Dienstes können Dialogvariable aus dem Profilpool in den Funktionspool kopiert werden. Dabei werden die Definition und der Wert im Profilpool nicht verändert. Mittels des VERASE-Dienstes können Dialogvariable im Profilpool gelöscht werden.

4.3 Programmaufbau einer Dialoganwendung

Jede Dialogmanager-Anwendung beginnt die Kommunikation mit dem Dialogmanager durch den Aufruf des DMOPEN-Dienstes und beendet sie durch den DMCLOSE-Aufruf. Dazwischen können weitere Dienste aufgerufen werden.

Der DMOPEN-Aufruf initialisiert die Verbindung zum Dialogmanager. Das Anwendungsprogramm darf keinen anderen Dialog-Dienst aufrufen, wenn der DMOPEN-Aufruf nicht erfolgreich war.

Der INIT-Operand muß beim ersten DMOPEN-Aufruf einer Dialogmanager-Anwendung angegeben werden. Der als Übergabeparameter vom Anwendungsprogramm bereitzustellende Verständigungsbereich (DMCOMM) enthält hierbei einen nicht initialisierten Systemteil.

Dieser erste DMOPEN-Aufruf fordert globalen Speicherbereich für die weitere Arbeit des Dialogmanagers an und stellt die Adresse dieses Bereiches in den Systemteil des Verständigungsbereiches. Außerdem wird in den Verständigungsbereich ein Kennzeichen eingetragen, durch das dieser DMCOMM von anderen unterschieden werden kann. Diese Eintragungen dürfen vom Anwendungsprogramm nicht verändert werden.

Durch das Anwendungskennzeichen des PROFILE-Operanden im ersten DMOPEN-Aufruf kann wahlweise ein Profilpool eröffnet werden. Jeder DMOPEN-Aufruf eröffnet einen neuen Funktionspool, der mit dem Profilpool verkettet wird. Verkettet bedeutet, daß beim Lesen von Dialogvariablen zunächst der Funktionspool und, falls dort die Variable nicht gefunden wird, anschließend der Profilpool durchsucht wird.

Alle nach dem ersten DMOPEN-Aufruf folgenden Aufrufe für Dialogdienste müssen den vom Dialogmanager initialisierten Verständigungsbereich (siehe Seite 118) benutzen. Dadurch wird erreicht, das diese Dienste mit einem Funktionspool und dem damit verketteten Profilpool verbunden sind.

Der DMCLOSE-Dienst wird ebenfalls mit dem beim DMOPEN initialisierten Verständigungsbereich aufgerufen. DMCLOSE schließt den entsprechenden Funktionspool und schreibt die Dialogvariablen des Profilpools in die Profilbibliothek. Der globale Speicherbereich des Dialogmanagers wird freigegeben.

Mehrfacher DMOPEN-Aufruf

Normalerweise enthält eine Anwendung nur einen DMOPEN/DMCLOSE-Aufruf. Eine komplexere Anwendung kann Unterroutinen rufen, die ihrerseits ebenfalls DMOPEN/DMCLOSE-Aufrufe enthalten. Es kann auch ein BS2000-Systemunterprogramm gerufen werden, das Dialogdienste aufruft. Auf diese Weise entsteht eine Hierarchie von DMOPEN-Aufrufen.

Im folgenden wird beschrieben, wie solche mehrfachen DMOPEN anzuwenden sind.

- Der erste DMOPEN-Aufruf einer Anwendung muß den INIT-Operanden enthalten und damit einen Dialogkomplex eröffnen. Wenn der Dialogkomplex Profilvariable verwenden will, muß der PROFILE-Operand angegeben werden.
Der Dialogmanager eröffnet daraufhin einen Funktionspool und einen Profilpool und stellt in den Systemteil des übergebenen DMCOMM globale Informationen. Der Profilpool steht für alle Dialogmanagerdienste dieses Dialogkomplexes zur Verfügung. Er wird erst durch ein den Dialogkomplex beendendes DMCLOSE geschlossen.
- In einem Dialogkomplex kann es z.B. Unterroutinen geben, die einen eigenen Funktionspool benutzen möchten. Dies ist durch einen DMOPEN-Aufruf ohne INIT-Operanden möglich. Ein DMOPEN ohne INIT eröffnet einen Dialogabschnitt. Bei einem DMOPEN-Aufruf ohne den INIT-Operanden muß der dazugehörige DMCOMM einen gültigen Systemteil enthalten. Dazu muß das Anwendungsprogramm den Systemteil des DMCOMM des ersten DMOPEN kopieren, wenn für jedes DMOPEN ein eigener DMCOMM verwendet werden soll.
Für streng geschachtelte mehrfache DMOPEN-Aufrufe, ist es nicht notwendig bei jedem DMOPEN einen eigenen DMCOMM bereitzustellen. Es kann der beim ersten DMOPEN verwendete DMCOMM benutzt werden. Der Dialogmanager rettet beim DMOPEN den bisherigen Systemteil und speichert in beim nächsten DMCLOSE wieder zurück. In diesem Fall darf aber nur mit dem zuletzt eröffneten Dialogabschnitt gearbeitet werden. Wenn verschiedene Funktionspools gleichzeitig verwendet werden sollen, müssen auch mehrere Verständigungsbereiche parallel existieren, d.h. es müssen Kopien den DMOPEN übergeben werden.
Ein DMCLOSE mit dem Verständigungsbereich des Dialogkomplexes (mit INIT) beendet auch alle untergeordnete Dialogabschnitte.
Bei jedem DMOPEN-Ruf im gleichen Dialogkomplex wird ein eigener Funktionspool bereitgestellt und mit dem Profilpool des Dialogkomplexes verkettet.
- DMCLOSE gibt den beim DMOPEN angeforderten Funktionspool wieder frei. Das zum ersten DMOPEN gehörende DMCLOSE gibt auch den Profilpool frei und schreibt die Profilvariablen in die Profilbibliothek. Der globale Speicher wird ebenfalls freigegeben und damit der Dialogkomplex beendet. Danach kann kein Dialogdienst mehr für diesen Komplex angefordert werden.

- Durch mehrere DMOPEN-Aufrufe mit dem INIT-Operanden entstehen mehrere Dialogkomplexe. Existieren mehrere Dialogkomplexe nebeneinander, ist ein Austausch von Dialogvariablen über Variablenpools untereinander nicht möglich. D.h. ein Dialogkomplex ist für den Dialogmanager von einem anderen Dialogkomplex vollständig getrennt.
- Wenn mehrere Dialogkomplexe mit dem gleichen Anwendungskennzeichen im PROFILE-Operanden erzeugt wurden, existieren auch mehrere Profilpools, die mit dem gleichen Element der Profilbibliothek korrespondieren. In diesem Fall wird beim Schliessen des Pools eventuell der Inhalt des durch einen anderen Dialogkomplex erzeugten Profilelementes überschrieben.

Beispiel einer Dialoganwendung:

```

Dialogkomplex 1
ispci (dmcomm1,buflen,"DMOPEN INIT PROFILE(ABC)");
call subr1
    Uebergabe dmcomm1
call dialogkomplex2
ispci (dmcomm1,buflen,"DMCLOSE");

subr 1
ispci (dmcomm1,buflen,"DMOPEN")
ispci (dmcomm1,buflen,"DISPLAY...")
ispci (dmcomm1,buflen,"DMCLOSE")

Dialogkomplex2 (ohne Profilpool)
ispci (dmcomm2,buflen,"DMOPEN INIT")
ispci (dmcomm2,buflen,"DISPLAY...")
ispci (dmcomm2,buflen,"DMCLOSE")
    
```

Beispiel einer Dialoganwendung mit parallelen Dialogabschnitten:

Dialogkomplex 1

ispci (dmcomm1,buflen,"DMOPEN INIT PROFILE(ABC)");	Eröffnen des Dialogkomplexes
ispci (dmcomm1,buflen,"DISPLAY...")	DISPLAY-Aufruf
dmcomm2 = dmcomm1	Kopien des DMCOMM
dmcomm3 = dmcomm1	
ispci (dmcomm2,buflen,"DMOPEN")	Eröffnen der beiden Dialogabschnitte
ispci (dmcomm3,buflen,"DMOPEN")	Anlegen Funktionspool 2 und 3
call subr1	Wechselweises Arbeiten in den verschiedenen Dialogabschnitten und Pools
Uebergabe dmcomm 2	
call subr2	
Uebergabe dmcomm 3	
ispci (dmcomm1,buflen,"DISPLAY...")	
call subr1	
Uebergabe dmcomm 2	
ispci (dmcomm2,buflen,"DISPLAY...")	
call subr2	
Uebergabe dmcomm 3	
call subr2	
ispci (dmcomm1,buflen,"DMCLOSE");	Abschließen des Dialogkomplexes einschließlich der Abschnitte

subr1 (Parameter par_dmcomm)

ispci (par_dmcomm,buflen,"VDEFINE...")	Arbeit mit Funktionspool 2
ispci (par_dmcomm,buflen,"DISPLAY...")	

subr2 (Parameter par_dmcomm)

ispci (par_dmcomm,buflen,"VCOPY....")	Arbeit mit Funktionspool 3
ispci (par_dmcomm,buflen,"DISPLAY...")	

4.4 Dialogdienste aufrufen

Die Dienste des Dialogmanagers stehen für Anwendungsprogramme zur Verfügung.

Die allgemeine Form der Aufrufe in einem Programm, sowie die Arten und die Schreibweise von Parametern und Operanden, werden beschrieben, ohne dabei jeden einzelnen Parameter oder Operanden darzustellen.

Ab Seite 125 werden die Dialogdienste in alphabetischer Reihenfolge aufgeführt und ihre Parameter und Operanden beschrieben.

4.4.1 Interface-Routine ISPCI

In einem Programm werden die Dienste des Dialogmanagers über die Interface-Routine ISPCI (3 Parameter) oder ISPCI2 (5 Parameter) als CALL-Schnittstelle gerufen. Diese Schnittstelle orientiert sich an internationale Empfehlungen für ein Dialog Interface (SAA CPI - Systems Application Architecture Common Programming Interface). An dieser Schnittstelle werden die Parameter auf folgende Art übergeben:

Register 1 -->	A(Verständigungsbereich)	
	A(Länge des Operandenbereiches)	
	A(Operandenbereich)	
	A(Längenbereich)	1)
	A(Wertebereich)	1)

- 1) Diese Parameter werden nur von einigen Diensten benötigt. Für den Aufruf dieser Dienste muß der Eingangspunkt ISPCI2 der Interface-Routine verwendet werden. Die Bedeutung dieser zusätzlichen Parameter wird bei der Beschreibung des entsprechenden Dienstes dargestellt. Diese Art der Parameterübergabe wird von allen Compilern beim Aufruf eines Unterprogrammes realisiert, oder kann zumindest gefordert werden (CALL-Anweisung in COBOL und FORTRAN, Funktionsaufruf in C).

Der Speicherplatz für den Verständigungsbereich und den Operandenbereich muß vom Anwendungsprogramm bereitgestellt werden.

Der Operandenbereich ist ein Speicherbereich, der den Namen und die Operanden des Dialogdienstes enthält. Eine genaue Beschreibung des Operandenbereiches erfolgt bei der Beschreibung der einzelnen Dialogdienste.

Die Länge des Operandenbereiches in Bytes ist als binäre Ganzzahl mit Vorzeichen anzugeben (4 Bytes auf Wortgrenze ausgerichtet). Die maximale Länge beträgt 512 Bytes. Im Operandenbereich selbst sind nur gültige Zeichen erlaubt (z.B. X'00' führt zum Fehler).

Vor dem Aufruf der Interface-Routine muß der Operandenbereich mit dem Namen und den Operanden des gewünschten Dienstes gefüllt werden. Die Angaben erfolgen als Zeichenketten. Dadurch ist eine einfache Beschreibung in den üblichen Programmiersprachen möglich.

Die angegebene Länge des Operandenbereiches kann größer als die Anzahl der in den Operandenbereich eingetragenen Zeichen sein, wenn der Operandenbereich bis zur angegebenen Länge mit Leerzeichen gefüllt ist.

Das folgende Beispiel zeigt einen Ausschnitt aus einem in COBOL geschriebenen Anwendungsprogramm. In dem Programm wird der Zugriff auf den Dialogmanager eröffnet und anschließend ein Format angezeigt.

```
01 DMMCOMM      PIC X(128) VALUE SPACE.
01 BUFLN        PIC S9(7) COMP VALUE 512.
01 BUFFER       PIC X(512) VALUE SPACE.

MOVE "DMOPEN INIT PROFILE(ABC)" TO BUFFER.
CALL "ISPCI" USING DMMCOMM BUFLN BUFFER.

MOVE "DISPLAY PANEL( FORM001 )" TO BUFFER.
CALL "ISPCI" USING DMMCOMM BUFLN BUFFER.
```

Weitere Beispiele zum Aufruf von Dialogdiensten sind bei der Beschreibung der Dienste angeführt.

4.4.2 Verständigungsbereich

Der Verständigungsbereich enthält Daten zur Verständigung zwischen Programm und Dialogmanager. Er ist ein Speicherbereich des Anwendungsprogrammes.

Der Verständigungsbereich (DMCOMM) ist 128 Bytes lang und hat folgenden Aufbau:

Offset	Länge	Name	Bedeutung
0	8	DMRC	Returncode
0	2	DMSC2	Subcode2 (reserviert)
2	2	DMSC1	Subcode1 (Fehlerklasse)
4	4	DMMC	Maincode (Hauptwert)
8	8	DMMSGID	Meldungsschlüssel
16	8	DMFLAG	Anzeige für spezielle Dienste
16	1	DMERR	Anzeiger für Fehlerkontrolle
24	104	DMSYS	Systemteil

Die Bestandteile des Verständigungsbereiches haben folgende Bedeutung:

DMRC

Der Returncode informiert über die erfolgreiche oder fehlerhafte Ausführung des aufgerufenen Dienstes. Eine Rückkehrinformation wird vom Dialogmanager in das Feld DMRC, respektive in die Teilfelder DMMC und DMSC1 des Verständigungsbereiches abgelegt. Das Feld DMSC2 ist als reserviert zu betrachten. Nur wenn der SC1 den Wert Null hat, hat der SC2 für den Anwender eine Bedeutung. Im Falle einer Warnung ist dessen Wert 2.

Der Maincode ist der Hauptwert des Returncodes.

Der Subcode1 gibt die Fehlerklasse an, zu der der Hauptwert gehört.

Es gibt folgende Fehlerklassen (dezimaler Wert des Subcodes1):

- 0 erfolgreiche Ausführung (einschließlich Warnungen)
- 1 Parameterfehler (Syntaxfehler oder keine sinnvollen Werte)
- 32 interne Fehler des Dialogmanagers
- 64 sonstiger Fehler

Die möglichen Werte des Maincodes der Fehlerklasse 0 werden bei der Beschreibung der Dialogdienste aufgeführt. Alle Returncodes bei einer nicht erfolgreichen Ausführung eines Dienstes sind im Anhang enthalten.

DMMSGID

DMMSGID kann nach Ausführung eines Dienstes einen Meldungsschlüssel enthalten. Die damit verbundene Meldung erklärt die Fehlerursache für einen Fehlercode. Wenn der Returncode Null ist, enthält DMMSGID Leerzeichen.

Bei eingeschalteter Fehlerkontrolle durch den Dialogmanager wird bei einem Fehler, der zur Ausgabe der Fehlermaske führt, der Text der Meldung bereits in der Maske ausgegeben.

DMFLAG

Der Inhalt dieses Feldes wird folgendermaßen verwendet:

DMERR Anzeiger zum Ein/Ausschalten der Fehlerkontrolle

Y Fehlerkontrolle für Fehler

W Fehlerkontrolle für Warnungen und für Fehler

N (oder Leerzeichen) keine Fehlerkontrolle

empfohlene Einstellung: Y

DMSYS

Enthält den Systemteil des Verständigungsbereiches. Diese Informationen dürfen durch Anwendungsprogramme nicht zerstört werden.

Verwendung des Verständigungsbereiches

In einem Programm ist der Verständigungsbereich als Parameter beim Aufruf der Interface-Routine ISPCI anzugeben. In den Programmiersprachen kann der Verständigungsbereich auf folgende Weise definiert werden, wobei die hier aufgeführten Programmiersprachen nur als Beispiel dienen. In anderen Sprachen ist, wenn dort möglich, analog zu verfahren.

Programmiersprache COBOL:

```
DATA DIVISION
WORKING-STORAGE SECTION
01 DMCOMM    IS COMP.
   02 DMRC .
     03 DMSC2 PIC S9(4).
     03 DMSC1 PIC S9(4).
     03 DMCC  PIC S9(7).
02 DMMSGID PIC X(8).
02 DMFLAG .
   03 DMERR  PIC X(1).
   03 FILLER PIC X(7).
02 DMSYS    PIC X(104).
```

Programmiersprache C:

```
typedef struct
{
    short int      sc2;
    short int      sc1;
    long  int      mc;
    char           msgid[8];
    char           dmerr;
    char           filler[7];
    char           sys[104];
} DMCOMM_T;
DMCOMM_T dmcomm;
```

Programmiersprache FORTRAN:

```
C  Definition als INTEGER*2-Feld mit 64 Elementen
INTEGER*2      DMCOMM(64)
INTEGER*2      DMSC2,DMSC1
INTEGER*4      DMMC
EQUIVALENCE   (DMCOMM(1),DMSC2),(DMCOMM(2),DMSC1)
EQUIVALENCE   (DMCOMM(3),DMMC)
CHARACTER*8    DMMSGID
EQUIVALENCE   (DMCOMM(5),DMMSGID)
CHARACTER*8    DMFLAG
CHARACTER*1    DMERR
EQUIVALENCE   (DMCOMM(9),DMFLAG),(DMCOMM(9),DMERR)
CHARACTER*104  DMSYS
EQUIVALENCE   (DMCOMM(13),DMSYS)
```

4.4.3 Fehlerbehandlung durch den Dialogmanager

Die Dialogdienste liefern einen Rückkehrcode, der im Verständigungsbereich abgelegt wird. Dieser Code enthält Informationen über die erfolgreiche oder nicht erfolgreiche Ausführung des Dialogdienstes.

Im Anwendungsprogramm muß der Code nach jeder Rückkehr von einem Dialogdienst ausgewertet werden. Durch die Unterteilung des Rückkehrcodes in Klassifizierungs- und BeschreibungsCodes wird diese Auswertung vereinfacht.

Der Dialogmanager bietet dem Programmierer einer Anwendung die Möglichkeit beim Auftreten von Fehlern Informationen zum Fehler in einer Maske anzuzeigen (Fehlerreport).

Folgende Voraussetzungen müssen erfüllt sein, damit der Dialogmanager eine Fehlermaske anzeigt:

1. Der Anzeiger DMERR (im DMCOMM) muß auf Y oder W gesetzt sein (empfohlene Einstellung: Y).
2. Der Klassifizierungscode SC1 muss ungleich 0 (Fehler) oder SC1=0 und SC2=2 (Warnung) sein.
3. Die Fehlersituation darf die Arbeitsfähigkeit des Dialogmanagers nicht verhindern.

Als Maske im Hauptfenster (Vollbild) wird eine Fehlerbeschreibung ausgegeben. Der Inhalt der Maske enthält mindestens folgendes:

- vom Anwender gelieferter Operandenbereich nach Variablensubstitution durch Dialogmanager
- Rückkehrcode
- eine Beschreibung des Fehlers

Die Beschreibung ist in der Meldung enthalten, deren Meldungsschlüssel im Verständigungsbereich steht. Die Meldungsschlüssel finden Sie im Anhang ab Seite 232.

Nach der Anzeige eines Fehlers kann das Anwendungsprogramm die Ausführung der Anwendung abbrechen oder, falls sinnvoll, fortsetzen. Tritt ein Fehlercode > 200000 auf, kann keine Fehlermaske angezeigt werden.

Im Kommandoingabefeld der Fehlermaske kann das Systemkommando ABORT eingegeben werden. Dies bewirkt, das der Maincode im DMCOMM auf 399999 gesetzt wird. Das Anwendungsprogramm sollte auf diesen Code geeignet reagieren.

4.4.4 Struktur des Operandenbereiches

Der Operandenbereich muß den Namen des Dialogdienstes enthalten. Er kann Stellungsoperanden und Schlüsselwortoperanden enthalten.

Stellungsoperand

Der Stellungsoperand ist bei einigen Variablendiensten notwendig und enthält eine Liste von Namen. Die Namensliste wird in Klammern eingeschlossen. Als Trennzeichen in der Liste sind ein oder mehrere Leerzeichen, ein Komma oder Leerzeichen und Komma zulässig. Besteht die Namensliste nur aus einem Namen, können die Klammern weggelassen werden.

Beispiele:

Die folgenden Paare von Inhalten des Operandenbereiches sind jeweils identisch.

```
VDELETE (V111,V222,V333)
VDELETE ( V111 V222 V333 )

VDELETE ABC
VDELETE (ABC)
```

Schlüsselwortoperanden

Schlüsselwortoperanden bestehen aus der Zeichenfolge, die den Operanden kennzeichnet und evtl. einem Wert. Der Wert wird in Klammern hinter das Schlüsselwort gesetzt. (Die Klammer kann unmittelbar oder durch Leerzeichen getrennt dem Schlüsselwort folgen). Besteht der Wert in der Klammer nur aus Leerzeichen, wird der Operand als nicht angegeben betrachtet.

Manche Operanden erlauben die Angabe einer Liste von Werten. Die Werte werden in dieser Liste ohne eine zusätzliche Klammerung ebenfalls durch Leerzeichen, ein Komma oder durch ein in Leerzeichen eingefaßtes Komma getrennt. Schlüsselwortoperanden können in beliebiger Reihenfolge angegeben werden.

4.4.5 Variablensubstitution im Operandenbereich und in Meldungen

Beginnt eine Zeichenfolge im Operandenbereich oder im Text einer Meldung mit dem Zeichen "&", so wird die diesem Zeichen folgende Zeichenfolge als Name einer Dialogvariablen angesehen. Der Dialogmanager nimmt dann an, daß an dieser Stelle im Operandenbereich der Wert dieser Dialogvariablen stehen soll.

Für die Variablensubstitution gilt folgende Syntax:

`&(varname)` oder `&(varname#index)`

"varname" ist der Name einer Dialogvariablen. (In einer SDF-P-Anwendung: Name einer S-Variablen). Die Dialogvariable kann von jedem Typ sein, der sich in eine Zeichenkette konvertieren läßt. Der substituierte Wert ist immer eine Zeichenkette.

Die Klammern können weggelassen werden, wenn der Name der Dialogvariablen keinen Punkt enthält und keine Indexangabe erfolgt. In diesem Fall wird das Ende des Namens durch ein Zeichen, das nicht den Regeln der Namensbildung von Dialogvariablen entspricht oder durch einen Punkt bestimmt. Der Punkt hat in diesem Fall eine besondere Bedeutung: er dient nur als Endekennzeichen und geht verloren (kompatibel zur bisherigen Variablensubstitution in FHS V8.0).

Ein einfaches "&" gefolgt von einem Leerzeichen wird als Zeichen "&" interpretiert, es erfolgt keine Substitution.

Ein doppeltes "&" (&&) wird zu einem einfachen "&".

Eine rekursive Substitution erfolgt nicht.

Beispiele:

```
"DISPLAY PANEL( ABC )"
```

Ein Aufruf des Anzeigedienstes mit diesem Operandenbereich bewirkt die Ausgabe des Formates ABC.

```
"DISPLAY PANEL (&ABC)" "DISPLAY PANEL(&(A.B))"
```

Ein Aufruf des Anzeigedienstes mit diesem Operandenbereich bewirkt die Ausgabe des Formates, dessen Name in der Dialogvariablen ABC oder A.B enthalten ist.

```
"DISPLAY PANEL(&NFL#5)"
```

Ein Aufruf des Anzeigedienstes mit diesem Operandenbereich bewirkt die Ausgabe des Formates, dessen Name in der Dialogvariablen NFL im 5. Element des Arrays von Dialogvariablen enthalten ist.

```
"DISPLAY PANEL(&(DEF)#5)"
```

Weitere gültige Angaben sind: &ABC&DEF NO&ABC.D &ABC;TEXT &(a.b).LISTE

Wertesetzungen sind im Operandenbereich eines DMOPEN-Dienstes nicht möglich, da dem Anwendungsprogramm ja erst durch diesen Dialogdienst ein Variablenpool zugeordnet wird.

Der ganze Operandenbereich kann durch einen Variablennamen beschrieben werden.

Die eingefügten Zeichenfolgen werden nicht auf weitere Wertesetzungen geprüft. Der Operandenbereich darf nach der Variablenersetzung 2048 Zeichen nicht überschreiten.

4.5 Beschreibung der Dialogdienste

Für jeden Dialogdienst folgen einer Kurzbeschreibung des Dienstes die Parameter der Interface-Routine sowie die Struktur und die Operanden des Operandenbereiches. Desweiteren werden die Funktion des Dienstes ausführlich dargestellt und Beispiele zur Verwendung des Dienstes aufgeführt.

Wenn ein Dialogdienst Auswirkungen auf einen nachfolgenden Dialogdienst hat, so ist, wenn nicht ausdrücklich anders beschrieben, ein Dialogdienst desselben Dialogabschnittes gemeint (gleicher Verständigungsbereich).

Dialogdienste verschiedener Dialogabschnitte beeinflussen sich, bis auf wenige Einschränkungen (z.B. SETP siehe unten) nicht. Wenn zwei DISPLAY-Aufrufe unterschiedlicher Dialogabschnitte folgen, wird keine Differenzausgabe generiert, auch wenn dasselbe Format ausgegeben wird.

Verwendete Metasyntax:

Geschweifte Klammern schließen mehrere Werte ein, die alternativ auftreten können.

Hinweis zur Verwendung des Begriffes "feldname":

Durch IFG wird einem Maskenfeld der Namen einer Dialogvariablen zugeordnet, die in diesem Feld angezeigt werden soll. In einer Maske wird zur Benennung eines Feldes der entsprechende Name der Dialogvariablen als Feldname verwendet. Folgt einem Feldnamen ein "#" und ein Index, wird damit für ein Listenfeld eine bestimmte Listenzeile bezeichnet, da dadurch ein Element eines Arrays von Dialogvariablen angesprochen wird.

4.5.1 ADDPOP - Definieren der Position einer Dialogbox

ADDPOP teilt dem Dialogmanager mit, daß die Anzeige aller nachfolgenden Formate in einer Box erfolgen soll. Die durch den ADDPOP-Aufruf initialisierte Box bleibt solange wirksam bis ein REMPOP-Aufruf erfolgt oder bis zu einem anderen ADDPOP-Aufruf. Ein ADDPOP-Aufruf hat erst beim nächsten DISPLAY-Aufruf Auswirkungen auf das Bildschirmbild.

Die Größe der Dialogbox wird durch die Größe der Maske bestimmt. Ist diese größer als der an der gewünschten Position zur Verfügung stehende Platz, sucht der Dialogmanager einen anderen Platz, abhängig davon, ob der POPLOC-Operand angegeben wurde oder nicht, siehe auch Abschnitt "Erzeugen und Entfernen von expliziten Dialogboxen" Seite 16.

Das Beseitigen einer Dialogbox erfolgt mit dem REMPOP-Dienst. Die Box wird erst mit dem nächsten DISPLAY-Aufruf entfernt.

Einem ADDPOP-Aufruf darf erst nach einem DISPLAY-Aufruf mit Formatname ein erneuter ADDPOP-Aufruf folgen.

Parameter der Interface-Routine (ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
ADDPOP    [POPLOC( { fieldname } ) ]
           [ROW(zeile)]
           [COLUMN(spalte)]
```

Operanden:

ADDPOP

Name des Dienstes

POPLOC(feldname)

POPLOC definiert den Bezugspunkt für eine feldbezogene Positionierung der Dialogbox.

"feldname" muß der Name eines Feldes der aktuellen Maske sein. Die aktuelle Maske ist das zuletzt ausgegebene Format. Der Bezugspunkt ist das erste Zeichen des angegebenen Feldes.

Wenn sich der Feldname auf eine Listenzeile bezieht, kann ein Index zur Festlegung einer bestimmten Listenzeile angegeben werden. Ist in diesem Fall kein Index angegeben, wird die oberste Zeile im aktuellen Listenbereich angenommen.

Zur Festlegung der obersten Zeile im Listenbereich (Top-Index) wird der Wert der TOPINDEX-Variablen zum Zeitpunkt des ADDPOP-Aufrufes verwendet. Bei dem nachfolgenden DISPLAY-Aufruf für die Ausgabe einer Maske in einer Dialogbox, wird der unterliegende Listenbereich entsprechend der TOPINDEX-Angabe positioniert (siehe auch Listenverarbeitung, TOPINDEX-Variable).

Existieren mehrere Felder des angegebenen Namens, wird das erste Feld verwendet.

*CENTRAL gibt an, daß die Box zentriert in der Mitte des Bildschirmes auszugeben ist. Die Operanden ROW und COLUMN werden in diesem Fall ignoriert.

Fehlt der Operand POPLOC ist der Bezugspunkt die linke obere Ecke (der Startpunkt) der aktuellen Maske.

Wird der POPLOC-Operand ohne Feldname angegeben, so darf eine Box auf einen "leeren" Bildschirm ausgegeben werden, d.h. es muß kein Vollbild existieren.

ROW(zeile)

gibt die Verschiebung der Dialogbox nach unten (positiver Wert) oder oben (negativer Wert) bezüglich des Bezugspunktes an.

Wird der Operand ROW weggelassen, erfolgt eine Verschiebung um 2 Zeilen nach unten (Standardverschiebung).

COLUMN(spalte)

gibt die Verschiebung der Dialogbox nach rechts (positiver Wert) oder links (negativer Wert) bezüglich des Bezugspunktes an.

Wird der Operand COLUMN weggelassen, erfolgt eine Verschiebung um 2 Spalten nach rechts (Standardverschiebung).

Beispiel:

Mittels folgender COBOL-Anweisungen wird das Format FORM002 in eine das Format FORM001 überlagernde Dialogbox ausgegeben.

```
01 BUFLN      PIC S9(7) COMP VALUE 512.
01 BUFFER     PIC X(512) VALUE SPACE.

MOVE "DISPLAY PANEL( FORM001 )" TO BUFFER.
CALL "ISPCI" USING DMCMM BUFLN BUFFER.

MOVE "ADDPOP" TO BUFFER.
CALL "ISPCI" USING DMCMM BUFLN BUFFER.

MOVE "DISPLAY PANEL( FORM002 )" TO BUFFER.
CALL "ISPCI" USING DMCMM BUFLN BUFFER.
```

4.5.2 ATTR - Definieren dynamischer Attribute für Maskenfelder

Die Feldattribute von Maskenfeldern (z.B. Helligkeit, Farbe usw.) sind im allgemeinen statisch und werden bei der Definition eines Formates mittels IFG festgelegt. Diese Attribute sind in der Formatbeschreibung abgelegt. In besonderen Situationen kann es notwendig sein, durch das Anwendungsprogramm die Feldattribute für bestimmte Felder zu ändern.

Der ATTR-Dienst ermöglicht das dynamische Ändern von Feldattributen für Maskenfelder. Ein ATTR-Aufruf hat erst dann Auswirkungen auf das Bildschirmbild, wenn der nächste DISPLAY-Dienst ausgeführt wird und ist nur für diesen einen Aufruf wirksam.

Der ATTR-Dienst kann vor einem DISPLAY-Aufruf mehrmals aufgerufen werden. Die einzelnen ATTR-Rufe werden dabei verkettet, d.h. alle definierten Attributzuordnungen werden gesammelt und sind beim nächsten DISPLAY-Aufruf wirksam. Die dynamischen Feldattribute haben Vorrang vor den statisch definierten Attributen.

Der DISPLAY-Dienst löscht vor der Rückkehr zur Anwendung alle dynamischen Attributzuordnungen.

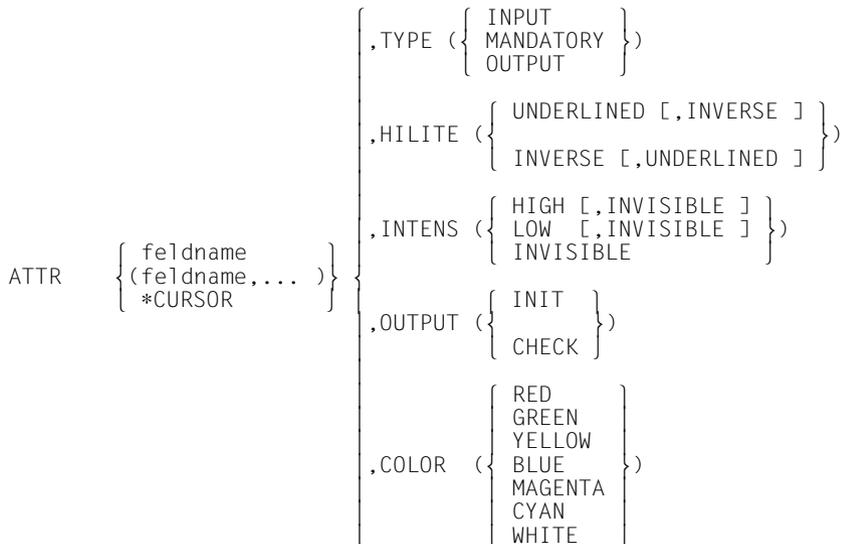
Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:



Operanden:

ATTR

Name des Dienstes

feldname

gibt den Namen eines Maskenfeldes an, für das dynamisch Feldattribute geändert werden sollen. Es kann eine Liste von Namen (in Klammern eingeschlossen) angegeben werden, für die alle die gleichen Attribute gelten.

Wenn sich der Feldname auf eine Listenzeile bezieht, kann ein Index zur Festlegung einer bestimmten Listenzeile angegeben werden. Ist in diesem Fall kein Index angegeben, wird die oberste Zeile im Listenbereich angenommen (siehe auch Listenverarbeitung, TOPINDEX-Variable).

Für Listenfelder kann als Index auch ein Stern (*) angegeben werden. In diesem Fall werden alle Indizes des angegebenen Feldes (Spalte einer Liste) mit den gleichen Attributen versehen. Stern darf nicht in Verbindung mit TYPE(MANDATORY) angegeben werden.

***CURSOR**

Die Feldattribute des Feldes, in dem sich der Cursor beim nachfolgenden DISPLAY-Aufruf befinden wird, sollen geändert werden. Wird beim nachfolgenden DISPLAY-Aufruf festgestellt, daß für das Feld auf dem der Cursor positioniert worden ist, auch für den Feldnamen ein dynamisches Attribut angefordert ist, werden die Angaben kombiniert. Bei gleichen Angaben hat die *CURSOR-Angabe den Vorrang.

TYPE()

Gibt den einzustellenden Feldtyp an:

INPUT

Das entsprechende Maskenfeld wird zum ungeschützten Eingabefeld ohne Eingabepflicht. Das Feld kann über die Tastatur verändert werden.

MANDATORY

Für das Maskenfeld besteht Eingabepflicht. Das Feld muß vom Terminal-Benutzer an der Datensichtstation modifiziert werden.

OUTPUT

Das Maskenfeld wird ein geschütztes, nicht markierbares Ausgabefeld. Der Feldinhalt kann über die Tastatur nicht verändert werden.

HILITE()

Beschreibt die Art der Hervorhebung des Maskenfeldes:

UNDERLINED

Der Feldinhalt wird unterstrichen dargestellt.

INVERSE

Der Feldinhalt wird invers dargestellt.

Hinweis:

Wenn bereits bei der Formatdefinition eine Hervorhebung von Feldern vereinbart wurde, kann diese nicht durch dynamische Attribute aufgehoben werden. Felder, die dynamisch hervorgehoben werden sollen, dürfen deshalb nicht schon bei der Formatdefinition hervorgehoben werden.

INTENS()

Gibt die Helligkeitsstufe des Feldes an:

HIGH

Das Feld wird mit hoher Intensität dargestellt.

LOW

Das Feld wird mit geringer Intensität dargestellt.

INVISIBLE

Der Feldinhalt ist nicht sichtbar und nicht druckbar.

OUTPUT()

Beschreibt die Verwendung der Voreinstellung von Maskenfeldern.

Ohne Angabe von OUTPUT werden Voreinstellungswerte der Maske immer dann verwendet, wenn beim DISPLAY-Aufruf die dem Maskenfeld zugeordnete Dialogvariable nicht existiert oder deren Wert die relevante Länge Null hat. Bei der Rückkehr vom DISPLAY-Dienst enthalten die entsprechenden Dialogvariablen den Voreinstellungswert des jeweiligen Maskenfeldes. Falls eine Dialogvariable nicht existiert wird sie mit dem Typ CHAR implizit angelegt.

Für Listenfelder werden Angaben zur Voreinstellung ignoriert.

INIT

Unabhängig vom Inhalt der mit dem Maskenfeld verbundenen Dialogvariablen, wird beim Anzeigen der Maske der beim Definieren des Formates festgelegte Voreinstellungswert für das spezifizierte Feld ausgegeben. Ist im Format kein Wert hierfür vorgegeben, wird das Feld mit Füllzeichen angezeigt.

CHECK

Für eine existierende Dialogvariable, deren Wert die relevante Länge größer Null hat, wird die Voreinstellung nur dann verwendet, wenn der Inhalt der mit dem Maskenfeld verbundenen Dialogvariablen den Wert Null hat.

Ohne Angabe des OUTPUT-Operanden, wird eine Voreinstellung eines Maskenfeldes immer verwendet, wenn die mit dem Maskenfeld verbundene Dialogvariable nicht existiert oder die relevante Länge Null hat.

COLOR()

Dient der Farbangabe für das Maskenfeld. Für Geräte, bei denen keine Farbdarstellung möglich ist, wird die Angabe ignoriert.

RED	rot
GREEN	grün
YELLOW	gelb
BLUE	blau
MAGENTA	purpur
CYAN	blaugrün
WHITE	weiß

Beispiel:

Mittels folgender COBOL-Anweisungen werden für einige Maskenfelder die Feldattribute zeitweilig geändert.

```
01 BUFLen      PIC S9(7) COMP VALUE 512.
01 BUFFER      PIC X(512) VALUE SPACE.

MOVE "ATTR FELD1,TYPE(MANDATORY),HILITE(UNDERLINED)" INTO BUFFER.
CALL "ISPCI" USING DMCOMM BUFLen BUFFER.

MOVE "ATTR (FELD2,FELD3),TYPE(OUTPUT)" INTO BUFFER.
CALL "ISPCI" USING DMCOMM BUFLen BUFFER.
```

4.5.3 CONTROL - Einstellen von Arbeitsmodi

Der CONTROL-Dienst erlaubt, bestimmte Optionen für die Verarbeitung zu vereinbaren.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
CONTROL DISPLAY { REFRESH
                  SAVE
                  RESTORE
                  CCSN( ccsname/*EXTEND )
                  NOCCSN
                  ACK }
```

Operanden:

CONTROL

Name des Dienstes

DISPLAY SAVE

Bewirkt, daß die Informationen über die augenblickliche Bildschirmanzeige gerettet werden. Dies erlaubt eine geschachtelte Anzeige von Masken im Hauptfenster.

Der SAVE-Aufruf muß nach dem DISPLAY-Aufruf erfolgen, dessen Bild gerettet werden soll. Die Wiederherstellung der Anzeige erfolgt durch CONTROL DISPLAY RESTORE.

SAVE und RESTORE müssen paarig und mit dem gleichen Verständigungsbereich aufgerufen werden, der auch beim DISPLAY-Aufruf verwendet wurde.

DISPLAY RESTORE

Das durch ein vorheriges CONTROL DISPLAY SAVE gerettete Bild wird wieder so hergestellt wie es zum Zeitpunkt des Rettens war. Anschließend kann mit der Verarbeitung fortgesetzt werden, wie es vor dem SAVE möglich war.

DISPLAY REFRESH

Bewirkt, daß das Bild beim nächsten DISPLAY-Aufruf vollständig aufgebaut wird (keine Differenzausgabe). P-Tasten und Zeichensätze werden bei Bedarf neu geladen. Diese Angabe ist nur für den nachfolgenden DISPLAY-Aufruf im Dialogkomplex unabhängig vom Dialogabschnitt (Verständigungsbereich) gültig.

DISPLAY ACK

Bewirkt, daß beim nächsten DISPLAY-Aufruf, vor der Ausgabe der Maske, im Zeilenmodus die Zeichen "ACK" ausgegeben werden und eine Eingabe angefordert wird. Erst nach dieser Eingabe wird das vollständig aufgebaute Bild ausgegeben (keine Differenzausgabe).

Damit kann der Anwender verhindern, daß Meldungen, die nicht durch den Dialogmanager ausgegeben worden sind, überschrieben werden, ohne daß der Bediener diese wahrnehmen konnte.

P-Tasten und Zeichensätze werden nicht neu geladen (vergleiche REFRESH). Diese Angabe ist nur für den nachfolgenden DISPLAY-Aufruf im Dialogkomplex unabhängig vom Dialogabschnitt (Verständigungsbereich) gültig.

DISPLAY CCSN(ccsname)

Bewirkt, daß für alle nachfolgenden DISPLAY-Aufrufe der angegebene ccsname als CCSNAME (Coded Character Set Name) verwendet wird. *EXTEND gibt an, daß der augenblicklich für die Datensichtstation gültige CCSN verwendet werden soll. Wurde bei der Formatdefinition ein CCSNAME für ein Format definiert und dieser Name ist vom angegebenen Namen verschieden, wird eine Warnung vom DISPLAY-Aufruf geliefert. Vorrang hat der CCSNAME des CCSN-Operanden.

Diese Angabe ist nur für den aktuellen Dialogabschnitt und nur solange gültig, bis ein neuer CONTROL-Aufruf mit CCSN oder NOCCSN und demselben Verständigungsbereich erfolgt.

Die Prüfung, ob der CCS-Name bekannt ist oder ob das Subsystem XHCS geladen ist, erfolgt erst beim nachfolgenden DISPLAY-Aufruf.

DISPLAY NOCCSN

Die Definition eines CCSN wird aufgehoben. Diese Angabe ist an den Verständigungsbereich gebunden und ist solange gültig, bis ein neuer CONTROL-Aufruf mit CCSN oder NOCCSN und demselben Verständigungsbereich erfolgt.

Beispiel:

Mittels folgender COBOL-Anweisungen werden bestimmte Optionen eingestellt.

```
01 BUFLN      PIC S9(7) COMP VALUE 25.  
01 BUFFER     PIC X(25) VALUE SPACE.  
  
MOVE "CONTROL DISPLAY SAVE" INTO BUFFER.  
CALL "ISPCI" USING DMMCOMM BUFLN BUFFER.  
  
MOVE "CONTROL DISPLAY RESTORE" INTO BUFFER.  
CALL "ISPCI" USING DMMCOMM BUFLN BUFFER.
```

4.5.4 DISPLAY - Anzeigen eines Formates und/oder einer Meldung

Der DISPLAY-Dienst realisiert die Anzeige eines Formates und/oder einer Meldung. Die Formatdefinition wird jedoch nur dann aus der Formatbibliothek gelesen, wenn der PANEL-Operand angegeben worden ist. Wenn ein Format geladen wird, bleibt es für den gesamten Dialogkomplex geladen.

Nach der Anzeige hat der Terminal-Benutzer der Anwendung verschiedene Möglichkeiten. Er kann u.a. Feldinhalte und damit die Werte von Dialogvariablen ändern, er kann die Ausgabe von Hilfeformaten veranlassen und er kann über die Menüleiste Aktionen anfordern. Erst wenn alle seine Aktionen und die in der Formatdefinition beschriebenen Prüfungen und Abläufe abgeschlossen sind, wird der DISPLAY-Dienst beendet.

Zur Verarbeitung der Voreinstellungswerte eines Maskenfeldes siehe ATTR-Dienst (OUTPUT-Operand).

Folgende Tabelle erläutert den Ablauf des DISPLAY-Dienstes in Abhängigkeit von den Angaben des PANEL- oder des MSG-Operanden.

PANEL	MSG	Ablauf
ja	nein	Lesen der Formatdefinition und der Dialogvariablen Anzeige des Formates
ja	ja	Lesen der Formatdefinition und der Dialogvariablen Lesen der Meldung Anzeige des Formates mit der Meldung im Meldebereich oder in einer Meldebox
nein	ja	Lesen der Meldung Wiederanzeige des zuletzt angezeigten Bildes ohne erneutes Lesen der Dialogvariablen (auch der Steuervariablen) mit Überlagerung durch eine Meldung im Meldebereich oder in einer Meldebox
nein	nein	Wiederanzeige des zuletzt angezeigten Bildes ohne erneutes Lesen der Dialogvariablen (auch der Steuervariablen)

Die Arbeit des DISPLAY-Dienstes wird beendet,

- bei Betätigung der ENTER-Taste, wobei der Cursor sich nicht in der Menüleiste befinden und im Kommandobereich kein Systemkommando stehen darf,
- durch eines der Kommandos CANCEL oder EXIT, mit denen auch Funktionstasten belegt sein können,
- durch ein Anwendungskommando, mit dem eine Funktionstaste belegt ist,

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```

DISPLAY   [ PANEL( formatname ) ]

           [ MSG ( msgid ) [ MSGLOC( { fieldname
                                     *CENTRAL } ) ] ]
           [ CURSOR( fieldname ) [ CSRPOS( pos ) ] ]
           [ LOCK ] [ ALARM ] [ HARDCOPY ]
           [ MANDATORY ] [ NOAUTOTAB ]
           [ NODISPLAY ]

```

Operanden:

DISPLAY

Name des Dienstes

PANEL(formatname)

Name des anzuzeigenden Formates. Wenn der Operand nicht angegeben ist, wird das zuletzt im PANEL-Operanden angegebene Format reaktiviert. Die Dialogvariablen werden dabei nicht erneut gelesen, d.h. die Anzeige erfolgt mit den vorher angezeigten Werten.

MSG(msgid)

Meldungsschlüssel einer Meldung, die zusammen mit dem Format anzuzeigen ist.

MSGLOC(feldname)

Name eines Feldes in der Maske, das für die Positionierung einer Meldebox verwendet wird (siehe auch Meldungsausgabe). Wenn der Feldname sich auf ein Feld eines Listenbereiches bezieht, kann ein Index zur Festlegung einer bestimmten Listenzeile angegeben werden (siehe auch Listenverarbeitung).

Das Feld muß in dem Format enthalten sein, das im PANEL-Operanden angegeben ist oder, falls kein PANEL-Operand angegeben ist, das durch den letzten DISPLAY-Aufruf mit dem PANEL-Operanden angezeigt wurde.

Existieren in dem Format mehrere Felder des angegebenen Namens, wird das erste Feld verwendet.

Eine Meldebox wird 2 Zeichen nach rechts und 2 Zeilen unterhalb des im MSGLOC-Operanden angegebenen Feldes ausgegeben. Ist unterhalb des definierten Feldes zu wenig Platz für die auszugebende Meldung, sucht der Dialogmanager nach einem bestimmten Algorithmus einen anderen Platz (siehe auch Meldungsausgabe).

Wird als feldname *CENTRAL angegeben, wird eine Meldebox in der Mitte des Bildschirms ausgegeben.

Die Angabe \$zzz#sss erlaubt eine absolute Positionierung der Meldebox.

Fehlt der MSGLOC-Operand, wird eine Meldebox bezüglich der Cursorposition platziert.

CURSOR(feldname)

Name des Feldes, in das der Cursor gestellt werden soll.

Existieren in dem Format mehrere Felder des angegebenen Namens, wird das erste Feld verwendet.

Der angegebene Name darf nur der Name eines Eingabefeldes oder eines markierbaren Textfeldes sein.

Wird der Operand CURSOR weggelassen, wird der Cursor in das erste Eingabefeld des Formates gestellt.

Wenn der Feldname sich auf ein Feld eines Listenbereiches bezieht, kann ein Index zur Festlegung einer bestimmten Listenzeile angegeben werden, z.B. CURSOR (EINFELD#3) (siehe auch Listenverarbeitung). Ist in diesem Fall kein Index angegeben, wird die oberste Zeile im Listenbereich angenommen.

Alternativ zum Feldnamen kann eine absolute Cursorposition in der Form \$zzz#sss angegeben werden. "zzz" ist die absolute Zeilenposition auf dem Bildschirm und "sss" die absolute Spaltenangabe.

CSRPOS(pos)

Position des Cursors in dem Feld, das durch den CURSOR-Operanden beschrieben wird. Das erste Zeichen im Feld hat die Position 1.

Der CSRPOS-Operand hat nur Bedeutung, wenn der Wert im CURSOR-Operanden der Name eines Eingabefeldes ist.

Wenn CSRPOS weggelassen wird oder einen unzulässigen Wert hat (kleiner als 1 oder größer als die Feldlänge), wird der Wert 1 angenommen.

LOCK

Bei Angabe dieses Operanden wird nach der Anzeige des Formates sofort zum Anwendungsprogramm zurückgekehrt. Diese Funktion kann z.B. zum Anzeigen eines Logos oder einer "Bitte warten"-Meldung verwendet werden.

ALARM

Bei Angabe dieses Operanden ertönt bei der Anzeige des Formates das akustische Signal.

HARDCOPY

Es wird automatisch der Bildschirminhalt auf ein vorhandenes Hardcopygerät ausgegeben.

MANDATORY

Alle Maskenfelder mit dem statischen Attribut "Eingabepflicht" werden wieder auf Eingabepflicht gesetzt. Dies ist nützlich, wenn nach einem DISPLAY-Aufruf die Eingabepflicht erfüllt wurde (und dadurch auf nicht mehr wirksam gesetzt ist), bei einer Wiederanzeige des gleichen Formates aber für alle Felder wieder wirksam werden soll. Diese Angabe wirkt nicht für Listfelder.

NOAUTOTAB

Der Cursor kann durch die Pfeiltasten auf geschützte Felder des Bildschirmes positioniert werden. Ist NOAUTOTAB nicht angegeben, kann der Cursor nur auf ungeschützte und markierbare Felder des Bildschirmes positioniert werden.

NODISPLAY

Das Bildschirmbild wird intern genau so aufgebaut wie ohne Angabe NODISPLAY. Es wird jedoch nicht zur Datensichtstation übertragen. Der DISPLAY-Aufruf wird beendet als ob die ENTER-Taste betätigt worden wäre. Ist der Kommandobereich mit einem FHS-Systemkommando vorbelegt, so wird dieses ausgeführt. Treten dabei Fehler auf, werden diese ignoriert.

Beispiel:

Mittels folgender COBOL-Anweisungen wird das Format FORM001 und die Meldung MSGI123 ausgegeben.

```
01 BUFLN      PIC S9(7) COMP VALUE 512.
```

```
01 BUFFER     PIC X(512) VALUE SPACE.
```

```
MOVE "DISPLAY PANEL(FORM001) MSG(MSGI123) CURSOR(FELD1)" TO BUFFER.
```

```
CALL "ISPCI" USING DMCMM BUFLN BUFFER.
```

4.5.5 DMCLOSE - Ende der Nutzung von Dialogdiensten

Der DMCLOSE-Dienst beendet den durch DMOPEN initialisierten Zugriff zu Dialogdiensten.

Durch den DMCLOSE-Aufruf wird der Funktionspool freigegeben, dessen Kennzeichen im Systemteil des Verständigungsbereiches enthalten ist, der mit dem DMOPEN-Aufruf dem Dialogmanager übergeben wird. Hat der zugehörige DMOPEN-Aufruf einen Profilpool eröffnet (DMOPEN INIT PROFILE(...)), werden, falls nicht NOSAVE angegeben wurde, die Profilvariablen in die Profilbibliothek geschrieben und alle Speicherbereiche des Dialogmanagers für den Dialogkomplex und der Profilpool freigegeben. Es kann dann kein Dialogdienst, außer einem neuen DMOPEN, ausgeführt werden.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
DMCLOSE [ NOSAVE ]
```

Operanden:

DMCLOSE

Name des Dienstes

NOSAVE

Diese Angabe darf nur für ein den Dialogkomplex abschließendes DMCLOSE (analog DMOPEN INIT) angegeben werden (ansonst wird dieser Operand ignoriert). NOSAVE bewirkt, daß der Profilpool nicht in die Profilbibliothek zurückgeschrieben wird.

Beispiel:

Mittels folgender COBOL-Anweisungen wird der Dialog beendet, dessen Verständigungsbereich in der COBOL-Variablen DMCOMM steht.

```
01 BUFLN      PIC S9(7) COMP VALUE 512.  
01 BUFFER     PIC X(512) VALUE SPACE.  
  
MOVE "DMCLOSE" TO BUFFER.  
CALL "ISPCI" USING DMCOMM BUFLN BUFFER.
```

4.5.6 DMOPEN - Beginn der Nutzung von Dialogdiensten

Der erste Dialogdienst, den ein Anwendungsprogramm aufruft, muß der DMOPEN-Dienst mit dem Operanden INIT sein. Dieser DMOPEN-Aufruf eröffnet einen Dialogkomplex (Siehe Eröffnen und Beenden einer Dialoganwendung).

Durch einen DMOPEN-Aufruf wird immer ein neuer Funktionspool eröffnet. Die Informationen zum Funktionspool werden im Systemteil des Verständigungsbereiches hinterlegt, der mit dem DMOPEN-Aufruf dem Dialogmanager übergeben wird.

Außerdem kann beim DMOPEN für einen Dialogkomplex ein Profilpool für den Dialogkomplex eröffnet werden. Dazu muß jedem Terminal-Benutzer vor der Nutzung des Dialogmanagers eine persönliche Profilbibliothek mittels des Dateikettungsnamens IDHPROF zugewiesen werden siehe auch Abschnitt "Bibliotheken zuweisen" auf Seite 170. Das Element "elementname" in der Profilbibliothek enthält die Dialogvariablen des Anwendungsprofils. Diese werden in den Profilpool eingelesen. Fehlt das Profil-Bibliothekselement, wird mit einem leeren Profilpool begonnen. Bei Ausführung des zum DMOPEN gehörenden DMCLOSE-Dienstes für einen Dialogkomplex können die Dialogvariablen des Profilpools in das Element zurückgeschrieben oder das Element erzeugt werden.

Ein DMOPEN ohne INIT eröffnet einen Dialogabschnitt. Alle Dialogabschnitte des Dialogkomplexes werden mit dem DMCLOSE, das zum DMOPEN mit INIT gehört, implizit abgeschlossen.

Jedes Anwendungsprogramm sollte vor dem Freigeben des für den Verständigungsbereich verwendeten Speicherplatzes den DMCLOSE-Dienst aufrufen.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Wenn der INIT-Operand nicht angegeben ist, muß der Systemteil des Verständigungsbereiches initialisiert bereitgestellt werden.

Eine Variablensubstitution im Operandenbereich ist nicht erlaubt.

Aufbau des Operandenbereiches:

```
DMOPEN [ INIT [ PROFILE(elementname) ] ]
```

Operanden:

DMOPEN

Name des Dienstes

INIT

Ein Dialogkomplex wird eröffnet. Diese Angabe muß beim ersten DMOPEN erfolgen.

PROFILE(elementname)

Zusätzlich zum Funktionspool wird ein Profilpool für einen Dialogkomplex eröffnet. Die Profilvariablen befinden sich im Bibliothekselement "elementname" der Profilbibliothek. Die Namenskonventionen der Profilbibliothek entsprechen denen von SDF-P.

Wird der PROFILE-Operand weggelassen, wird ein leerer, namenloser Profilpool eröffnet. Dieser wird beim DMCLOSE nicht zurückgeschrieben.

Die Operanden sind in der beschriebenen Reihenfolge anzugeben.

Beispiel:

Mittels folgender COBOL-Anweisungen wird eine Nutzung des Dialogmanagers für eine Dialogkomplex begonnen.

```
01 BUFLN      PIC S9(7) COMP VALUE 512.
```

```
01 BUFFER     PIC X(512) VALUE SPACE.
```

```
MOVE "DMOPEN INIT PROFILE(ABC)" TO BUFFER.
```

```
CALL "ISPCI" USING DMMCOMM BUFLN BUFFER.
```

4.5.7 REMPOP - Entfernen der Definition einer Dialogbox

Durch den REMPOP-Dienst wird die Definition der durch den letzten ADDPOP-Aufruf initialisierten Dialogbox oder alle Definitionen entfernt.

Dieser Dienst hat keine Auswirkungen auf den aktuellen Bildschirminhalt. Der Bildschirminhalt wird erst beim nächsten DISPLAY-Aufruf verändert. Soll die unveränderte unterliegende Maske angezeigt werden, ist der DISPLAY-Dienst ohne den Operand PANEL aufzurufen.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge der Operandenbereich
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
REMPOP [ ALL/*ALL ]
```

Operanden:

REMPOP

Name des Dienstes

ALL oder *ALL

es werden die Definitionen aller Dialogboxen gelöscht.

4.5.8 VCOPY - Kopieren von Dialogvariablen in das Anwendungsprogramm

Der VCOPY-Dienst liest den Wert einer oder die Werte mehrerer Dialogvariablen und überträgt diesen Werte oder diese Werte hintereinander in den als Zielbereich angegebenen Speicherplatz. Die Länge der zu schreibenden Werte wird durch die entsprechende Längenangabe im Längenbereich beim Aufruf bestimmt. Der Zielbereich ist ein im Anwendungsprogramm definierter Speicherbereich.

Die Dialogvariable wird zuerst im Funktionspool und, falls dort nicht gefunden, im Profilpool gesucht. Spezielle Systemvariable werden immer aktuell berechnet oder aus einem Systempool kopiert.

Der kopierte Wert wird entsprechend dem im Operanden FORMAT angegebenen Datentyp gespeichert. Wenn die zu lesende Dialogvariable einen anderen Datentyp hat als für das Zielfeld angegeben wurde, erfolgt eine Datenumwandlung.

Der durch das Längenfeld vorgegebene Platz wird mit Füllzeichen aufgefüllt, wenn der Wert der Dialogvariablen kürzer als die vorgegebene Länge ist. Ist der Wert länger, wird dieser abgeschnitten.

Parameter der Interface-Routine (ISPCI2):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich
- Längenbereich
- Zielbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Längenbereich

Ist ein Speicherbereich des Anwendungsprogrammes mit einem oder mehreren Feldern. Jedes Feld besteht aus einem Vollwort (4 Bytes, Wortgrenze) und muß eine binäre vorzeichenbehaftete Ganzzahl enthalten.

Für jeden Namen einer Dialogvariablen im Operanden "namensliste" und für jeden Füllbereich muß dieser Bereich einen Wert enthalten. Die n-te Angabe in der Namensliste und der n-te Wert im Längenbereich gehören zusammen.

Die Längenangaben im Längenbereich geben die maximale Länge des Speicherplatzes im "Zielbereich" an, der für den zu kopierenden Wert zur Verfügung steht. Mittels der Längenangaben im Längenbereich wird somit der Zielbereich strukturiert.

Nach dem VCOPY-Aufruf enthält das jeweilige Feld im Längenbereich in Abhängigkeit vom angegebenen Datentyp des Wertes im Zielbereich für die Typen CHAR und BINSTR die Länge des kopierten Wertes der Dialogvariablen ohne nachfolgende nichtbedeutsame Leerzeichen (relevante Länge). Wenn der zu kopierende Wert der Dialogvariablen nur aus nichtbedeutsamen Leerzeichen besteht (siehe Operand NOBSCAN des VDEFINE-Dienstes) oder durch die Endekennzeichnung beim Typ BINSTR die Länge Null hat, ist die relevante Länge 0. Für andere Datentypen bleibt das entsprechende Längenfeld unverändert.

Existiert die Dialogvariable weder im Funktions- noch im Profilpool, enthält das entsprechende Längenfeld nach dem VCOPY-Aufruf den Wert 0 und der zugehörige Zielbereich für den Wert ist mit Füllzeichen gefüllt. Außerdem wird ein Rückkehrcode als Warnung gesetzt.

Wird in der Namensliste als Name * angegeben, wird ein Füllbereich der angegebenen Länge definiert.

Zielbereich

Der Zielbereich ist ein Speicherbereich des Anwendungsprogrammes, in den die zu kopierenden Daten abgelegt werden sollen. Der Zielbereich wird entsprechend den maximalen Längenangaben im Längenbereich in Felder für jeweils einen Wert unterteilt. Die Werte werden in dem angegebenen Format gespeichert und gegebenenfalls bis zur maximalen Länge des Feldes mit Füllzeichen aufgefüllt.

Aufbau des Operandenbereiches:

```
VCOPY    namensliste
          [ FORMAT( { CHAR
                     BINSTR
                     NUMS
                     NUMU
                     FIXEDS
                     FIXEDU
                     PACK
                     *
                   } ,... ) ]
```

Operanden:

VCOPY

Name des Dienstes

namensliste

Ein oder mehrere Namen von Dialogvariablen, deren Werte in den Zielbereich zu kopieren sind.

Folgt einem Namen ein "#" und ein Index, wird der Wert der Dialogvariablen des entsprechenden Elementes eines Arrays von Dialogvariablen kopiert.

Ein * in der Namensliste definiert einen Füllbereich. An der gleichen Position in der Liste der Datentypen kann auch ein * stehen. Die Länge des Füllbereiches wird durch eine entsprechende Längenangabe im Längenbereich bestimmt.

FORMAT(datentyp)

Gibt den Datentyp an, den der kopierte Wert im Zielbereich haben soll (siehe auch VDEFINE). Stimmen der Datentyp der Quelle und des Zieles nicht überein, wird, falls möglich, eine Datenumwandlung durchgeführt.

"datentyp" ist entweder eine einzige Datentypangabe oder eine Liste von Datentypen. Diese Liste muß genau so viele Elemente besitzen, wie Namen in der Namensliste eingetragen sind.

Ist nur ein einziger Wert angegeben, gilt für alle Dialogvariablen dieser Datentyp.

Wird der FORMAT-Operand nicht angegeben, wird der Datentyp CHAR für alle Werte angenommen.

Ein * kann angegeben werden, wenn in der Namensliste an dieser Stelle ein Füllbereich definiert wurde.

Die zulässigen Datentypen für die Datenumwandlungen werden im folgenden dargestellt:

Zielfeld Typ	Zu kopierende Dialogvariable vom Typ (Quelle)						
	CHAR	BINSTR	NUMS	NUMU	FIXEDS	FIXEDU	PACK
CHAR	+	+	3)	3)	3)	3)	3)
BINSTR	+	+	3)	3)	3)	3)	3)
NUMS	1)	1)	+	+	+	+	+
NUMU	2)	2)	4)	+	4)	+	4)
FIXEDS	1)	1)	+	+	+	+	+
FIXEDU	2)	2)	4)	+	4)	+	4)
PACK	1)	1)	+	+	+	+	+

Dabei gilt:

+ Datenumwandlung möglich

- Datenumwandlung nicht unterstützt

1. Der Wert des Quellfeldes muß folgendes Format haben:

[+/-]9[9...][b...] linksbündig

2. Der Wert des Quellfeldes muß folgendes Format haben:

[+]9[9...][b...] linksbündig

3. Der Wert des Zielfeldes hat folgendes Format:

[-]9[9...][b...] linksbündig

4. Vorzeichenfehler möglich

Es bedeuten:

9 Ziffer 0 - 9

b Leerzeichen (oder NIL für BINSTR)

Beispiele:

Mittels folgender COBOL-Anweisungen werden die Dialogvariablen ST1, ST2 und die Systemvariable SYS-CCS-NAME dem Programm zur Verfügung gestellt. Die Dialogvariablen haben folgende Werte vom Typ CHAR:

```
ST1          : existiert nicht
ST2          : Db                b = Leerzeichen
SYS-CCS-NAME : EDF041bb

01 BUFLLEN   PIC S9(7) COMP VALUE 100.
01 BUFFER    PIX X(100) VALUE SPACE.

01 LENBER.
  02 VLEN     PIC S9(7) COMP OCCURS 3 TIMES.
01 TARGET.
  02 DCCS-NAME PIC X(8).
  02 VST.
    03 VST1   PIC X(6).
    03 VST2   PIC X(5).

MOVE 8 TO VLEN(1).
MOVE 6 TO VLEN(2).
MOVE 5 TO VLEN(3).

MOVE "VCOPY (SYS-CCS-NAME ST1 ST2)" TO BUFFER.
CALL "ISPCI2" USING DMCOMM BUFLLEN BUFFER LENBER TARGET.
```

Nach Ausführung des VCOPY-Dienstes befinden sich die Werte in den COBOL-Variablen VST1, VST2 und DCCS-NAME Die COBOL-Variable haben dann folgende Werte:

```
DCCS-NAME: EDF041bb
VST1      : bbbbbb
VST2      : Dbbbb
VLEN(1)   : 6
VLEN(2)   : 0
VLEN(3)   : 1
```

4.5.9 VDEFINE - Definieren von expliziten Dialogvariablen

Mittels des VDEFINE-Dienstes wird eine explizite Dialogvariable definiert und ihr Speicherplatz zugeordnet, der vom Anwendungsprogramm bereitgestellt wird.

Der Funktionspool enthält nach der Definition einer expliziten Dialogvariablen deren Namen, den Datentyp, den Vielfachfaktor und die Adresse des Wertes.

Mit einem einzigen VDEFINE-Aufruf kann eine Liste von Dialogvariablen definiert werden. Jede Dialogvariable dieser Liste kann einen eigenen Datentyp und eine eigene Länge haben, es können aber auch alle Variablen mit dem gleichen Datentyp und der gleichen Länge definiert werden.

Die Speicherplätze für alle Variablen, die durch einen Aufruf des VDEFINE-Dienstes definiert werden sollen, müssen unmittelbar hintereinander liegen. Einzelne Teile des Speicherplatzes können als Füllbereich definiert werden.

Mittels DIM-Operanden kann ein Array von Dialogvariablen definiert werden.

Parameter der Interface-Routine (ISPCI2):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich
- Längenbereich
- Wertebereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Längenbereich

Ist ein Speicherbereich des Anwendungsprogrammes mit einem oder mehreren Feldern. Jedes Feld besteht aus einem Vollwort (4 Bytes, Wortgrenze) und muß eine binäre vorzeichenbehafte Ganzzahl enthalten.

Wird im OPTION-Operanden LIST angegeben, muß dieser Bereich für jeden Namen im Operanden "namensliste" einen Wert enthalten. Der n-te Namen in der Namensliste und der n-te Wert im Längenbereich gehören zusammen.

Wird im OPTION-Operanden LIST nicht angegeben, braucht dieser Bereich nur ein Feld zu haben. Der Wert gilt dann für alle Dialogvariablen der Namensliste.

Die Werte im Längenbereich geben die Längen der Speicherplätze im Wertebereich an. Mittels des Längenbereiches wird somit der Wertebereich strukturiert.

Wird in der Namensliste als Name * angegeben, wird ein Füllbereich der angegebenen Länge definiert.

Wertebereich

Der Wertebereich ist ein Speicherbereich des Anwendungsprogrammes, der die Werte der angegebenen Dialogvariablen enthält.

Das Anwendungsprogramm muß sicherstellen, daß die Werte der Dialogvariablen in der angegebenen Reihenfolge und in dem angegebenen Format in diesem Speicherbereich stehen. Die Länge des n-ten Wertes im Wertebereich wird durch den Inhalt des n-ten Feldes im Längenbereich bestimmt.

Einige Programmiersprachen stellen dazu das Sprachmittel "Struktur" bereit. In COBOL müssen die COBOL-Programmvariablen, die mittels eines VDEFINE-Dienstes zu expliziten Dialogvariablen erklärt werden sollen, hintereinander definiert werden. Es ist darauf zu achten, daß die Werte lückenlos im Speicher stehen.

Aufbau des Operandenbereiches:

VDEFINE namensliste

```

[ FORMAT( {
            CHAR
            BINSTR
            NUMS
            NUMU
            FIXEDS
            FIXEDU
            PACK
            *
          } ,... ) ]

[ DIM ( zahl ) ]

[ OPTION( {
            COPY
            NOBSCAN
            LIST
          } ,... ) ]

```

Operanden:

VDEFINE

Name des Dienstes

namensliste

Die in dieser Liste angegebenen Namen definieren explizite Dialogvariable, verbinden also einen Variablennamen mit Speicherplatz des Programmes.

Eine Indexangabe ist nicht erlaubt.

Ein * in der Namensliste definiert einen Füllbereich. An der gleichen Position in der Liste der Datentypen kann auch ein * stehen. Die Länge des Füllbereiches wird durch den entsprechenden Wert im Längenbereich bestimmt.

FORMAT(datentyp)

Gibt die Darstellung des Wertes im Wertebereich an. "datentyp" ist entweder eine einzige Datentypangabe oder eine Liste von Datentypen. Diese Liste muß genau so viele Elemente besitzen, wie Namen in der Namensliste eingetragen sind.

Ist nur ein einziger Wert angegeben, haben alle definierten Dialogvariablen diesen Datentyp.

Ein * kann angegeben werden, wenn in der Namensliste an dieser Stelle ein Füllbereich definiert wurde.

Die zulässigen Datentypen werden im folgenden gesondert dargestellt.

DIM(zahl)

Der DIM-Operand (Vielfachfaktor) gibt an, daß das durch die Variablennamen, die Datentypen und die Werte im Längenbereich definierte Speicherlayout vervielfacht auftritt. D.h. es wird dadurch ein Array von Dialogvariablen definiert (siehe auch Arten von Dialogvariablen). Besteht "namensliste" aus mehreren Namen, wird ein Array von Strukturen definiert. "zahl" gibt die Anzahl der Arrayelemente an. Besteht "namensliste" nur aus einem einzigen Namen, besteht das Array aus einfachen Dialogvariablen.

Jede der in der Namensliste angegebenen Dialogvariablen hat "zahl" Elemente. Diese Elemente können in einigen Dialogdiensten als indizierte Dialogvariable angesprochen werden.

Die Angabe von COPY im Operand OPTION wird bei Angabe von DIM ignoriert.

Der Wert von "zahl" kann 1- 32767 sein.

OPTION(option)

Als "option" kann eine Kombination der folgenden Werte angegeben werden:

COPY

Für die Werte der mit diesem VDEFINE-Dienst definierten expliziten Dialogvariablen werden die Werte von gleichnamigen impliziten Dialogvariablen im Funktionspool oder die Werte von im Profilpool existierenden gleichnamigen Dialogvariablen oder gleichnamigen Systemvariablen des Systempools (siehe Anhang) kopiert. Beim Kopieren der Werte findet evtl. eine Datenkonvertierung statt. Es wird erst der Funktionspool, dann der Systempool und danach der Profilpool durchsucht. Implizite Dialogvariable werden nach dem Kopieren ihres Wertes aus dem Funktionspool gelöscht. Dadurch wird ein Namenskonflikt vermieden. Wenn der DIM-Operand angegeben ist, wird COPY ignoriert.

NOBSCAN

Dieser Wert hat nur für die Datentypen CHAR und BINSTR eine Bedeutung.

Wird eine explizite Dialogvariable mit dem Attribut NOBSCAN definiert, so werden nachfolgende Leerzeichen als bedeutsam erklärt und beim Lesen dieser Variablen wird als Länge der Daten für CHAR die bei VDEFINE definierte Länge und für BINSTR eine Länge entsprechend der C-Funktion strlen geliefert. Ist NOBSCAN nicht vereinbart, werden nachfolgende Leerzeichen als nicht bedeutsam angesehen und als Länge wird die Länge der Daten ohne nachfolgende Leerzeichen bereitgestellt. Man kann sich den Unterschied auch als Daten mit fester oder variabler Länge vorstellen.

Folgende Beispiele sollen dies verdeutlichen:

				Daten der Variablen	Länge beim Lesen der Dialogvariable
VDEFINE	...	CHAR	4 NOBSCAN	Abbb	4
VDEFINE	...	CHAR	4	Abbb	1
VDEFINE	...	CHAR	4 NOBSCAN	bbbb	4
VDEFINE	...	CHAR	4	bbbb	0
VDEFINE	...	BINSTR	5 NOBSCAN	Abbb0	4
VDEFINE	...	BINSTR	5	Abbb0	1
VDEFINE	...	BINSTR	5 NOBSCAN	bbbb0	4
VDEFINE	...	BINSTR	5	bbbb0	0
VDEFINE	...	BINSTR	5 NOBSCAN	A0000	1
VDEFINE	...	BINSTR	5	A0000	1
VDEFINE	...	BINSTR	5 NOBSCAN	00000	0
VDEFINE	...	BINSTR	5	00000	0

Dieser Operand hat in Verbindung mit VCOPY und dem Anzeigen der Dialogvariablen in einem Maskenfeld und bei der Variablenersetzung eine Bedeutung.

LIST

LIST ist anzugeben, wenn die Dialogvariablen der Namensliste unterschiedliche Länge haben. Damit kann jeder Dialogvariablen der Namensliste eine Länge zugeordnet werden.

Wird LIST nicht angegeben, haben alle Variablen der Namensliste die gleiche Länge. Wird in diesem Fall mehr als eine Länge angegeben, wird die erste Längenangabe für alle Variablen der Namensliste verwendet. Die weiteren Angaben werden ignoriert.

Nur wenn LIST angegeben wurde, können in der zu definierenden Struktur von Dialogvariablen Füllbereiche verwendet werden.

Datentypen:

Die in den Beschreibungen der Datentypen gemachten Längenbegrenzungen gelten für den Dialogmanager. Im Anwendungsprogramm können durch die verwendete Programmiersprache stärkere Einschränkungen gelten.

Wenn durch den DISPLAY-Dienst eine explizite Dialogvariable in einer Maske angezeigt werden soll, muß beim Lesen der Variablen eine Datenumwandlung in eine als externes Format bezeichnete Zeichenfolge erfolgen. Beim Speichern eines in einer Maske eingegeben Wertes in eine explizite Dialogvariable muß das als Zeichenfolge angegebene externe Format in das interne Format der Dialogvariablen umgewandelt werden (siehe auch Anwendung der Anzeigedienste).

Eine Datenumwandlung kann auch beim Kopieren oder Ersetzen einer Dialogvariablen notwendig werden.

Im folgenden wird die Darstellung des internen Wertes in Abhängigkeit vom Datentyp beschrieben:

CHAR

Zeichenkette. Der Inhalt wird nach links ausgerichtet und mit Leerzeichen aufgefüllt.

CHAR-Variable können 1 - 32767 Bytes lang sein.

Es erfolgt keine Prüfung auf zulässige Zeichen beim Zugriff auf die Dialogvariable.

NUMS

Numerische Zeichenkette mit Vorzeichen. Es sind nur die Ziffern 0 bis 9 zulässig und ein Vorzeichen (+/-). Der Wert wird rechts ausgerichtet und mit Nullen (C'0') aufgefüllt. Das Vorzeichen befindet sich unmittelbar rechts neben der letzten Ziffer.

Eine NUMS-Variable darf 2-16 Bytes lang sein.

Dieser Variablentyp unterstützt die bisherige numerische Datendarstellung in FHS-Adressierungshilfen. Die externe Darstellung der Variablen ist nur in einem arithmetischen Maskenfeld mit entsprechenden Aufbereitungseigenschaften möglich.

NUMU

Numerische Zeichenkette ohne Vorzeichen. Es sind nur die Ziffern 0 bis 9 zulässig. Der Wert wird rechts ausgerichtet und mit Nullen (C'0') aufgefüllt.

Eine NUMU-Variable darf 1-15 Bytes lang sein.

Dieser Variablentyp unterstützt die bisherige numerische Datendarstellung in FHS-Adressierungshilfen. Die externe Darstellung der Variablen ist nur in einem arithmetischen Maskenfeld mit entsprechenden Aufbereitungseigenschaften möglich.

FIXEDS

Binäre vorzeichenbehaftete Ganzzahl, die 2 oder 4 Bytes belegen kann.

FIXEDU

Binäre Ganzzahl ohne Vorzeichen, die 2 oder 4 Bytes belegen kann.

BINSTR

Dieser Variablentyp unterscheidet sich vom Typ CHAR nur durch das Füllzeichen. Als Füllzeichen wird X'00' verwendet.

Beim Speichern des Wertes einer BINSTR-Variablen wird der für den Wert vorgesehene Platz maximal bis maximale Länge-1 mit dem Wert belegt, das letzte Zeichen enthält das Füllzeichen. Beim Lesen einer BINSTR-Variablen ist beim ersten Zeichen X'00' das Ende des Wertes erreicht.

Mit diesem Datentyp wird der Typ STRING der Sprache C unterstützt.

PACK

Gepackte Dezimalzahl, die 1 - 8 Bytes belegen kann.

Enthält das letzte Halbbyte X'C' ist der Wert positiv und bei X'D' negativ. Bei anderem Inhalt wird der Wert der Variablen als vorzeichenlos betrachtet.

Mit diesem Datentyp wird der Typ PACKED DECIMAL von COBOL unterstützt.

Beispiele:

1. Mittels folgender COBOL-Anweisungen wird ein Array von Dialogvariablen X, Y und Z in der Struktur S explizit deklariert.

```
01 LBUF          PIC S9(7)  COMP VALUE 512.
01 BUF           PIC X(512) VALUE SPACE.
01 LENFELD.
  02 VLEN        PIC S9(7)  COMP OCCURS 3 TIMES.
01 S.
  02 SE          OCCURS 3 TIMES.
    03 A         PIC X(10).
    03 B         PIC S9(7)  COMP.
    03 C         PIC X(2).

MOVE 10 TO VLEN(1).
MOVE 4 TO VLEN(2).
MOVE 2 TO VLEN(3).
MOVE "VDEFINE (X Y Z) FORMAT(CHAR FIXEDS CHAR) DIM(3)
      OPTION(LIST)" TO BUF.
CALL "ISPCI2" USING DMCOMM LBUF BUF LENFELD S.
```

2. Mittels folgender COBOL-Anweisungen wird die Dialogvariable TEXT deklariert, die im COBOL-Programm als Variable FEHLER angesprochen wird.

```
01 LBUF          PIC S9(7)  COMP VALUE 512.
01 BUF           PIC X(512) VALUE SPACE.
01 FEHLER        PIC X(80).
01 VLEN          PIC S9(7)  COMP VALUE 80.

MOVE "VDEFINE TEXT FORMAT(CHAR)" TO BUF.
CALL "ISPCI2" USING DMCOMM LBUF BUF VLEN FEHLER.
```

4.5.10 VDELETE - Löschen von Dialogvariablen im Funktionspool

Mit dem VDELETE-Dienst kann das Anwendungsprogramm implizite oder explizite Dialogvariable im Funktionspool löschen.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
VDELETE {  namensliste  }
        {  *ALL        }
```

Operanden:

VDELETE

Name des Dienstes

namensliste

Eine Liste von Namen der Dialogvariablen, die gelöscht werden sollen.

Soll eine Struktur von Dialogvariablen gelöscht werden, sind alle Namen der Dialogvariablen der Struktur anzugeben.

Soll ein Array von Dialogvariablen entfernt werden, sind alle Namen der Dialogvariablen eines Arrayelementes (Struktur oder einfache Dialogvariable) ohne Index anzugeben.

*ALL

Bei Angabe von *ALL werden alle Definitionen von Dialogvariablen im Funktionspool gelöscht.

Beispiel:

```
01 LBUF          PIC S9(7)  COMP VALUE 512.  
01 BUF          PIC X(512) VALUE SPACE.
```

```
MOVE "VDELETE *ALL" TO BUF.  
CALL "ISPCI" USING DMMCOMM LBUF BUF.
```

4.5.11 VERASE - Löschen von Dialogvariablen im Profilpool

Mit dem VERASE-Dienst kann das Anwendungsprogramm Dialogvariable aus dem Profilpool löschen.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
VERASE { namensliste }  
      { *ALL }
```

Operanden:

VERASE

Name des Dienstes

namensliste

In "namensliste" werden die Namen der Dialogvariablen angegeben, die im Profilpool gelöscht werden sollen. Eine Indexangabe ist nicht erlaubt.

*ALL

Bei Angabe von *ALL werden alle Dialogvariablen im Profilpool gelöscht.

4.5.12 VGET - Kopieren von Variablen aus Profilpool in den Funktionspool

Die Werte von Dialogvariablen aus dem Profilpool oder von S-Variablen aus einem SDF-P-Variablenpool werden in den Funktionspool kopiert.

Existiert eine gleichnamige Dialogvariable schon im Funktionspool, wird nur ihr Wert geändert. Dabei findet eine Datenumwandlung statt, wenn die Datentypen der Variablen nicht übereinstimmen.

Existiert eine der angegebenen Dialogvariablen nicht im Funktionspool, wird sie implizit angelegt und erhält den zu kopierenden Wert.

Existiert eine zu kopierende Dialogvariable oder S-Variable nicht, bleibt eine gleichnamige Dialogvariable im Funktionspool unverändert. Eine implizite Dialogvariable wird nicht angelegt.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
VGET namensliste [ { PROFILE
                   PROCEDURE } ]
                   TASK
```

Operanden:

VGET

Name des Dienstes

namensliste

Eine Liste mit Namen von Dialogvariablen (oder S-Variablen), deren Werte in den Funktionspool kopiert werden sollen. Eine Indexangabe ist nicht erlaubt.

PROFILE

Die in der Namensliste angegebenen Dialogvariablen sind aus dem Profilpool zu lesen. Diese Angabe kann weggelassen werden (Voreinstellung).

PROCEDURE

Die in der Namensliste angegebenen S-Variablen sind aus dem aktuellen SDF-P-Prozedurpool zu kopieren.

TASK

Die in der Namensliste angegebenen S-Variablen sind aus dem Task-spezifischen SDF-P-Pool zu kopieren.

4.5.13 VPUT - Kopieren von Variablen aus Funktionspool in Profilpool oder in SDF-P-Variablenpool

Die augenblicklichen Werte von Dialogvariablen des Funktionspools werden in den Profilpool oder in einen SDF-P-Variablenpool kopiert.

Existiert bereits eine gleichnamige Dialogvariablen im Ziel-Pool wird nur ihr Wert geändert. Dabei findet eine Datenumwandlung statt, wenn die Datentypen der Variablen nicht übereinstimmen. Ist das Ziel der Profilpool, wird eine implizite Dialogvariable im Profilpool angelegt, wenn sie dort nicht existiert. Diese implizite Dialogvariable erhält den Datentyp FIXEDS, wenn die zu kopierende Variable vom Typ FIXEDS oder FIXEDU ist. Anderenfalls erhält sie den Datentyp CHAR.

Existiert eine angegebene Dialogvariable nicht im Funktionspool oder in einem SDF-P-Variablenpool, wird als Warnung ein entsprechender Rückkehrcode geliefert.

Parameter der Interface-Routine(ISPCI):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Aufbau des Operandenbereiches:

```
VPUT namensliste [ { PROFILE
                   { PROCEDURE }
                   TASK } ]
```

Operanden:

VPUT

Name des Dienstes

namensliste

Eine Liste mit Namen von Dialogvariablen, deren Werte kopiert werden sollen. Eine Indexangabe ist nicht erlaubt.

PROFILE

Die in der Namensliste angegebenen Dialogvariablen sind in den Profilpool zu kopieren. Diese Angabe kann weggelassen werden (Voreinstellung).

PROCEDURE

Die in der Namensliste angegebenen Dialogvariablen sind in den aktuellen SDF-P-Prozedurpool zu kopieren..

TASK

Die in der Namensliste angegebenen Dialogvariablen sind in den Task-spezifischen SDF-P-Pool zu schreiben.

Es existieren folgende Längeneinschränkungen:

- Profilepool: maximale Länge für den Datentyp CHAR ist 16383;
- SDF-P-Variablenpool: maximale Länge ist 4096.

4.5.14 VREPLACE - Ersetzen von Dialogvariablen im Funktionspool

Mittels des VREPLACE-Dienstes werden die Werte von impliziten oder expliziten Dialogvariablen im Funktionspool verändert.

Eine zu verändernde Dialogvariable kann bereits vorher als explizite oder implizite Dialogvariable definiert worden sein. Wenn eine der angegebenen Dialogvariablen noch nicht existiert, wird sie als implizite Dialogvariable erzeugt.

Die implizite Dialogvariable erhält den Datentyp FIXEDS, wenn der Datentyp des neuen Wertes (Quellwert) FIXEDS oder FIXEDU ist. Ansonsten erhält die implizite Dialogvariable den Datentyp CHAR.

Wenn eine zu verändernde Dialogvariable einen anderen Datentyp hat als der neue Wert, wird zur Abspeicherung des neuen Wertes eine Datenumwandlung durchgeführt. Die bestehende Dialogvariable behält ihren Datentyp.

Eine implizit existierende Dialogvariable wird entfernt, wenn der neue Wert die Länge 0 hat.

Parameter der Interface-Routine (ISPCI2):

- Verständigungsbereich
- Länge des Operandenbereiches
- Operandenbereich
- Längenbereich
- Wertebereich

Beschreibung der Parameter:

Die Parameter müssen so angegeben werden, wie es im Abschnitt "Interface-Routine (ISPCI)" Seite 116 beschrieben wird.

Längenbereich

Ist ein Speicherbereich des Anwendungsprogrammes mit einem oder mehreren Feldern. Jedes Feld besteht aus einem Vollwort (4 Bytes, Wortgrenze) und muß eine binäre vorzeichenbehafte Ganzzahl enthalten.

Für jeden Namen einer Dialogvariablen im Operanden "namensliste" muß dieser Bereich einen Wert enthalten. Der n-te Namen in der Namensliste und der n-te Wert im Längenbereich gehören zusammen.

Diese Werte geben die Länge der Werte im Wertebereich an.

Wertebereich

Der Wertebereich ist ein Speicherbereich im Anwendungsprogramm, der die neuen Werte (Quellwerte) für die Dialogvariablen enthält. Für jeden Namen der Namensliste muß ein Wert vorhanden sein.

Das Anwendungsprogramm muß sicherstellen, daß die Werte der Dialogvariablen lückenlos in der durch die Namensliste vorgegebenen Reihenfolge und in dem angegebenen Format in diesem Speicherbereich stehen.

Die Länge des n-ten Wertes wird durch den Inhalt des n-ten Feldes des Längenbereiches bestimmt.

Aufbau des Operandenbereiches:

VREPLACE namensliste

```

          [ CHAR      ]
          [ BINSTR   ]
          [ NUMS     ]
          [ NUMU     ]
[ FORMAT( { FIXEDS } , ... ) ]
          [ FIXEDU   ]
          [ PACK     ]
          [ *        ]

```

Operanden:

VREPLACE

Name des Dienstes

namensliste

Ein oder mehrere Namen von Dialogvariablen, deren Werte im Funktionspool verändert werden sollen.

Folgt einem Namen ein "#" und ein Index, wird der Wert der Dialogvariablen des entsprechenden Elementes eines Arrays von Dialogvariablen verändert.

Ein * in der Namensliste definiert einen Füllbereich. An der gleichen Position in der Liste der Datentypen kann auch ein * stehen. Die Länge des Füllbereiches wird durch den entsprechenden Wert im Längenbereich bestimmt.

FORMAT(datentyp)

Gibt den Datentyp an, den der neue Wert (Quellwert) im Wertebereich hat (siehe auch VDEFINE). Stimmen der Datentyp der Quelle und des Zieles nicht überein, wird, falls möglich, eine Datenumwandlung durchgeführt.

"datentyp" ist entweder eine einzige Datentypangabe oder eine Liste von Datentypen. Diese Liste muß genau so viele Elemente besitzen, wie Namen in der Namensliste eingetragen sind.

Ist nur ein einziger Wert angegeben, haben alle Quellwerte diesen Datentyp.

Wird der FORMAT-Operand nicht angegeben, wird für alle Quellwerte der Datentyp CHAR angenommen.

Ein * kann angegeben werden, wenn in der Namensliste an dieser Stelle ein Füllbereich definiert wurde.

Die zulässigen Datentypen für die Datenumwandlungen werden im folgenden dargestellt:

Quellwert Typ	Zu ersetzende Dialogvariable (Ziel) vom Typ:							impl. Dialogva- riable
	CHAR	BINSTR	NUMS	NUMU	FIXEDS	FIXEDU	PACK	
CHAR	+	+	1)	2)	1)	2)	1)	CHAR
BINSTR	+	+	1)	2)	1)	2)	1)	CHAR
NUMS	3)	3)	+	4)	+	4)	+	CHAR
NUMU	3)	3)	+	+	+	+	+	CHAR
FIXEDS	3)	3)	+	4)	+	4)	+	FIXEDS
FIXEDU	3)	3)	+	+	+	+	+	FIXEDS
PACK	3)	3)	+	4)	+	4)	+	CHAR

Dabei gilt:

+ Datenumwandlung möglich

- Datenumwandlung nicht unterstützt

1. Der Wert des Quellfeldes muß folgendes Format haben:

[+/-]9[...][b...] linksbündig

2. Der Wert des Quellfeldes muß folgendes Format haben:

[+]9[...][b...] linksbündig

3. Der Wert des Zielfeldes hat folgendes Format:

[-]9[...][b...] linksbündig

4. Vorzeichenfehler möglich

Es bedeuten:

- 9 Ziffer 0 - 9
- b Leerzeichen (oder NIL für BINSTR)

Beispiele:

```
01 LBUF            PIC S9(7)  COMP VALUE 512.
01 BUF            PIC X(512) VALUE SPACE.
01 WERTE          PIC X(80).
01 VLEN.
03 LV             PIC S9(7)  COMP OCCURS 4 TIMES.
MOVE "1234567890ABCDEFGHIJKL" TO WERTE.
MOVE 10 TO LV(1).
MOVE 10 TO LV(2).
MOVE "VREPLACE (V1 V2)" TO BUF.
CALL "ISPCI2" USING DMCOMM LBUF BUF VLEN WERTE.
MOVE 5 TO LV(1).
MOVE 10 TO LV(2).
MOVE "VREPLACE (V3 V4) FORMAT(CHAR)" TO BUF.
CALL "ISPCI2" USING DMCOMM LBUF BUF VLEN WERTE.
```

Die Dialogvariablen haben nach Ausführung der VREPLACE-Dienste folgende

Werte:

```
V1    1234567890
V2    ABCDEFGHIJ
V3    12345
V4    67890ABCDE
```

4.6 Bibliotheken zuweisen

Folgende Bibliotheken werden für die Arbeit mit dem Dialogmanager benötigt und müssen vor dem ersten Aufruf von Dialogdiensten zugeordnet werden. Die Zuordnung ist bis zum Ende der Nutzung gültig.

- Formatbibliothek

Die Formatbibliothek enthält alle notwendigen Dialogelemente. Es muß eine Hauptbibliothek mit dem Dateikettungsnamen IDHPLIB zugewiesen werden. Alternativbibliotheken können über einen Dateikettungsnamen BLSLIBnn zugewiesen werden. Dabei gibt nn eine Nummer von 00 bis 99 an. Die Bibliotheken werden nach aufsteigenden Nummern durchsucht. Eine lückenlose Reihenfolge ist nicht erforderlich.

- Profilbibliothek

Die Profilbibliothek enthält die Profilvariablen und ist benutzerspezifisch. Jedem Terminal-Benutzer muß seine Profilbibliothek mit dem Dateikettungsnamen IDHPROF zugewiesen werden.

Nutzung verschiedener Landessprachen: Die sprachspezifischen Teile (Formate, Tastenbelegungen, Hilfen und Meldungen) befinden sich in einer Formatbibliothek. Durch Zuordnung der jeweiligen sprachspezifischen Formatbibliothek beim Aufruf von FHS-DM, wird für den jeweiligen Terminal-Benutzer seine Sprachvariante wirksam.

4.7 Kompatibilität

Der Dialogmanager des FHS ist nur für TIAM-Anwendungen nutzbar. Es besteht keine Kompatibilität zu Anwendungsprogrammen, die FHS in der bisherigen Weise nutzen. Solche Anwendungen sind ohne Einschränkungen weiterhin lauffähig.

Die Nutzung des Dialogmanagers ist nur mit Formaten möglich, die vom IFG ab Version V08.1A für die Nutzung durch den Dialogmanager vorgesehen sind. Diese Formate werden hier als DM-Formate bezeichnet.

Die für die Dialogerweiterung des FHS Version 8.0 erzeugten DE-Formate zur Nutzung unter UTM können nur eingeschränkt durch den Dialogmanager verarbeitet werden. Dies ist im allgemeinen auch nicht sinnvoll, da DE-Formate für UTM-Anwendungen vorgesehen sind.

DM-Formate sind erweiterte DE-Formate und unterscheiden sich durch:

- Vom IFG überprüfte Zuordnung von Feldnamen für alle Ein- und Ausgabefelder sowie für markierbare Textfelder. Dies ist wegen der Verbindung mit den Dialogvariablen zwingend notwendig.
- Zusätzliche Namen für Dialogvariablen für Auswahlfelder, Listenbereiche und Pull-down-Box. Ohne diese Angaben kann der Dialogmanager diese Elemente nur eingeschränkt verarbeiten.
- Führungstext für Verschiebeinformation in Hilfeformaten.

Keine Unterschiede ergeben sich für Meldungen und Tastenbelegungstabellen.

Durch den Wegfall von Adressierungshilfen ist die Arbeit mit globalen und feldbezogenen Attributen nicht in der bisherigen Weise möglich. Der Dialogmanager bietet dazu andere Möglichkeiten.

Nicht alle mit dem IFG beschreibbaren Funktionen werden durch den Dialogmanager verarbeitet. Es werden folgende Möglichkeiten ignoriert:

- Angabe zum Eingabefüllzeichen
- Angabe undefinierter Werte
- Bereitstellung einer Exit-Routine.

4.8 SDF-P-Variable in FHS-TIAM-Programmen

Zugriff auf SDF-P-Variable

Die Variablendienste VGET und VPUT erlauben einem Anwendungsprogramm, einfache SDF-P-Variable zu lesen und zu schreiben. Damit ist eine Kommunikation zwischen einer SDF-P-Prozedur und einem von dieser Prozedur aufgerufenen Anwendungsprogramm möglich. Der Einsatz von SDF-P-Variablen wird ausführlich im nächsten Kapitel beschrieben.

5 SDF-P-Schnittstelle

Ab FHS V8.1 kann FHS die Rolle des Ausgabe-Servers für S-Variablenströme übernehmen. Dazu können mit Hilfe von S-Prozeduren Anwendungen für FHS V8.1 benutzt und gesteuert werden.

Beide Funktionen werden in den folgenden zwei Abschnitten beschrieben.

5.1 FHS als Ausgabe-Server

Als Ausgabe-Server ist FHS sowohl auf Programm- wie auch auf Kommandoebene im SDF-P-Kontext verfügbar, d.h.:

- FHS-PRIV V8.1 kann auf Kommandoebene durch TRANSMIT-BY-STREAM aufgerufen werden.
- FHS-PRIV V8.1 kann durch ein TU-Programm unter Benutzung eines TRANSVV SVC aufgerufen werden.

Voraussetzung für beide Möglichkeiten ist, daß der Variablenstrom, der vom Programm benutzt wird, im Operanden SERVER des Kommandos ASSIGN-STREAM FHS zugewiesen wird.

Wenn durch ein TRANSMIT-BY-STREAM-Kommando Informationen bei FHS V8.1 eingehen, stellt FHS Anzeigedienste zur Verfügung, die den Diensten entsprechen, die FHS im TU-Modus durch die Dienste DISPLAY, ADDPOP and REMPOP anbietet.

5.1.1 ASSIGN-STREAM

Um FHS als Server für S-Variablenströme zuzuweisen, müssen Sie bei dem Kommando ASSIGN-STREAM im Operanden TO die Zuweisung SERVER(...) FHS vornehmen. Im Operanden STREAM-NAME können Sie für FHS-Anwendungen eigene S-Variablenströme zuweisen. Weitere Informationen hierzu finden Sie im Handbuch "SDF-P V2.0".

Beispiel

```
ASSIGN-STREAM STREAM-NAME=<structured-name 1...20>
                TO=*SERVER(SERVER-NAME=FHS,
                           SERVER-INFORMATION="FHS-LIB=format library
                                                [,P-KEY-SETTING=(P1,...,P20)]")
```

Wird in SDF-P dieses Kommando eingegeben, wird FHS aufgerufen. FHS wertet die Angabe in SERVER-INFORMATION aus:

- FHS-LIB ist die Format-Bibliothek von FHS. Der Name der FHS-Bibliothek muß wie folgt angegeben werden:

```
SERVER-INFORMATION = "FHS-LIB=$USER-ID.library-name".
```

An dieser Stelle müssen Sie den Namen einer Formatbibliothek angeben. Mit einem LINK-Kommando können Sie auch eine andere Formatbibliothek angeben, z.B. um die deutschen Standardformate IDHx.... aus \$.SYSFHS.FHS.081.FHS-DM.D einzubinden.

Die Formate werden in der unter SERVER-INFORMATION angegebenen Bibliothek gesucht und dann in die benutzte BLSLIBxx (00 - 99) geladen.

- P-KEY-SETTING gibt an welche F-Taste durch eine P-Taste simuliert wird. Diese Angabe ist optional, sie dient hier zur Veranschaulichung des Assign-Stream-Kommandos.

5.1.2 TRANSMIT-BY-STREAM

Die Übertragung der S-Variablen wird in S-Prozeduren mit dem Kommando TRANSMIT-BY-STREAM geregelt. Dieses Kommando steuert von Client-Seite die Variablenübermittlung über einen ausgewählten S-Variablenstrom zum bzw. vom angegebenen Server.

Man gibt dabei im Operanden STREAM-NAME den ausgewählten Variablenstrom an.

In den restlichen Operanden wird in Entsprechung zur Einstellung bei ASSIGN-STREAM ...,TO=*VARIABLE() bestimmt, welche Variablen zum bzw. vom Server übertragen werden. D.h. analog zu ASSIGN-STREAM müssen hier die angegebenen Variablen als Strukturen deklariert sein.

Beispiel

```
TRANSMIT-BY-STREAM  STREAM-NAME = <structured-name 1..20>
                    ,VARIABLE-NAME = <composed-name 1..255>
                    ,RETURN-VARIABLE-NAME = *SAME / <composed-name 1..255>
                    ,CONTROL-VAR-NAME = <composed-name 1..255>
                    ,RET-CONTROL-VAR-NAME = *SAME / <composed-name 1..255>
```

Wird in SDF-P dieses Kommando eingegeben und ist FHS als Server zugeordnet, wird FHS aufgerufen und der Inhalt der S-Variablenströme wird in FHS-Masken ausgegeben.

FHS kann selber Variablen generieren, die an den Aufrufer zurückgeliefert werden. Diese kommen entweder aus Eingabefeldern (Werte vom Terminal-Benutzer eingegeben) oder von FHS (z.B. Position des Cursors,...).

Der Inhalt der Variablen gliedert sich in zwei unterschiedlich Typen:

- Daten
- Kontroll-Variablen

Für FHS werden die Kontroll-Variablen in der S-Variablen SYSFHS-CONTROL beschrieben (siehe auch Seite 193).

Das Layout wird als Prozedur in der Bibliothek SYSPRC.FHS.081 im Element DISPLAY-CONTROL zur Verfügung gestellt. Die Elemente sind reservierte Namen mit einer festgelegten Semantik. Variablen, die mit dieser Prozedur deklariert werden, müssen mit SCOPE=VISIBLE deklariert sein.

In SYSFHS-CONTROL sind 7 verschiedene Informationen enthalten:

- Layout und Version
- Information über die Aktion
- Resource (Bezeichnung der Maske/Meldung)
- Information über das Format
- Information über Feldattribute
- Information über Eingabeformatierung
- Rückgabe-Werte (Return Code)

Nach dem Aufruf setzt FHS die Elemente der Kontroll-Variablen auf Standardwerte zurück, falls solche definiert wurden.

5.2 FHS-Dienste für SDF-P-Anwendungen

Um aus einer SDF-P-Prozedur einen Variablen-Dienst aufzurufen, müssen Sie beim TRANSMIT-BY-STREAM-Kommando in der Variablen-Struktur den entsprechenden Wert einsetzen. Die möglichen Werte sind im folgenden Abschnitt aufgelistet.

Zur Vereinfachung sind in den nachfolgenden Tabellen die Hochkommata bei den String-Werten weggelassen. Beachten Sie bitte, daß einer String-Variablen ein in Hochkomma eingeschlossener Wert zugeordnet werden muß.

Beispiel:

```
/MYFHS-CONTROL.SERVICE = "*ADD-POPUP"  
/MYFHS-CONTROL.POP-LOCATION = "MYFIELD"
```

An den Stellen an denen ein Feldname angegeben werden kann, ist beim Arbeiten mit Listen eine Indexangabe möglich. Der Index muß durch einen "*" (Stern) ersetzt und der Index muß angegeben werden.

5.2.1 Layout und Version

Diese Variablen werden im Standard-Header von SDF-P ausgeliefert. Nachfolgend werden die Werte für FHS aufgeführt. FHS V8.1 implementiert Layout Version 1.

name	type	possible value
UNIT	string	'FHS'
FUNCTION	string	'DISPLAY'
VERSION	integer	1

5.2.2 Information über Anzeigendienste

Um aus einer SDF-P-Prozedur einen Variablen-Dienst (DISPLAY-, ADDPOP- und REMPOP-Dienst) aufzurufen, müssen Sie beim TRANSMIT-BY-STREAM-Kommando bei der Service-Angabe den gewünschten Dienst eintragen. Die nachfolgend aufgeführten Angaben sind möglich.

Name	Typ	Standardwert	möglicher Wert
SERVICE	string	*DISPLAY	*ADD-POPUP / *REMOVE-POPUP / *ADDDPOP-DISPLAY / *REMPPOP-DISPLAY
DIAGINFO	string	*NO	*YES / *WARNING
POP-LOCATION	string	*NONE	<field-name 1..255> / *CENTRAL
POP-LOC-IND	integer	0	<integer>
ROW	integer	2	<integer>
COLUMN	integer	2	<integer>

SERVICE:

der von FHS auszuführende Dienst:

- *DISPLAY Ausgabe eines Formates der aktuellen Box-Hierarchiestufe (wie bei TU-FHS, siehe Seite 137).
- *ADD-POPUP hinzufügen einer Pop-up Hierarchie (wie bei TU-FHS, siehe Seite 126).
- *REMOVE-POPUP entfernen/löschen einer Pop-up Hierarchie (wie bei TU-FHS, siehe Seite 146).
- *ADDDPOP-DISPLAY kombinierte Aktion; hinzufügen einer Pop-up Hierarchie und Ausgabe eines Formates dieser Hierarchiestufe.
- *REMPPOP-DISPLAY kombinierte Aktion; löschen einer Pop-up Hierarchie und Ausgabe eines Formates einer niedrigeren Hierarchiestufe.

DIAGINFO

Anzeige von zusätzlicher Diagnoseinformation (wie DMERR bei TU-FHS).

- *NO keine Diagnoseinformationen
- *YES Diagnoseinformationen im Fehlerfall
- *WARNING Diagnoseinformationen im Fehlerfall und bei Warnungen

POP-LOCATION:

Definition des Referenzpunktes einer Box.

*NONE Referenzpunkt ist die obere linke Ecke des aktuellen Formats.

<field name 1..255>

Name des Feldes dessen erster Buchstabe den Startpunkt der Box darstellt.

*CENTRAL Mitte des Bildschirms

POP-LOC-IND:

Indexangabe für POP-LOCATION

Wenn der Feldname sich auf ein Feld eines Listenbereiches bezieht, können Sie einen Index zur Festlegung einer bestimmten Listenzeile angeben.

Beispiel:

Für POP-LOCATION müssen Sie folgendes angeben:

POP-LOCATION = 'A#*'

POP-LOC-IND = 4

ROW:

Senkrechte Distanz der Box zum Referenzpunkt:
positiver Wert: unterhalb, negativer Wert: oberhalb.

COLUMN:

Waagrechte Distanz der Box zum Referenzpunkt:
positiver Wert: rechts, negativer Wert: links.

5.2.3 Angaben zu Formatnamen und Meldungen

Diese Variablen beschreiben das IFG-Format und falls vorhanden die Meldung und den Ausgabeort.

Name	Typ	Standardwert	mögliche Werte
RESOURCE	string	*SAME	<panel-name> / *ALL
MESSAGE-ID	string	*NONE	<msg-id>
MESSAGE-FIELD	string	*NONE	<field-name 1..255> / \$zzz#sss / *CENTRAL
MSG-FIELD-IND	integer	0	<integer>

RESOURCE:

Name des Formats.

*SAME das zuletzt ausgegebene Format wird neu ausgegeben

<panel-name> Name des Formates

*ALL wird beim REMPOP Dienst verwendet, um alle Formate zu entfernen.

MESSAGE-ID:

Kennzeichen der Meldung, die im Format ausgegeben/angezeigt werden soll.

MESSAGE-FIELD:

Name des Feldes für das die Meldung ausgegeben/angezeigt werden soll.

*NONE Referenzpunkt ist die obere linke Ecke des aktuellen Formates

<field-name 1..255>

Name des Feldes dessen erster Buchstabe den Startpunkt der Box darstellt

\$zzz#sss Koordinaten des Feldes (zzz = Zeile, sss = Spalte)

*CENTRAL Mitte des Bildschirms

MSG-FIELD-IND

Indexangabe für MESSAGE-FIELD.

5.2.4 Angaben zu Formateigenschaften bei der Ausgabe

Die Variablen beschreiben die Ausgabeattribute eines Formats.

Name	Typ	Standardwert	mögliche Werte
CURSOR-OUTPUT-INDEX	integer	0	<integer>
CURSOR-OUTPUT	string	*NONE	<field-name 1..255> / \$zzz#sss
CURSOR-OUTPUT-POS	integer	0	<integer>
LOCK	string	*NO	*YES
ALARM	string	*NO	*YES
HARDCOPY	string	*NO	*YES
AUTOTAB	string	*YES	*NO
MANDATORY	string	*NO	*YES
REFRESH	string	*NO	*YES
ACK	string	*NO	*YES

CURSOR-OUTPUT-INDEX:

Indexangabe für CURSOR-OUTPUT

Wenn der Feldname sich auf ein Feld eines Listenbereiches bezieht, können Sie einen Index zur Festlegung einer bestimmten Listenzeile angeben.

CURSOR-OUTPUT:

Name des Feldes, in dem der Cursor bei der Ausgabe stehen soll:

*NONE der Cursor steht im ersten Feld.

<field-name 1..255>
 Name des Feldes.

\$zzz#sss Koordinaten des Feldes (zzz = Zeile, sss = Spalte).

CURSOR-OUTPUT-POS:

Position des Cursors in dem Feld, das durch den CURSOR-Operanden beschrieben wird. Das erste Zeichen im Feld hat die Position 1.

Der CSRPOS-Operand hat nur Bedeutung, wenn der Wert im CURSOR-Operanden der Name eines Eingabefeldes ist.

Wenn CSRPOS weggelassen wird oder einen unzulässigen Wert hat (kleiner als 1 oder größer als die Feldlänge), wird der Wert 1 angenommen.

LOCK:

*YES: Bei Angabe dieses Operanden wird nach der Anzeige des Formates sofort zum Anwendungsprogramm zurückgekehrt. Diese Funktion kann z.B. zum Anzeigen eines Logos oder einer "Bitte warten"-Meldung verwendet werden.

ALARM:

*YES: Bei Angabe dieses Operanden ertönt bei der Anzeige des Formates ein akustische Signal.

HARDCOPY:

*YES: Bei Angabe dieses Wertes wird automatisch der Bildschirminhalt auf ein vorhandenes Hardcopygerät ausgegeben

AUTOTAB:

*YES: Automatisches Springen des Cursors von Feld zu Feld.

*NO: Der Cursor kann mit den Pfeiltasten auf geschützte Felder des Formats positioniert werden.

MANDATORY:

*YES: die Formatfelder mit der Eigenschaft "Eingabepflicht" haben bei einer Neuaustrage des selben Formats wieder die Eigenschaft "Eingabepflicht".

REFRESH:

*YES: die Bildschirmausgabe wird neu aufgebaut (keineDifferenzausgabe).

ACK:

Vor der Formatausgabe wird "ACK" im Zeilenmodus ausgegeben und eine Eingabe des Anwenders wird abgewartet.

5.2.5 Feldattribute (TU ATTR) ändern

Diese Variablen ermöglichen das dynamische Ändern einiger Feldattribute

Name		Typ	Standardwert	mögliche Werte
ATTR		list	<list of structure>	
	FIELD	string	*CURSOR	<field-name 1..255> / (field-name1,.....,field-name8)
	FIELD-IND		0	<integer> / (<integer1>,.....,<integer8>)
	TYPE	string	*UNCHANGED	*INPUT / *OUTPUT / *MANDATORY
	HILITE	string	*UNCHANGED	*UNDERLINE / *INVERSE / *INV-UND
	INTENSITY	string	*UNCHANGED	*HIGH / *LOW / *INVISIBLE / *HIGH-INV / *LOW-INV
	OUTPUT	string	*UNCHANGED	*INIT / *CHECK
	COLOR	string	*UNCHANGED	*RED / *GREEN / *YELLOW / *BLUE / *MA- GENTA / *CYAN / *WHITE

ATTR:

Liste der Felder, die während eines DISPLAY-Dienstes geändert werden sollen. Die Attribute sind in der folgenden Struktur enthalten:

FIELD:

Feldname des zu ändernden Feldes.

***CURSOR** Die Feldattribute des Feldes, in dem sich der Cursor beim nachfolgenden DISPLAY-Aufruf befinden wird, sollen geändert werden. Wird beim nachfolgenden DISPLAY-Aufruf festgestellt, daß für das Feld auf dem der Cursor positioniert worden ist, auch für den Feldnamen ein dynamisches Attribut angefordert ist, werden die Angaben kombiniert. Bei gleichen Angaben hat die *CURSOR-Angabe den Vorrang.

<field-name 1..255> / (field-name1,.....,field-name8)
Feldname des zu ändernden Feldes

FIELD-IND:

Indexangabe für das in FIELD angegebene Feld.

Wenn der Feldname sich auf ein Feld eines Listenbereiches bezieht, können Sie einen Index zur Festlegung einer bestimmten Listenzeile angeben.

TYPE:

Gibt den einzustellenden Feldtyp an:

***INPUT**

Das entsprechende Maskenfeld wird zum ungeschützten Eingabefeld ohne Eingabepflicht. Das Feld kann über die Tastatur verändert werden.

***MANDATORY**

Für das Maskenfeld besteht Eingabepflicht. Das Feld muß vom Terminal-Benutzer an der Datensichtstation modifiziert werden.

***OUTPUT**

Das Maskenfeld wird ein geschütztes, nicht markierbares Ausgabefeld. Der Feldinhalt kann über die Tastatur nicht verändert werden.

HILITE:

Beschreibt die Art der Hervorhebung des Maskenfeldes:

***UNDERLINED**

Der Feldinhalt wird unterstrichen dargestellt.

***INVERSE**

Der Feldinhalt wird invers dargestellt.

***INV-UND**

Der Feldinhalt wird invers und unterstrichen dargestellt.

Hinweis:

Wenn bereits bei der Formatdefinition eine Hervorhebung von Feldern vereinbart wurde, kann diese nicht durch dynamische Attribute aufgehoben werden. Felder, die dynamisch hervorgehoben werden sollen, dürfen deshalb nicht schon bei der Formatdefinition hervorgehoben werden.

INTENSITY:

Gibt die Helligkeitsstufe des Feldes an:

***HIGH**

Das Feld wird mit hoher Intensität dargestellt.

***LOW**

Das Feld wird mit geringer Intensität dargestellt.

***INVISIBLE**

Der Feldinhalt ist nicht sichtbar und nicht druckbar.

***HIGH-INV**

Der Hintergrund wird mit hoher Intensität dargestellt.

***LOW-INV**

Der Hintergrund wird mit geringer Intensität dargestellt.

OUTPUT:

Beschreibt die Verwendung der Voreinstellung von Maskenfeldern.

Ohne Angabe von OUTPUT werden Voreinstellungswerte der Maske immer dann verwendet, wenn beim DISPLAY-Aufruf die dem Maskenfeld zugeordnete Dialogvariable nicht existiert oder deren Wert die relevante Länge Null hat. Bei der Rückkehr vom DISPLAY-Dienst enthalten die entsprechenden Dialogvariablen den Voreinstellungswert des jeweiligen Maskenfeldes. Falls eine Dialogvariable nicht existiert wird sie mit dem Typ CHAR implizit angelegt.

Für Listenfelder werden Angaben zur Voreinstellung ignoriert.

***INIT**

Unabhängig vom Inhalt der mit dem Maskenfeld verbundenen Dialogvariablen, wird beim Anzeigen der Maske der beim Definieren des Formates festgelegte Voreinstellungswert für das spezifizierte Feld ausgegeben. Ist im Format kein Wert hierfür vorgegeben, wird das Feld mit Füllzeichen angezeigt.

***CHECK**

Für eine existierende Dialogvariable, deren Wert die relevante Länge größer Null hat, wird die Voreinstellung nur dann verwendet, wenn der Inhalt der mit dem Maskenfeld verbundenen Dialogvariablen den Wert Null hat.

Ohne Angabe des OUTPUT-Operanden, wird eine Voreinstellung eines Maskenfeldes immer verwendet, wenn die mit dem Maskenfeld verbundene Dialogvariable nicht existiert oder die relevante Länge Null hat.

COLOR:

Dient der Farbangabe für das Maskenfeld. Für Geräte, bei denen keine Farbdarstellung möglich ist, wird die Angabe ignoriert.

*RED	rot
*GREEN	grün
*YELLOW	gelb
*BLUE	blau
*MAGENTA	purpur
*CYAN	blaugrün
*WHITE	weiß

5.2.6 Eingabeformatierung

Diese Variablen beschreiben die Informationen, die FHS nach der Formatierung zurückgibt..

Name	Typ	Rückgabewert
COMMAND	string	<string 1..255>
FHS-VERSION	string	<string 1..30>
CURSOR-INPUT	string	<string 1..255>
CURSOR-INPUT-INDEX	integer	<integer>
CURSOR-INPUT-POS	integer	<integer>

COMMAND:

Blätter- oder Anwendungskommando, auf das die Anwendung reagieren muß.

FHS-VERSION:

Version des Dialog Managers.

CURSOR-INPUT:

Name des Feldes in dem der Cursor steht. Befindet sich der Cursor in einem unbekanntem Feld oder zwischen den Feldern, wird "\$ZZZ#SSS" zurückgegeben. Z steht für Linie und S für Spalte.

CURSOR-INPUT-INDEX:

Index des Cursors in einer Liste.

CURSOR-INPUT-POS:

Position des Cursors in dem in CURSOR-POSITION vorgegebenen Feld.

5.2.7 FHS-Rückgabe-Werte (return codes)

Diese Variablen werden in einem Standard-Header von SDF-P ausgegeben. FHS verwendet sie, um Ausgabe-Informationen zurückzugeben.

Name	Typ	Rückgabe-Wert
SUBCODE2	integer	<integer>
SUBCODE1	integer	<integer>
MAINCODE	string	<name 1..7>

5.3 Namenskonventionen für Listenverarbeitung

Konventionen für SDF-P-Variablen in der Modellzeile (Listenverarbeitung)

Name ohne Indexangabe

Beispiel: ABC

 ABC: ist die aktuelle Liste

SDF-P Variablendefinition:

```
/ declare-variable name = ABC (type=*string),-
/ multiple-elements = *list(limit=100)
```

Strukturierter Name ohne Indexangabe

Beispiel: A.B-C

 B-C: ist die aktuelle Liste

SDF-P Variablendefinition:

```
/ declare-variable name = A (type = *structure(*by-syscmd))
/ begin-structure
/ declare-element name = B-C (type=*string),-
/ multiple-elements = *list(limit=100)
/ end-structure
```

Strukturierter Name mit konstanter Indexangabe

Beispiel: A#3.B-C

 A: Liste oder ARRAY von Strukturen, die selbst eine Liste enthalten
(Index=3)

 B-C: aktuelle Liste zum Anzeigen

SDF-P Variablendefinition:

```
/ declare-variable name = A (type = *structure(*by-syscmd)),-
/ multiple-elements = *array(upper-bound = 10)
/ begin-structure
/ declare-element name = B-C (type=*string),-
/ multiple-elements = *list(limit=100)
/ end-structure
```

Strukturierter Name mit variabler Indexangabe

Beispiel: A#*.B-C
 A: ist aktuelle Liste
 B-C: ist skalar

SDF-P Variablendefinition:

```
/ declare-variable name = A (type = *structure(*by-syscmd)),-  
/ multiple-elements = *list (limit = *NONE)  
/ begin-structure  
/ declare-element name = B-C (type=*string)  
/ end-structure
```

Strukturierter Name mit variabler und konstanter Indexangabe

Beispiel: A#*.B-C#3
 A: aktuelle Liste, Liste von Strukturen, von denen ein Element angezeigt
 werden soll
 B-C: Liste (konstanter Index)

SDF-P Variablendefinition:

```
/ declare-variable name = A (type = *structure(*by-syscmd)),-  
/ multiple-elements = *list (limit = *NONE)  
/ begin-structure  
/ declare-element name = B-C (type=*string),-  
/ multiple-elements = *list (limit=100)  
/ end-structure
```

5.4 FHS-Anwendungen durch S-Prozeduren steuern

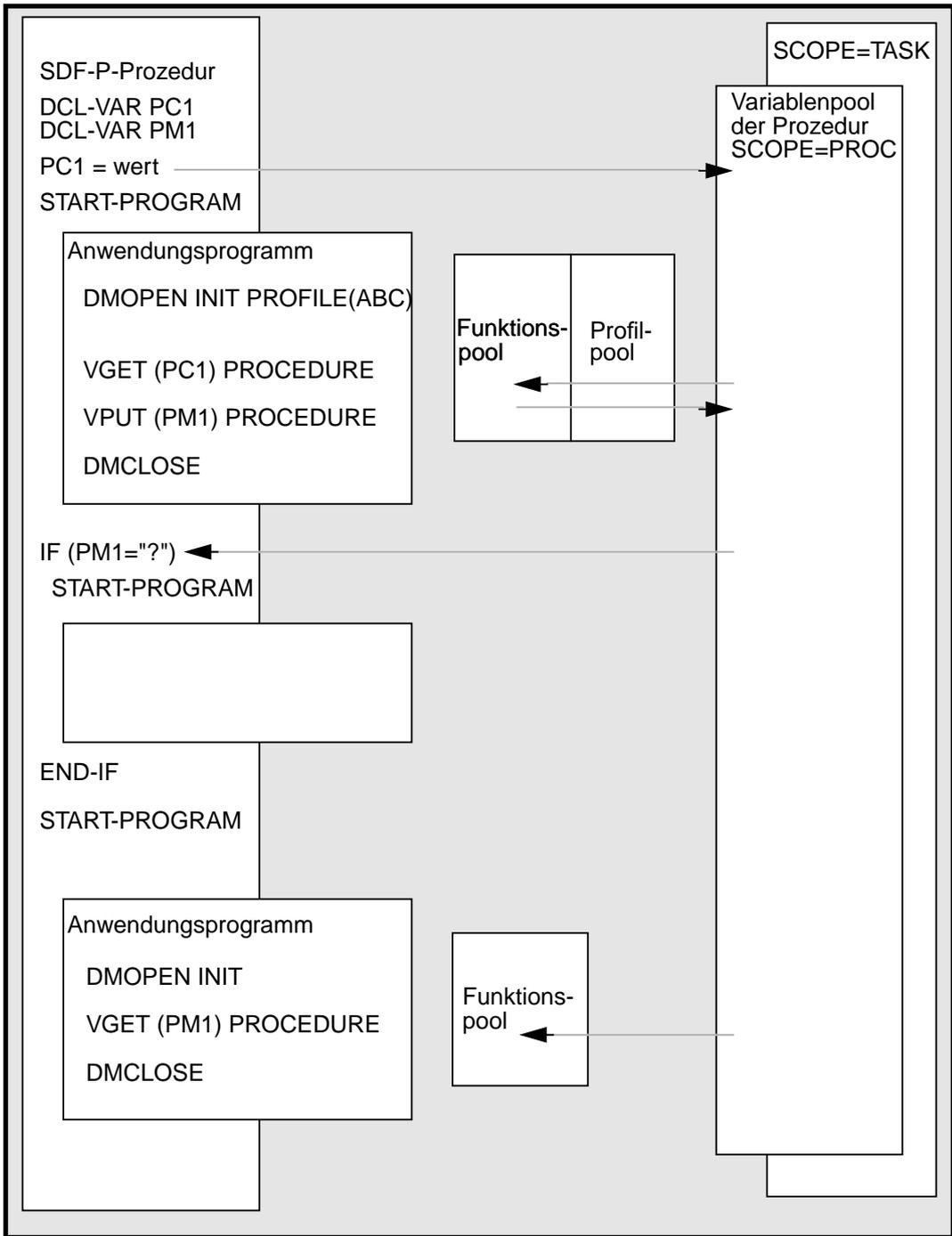
5.4.1 S-Variablen mit FHS 8.1 ausgeben

S-Prozeduren können die Kommandos ASSIGN-STREAM und TRANSMIT-BY-STREAM benutzen, um einen Dialog mit dem Terminal-Benutzer mittels FHS-Formatfeldern zu starten, die mit FHS-Dialog-Variablen gefüllt sind.

5.4.2 S-Variablen in FHS-TIAM-Programmen ausgeben und erzeugen

Die Variablendienste VGET und VPUT in FHS V8.1 erlauben einem Anwendungsprogramm einfache S-Variable zu lesen und zu schreiben. Damit ist eine Kommunikation zwischen einer S-Prozedur und einem von dieser Prozedur aufgerufenen Anwendungsprogramm möglich.

Das folgende Bild zeigt den Variablen austausch zwischen Anwendungsprogramm und einer S-Prozedur.



5.4.3 FHS-Anwendungen in geschachtelten S-Prozeduren steuern

In geschachtelten S-Prozeduren werden Zuweisungen von S-Variablenströmen genauso wie bei Systemdateien (SYSDTA,SYSOUT,..) gestapelt. Der Anwender sollte darum im Kommando SET-PROCEDURE-OPTIONS die Operandeneinstellung SYSTEM-FILE-CONTEXT=STD oder OWN benützen, falls er seine string-Anweisungen auch gestapelt bearbeiten will.

Im TPR-Modus sichert FHS seine Anzeige-Umgebung in Übereinstimmung mit der Zuweisung der S-Variablenströme im Kommando TRANSMIT-BY-STREAM. Zur Zuweisungszeit initialisiert FHS einen dem Variablenstrom spezifischen Kontext, der automatisch bei jeder FHS-Operation mit demselben Variablenstrom benützt wird. Dies passiert bis zum Ende der Zuweisung, z.B. wenn demselben Variablenstrom *DUMMY oder ein anderer Server zugewiesen wird - entweder explizit oder implizit bei Prozedurende (gemäß der SYSTEM-FILE-CONTEXT-Zuweisung).

Dieser Mechanismus schließt ein, daß geschachtelte Prozeduren unabhängig codiert sein können ohne Überlagerungen von Anzeigebefehlen in verschiedenen FHS-Kontexten.

Um die Unabhängigkeit von der aufrufenden Prozedur zu sichern, ist es notwendig, der Variablen SYSFHS-CONTROL.REFRESH den Wert *YES* zuzuweisen, um ein Bedienfeld nach dem Aufruf dieser Prozedur auszugeben.

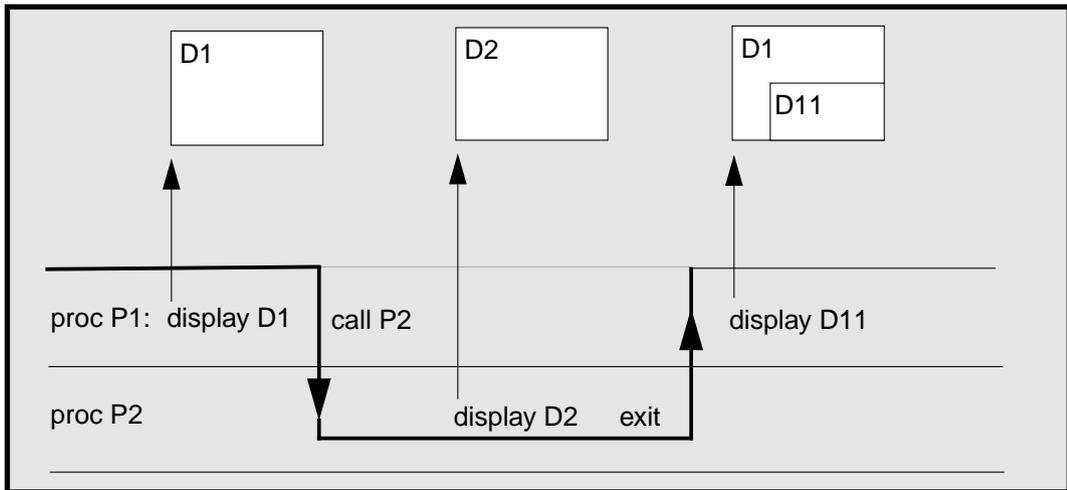
Beispiel

In der Prozedur P1 wird der Variablenstrom S1 FHS zugewiesen; das Bedienfeld D1 wird in der Prozedur P1 angezeigt. P1 ruft die Prozedur P2 auf ; der Variablenstrom S2 in P2 ist ebenfalls FHS zugewiesen; P2 zeigt das Bedienfeld D2 an. D2 überschreibt D1 vollkommen.

```
Proc P1 : assign  S1 -----> FHS
           displays D1
           calls   P2
           pop-up  D11

Proc P2 : assign  S2 -----> FHS
           displays D2
           exit
```

Nach Beendigung von P2 geht FHS in die Anzeige-Umgebung von S1 zurück. Dafür wird das "Pop up-Menü" auf dem gegenwärtigen Bildschirm (mit oder ohne Abzweigung, z.B. Bedienfeld D11) wieder implizit auf dem gegenwärtigen Bedienfeld von P1 (D1) wiederhergestellt.



Das Bedienfeld D2 ist gelöscht worden und das Bedienfeld D1 ist bei der nächsten Anzeige der Prozedur P1 wiederhergestellt worden.

Hinweis

Variablen, die in einem Bedienfeld angezeigt werden sollen, müssen in der gegenwärtigen S-Prozedur sichtbar sein (siehe Kapitel "Variablen in S-Prozeduren verwenden", Abschnitt "Geltungsbereich von Variablen").

5.5 SYSFHS-CONTROL - Struktur für Layout und Initialisierung

Die folgende S-Prozedur wird mit FHS ausgeliefert. Sie definiert das Layout der Kontroll-Variablen und initialisiert sie.

```

/set-procedure-options caller=include
/begin-parameter-declaration
/ declare-parameter -
/ "----- std param -----*"-
/      (PREFIX      (type=string,init='SYSFHS-') -
/      ,INCLUDE-FORM (type=string,init='LAYOUT') "/initialize" -
/      ,VARIABLE-NAME(type=string,init='') -
/ "----- include specific param -----*"-
/ " action variables " -
/      ,SERVICE      (type=string,init='*DISPLAY') -
/      ,DIAGINFO      (type=string,init='*NO') -
/      ,POP-LOCATION    (type=string,init='*NONE') -
/      ,POP-LOC-IND   (type=integer,init=0) -
/      ,ROW            (type=integer,init=2) -
/      ,COLUMN        (type=integer,init=2) -
/ " resource variables " -
/      ,RESOURCE      (type=string,init='*SAME') -
/      ,MESSAGE-ID    (type=string,init='*NONE') -
/      ,MESSAGE-FIELD (type=string,init='*NONE') -
/      ,MSG-FIELD-IND (type=integer,init=0) -
/ " panel variables " -
/      ,CURSOR-OUTPUT-INDEX (type=integer,init=0) -
/      ,CURSOR-OUTPUT      (type=string,init='*NONE') -
/      ,CURSOR-OUTPUT-POS  (type=integer,init=0) -
/      ,LOCK                (type=string,init='*NO') -
/      ,ALARM               (type=string,init='*NO') -
/      ,HARDCOPY            (type=string,init='*NO') -
/      ,AUTOTAB             (type=string,init='*YES') -
/      ,MANDATORY          (type=string,init='*NO') -
/      ,REFRESH            (type=string,init='*NO') -
/      ,ACK                 (type=string,init='*NO') -
/ " field attributes " -
/      ,ATTR-LIST      (type=integer,init=0) -
/      " number of list elements to reset "-
/      ,FIELD          (type=string,init='*CURSOR') -
/      ,FIELD-IND      (type=string,init='0') -
/      ,TYPE           (type=string,init='*UNCHANGED') -
/      ,HILITE        (type=string,init='*UNCHANGED') -
/      ,INTENSITY     (type=string,init='*UNCHANGED') -
/      ,COLOR         (type=string,init='*UNCHANGED') -
/      ,OUTPUT        (type=string,init='*UNCHANGED') -

```

```

/   " input information " -
/       ,COMMAND          (type=string,init='') -
/       ,FHS-VERSION      (type=string,init='') -
/       ,CURSOR-INPUT     (type=string,init='') -
/       ,CURSOR-INPUT-INDEX (type=integer,init=0) -
/       ,CURSOR-INPUT-POS (type=integer,init=0) -
/       )
/end-parameter-declaration
/
/ if (upper-case(INCLUDE-FORM) == 'LAYOUT')
/
/ begin-structure ATTR,scope=proc
/   declare-element -
/       (FIELD          (type=string) -
/       ,FIELD-IND      (type=string) -
/       ,TYPE           (type=string) -
/       ,HILITE        (type=string) -
/       ,INTENSITY     (type=string) -
/       ,COLOR         (type=string) -
/       ,OUTPUT        (type=string) -
/       )
/ end-structure // begin-structure name=&PREFIX.LAYOUT,scope=proc
/   declare-element STD-HEADER(type=structure(&PREFIX.FHDR))
/   declare-element -
/   " action variables " -
/       (SERVICE      (type=string) -
/       ,DIAGINFO      (type=string) -
/       ,POP-LOCATION    (type=string) -
/       ,POP-LOC-IND   (type=integer) -
/       ,ROW           (type=integer) -
/       ,COLUMN        (type=integer) -
/   " resource variables " -
/       ,RESOURCE      (type=string) -
/       ,MESSAGE-ID    (type=string) -
/       ,MESSAGE-FIELD (type=string) -
/       ,MSG-FIELD-IND (type=integer) -
/   " panel variables " -
/       ,CURSOR-OUTPUT-INDEX (type=integer) -
/       ,CURSOR-OUTPUT      (type=string) -
/       ,CURSOR-OUTPUT-POS  (type=integer) -
/       ,LOCK                (type=string) -
/       ,ALARM               (type=string) -
/       ,HARDCOPY            (type=string) -
/       ,AUTOTAB             (type=string) -
/       ,MANDATORY          (type=string) -
/       ,REFRESH            (type=string) -
/       ,ACK                (type=string) -
/       )

```

```

/ " field attributes "
/ declare-element ATTR (type=struct(attr))-
/      ,mult-elem=list
/
/ " input information " -
/ declare-element -
/      (COMMAND (type=string) -
/      ,FHS-VERSION (type=string) -
/      ,CURSOR-INPUT (type=string) -
/      ,CURSOR-INPUT-INDEX (type=integer) -
/      ,CURSOR-INPUT-POS (type=integer) -
/      )
/end-structure
/
/else-if (upper-case(INCLUDE-FORM) == 'INITIALIZE')
/
/   if (VARIABLE-NAME == '')
/     write-text '% mandatory parameter variable-name missing.'
/     raise-error
/   end-if
/ declare-variable PARAM(type=string)
/ include-procedure *lib-elem(lib=$.SYSPRC.SDF-P-BASYS.020,e1=FHDR) -
/ "----- std param -----*"-
/      ,(INCLUDE-FORM='INITIALIZE' -
/      ,VARIABLE-NAME='&VARIABLE-NAME..STD-HEADER' -
/      "----- include specific param -----*"-
/      ,UNIT ='FHS', "fhs unit name" -
/      ,FUNCTION ='DISPLAY', "fhs fc for display?" -
/      ,VERSION = 1 "control variable layout version"-
/      )
/
/   for PARAM = -
/      ('SERVICE' -
/      , 'DIAGINFO' -
/      , 'POP-LOCATION' -
/      , 'POP-LOC-IND' -
/      , 'ROW' -
/      , 'COLUMN' -
/      , 'RESOURCE' -
/      , 'MESSAGE-ID' -
/      , 'MESSAGE-FIELD' -
/      , 'MSG-FIELD-IND' -
/      , 'CURSOR-OUTPUT-INDEX' -
/      , 'CURSOR-OUTPUT' -
/      , 'CURSOR-OUTPUT-POS' -
/      , 'LOCK' -
/      , 'ALARM' -
/      , 'HARDCOPY' -

```

```

/          , 'AUTOTAB'          -
/          , 'MANDATORY'       -
/          , 'REFRESH'         -
/          , 'ACK'             -
/          , 'COMMAND'         -
/          , 'FHS-VERSION'     -
/          , 'CURSOR-INPUT'    -
/          , 'CURSOR-INPUT-INDEX' -
/          , 'CURSOR-INPUT-POS' -
/          )
/          &VARIABLE-NAME..&PARAM = &PARAM
/      end-for
/
/      if (ATTR-LIST > 0)
/          I = 1
/          while ( I <= ATTR-LIST )
/              for PARAM = -
/                  ( 'FIELD'      -
/                    , 'FIELD-IND' -
/                    , 'TYPE'     -
/                    , 'HILITE'   -
/                    , 'INTENSITY' -
/                    , 'COLOR'    -
/                    , 'OUTPUT'   -
/                  )
/                  &VARIABLE-NAME..ATTR#I.&PARAM = &PARAM
/              end-for
/              I=I+1
/          end-while
/      end-if
/
/  else
/      write-text '% form=&INCLUDE-FORM not supported; include aborts'
/      raise-error
/  end-if
/EXIT-PROCEDURE

```

Hinweis

In einem Include wird in dieser Prozedur der Standard Header eingebunden. Das ist eine S-Prozedur, die mit SDF-P mitgeliefert wird und für Funktionsidentifikationen und Returncode-Informationen zuständig ist. Die Beschreibung finden Sie auf den folgenden Seiten.

Include für den Standard-Header

Dieser Standard-Header wird mit SDF-P ausgeliefert. Er wird in der Prozedur SYSFHS aufgerufen (mit /INCLUDE-PROCEDURE).

Die Prozedur steht in der Bibliothek SYSPRC.SDF-P-BASYS.020 im Element FHDR.

Dieser Standard-Header kann als erstes Element der Struktur der Kontroll-Variablen definiert werden.

```

/SET-PROCEDURE-OPTIONS &* CALLER=INCLUDE
/"WIRD NICHT VON SDF-P-BASYS UNTERSTUETZT"
/BEGIN-PARAMETER-DECLARATION
/  /DECLARE-PARAM -
/  "----- STD PARAM -----*"
/      (PREFIX              (INITIAL-VALUE="SYSSDP") -
/      ,INCLUDE-FORM      (INITIAL-VALUE="LAYOUT") "INITIALIZE" -
/      ,VARIABLE-NAME (INITIAL-VALUE="") -
/  "----- INCLUDE SPSECIFIC PARAM -----*"
/      ,UNIT              (INITIAL-VALUE="") -
/      ,FUNCTION          (INITIAL-VALUE="") -
/      ,VERSION           (INITIAL-VALUE=0) -
/      ,SUBCODE2         (INITIAL-VALUE=0) -
/      ,SUBCODE1         (INITIAL-VALUE=0) -
/      ,MAINCODE         (INITIAL-VALUE="CMD0001") -
/      )
/END-PARAMETER-DECLARATION
/
/IF (NOT IS-SDF-P( ))
/  EXIT-PROCEDURE ERROR=*YES(SUBCODE1=41,MAINCODE=CMD2241)
/END-IF
/
/IF (UPPPER-CASE(INCLUDE-FORM) == "LAYOUT")
/
/BEGIN-STRUCTURE NAME=&PREFIX.IFID-MDL,SCOPE=*PROCEDURE
/  DECLARE-ELEMENT -
/      (UNIT              (TYPE=*STRING) -
/      ,FUNCTION          (TYPE=*STRING) -
/      ,VERSION           (TYPE=*INTEGER) -
/      )
/END-STRUCTURE
/BEGIN-STRUCTURE NAME=&PREFIX.RETC-MDL,SCOPE=*PROCEDURE
/  DECLARE-ELEMENT -
/      ,SUBCODE2         (TYPE=*INTEGER) -
/      ,SUBCODE1         (TYPE=*INTEGER) -
/      ,MAINCODE         (TYPE=*STRING) -
/  )
/END-STRUCTURE
/BEGIN-STRUCTURE NAME=&PREFIX.FHDR,SCOPE=*PROCEDURE

```

```

/   DECLARE-ELEMENT INTERFACE-ID(TYPE=*STRUCTURE(&PREFIX.IFID-MDL))
/   DECLARE-ELEMENT RETURNCODE(TYPE=*STRUCTURE(&PREFIX.RETC-MDL))
/END-STRUCTURE
/
/ELSE-IF (UPPER-CASE(INCLUDE-FORM) == "INITIALIZE")
/
/   IF (VARIABLE-NAME == "")
/       WRITE-TEXT "% OBLIGATORISCHER OPERAND VARIABLE-NAME GEHT AB."
/       RAISE-ERROR
/   END-IF
/   DECLARE-VARIABLE PARAM(TYPE=*STRING)
/   FOR PARAM = ("UNIT", "FUNCTION")
/       &VARIABLE-NAME..INTERFACE-ID.&PARAM = &PARAM
/   END-FOR
/   &VARIABLE-NAME..INTERFACE-ID.VERSION=INTEGER(VERSION)
/   FOR PARAM = ("SUBCODE2", "SUBCODE1")
/       &VARIABLE-NAME..RETURNCODE.&PARAM = INTEGER(&PARAM)
/   END-FOR
/   &VARIABLE-NAME..RETURNCODE.MAINCODE=MAINCODE
/ELSE
/       WRITE-TEXT "% FORM = &INCLUDE-FORM WIRD NICHT UNTERSTUETZT; -
/INCLUDE WIRD ABGEBROCHEN"
/       RAISE-ERROR
/END-IF
/EXIT-PROCEDURE

```

Der Server prüft nicht, ob dieser Header existiert. Dennoch kann er bei der Angabe der Kontroll-Variablen nicht weggelassen werden.

Beschreibung der Parameter

UNIT (type=string)

beschreibt den Server der die Kontroll-Variable definiert hat. Das sollte der beim /ASSIGN-STREAM angegebene Name sein.

FUNCTION (type=string)

beschreibt die Funktion für die der Server das Layout der Kontroll-Variablen definiert hat. Der Name wird vom Server definiert und verarbeitet.

VERSION (type=integer)

Version der Kontroll-Variablen. Dadurch können frühere Versionen der Kontroll-Variablen vom Server kompatibel unterstützt werden.

SUBCODE2 (type=integer)

vom Server zurückgegebener Wert "Subcode2" (in RET-CONTROL-VAR-NAME).

SUBCODE1 (type=integer)

vom Server zurückgegebener Wert "Subcode1" (in RET-CONTROL-VAR-NAME).

MAINCODE (type=string)

vom Server zurückgegebenes Meldungs-Kennzeichen (in RET-CONTROL-VAR-NAME).

Für die Parameter SUBCODE2, SUBCODE1, MAINCODE gelten die gleichen Konventionen wie für TU-FHS, z.B. IDHA001 (siehe Anhang ab Seite 231).

Diese Kontroll-Variablen sollten vom Server zurückgeliefert werden, um eine genauere Fehlerbehandlung zu ermöglichen, als dies mit den Rückgabewerten des transmit-by-stream oder TRANSVV-Kommandos möglich ist.

5.6 Unterschiede zwischen FHS-DM und FHS-PRIV

P-Tasten-Belegung

Im Gegensatz zu FHS-TU kann der Wert von P-KEY-Setting nicht in einem Benutzerprofil gespeichert werden.

Die Systemvariablen KEY-LIST und KEY-AEREA existieren nicht.

CCSNAME und 8-bit-Modus

Der CCSNAME kann nicht angegeben werden; die Systemvariable CCSNAME existiert nicht.

Wenn ein Benutzer im 8-bit-Modus arbeiten will, muß er vor der Zuordnung das Kommando "MODIFY-TERMINAL-OPTION" eingeben.

SYS-Kommando

Im TPR-Modus wird das SYS-Kommando abgelehnt.

6 Beispiel-Programme

Nachfolgend finden Sie ein COBOL-Beispiel, das FHS-DM als Benutzer-Schnittstelle verwendet.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. COB7.
*****
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    TERMINAL IS VIDEO.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 DMCOMM.
    02 DMRC IS COMP.
        03 DMSC2 PIC S9(4).
        03 DMSC1 PIC S9(4).
        03 DMMC PIC S9(7).
    02 DMMSGID PIC X(8).
    02 DMFLAG.
        03 DMERR PIC X(1).
        03 FILLER PIC X(7).
    02 DMSYS PIC X(104).
77 BUFFER PIC X(512).
77 LEN PIC S9(5) COMP VALUE 512.
77 VAR1 PIC X(6).
77 VARILEN PIC S9(5) COMP.
*
***** PROGRAM START
*
PROCEDURE DIVISION.
*****
CONTR SECTION.
CONTR-0.
PERFORM MAIN.
CONTR-9.
DISPLAY "PROGRAM COB7 TERMINATED" UPON VIDEO.
STOP RUN.
*****

```

```
MAIN SECTION.
MAIN-0.
*
    MOVE " DMOOPEN INIT PROFILE(DATA1)" TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
    MOVE " VDEFINE VAR1 FORMAT(CHAR)" TO BUFFER.
    MOVE 6 TO VARILEN.
    CALL "ISPCI2" USING DMCOMM LEN BUFFER VARILEN VAR1.
    PERFORM CHECK-FHS-RC.
*
    MOVE " VGET VAR1 PROFILE      " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
    .
*
    .
*
    .
*
    vdefine of other variables
*
    .
*
    .
*
    .
*
* siehe (1) in Erläuterungen
    MOVE " DISPLAY PANEL(V81) " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
    MOVE " ADDPOP " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
* siehe (2) in Erläuterungen
    MOVE " DISPLAY PANEL(BOXV81) " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
* siehe (3) in Erläuterungen
    MOVE " DISPLAY MSG(IDHT017  ) " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
    MOVE " VPUT VAR1 PROFILE      " TO BUFFER.
    CALL "ISPCI" USING DMCOMM LEN BUFFER.
    PERFORM CHECK-FHS-RC.
*
```

```
MOVE " DMCLOSE " TO BUFFER.  
CALL "ISPCI" USING DMCOMM LEN BUFFER.  
PERFORM CHECK-FHS-RC.  
EXIT.
```

*

```
CHECK-FHS-RC SECTION.
```

```
CHECK-0.
```

```
    IF DMSC1 IN DMRC IN DMCOMM NOT = 0  
      THEN DISPLAY "RETURNCODE:" DMSC2 "/"  
                  DMSC1 "/" DMMC "      "  
                  DMMSGID UPON VIDEO.
```

```
EXIT.
```

Ein entsprechendes Beispiel-Programm in ASSEMBLER

```

MACRO
FILLBUF &STMT
MVI  BUFFER,' '
MVC  BUFFER+1(L'BUFFER-1),BUFFER
MVC  BUFFER(L'&STMT),&STMT
MEND

*
ASM7  CSECT
ASM7  AMODE ANY
ASM7  RMODE ANY
      USING *,10
      BALR 10,0
      BCTR 10,0
      BCTR 10,0
R1    EQU 1
R2    EQU 2
R7    EQU 7
R14   EQU 14
R15   EQU 15
*
      LA   R1,PL
*
      L    R15,ISPCI@
      FILLBUF STMT1
      BASR R14,R15
      BAL  R7,CHECKRC
*
      L    R15,ISPCI2@
      LA   R2,VAR1
      ST  R2,OPER2@
      LA   R2,VAR1L
      ST  R2,OPER2L@
      FILLBUF STMT2
      BASR R14,R15
      BAL  R7,CHECKRC
*
      L    R15,ISPCI@
      FILLBUF STMT3
      BASR R14,R15
      BAL  R7,CHECKRC
*
      .
*
      .
*
      .
*
      vdefine of other variables

```

```

*      .
*      .
*      .
*
*   siehe (1) in Erläuterungen
*       FILLBUF STMT4
*       BASR  R14,R15
*       BAL   R7,CHECKRC
*
*       FILLBUF STMT5
*       BASR  R14,R15
*       BAL   R7,CHECKRC
*   siehe (2) in Erläuterungen
*       FILLBUF STMT6
*       BASR  R14,R15
*       BAL   R7,CHECKRC
*   siehe (3) in Erläuterungen
*       FILLBUF STMT7
*       BASR  R14,R15
*       BAL   R7,CHECKRC
*
*       FILLBUF STMT8
*       BASR  R14,R15
*       BAL   R7,CHECKRC
*
*       FILLBUF STMT9
*       BASR  R14,R15
*       BAL   R7,CHECKRC*
*       TERM
*-----
CHECKRC DS    0H
        CLC   DMSC1,=X'0000'
        BE    OK
        TERMD
OK      BR    R7
*-----
*
STMT1  DC    C'DMOPEN INIT PROFILE (DATA1)
STMT2  DC    C'VDEFINE VAR1 FORMAT(CHAR)
STMT3  DC    C'VGET VAR1 PROFILE
*      .
*      .
*      .
*      other "vdefine" statements
*      .
*      .
*      .
STMT4  DC    C'DISPLAY PANEL(V81)

```

```
STMT5    DC    C'ADDDPOP
STMT6    DC    C'DISPLAY PANEL(BOXV81)
STMT7    DC    C'DISPLAY MSG(IDHT017)
STMT8    DC    C'VPUT VAR1 PROFILE
STMT9    DC    C'DMCLOSE
*
PL        DS    5A
          ORG   PL
DMCOMM@  DC    A(DMCOMM)
BUFLLEN@ DC    A(BUFLLEN)
BUFFER@  DC    A(BUFFER)
OPER2L@  DC    A(0)
OPER2@   DC    A(0)
          ORG
*
DMCOMM   DS    0F
DMSC2    DS    H
DMSC1    DS    H
DMMC     DS    F
DMMSGID  DS    CL8
DMERR    DS    C
RESERVD1 DS    XL7
DMSYS    DS    XL104
*
VAR1     DS    CL6
VAR1L    DC    A(L'VAR1)
BUFLLEN  DC    A(L'BUFFER)
BUFFER   DC    CL256''
*
ISPCI@   DC    V(ISPCI)
ISPCI2@  DC    V(ISPCI2)
*
          END
```

6.1 Aufruf von FHS-DM als Subsystem

Wenn FHS-DM als Subsystem geladen ist, kann es von jedem Haupt-Programm aufgerufen werden, das die nicht aufgelösten ISPCI und ISPCI2-Aufrufe aufruft.

Sie sollten dafür sorgen, daß diese Extern-Verweise während der Laufzeit vom DBL (Dynamic Binder Loader) dynamisch aufgelöst werden!

Wurde das FHS-DM V8.1 Subsystem an eine Adresse innerhalb oder oberhalb des 16 MB Adressraums, dann muß die Anwendung des Benutzers mit AMODE=ANY erstellt und mit /START-PROGRAM xxx,PROGRAM-MODE=ANY gestartet werden.

Wurde das FHS-DM V8.1 Subsystem exklusiv unterhalb des 16 MB Adressraums geladen, müssen die PROGRAM-MODE-Parameter im /START-PROGRAM Kommando nicht angegeben werden.

Beispiele für Link-Prozeduren

A. Hauptprogramm in COBOL

```

                /ASSIGN-SYSLST COB7.LNKLST
                /ASSIGN-SYSDTA *SYSCMD
                /START-BINDER
                //START-LLM-CREATION INTER-NAME=TESTCOB
                //BEGIN-SUB-LLM APPLIC
(1)             //INCLUDE-MODULE LIB=COB.LIB,E=COB7,TYPE=R
                //END-SUB-LLM
                //BEGIN-SUB-LLM RT
(2)             //RESOLVE-BY-AUTOLINK $.SYSLNK.CRTE,TYPE=R,SCOPE=*WHOLE-LLM
                //END-SUB-LLM
                //SAVE-LLM LIBRARY=COB.LIB,ELEM=TESTCOB,MAP=YES
                //END
                /ASSIGN-SYSLST *PRIMARY
                /ASSIGN-SYSDTA *PRIMARY
```

B. Hauptprogramm in ASSEMBLER

```
        /ASSIGN-SYSLST ASM7.LNKLST
        /ASSIGN-SYSDTA *SYSCMD
        /START-BINDER
        //START-LLM-CREATION INTER-NAME=TESTASM
        //BEGIN-SUB-LLM APPLIC
(1)     //INCLUDE-MODULE LIB=ASM.MODLIB,E=ASM7,TYPE=R
        //END-SUB-LLM
        //SAVE-LLM LIBRARY=ASM.MODLIB,ELEM=TESTASM,MAP=YES
        //END
        /ASSIGN-SYSLST *PRIMARY
        /ASSIGN-SYSDTA *PRIMARY
```

(1) Ihr Anwendungsprogramm

(2) Das CRTE Laufzeitsystem für COBOL

Dieses Programm ergibt zwei ungelöste Extern-Verweise bei ISPCI and ISPCI2. Diese werden während der Laufzeit mit dem FHS-DM Subsystem aufgelöst.

Sie können diese Programme mit folgendem Kommando starten:

```
/START-PROG *M(library,element,RUN-MODE=ADV)
```

Erläuterungen (1)

Management	View	Run			
FILE MANAGER					
FILE(S)	SELECTION		From: 1	Total: 13	
			To : 13	More :	
?	Size	Cat.	UserId	File name	
	381	P	SWN24FHS	SM.SS.FHS.V08.0A70.GCLIB.P	
	381	P	SWN24FHS	SM.SS.FHS.V08.0A70.GCLIB.UR	
	732	P	SWN24FHS	SM.SS.FHS.V08.0A70.SRC	
	573	P	SWN24FHS	SM.SS.FHS.V08.1A50.GCLIB.P	
	228	P	SWN24FHS	SM.SS.FHS.V08.1A50.GCLIB.UR	
	39579	P	SWN24FHS	SM.SS.FHS.V08.1A70.LIST	
	2973	P	SWN24FHS	SM.SS.FHS.V08.1A70.SRC	
	585	P	SWN24FHS	SM.SS.FHS.V08.1A80.GCLIB.P	
	228	P	SWN24FHS	SM.SS.FHS.V08.1A80.GCLIB.UR	
	987	P	SWN24FHS	SM.SS.FHS.V08.1A80.LIB	
	27	P	SWN24FHS	SM.SS.FHS.V08.1A80.LIST	
	12	P	SWN24FHS	SM.SS.FHS.V08.1A80.SRC	
	303	P	SWN24FHS	SM.SS.PROSOS-TU.V07.0A00.LIB	
COMMAND ==>					
F1=HELP F3=EXIT F7=BACKWARD F8=FORWARD F10=MENU F12=CANCEL					

Erläuterungen (2)

Management	View	Run
FILE M A N A G E R		
FILE(S)	SELECTION	From: 1 Total: 13 To : 13 More :
? Size	Cat UserId	File name
381		
381		
732		
573		
228		
39579		
2973		
585		
228		
987	P	SWN24FHS SM.SS.FHS.V08.1A80.LIB
27	P	SWN24FHS SM.SS.FHS.V08.1A80.LIST
12	P	SWN24FHS SM.SS.FHS.V08.1A80.SRC
303	P	SWN24FHS SM.SS.PROSOS-TU.V07.0A00.LIB
COMMAND ==>		
F1=HELP F3=EXIT F7=BACKWARD F8=FORWARD F10=MENU F12=CANCEL		

CONFIRMATION

Name: :P:\$SWN24FHS.SM.SS.FHS.V08.1A50.GCLIB.P

is going to be deleted. Continue: _ (Yes/No)

Erläuterungen (3)

Management	View	Run
FILE MANAGER		
FILE(S)	SELECTION	From: 1 Total: 13 To: 13 More :
? Size	Cat Userld	File name
381		
381		
732		
573		
228		
39579		
2973		
585		
228	P SW	
987	P SW	
27	P SW	
12	P SW	
303	P SW	
COMMAND ==>		F12=Remove
F1=HELP F3=EXIT F7=BACKWARD F8=FORWARD F10=MENU F12=CANCEL		

CONFIRMATION

Name: :P:\$SWN24FHS.SM.SS.FHS.V08.1A50.GCLIB.P

is going to be deleted. Continue: _ (Yes/No)

IDHT017

The specified character string contains at least one unallowed character. Only the following characters are allowed in this field :

" NY"

6.2 Beispielprozeduren für FHS in SDF-P

Das folgende Beispiel zeigt die Deklaration und die Initialisierung der Variablen MYVAR (ohne ATTR-LIST Element):

```
/include-procedure *lib-elem(lib=$.SYSPRC.SDF-P-BASYS.020,e1=FHDR) -
/
, (PREFIX='SYSFHS-')
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL)
/declare-variable MYVAR (type = *struct(SYSFHS-LAYOUT))
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL) , -
/
(include-form='INITIALIZE',variable-name='MYVAR', -
/
attr-list = 2 )
/
/show-variable MYVAR
```

Die Variable wird erzeugt und wie folgt initialisiert:

```
MYVAR.STD-HEADER.INTERFACE-ID.UNIT = FHS
MYVAR.STD-HEADER.INTERFACE-ID.FUNCTION = DISPLAY
MYVAR.STD-HEADER.INTERFACE-ID.VERSION = 1
MYVAR.STD-HEADER.RETURNCODE.SUBCODE2 = 0
MYVAR.STD-HEADER.RETURNCODE.SUBCODE1 = 0
MYVAR.STD-HEADER.RETURNCODE.MAINCODE = CMD0001
MYVAR.SERVICE = *DISPLAY
MYVAR.DIAGINFO = *NO
MYVAR.POP-LOCATION = *NONE
MYVAR.POP-LOC-IND = 0
MYVAR.ROW = 2
MYVAR.COLUMN = 2
MYVAR.RESOURCE = *SAME
MYVAR.MESSAGE-ID = *NONE
MYVAR.MESSAGE-FIELD = *NONE
MYVAR.MSG-FIELD-IND = 0
MYVAR.CURSOR-OUTPUT-INDEX = 0
MYVAR.CURSOR-OUTPUT = *NONE
MYVAR.CURSOR-OUTPUT-POS = 0
MYVAR.LOCK = *NO
MYVAR.ALARM = *NO
MYVAR.HARDCOPY = *NO
MYVAR.AUTOTAB = *YES
MYVAR.MANDATORY = *NO
MYVAR.REFRESH = *NO
MYVAR.ACK = *NO
MYVAR.ATTR(*LIST).FIELD = *CURSOR
MYVAR.ATTR(*LIST).FIELD-IND = 0
MYVAR.ATTR(*LIST).TYPE = *UNCHANGED
MYVAR.ATTR(*LIST).HILITE = *UNCHANGED
MYVAR.ATTR(*LIST).INTENSITY = *UNCHANGED
```

```
MYVAR.ATTR(*LIST).COLOR = *UNCHANGED
MYVAR.ATTR(*LIST).OUTPUT = *UNCHANGED
MYVAR.ATTR(*LIST).FIELD = *CURSOR
MYVAR.ATTR(*LIST).FIELD-IND = 0
MYVAR.ATTR(*LIST).TYPE = *UNCHANGED
MYVAR.ATTR(*LIST).HILITE = *UNCHANGED
MYVAR.ATTR(*LIST).INTENSITY = *UNCHANGED
MYVAR.ATTR(*LIST).COLOR = *UNCHANGED
MYVAR.ATTR(*LIST).OUTPUT = *UNCHANGED
MYVAR.COMMAND =
MYVAR.FHS-VERSION =
MYVAR.CURSOR-INPUT =
MYVAR.CURSOR-INPUT-INDEX =
MYVAR.CURSOR-INPUT-POS =
```

6.3 Beispiele für das Arbeiten mit FHS mittels S-Prozeduren

Das nächste Beispiel zeigt, wie eine FHS V8.1 Anwendung, die mit "FHS- Dialog-Variablen" arbeitet, durch eine S-Prozedur gesteuert werden kann. Die S-Prozedur kontrolliert die Felder der IFG-Maske die von der FHS-Anwendung verwendet wird. Die "FHS-Formate" werden durch "Variablen-Container"simuliert. Folgende Formate werden verwendet:

SCREEN1:

```

Dateiauswahl
-----
Auswahl des Dateinamens (*=Wildcard)

      a*

-----
Kmdo==>
F1=Hilfe F3=Beenden F12=Abbrechen
    
```

SCREEN2:

```

Aktion
-----
Dateiverwaltungssystem
-----
Dateiliste                               Zeilen 7 bis 8 von 8 Weiter:
? Größe Cat UserId  Dateiname
3      X  XXXXXX  AX
3      X  XXXXXX  AXX
*****Ende der Liste*****
-----
Kmdo==>
F1=Hilfe F3=Beenden F12=Abbrechen
    
```

```

/"main" set-procedure-options
/
/include-procedure *lib-elem(lib=$.SYSPRC.SDF-P-BASYS.020,e1=FHDR) -
/
, (PREFIX='SYSFHS-')
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL)
/declare-variable SYSPINFO (type = *struct(SYSFHS-LAYOUT))
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL) , -
/
(include-form='INITIALIZE',variable-name='SYSPINFO')
/
/declare-variable SCREEN1(type=*structure(*by-syscmd))
/begin-structure
/ declare-element FILE-NAME
/end-structure
/
/declare-variable SCREEN2(type=*structure(*by-syscmd))
/begin-structure
/ declare-element name = FILELIST (type = *structure(*dynamic)) -
/
, multiple-elements = *list
/ declare-element name = SDFPLIST-MODINDEX (type = integer) -
/
, multiple-elements = *list'
/ declare-element SDFPLIST-TOPINDEX
/ declare-element ACTION-CHOICE
/ declare-element ACTION-NUMBER
/ declare-element ACTION
/end-structure
/
/NUMBER-OF-ATTEMPT = 1
/
/ "Initialize MODINDEX List for 30 elements "
/LOOP: while (NUMBER-OF-ATTEMPT <= 30)
/
SCREEN2.SDFPLIST-MODINDEX#&(NUMBER-OF-ATTEMPT) = 0
/
NUMBER-OF-ATTEMPT = NUMBER-OF-ATTEMPT + 1
/end-while LOOP
/
/assign-stream -
/
stream-name = PRESENTATION,-
/
to = *SERVER(FHS,-
/
server-info = 'FHS-LIB = MAPLIB')
/
/SYSPINFO.RESOURCE = 'SDFPPFILE'
/SYSPINFO.SERVICE = '*DISPLAY'
/SYSPINFO.REFRESH = '*YES'
/SYSPINFO.ACK = '*YES'
/
/transmit-by-stream variable-name = SCREEN1, -
/
stream-name = PRESENTATION,-
/
control-var-name = SYSPINFO
/

```

```

/if (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE NE 'IDH0000')
/show-variable SYSPINFO.STD-HEADER
/ end-if
/
/if ((SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0004') -
/ OR (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0008'))
/ exit-procedure
/ end-if
/
/execute-cmd cmd=(SHOW-FILE-ATTRIBUTES f-name=&(SCREEN1.FILE-NAME) -
/ ,INFO=NAME-AND-SPACE) -
/ ,text-output = *NONE -
/ ,structure-output = SCREEN2.FILELIST
/
/repeat
/
/ SYSPINFO.RESOURCE = 'SDFPLIST'
/ SYSPINFO.REFRESH = '*YES'
/ SYSPINFO.ACK = '*YES'
/ SYSPINFO.SERVICE = '*DISPLAY'
/ SYSPINFO.ACK = '*YES'
/ SCREEN2.SDFPLIST-TOPINDEX = 2
/
/ transmit-by-stream variable-name = SCREEN2, -
/ stream-name = PRESENTATION,-
/ control-var-name = SYSPINFO
/
/ if (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE NE 'IDH0000')
/ show-variable SYSPINFO.STD-HEADER
/ end-if
/
/until ((SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0004') -
/ OR (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0008'))
/
/assign-stream stream-name = PRESENTATION, to = *dummy
/
/exit-procedure

```

6.4 Beispiel für die Verwendung von Feld-Attributen:

```

/"attt" set-procedure-options
/
/include-procedure *lib-elem(lib=$.SYSPRC.SDF-P-BASYS.020,e1=FHDR) -
/      ,(PREFIX='SYSFHS-')
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL)
/declare-variable SYSPINFO (type = *struct(SYSFHS-LAYOUT))
/include-procedure *lib-elem(lib=$.SYSPRC.FHS.081,e1=SYSFHS-CONTROL) , -
/      (include-form='INITIALIZE',variable-name='SYSPINFO',-
/      attr-list=3)
/
/declare-variable SCREEN1(type=*structure(*by-syscmd))
/begin-structure
/  declare-element INPUT-FLD
/end-structure
/
/
/assign-stream -
/      stream-name = PRESENTATION,-
/      to           = *SERVER(FHS,-
/      server-info = 'FHS-LIB = MAPLIB')
/
/repeat
/
/SYSPINFO.RESOURCE = 'V03      '
/SYSPINFO.SERVICE = '*DISPLAY'
/SYSPINFO.ATTR#1.FIELD = '(SYS-DATE,SYS-DAY)'
/SYSPINFO.ATTR#1.FIELD-IND = '0'
/SYSPINFO.ATTR#1.HILITE   = '*UNDERLINE'
/SYSPINFO.ATTR#2.FIELD = 'SYS-TIME'
/SYSPINFO.ATTR#2.INTENSITY = '*INVISIBLE'
/
/transmit-by-stream variable-name = SCREEN1, -
/      stream-name = PRESENTATION,-
/      control-var-name = SYSPINFO
/
/if (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE NE 'IDH0000')
/show-variable SYSPINFO.STD-HEADER
/  end-if
/
/if (SYSPINFO.COMMAND = 'usercmd')
/  include-procedure *l(lib=userlib,e1=userelem)
/  end-if
/

```

```
/until ((SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0004') -  
/ OR (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE EQ 'IDH0008'))  
/  
/assign-stream stream-name = PRESENTATION, to = *dummy  
/  
/exit-procedure
```

6.5 Beispiel: Erstellen eines graphischen Bibliotheksmanagers

In diesem Beispiel wird ausführlich eine Anwendungsmöglichkeit gezeigt, wie das Zusammenspiel von S-Prozeduren, S-Variablenströmen und FHS für die Erstellung eines graphischen Bibliotheksmanagers genutzt werden kann.

Die einzelnen dafür verantwortlichen S-Prozeduren sind (intern) als Elemente in der Bibliothek LIBRARY-MANAGER.PL unter den Namen RUN, SCREEN01 und SCREEN02 mit dem jeweiligen Typ J verwahrt. Dabei ist RUN die steuernde S-Prozedur, die durch CALL-PROCEDURE aufgerufen wird. SCREEN01 und SCREEN02 sind dagegen davon abhängig für die zwei möglichen Standard-Bildschirmanzeigen verantwortlich.

Im folgenden werden diese drei S-Prozeduren aufgelistet. Anschließend werden einige Anwendungen gezeigt, um zu demonstrieren, wie der FHS-unterstützte Bibliotheksmanager eingesetzt werden kann.

Prozedur: RUN

```

/SET-PROCEDURE-OPTIONS CALLER=CALL
/
/"-----"
/"FIRST EXTRACT LIBRARY NAME FROM WHICH THIS PROCEDURE IS CALLED"
/"-----"
/DECLARE-VARIABLE SYSOUT(TYPE=*STRING),MULTIPLE-ELEMENTS=*LIST
/ASSIGN-SYSOUT TO=*VARIABLE(SYSOUT)
/SHOW-SYSTEM-FILE-ASSIGNMENT SYSTEM-FILE=*SYSCMD
/ASSIGN-SYSOUT *PRIMARY
/
/LIB-ELEM-POS = INDEX(SYSOUT#2,'*LIB-ELEM')
/LIBRARY-NAME = SUBSTRING(SYSOUT#2,LIB-ELEM-POS + LENGTH('*LIB-ELEM('))
/COMMA = INDEX(LIBRARY-NAME,',')
/LIBRARY-NAME = SUBSTRING(LIBRARY-NAME,1,COMMA - 1)
/
/"-----"
/"INITIALIZE FHS CONTROL VARIABLES"
/"-----"
/WRITE-TEXT 'LIBRARY MANAGER V1.0 - LOADING'
/INCLUDE-PROCEDURE *LIBRARY-ELEMENT(LIBRARY = $.SYSPRC.SDF-P-BASYS.020 -
/                                     ,ELEMENT = FHDR) -
/                                     ,PROCEDURE-PARAMETERS = (PREFIX = 'SYSFHS-')
/INCLUDE-PROCEDURE *LIBRARY-ELEMENT(LIBRARY = $.SYSPRC.FHS.081 -
/                                     ,ELEMENT = SYSFHS-CONTROL)
/DECLARE-VARIABLE SYSPINFO (TYPE = *STRUCTURE(SYSFHS-LAYOUT))
/DECLARE-VARIABLE SYSPINFO-SAVE (TYPE = *STRUCTURE(SYSFHS-LAYOUT))
/INCLUDE-PROCEDURE *LIBRARY-ELEMENT(LIBRARY = $.SYSPRC.FHS.081 -
/                                     ,ELEMENT = SYSFHS-CONTROL) -
/                                     ,PROCEDURE-PARAMETERS = (INCLUDE-FORM='INITIALIZE' -

```

```

/                                     ,VARIABLE-NAME='SYSPINFO')
/
/"-----"
/"ASSIGN THE STREAM TO FHS                                     "
/"-----"
/ASSIGN-STREAM STREAM-NAME = PRESENTATION -
/          ,TO          = *SERVER(FHS -
/          ,SERVER-INFO = 'FHS-LIB = &LIBRARY-NAME.')
```

```

/
/"-----"
/"START LMS V3.0                                             "
/"-----"
/ASSIGN-SYSOUT TO=*DUMMY
/START-LMS
/HOLD-PROGRAM
/ASSIGN-SYSOUT TO=*PRIMARY
/
/"-----"
/"SET TIMEOUT TO 0 WHEN SWITCHING FROM LINE MODE TO FULL SCREEN "
/"-----"
/MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL = *TIME(TIMEOUT = 0)
/
/"-----"
/"CALL MAIN PROCEDURE (SCREEN01)                             "
/"-----"
/INCLUDE-PROCEDURE *LIBRARY-ELEMENT(LIBRARY = &LIBRARY-NAME. -
/                                     ,ELEMENT = SCREEN01)
/
/"-----"
/"STOP LMS V3.0                                             "
/"-----"
/RESUME-PROGRAM
//END
```

Prozedur: SCREEN01

```

/DECLARE-VARIABLE SCREEN01(TYPE=*STRUCTURE(*BY-SYSCMD))
/BEGIN-STRUCTURE
/  DECLARE-ELEMENT NAME = FILELIST (TYPE = *STRUCTURE(*BY-SYSCMD)) -
/      ,MULTIPLE-ELEMENTS = *LIST
/  BEGIN-STRUCTURE
/    DECLARE-ELEMENT CHOICE
/    DECLARE-ELEMENT F-SIZE
/    DECLARE-ELEMENT CAT-ID
/    DECLARE-ELEMENT USER-ID
/    DECLARE-ELEMENT SHORT-F-NAME
/  END-STRUCTURE
/  DECLARE-ELEMENT NAME = SDFPLIST-MODINDEX (TYPE = INTEGER) -
/      ,MULTIPLE-ELEMENTS = *LIST
/  DECLARE-ELEMENT SDFPLIST-TOPINDEX(INITIAL-VALUE = 1)
/  DECLARE-ELEMENT SDFPLIST-BOTINDEX
/  DECLARE-ELEMENT SDFPLIST-NUMROW
/  DECLARE-ELEMENT FILE-MENU
/  DECLARE-ELEMENT FILE-CHOICE
/END-STRUCTURE
/
/DECLARE-VARIABLE I(TYPE = *INTEGER)
/
/WHILE (TRUE)
/
/  "INITIALIZE MODINDEX LIST FOR 50 ELEMENTS "
/  " (FHS REQUIREMENT) "
/  I = 1
/  WHILE (I <= 50)
/    SCREEN01.SDFPLIST-MODINDEX#&(I) = 0
/    I = I + 1
/  END-WHILE
/
/  "GET LIBRARY NAMES"
/  EXEC-CMD CMD=(SHOW-FILE-ATTRIBUTES -
/      SELECT=*BY-ATTRIBUTES(TYPE-OF-FILES = *PLAM-LIBRARY) -
/      ,INFO=*NAME-AND-SPACE -
/      ) -
/      ,STRUCTURE-OUTPUT=SCREEN01.FILELIST -
/      ,TEXT-OUTPUT=*NONE -
/      ,RETURNCODE=*VARIABLE(SUBCODE2=SUB2 -
/          ,SUBCODE1=SUB1 -
/          ,MAINCODE=MAIN)
/
/  IF (SUB1 NE 0)
/    WRITE-TEXT 'ERROR &SUB2 &SUB1 &MAIN RETURNED BY EXEC-CMD'
/    WRITE-TEXT 'LIBRARY MANAGER V1.0 ABNORMALLY TERMINATED

```

```

/      EXIT-PROCEDURE
/      END-IF
/
/      SYSPINFO.RESOURCE = 'SCREEN01'
/      SYSPINFO.SERVICE = '*DISPLAY'
/      SYSPINFO.REFRESH = '*YES'
/      SYSPINFO.COMMAND = ''
/      SCREEN01.SDFPLIST-NUMROW = SIZE('SCREEN01.FILELIST')
/      SCREEN01.FILE-MENU=0
/      SCREEN01.FILE-CHOICE=0
/      TRANSMIT-BY-STREAM VARIABLE-NAME = SCREEN01 -
/          ,STREAM-NAME = PRESENTATION -
/          ,CONTROL-VAR-NAME = SYSPINFO
/
/      IF ( (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE == 'IDH0004') -
/          OR (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE == 'IDH0008'))
/          WRITE-TEXT 'LIBRARY MANAGER V1.0 NORMALLY TERMINATED'
/          EXIT-PROCEDURE
/      END-IF
/
/      IF (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE NE 'IDH0000')
/          SUB2 = SYSPINFO.STD-HEADER.RETURNCODE.SUBCODE2
/          SUB1 = SYSPINFO.STD-HEADER.RETURNCODE.SUBCODE1
/          MAIN = SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE
/          WRITE-TEXT 'ERROR &SUB2 &SUB1 &MAIN RETURNED BY FHS SERVER'
/          WRITE-TEXT 'LIBRARY MANAGER V1.0 ABNORMALLY TERMINATED'
/          EXIT-PROCEDURE
/      END-IF
/
/      IF SCREEN01.FILE-MENU NE 0
/          IF SCREEN01.FILE-CHOICE == 9
/              WRITE-TEXT 'LIBRARY MANAGER V1.0 NORMALLY TERMINATED'
/              EXIT-PROCEDURE
/          ELSE-IF SCREEN01.FILE-CHOICE == 1
/              I = 1
/              WHILE (SCREEN01.SDFPLIST-MODINDEX#I NE 0)
/                  SCREEN01-CURR-INDEX = SCREEN01.SDFPLIST-MODINDEX#I
/                  IF SCREEN01.FILELIST#SCREEN01-CURR-INDEX.CHOICE == '/'
/                      SYSPINFO-SAVE = SYSPINFO
/                      INCLUDE-PROCEDURE -
/                          NAME=*LIBRARY-ELEMENT(&LIBRARY-NAME. -
/                              ,SCREEN02) -
/                          ,PROCEDURE-PARAMETERS=(&(SCREEN01.FILELIST#SCREEN01-CURR-
INDEX.SHORT-F-NAME))
/                      IF-CMD-ERROR
/                          WRITE-TEXT 'LIBRARY MANAGER V1.0 ABNORMALLY TERMINATED'
/                          EXIT-PROCEDURE
/                      ELSE

```

```

/          SAVE-RETURNCODE
/          IF (MAINCODE() = 'STOPOOK')
/              WRITE-TEXT 'LIBRARY MANAGER V1.0 NORMALLY TERMINATED'
/              EXIT-PROCEDURE
/          END-IF
/          END-IF
/          SYSPINFO = SYSPINFO-SAVE
/          END-IF
/          I = I + 1
/          END-WHILE
/          END-IF
/      END-IF
/
/      IF (SYSPINFO.COMMAND NE '')
/          EXEC-CMD CMD=(&(SYSPINFO.COMMAND)) -
/              ,TEXT-OUTPUT=*NONE -
/              ,RETURNCODE=*VARIABLE(SUBCODE2=SUB2 -
/                                      ,SUBCODE1=SUB1 -
/                                      ,MAINCODE=MAIN)
/
/          IF (SUB1 NE 0)
/              WRITE-TEXT 'ERROR &SUB2 &SUB1 &MAIN RETURNED BY COMMAND SERVER'
/              WRITE-TEXT 'LIBRARY MANAGER V1.0 ABNORMALLY TERMINATED'
/              EXIT-PROCEDURE
/          END-IF
/      END-IF
/END-WHILE

```

Prozedur: SCREEN02

```

/BEGIN-PARAMETER-DECLARATION
/ DECLARE-PARAMETER LIBRARY
/END-PARAMETER-DECLARATION
/
/DECLARE-VARIABLE SCREEN02(TYPE=*STRUCTURE(*BY-SYSCMD))
/BEGIN-STRUCTURE
/ DECLARE-ELEMENT NAME = ELEMLIST(TYPE = *STRUCTURE(*DYNAMIC)) -
/     ,MULTIPLE-ELEMENT = *LIST
/ DECLARE-ELEMENT NAME = SDFPLIST-MODINDEX(TYPE = *INTEGER) -
/     ,MULTIPLE-ELEMENT = *LIST
/ DECLARE-ELEMENT SDFPLIST-TOPINDEX(INITIAL-VALUE = 1)
/ DECLARE-ELEMENT SDFPLIST-BOTINDEX
/ DECLARE-ELEMENT SDFPLIST-NUMROW
/ DECLARE-ELEMENT FILE-MENU
/ DECLARE-ELEMENT FILE-CHOICE
/END-STRUCTURE
/

```

```

/DECLARE-VARIABLE SYSOUT(TYPE = *STRING), MULTIPLE-ELEMENTS = *LIST
/DECLARE-VARIABLE ERROR-ON-PRINT( TYPE = *BOOLEAN, INITIAL-VALUE = FALSE )
/DECLARE-VARIABLE I(TYPE = *INTEGER)
/
/RESUME-PROGRAM
//OPEN-LIBRARY LIBRARY = &LIBRARY.,MODE = *UPDATE
/HOLD-PROGRAM
/
/WHILE (TRUE)
/
/  "INITIALIZE MODINDEX LIST FOR 50 ELEMENTS "
/  " (FHS REQUIREMENT) "
/  I = 1
/  WHILE ( I <= 50)
/    SCREEN02.SDFPLIST-MODINDEX#&(I) = 0
/    I = I + 1
/  END-WHILE
/
/  ASSIGN-SYSOUT TO = *VARIABLE(SYSOUT)
/  RESUME-PROGRAM
//  SHOW-ELEMENT-ATTRIBUTES -
//    ELEMENT = *LIBRARY-ELEMENT(LIBRARY = *STD -
//                                     ,ELEMENT = *ALL ( VERSION = *ALL ) -
//                                     ,TYPE = *ALL) -
//    ,INFORMATION = *MAXIMUM -
//    ,SORT = *BY-NAME -
//    ,STRUCTURE-OUTPUT = SCREEN02.ELEMLIST
/  HOLD-PROGRAM
/  ASSIGN-SYSOUT TO = *PRIMARY
/
/  IF ( STMT-SPINOFF() == 'YES' )
/    SHOW-VARIABLE SYSOUT, INFORMATION = *PARAMETERS( NAME = *NONE )
/    MAINCODE = 'LMSOERR'
/    GOTO END
/  END-IF
/
/  SYSPINFO.RESOURCE = 'SCREEN02'
/  SYSPINFO.SERVICE = '*DISPLAY'
/  SYSPINFO.REFRESH = '*YES'
/  SYSPINFO.COMMAND = ''
/  SCREEN02.SDFPLIST-NUMROW = SIZE('SCREEN02.ELEMLIST')
/  SCREEN02.FILE-MENU=0
/  SCREEN02.FILE-CHOICE=0
/  TRANSMIT-BY-STREAM VARIABLE-NAME = SCREEN02 -
/                                     ,STREAM-NAME = PRESENTATION -
/                                     ,CONTROL-VAR-NAME = SYSPINFO
/

```

```

/   IF ( (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE == 'IDH0004') -
/       OR (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE == 'IDH0008'))
/       MAINCODE = 'FHSEXIT'
/       GOTO END
/   END-IF
/
/   IF (SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE NE 'IDH0000')
/       SUB2 = SYSPINFO.STD-HEADER.RETURNCODE.SUBCODE2
/       SUB1 = SYSPINFO.STD-HEADER.RETURNCODE.SUBCODE1
/       MAIN = SYSPINFO.STD-HEADER.RETURNCODE.MAINCODE
/       WRITE-TEXT 'ERROR &SUB2 &SUB1 &MAIN RETURNED BY FHS SERVER
/       MAINCODE = 'FHSOERR'
/       GOTO END
/   END-IF
/
/   IF SCREEN02.FILE-MENU NE 0
/       IF SCREEN02.FILE-CHOICE == 9
/           MAINCODE = 'FHSORET'
/           GOTO END
/       ELSE
/           I = 1
/           WHILE (SCREEN02.SDFPLIST-MODINDEX#I NE 0)
/               SCREEN02-CURR-INDEX = SCREEN02.SDFPLIST-MODINDEX#I
/               IF SCREEN02.ELEMLIST#SCREEN02-CURR-INDEX.CHOICE == '/'
/                   ELEMENT = SCREEN02.ELEMLIST#SCREEN02-CURR-INDEX.ELEM
/                   VERSION = SCREEN02.ELEMLIST#SCREEN02-CURR-INDEX.VERSION
/                   TYPE = SCREEN02.ELEMLIST#SCREEN02-CURR-INDEX.TYPE
/                   IF SCREEN02.FILE-CHOICE == 1 "DELETE ELEMENT"
/                       ASSIGN-SYSOUT TO = *VARIABLE(SYSOUT)
/                       RESUME-PROGRAM
//                      DELETE-ELEMENT ELEMENT = *LIBRARY-ELEMENT -
//                          ( LIBRARY = *STD -
//                            , ELEMENT = &ELEMENT.-
//                              ( VERSION = &VERSION. ) -
//                                , TYPE = &TYPE. )
/                       HOLD-PROGRAM
/                       ASSIGN-SYSOUT *PRIMARY
/                   ELSE-IF SCREEN02.FILE-CHOICE == 2 "EDIT ELEMENT"
/                       ASSIGN-SYSOUT TO = *VARIABLE(SYSOUT)
/                       RESUME-PROGRAM
//                      EDIT-ELEMENT ELEMENT = *LIBRARY-ELEMENT -
//                          ( LIBRARY = *STD -
//                            , ELEMENT = &ELEMENT. -
//                              ( VERSION = &VERSION. ) -
//                                , TYPE = &TYPE. )
/                       HOLD-PROGRAM
/                       ASSIGN-SYSOUT *PRIMARY

```

```

/          ELSE-IF SCREEN02.FILE-CHOICE == 3 "COPY ELEMENT"
/          WRITE-TEXT 'FUNCTION NOT IMPLEMENTED'
/          ELSE-IF SCREEN02.FILE-CHOICE == 4 "PRINT ELEMENT"
/          ASSIGN-SYSOUT TO = *VARIABLE(SYSOUT)
/          PRINT-FILE *LIBRARY-ELEMENT -
/              ( LIBRARY = &LIBRARY. -
/                , ELEMENT = &ELEMENT. -
/                  ( VERSION = &VERSION. ) -
/                    , TYPE = &TYPE. )
/          IF-CMD-ERROR
/          ERROR-ON-PRINT = TRUE
/          END-IF
/          ASSIGN-SYSOUT *PRIMARY
/          ELSE-IF SCREEN02.FILE-CHOICE == 5 "SELECT ELEMENT"
/          ASSIGN-SYSOUT TO = *VARIABLE(SYSOUT)
/          RESUME-PROGRAM
//          EXTRACT-ELEMENT ELEMENT = *LIBRARY-ELEMENT -
//              ( LIBRARY = *STD -
//                , ELEMENT = &ELEMENT. -
//                  ( VERSION = &VERSION. ) -
//                    , TYPE = &TYPE. ) -
//              ,TO-FILE = *STD
/          HOLD-PROGRAM
/          ASSIGN-SYSOUT *PRIMARY
/          ELSE-IF SCREEN02.FILE-CHOICE == 6 "ADD ELEMENT"
/          WRITE-TEXT 'FUNCTION NOT IMPLEMENTED'
/          END-IF
/
/          IF ( ERROR-ON-PRINT )
/          SHOW-VARIABLE SYSOUT, INFORMATION = *PARAMETERS( NAME =
*NONE )
/          MAINCODE = 'PRTOERR'
/          GOTO END
/          END-IF
/
/          IF ( STMT-SPINOFF() == 'YES' )
/          SHOW-VARIABLE SYSOUT, INFORMATION = *PARAMETERS( NAME =
*NONE )
/          MAINCODE = 'LMSOERR'
/          GOTO END
/          END-IF
/          END-IF
/          I = I + 1
/          END-WHILE
/          END-IF
/          END-IF
/
/

```

```
/ IF (SYSPINFO.COMMAND NE '')
/   EXEC-CMD CMD=(&(SYSPINFO.COMMAND)) -
/     ,TEXT-OUTPUT=*NONE -
/     ,RETURNCODE=*VARIABLE(SUBCODE2=SUB2 -
/                           ,SUBCODE1=SUB1 -
/                           ,MAINCODE=MAIN)
/
/   IF (SUB1 NE 0)
/     WRITE-TEXT 'ERROR &SUB2 &SUB1 &MAIN RETURNED BY COMMAND SERVER'
/     MAINCODE = 'CMDOERR'
/     GOTO END
/   END-IF
/ END-IF
/END-WHILE
/
/END:
/IF ( (MAINCODE = 'CMDOERR') -
/   OR (MAINCODE = 'PRTOERR') -
/   OR (MAINCODE = 'LMSOERR') -
/   OR (MAINCODE = 'FHSOERR') -
/ )
/   EXIT-PROCEDURE ERROR = *YES(SUBCODE2 = 0 -
/                               ,SUBCODE1 = 64 -
/                               ,MAINCODE = STOPERR)
/ELSE-IF (MAINCODE = 'FHSEXIT')
/   EXIT-PROCEDURE ERROR = *YES(SUBCODE2 = 0 -
/                               ,SUBCODE1 = 0 -
/                               ,MAINCODE = STOPOOK)
/ELSE-IF (MAINCODE = 'FHSORET')
/   EXIT-PROCEDURE
/ELSE
/   WRITE-TEXT 'ERROR &(SC2()) &(SC1()) &(MC()) REPORTED'
/   EXIT-PROCEDURE ERROR = *YES(SUBCODE2 = 0 -
/                               ,SUBCODE1 = 64 -
/                               ,MAINCODE = STOPERR)
/END-IF
```

Unter der Voraussetzung, daß FHS-PRIV geladen ist, kann der Bibliotheksmanager folgendermaßen aufgerufen werden:

```
CALL-PROCEDURE FROM-FILE=*LIBRARY-ELEMENT(LIBRARY-MANAGER.PL, RUN)
```

Es muß dann eine Bildschirmanzeige (unter Benutzung von SCREEN01) erscheinen, die die Namen der in der Kennung enthaltenen Bibliotheken auflistet, wie z.B:

```

File
-----
                                L I B R A R Y   M A N A G E R
-----
FILE(S) SELECTION                From:      1      Total:      5
                                To  :      5      More  :
?  Size  Cat.  UserId  File name
-----
  210   20S2 QM211  ALF.ASS.PLAMLIB
   30   20S2 QM211  ALF.LIB
  342   20S2 QM211  LIB-MAN
  342   20S2 QM211  LIBRARY-MANAGER.PL
   12   20S2 QM211  SCREEN02
===== [ N O   M O R E   D A T A   ] =====

-----
COMMAND ==>
F1=HELP  F3=EXIT
-----
LTG                                           TAST

```

Man hat nun z.B. die Möglichkeit, eine der aufgelisteten Bibliotheken zu öffnen und sich die jeweiligen Elemente anzeigen zu lassen. Dazu kennzeichnet man mit einem "/" am Zeilenanfang die zu öffnende Bibliothek, springt dann mit der Tabulatortaste zum Feld "FILE" (links oben), drückt die DÜ-Taste und gibt in dem erscheinenden Pull-Down-Menü "1" an. Die folgende BildschirmAusgabe soll dies verdeutlichen:

```

File
-----
: 1 1. Open library      : A R Y      M A N A G E R
: 9. Exit Library-manager :
:.....:
:.....: From:      1      Total:      5
:.....: To   :      5      More   :
-----
? Size  Cat.  UserId  File name
-----
 210   20S2  QM211   ALF.ASS.PLAMLIB
 30    20S2  QM211   ALF.LIB
 / 342   20S2  QM211   LIB-MAN
 342   20S2  QM211   LIBRARY-MANAGER.PL
 12    20S2  QM211   SCREEN02
===== [ N O M O R E D A T A ] =====

-----
COMMAND ==>
F1=HELP  F3=EXIT
    
```

Drückt man nun nochmal die DÜ-Taste, erscheinen in einer weiteren Bildschirmausgabe (unter Benutzung von SCREEN02) die jeweiligen Elementnamen dieser Bibliothek (zusammen mit Datum und Zeit ihrer Erstellung sowie dem Typ usw.).

```

File
-----
                                L I B R A R Y   M A N A G E R
-----
ELEMENT(S) SELECTION                From:      1      Total:      86
                                      To   :      6      More   :  +
? Element                               Date      Time                               Type
  Version
-----
LISTHLP
 *UP-LIM                1994-10-06 14:11:07                F
PHKEY
 001                    1994-10-06 14:17:57                F
SCREEN01
 001                    1994-10-06 14:10:19                F
SCREEN02
 001                    1994-10-06 14:10:28                F
-----
COMMAND ==>
F1=HELP  F3=EXIT
    
```

Weiter hat man nun die Möglichkeit, eines der aufgelisteten Elemente zu bearbeiten. Dazu kennzeichnet man mit einem "/" am Zeilenanfang das zu bearbeitende Element, springt dann mit der Tabulatortaste zum Feld "FILE" (links oben), drückt die DÜ-Taste und gibt in dem erscheinenden Pull-Down-Menü die Zahl an, die markiert, wie man das Element bearbeiten will: z.B. "4", um es auszudrucken.

```

File
-----
: 4 1. Delete           : R A R Y   M A N A G E R
: 2. Edit              :
: 3. Copy              :
: 4. Print             : From:      1      Total:   86
: 5. Select element   : To   :      6      More :   +
: 6. Add element      : Date       Time   Type
: 9. Return to main menu :
: .....:
: *UP-LIM             1994-10-06 14:11:07
/ PHKEY
  001                 1994-10-06 14:17:57
  SCREEN01           1994-10-06 14:10:19
  001
  SCREEN02           1994-10-06 14:10:28
  001
-----
COMMAND ==>
F1=HELP  F3=EXIT

```

Mit Betätigung der DÜ-Taste wird dies dann ausgeführt.

Soviel zu einigen Anwendungsmöglichkeiten: Mit der F3-Taste kann man den Bibliotheksmanager verlassen bzw. zum Ausgangsmenü zurückkehren.

Im übrigen kann man mit Eingabe von "+" oder "-" in der Zeile "COMMAND" nach dem Pfeil scrollen. Ein Hilfemenü ist durch Auslösen der F1-Taste zu erreichen.

7 Anhang

7.1 Übersicht der Rückkehrcodes

Der Hauptwert des Rückkehrcodes im DMMCOMM hat als Dezimalzahl die Struktur: gffnnn. Dabei steht "g" für Fehlertyp, "ff" für Funktionsgruppe und "nnn" für Fehlernummer

Fehlertyp	Bedeutung
0	kein Fehler
1	Fehler/Warnung (Fehleranzeige möglich)
2	Fehler (Fehleranzeige nicht möglich)
3	schwerer Fehler (z.B. ABORT-Kommando)

Funktionsgruppe	Bedeutung	viertes Zeichen des Meldungszeichens (x)
00	DM-Kommando (CANCEL/EXIT)	
01	allgemeine Fehler	Z
02	Syntaxprüfung	A
03	DISPLAY-Dienst	D
04	sonstige Dienste	U
05	DMOPEN ,	B
06	SDF-P-Schnittstelle (FHS-PRIV)	P
07	Variablendienste	V
08	reserviert	O
09	DOORS	C

Die Meldungen leiten sich aus dem Rückkehrcode ab und haben folgende Form: IDHxnnn (Bedeutung von x und nnn siehe oben)

7.2 Liste der Meldungen

Die Meldungen sind folgendermaßen aufgebaut:

Meldungsschlüssel (Dezimalwert)

Meldungstext.

IDHA001 (102001)

Beim Aufruf eines Dialogdienstes ist der Operandenbereich größer als zulässig. Bitte informieren Sie den Anwendungsentwickler!

IDHA002 (102002)

Beim Aufruf eines Dialogdienstes oder nach einer Variablensubstitution enthält der Operandenbereich ein NIL-Byte (X'00'). Bitte informieren Sie den Anwendungsentwickler!

IDHA003 (102003)

Die Syntax des Variablennamens ist nicht korrekt. Bitte informieren Sie den Anwendungsentwickler!

IDHA004 (102004)

Beim Aufruf eines Dialogdienstes ist der Operandenbereich nach der Substitution aller Variablen länger als 2048 Bytes. Bitte informieren Sie den Anwendungsentwickler!

IDHA005 (102005)

Beim Aufruf eines Dialogdienstes ist der Operandenbereich nicht korrekt. Bitte informieren Sie den Anwendungsentwickler!

IDHA006 (102006)

Beim Aufruf eines Dialogdienstes ist der Operandenbereich leer. Bitte informieren Sie den Anwendungsentwickler!

IDHA007 (102007)

Ein Operand wurde mehr als einmal angegeben. Bitte informieren Sie den Anwendungsentwickler!

IDHA008 (102008)

Eine öffnende Klammer fehlt. Bitte informieren Sie den Anwendungsentwickler!

IDHA009 (102009)

Eine schliessende Klammer fehlt. Bitte informieren Sie den Anwendungsentwickler!

IDHA010 (102010)

Die Syntax des Dialogvariablennamens ist nicht korrekt. Bitte informieren Sie den Anwendungsentwickler!

IDHA011 (102011)

Das Zeichen "*" ist in diesem Dialogdienst als Index nicht zulässig. Bitte informieren Sie den Anwendungsentwickler!

IDHA012 (102012)

Der Wert des ROW-Operanden liegt außerhalb der gültigen Werte (-43...43). Bitte informieren Sie den Anwendungsentwickler!

IDHA013 (102013)

Der Wert des COLUMN-Operanden liegt außerhalb der gültigen Werte (-132...132). Bitte informieren Sie den Anwendungsentwickler!

IDHA014 (102014)

Ein unbekannter Operand wurde angegeben. Bitte informieren Sie den Anwendungsentwickler!

IDHA015 (102015)

Es wurde kein Variablenname angegeben, obwohl mindestens einer erforderlich ist. Bitte informieren Sie den Anwendungsentwickler!

IDHA016 (102016)

Es existiert eine öffnende Klammer, die nicht erwartet wird. Bitte informieren Sie den Anwendungsentwickler!

IDHA017 (102017)

Es existiert ein unbekannter Wert. Bitte informieren Sie den Anwendungsentwickler!

IDHA018 (102018)

Es wurden zwei sich ausschliessende Werte angegeben, oder ein Wert existiert doppelt. Bitte informieren Sie den Anwendungsentwickler!

IDHA020 (102020)

Der Operand "DISPLAY" fehlt. Bitte informieren Sie den Anwendungsentwickler!

IDHA021 (102021)

Die Syntax des Zeichensatznamens (CCSN) ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA022 (102022)

Es wurde kein Operand angegeben, obwohl mindestens ein Operand erforderlich ist. Bitte informieren Sie den Anwendungsentwickler!

IDHA023 (102023)

Die Syntax des Formatnamens im PANEL-Operanden ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA024 (102024)

Die Syntax des Meldungsschlüssels im MSG-Operanden ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA025 (102025)

Der MSGLOC-Operand darf nur zusammen mit dem Operanden MSG angegeben werden, dieser fehlt aber. Bitte informieren Sie den Anwendungsentwickler!

IDHA026 (102026)

Die Syntax des Feldnamens im MSGLOC-Operanden ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA027 (102027)

Die Syntax des Feldnamens im CURSOR-Operanden ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA028 (102028)

Der CSRPOS-Operand darf nur zusammen mit dem Operanden CURSOR angegeben werden, dieser fehlt aber. Bitte informieren Sie den Anwendungsentwickler!

IDHA029 (102029)

Für die Position im CSRPOS-Operanden wurde keine gültige Zahl angegeben. Bitte informieren Sie den Anwendungsentwickler!

IDHA030 (102030)

Die Position im CSRPOS-Operanden liegt außerhalb der gültigen Werte (1...32767). Bitte informieren Sie den Anwendungsentwickler!

IDHA031 (102031)

Vor dem PROFILE-Operanden muß der Operand INIT angegeben werden. Bitte informieren Sie den Anwendungsentwickler!

IDHA032 (102032)

Die Syntax des Elementnamens im PROFILE-Operanden ist falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA033 (102033)

Mindestens ein Datentyp im FORMAT-Operanden ist falsch angegeben. Bitte informieren Sie den Anwendungsentwickler!

IDHA034 (102034)

Da mehr als ein Datentyp im FORMAT-Operanden angegeben ist, muß die Anzahl der Datentypen mit der Anzahl der Namen in der Namensliste übereinstimmen. Bitte informieren Sie den Anwendungsentwickler!

IDHA035 (102035)

Wurde im FORMAT-Operanden ein "*" angegeben, muß der korrespondierende Eintrag in der Namensliste ebenfalls einen "*" enthalten. Bitte informieren Sie den Anwendungsentwickler!

IDHA036 (102036)

Der Wert des DIM-Operanden ist keine Zahl. Bitte informieren Sie den Anwendungsentwickler!

IDHA037 (102037)

Der Wert des DIM-Operanden liegt außerhalb der gültigen Werte (1...32767). Bitte informieren Sie den Anwendungsentwickler!

IDHA038 (102038)

Am Ende des Operandenbereiches existieren Zeichen, die nicht zu dem aufgerufenen Dienst gehören. Bitte informieren Sie den Anwendungsentwickler!

IDHA039 (102039)

Für diesen Dialogdienst darf in den Variablennamen kein Index angegeben werden. Bitte informieren Sie den Anwendungsentwickler!

IDHA040 (102040)

Für diesen Dialogdienst darf als Index kein "*" angegeben werden. Bitte informieren Sie den Anwendungsentwickler!

IDHA041 (102041)

Die interne Tabelle für die Speicherverwaltung ist voll. Bitte informieren Sie den Anwendungsentwickler!

IDHA042 (102042)

Dieser Dialogdienst muß über den Eingangspunkt ISPC12 gerufen werden. Bitte informieren Sie den Anwendungsentwickler!

IDHA043 (102043)

Die Dialogvariable, deren Name im Operandenbereich durch ihren Wert ersetzt werden soll (Variablensubstitution), existiert nicht. Bitte informieren Sie den Anwendungsentwickler!

IDHA044 (102044)

Im FORMAT-Operanden ist ein "*" anstelle eines Datentyps nur bei einer Liste von Datentypen zulässig. Bitte informieren Sie den Anwendungsentwickler!

IDHA045 (102045)

Es wurde kein Format definiert. Bitte informieren Sie den Anwendungsentwickler!

IDHA046 (102046)

Ein SDF-P-Variablenname ist syntaktisch falsch. Bitte informieren Sie den Anwendungsentwickler!

IDHA047 (102047)

Die Länge des Operandenbereiches ist negativ oder gleich Null. Bitte informieren Sie den Anwendungsentwickler!

IDHA048 (102048)

Im Namen einer SDF-P-Variablen darf kein Index angegeben werden. Bitte informieren Sie den Anwendungsentwickler!

IDHA049 (102049)

Beim DMOPEN-Dienst ist keine Variablensubstitution zulässig. Bitte informieren Sie den Anwendungsentwickler!

IDHD000

Kein Fehler! Alles korrekt!

IDHD004

Kein Fehler! Es wurde das Kommando CANCEL eingegeben.

IDHD008

Kein Fehler! Es wurde das Kommando EXIT eingegeben.

IDHD021 (103021)

Es wurde der DISPLAY-Dienst ohne Operand PANEL zur Wiederanzeige eines vorherigen Bildes aufgerufen. Es existiert aber kein Bild, das angezeigt werden kann.

IDHD022 (103022)

Interner Fehler! Diagnose: Die interne Feldkette ist zerstört. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD023 (103023)

Interner Fehler! Diagnose: Die erzeugte Bildschirmnachricht positioniert über die letzte Bildschirmzeile hinaus. Informieren Sie Ihren SNI-Betreuer!

IDHD024 (103024)

Interner Fehler! Diagnose: Rückwärtspositionierung in der internen Feldkette. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD025 (103025)

Interner Fehler! Diagnose: Rückwärtspositionierung in der erzeugten Bildnachricht. Bitte informieren Sie den SNI-Betreuer!

IDHD026 (103026)

Interner Fehler! Diagnose: Falsches Bildstapelniveau. Bitte informieren Sie den SNI-Betreuer!

IDHD027 (103027)

Das Format &SYS-PAR0 ist als Box definiert worden. Es existiert kein Vollbild, auf das die Box ausgegeben werden kann.

IDHD028 (103028)

Es soll eine Box ausgegeben werden. Das Format &SYS-PAR0 ist jedoch nicht als Box definiert worden.

IDHD029 (103029)

Interner Fehler! Diagnose: Vor der Rückkehr zur Anwendung wurden nicht alle impliziten Bildniveaus entfernt. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD030 (103030)

Bei der Ausgabe einer Meldung trat der Fehler &SYS-PAR7 auf. Der Fehler trat während einer Fehlerbehandlung auf!

IDHD031 (103031)

Fehler in der Fehlerbehandlung! Im Abschnitt Datenbereitstellung trat der Fehler "&SYS-PAR7" auf.

IDHD032 (103032)

Beim Aufruf des DISPLAY-Dienstes mit den Operanden "LOCK" oder "NODISPLAY" wurde auf Grund von Prüfungen eine Aktion oder Meldung des FHS-DM angefordert. MSGID=&SYS-PAR7, PAR1=&SYS-PAR1, PAR2=&SYS-PAR2, PAR3=&SYS-PAR3.

IDHD033 (103033)

Im vorhergehenden ATTR-Dienst wurde der Name eines Maskenfeldes angegeben (&SYS-PAR1). In der anzuzeigenden Maske &SYS-PAR0 gibt es kein Feld mit diesem Namen.

IDHD034 (103034)

Interner Fehler! Diagnose: Falsche Angaben im ATTR-Steuerblock. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD035 (103035)

Der interne Dienst &SYS-PAR2 liefert den Rückkehrcode &SYS-PAR1.

IDHD036 (103036)

Bei der Anzeige des Formates &SYS-PAR0 sollen dynamisch Feldattribute geändert werden. Die angegebenen Attribute für das Feld &SYS-PAR1 sind nicht zulässig. Bitte informieren Sie den Anwendungsentwickler!.

IDHD037 (103037)

Beim Laden des Formates &SYS-PAR1 wurde der Rückkehrcode &SYS-PAR3 geliefert.

IDHD038 (103038)

Das Format &SYS-PAR1 kann nicht gefunden werden.

IDHD042 (203042)

Es soll das Format &SYS-PAR0 angezeigt werden. Dabei liefert der WRTRD-Ruf (VTSU-B) den Rückkehrcode "&SYS-PAR1".

IDHD044 (103044)

Warnung! Es soll das ICE-Format &SYS-PAR9 geladen werden. Dabei tritt ein Fehler auf. Das ICE-Format wurde nicht gefunden oder der R-Modul ist kein ICE-Format. Der Zeichensatz wird nicht geladen.

IDHD070 (103070)

Interner Rückkehrcode! Es soll die Meldung IDHF186, "Syntaxfehler im SETP-Kommando" ausgegeben werden.

IDHD071 (103071)

Interner Rückkehrcode! Es soll die Meldung IDHF184, "Syntaxfehler im PANELID-Kommando" ausgegeben werden.

IDHD072 (103072)

Interner Rückkehrcode! Es soll die Meldung IDHF185, "Syntaxfehler im KEYAREA-Kommando" ausgegeben werden.

IDHD075 (103075)

Interner Rückkehrcode! Es soll die Meldung IDHF192, "Fehler in Einfachauswahl-Eingabefeld" ausgegeben werden.

IDHD076 (103076)

Interner Rückkehrcode! Es soll die Meldung IDHF094, "Fehler in Mehrfachauswahl-Eingabefeld" ausgegeben werden.

IDHD078 (103078)

Interner Rückkehrcode! Es soll die Meldung IDHF198, "Index nicht gefunden" ausgegeben werden.

IDHD079 (103079)

Interner Rückkehrcode! Es soll die Meldung IDHF188, "Hilfe nicht verfügbar" ausgegeben werden.

IDHD080 (103080)

Interner Rückkehrcode! Es soll die Meldung IDHF191, "Hilfe nicht gefunden" ausgegeben werden.

IDHD081 (103081)

Interner Rückkehrcode! Es soll die Meldung IDHF180, "Falsches Kommando" ausgegeben werden.

IDHD082 (103082)

Interner Rückkehrcode! Es soll die Meldung IDHF193, "Falsche Cursorposition in der Menüleiste" ausgegeben werden.

IDHD083 (103083)

Interner Rückkehrcode! Es soll die Meldung IDHF194, "Falsches Kommando in der Menüleiste" ausgegeben werden.

IDHD084 (103084)

Interner Rückkehrcode! Es soll die Meldung IDHF187, "Falsches Verschiebe-Kommando" ausgegeben werden.

IDHD090 (103090)

Interner Fehler! Diagnose: Der Bildstapel ist zerstört. Es wird kein explizites Bild gefunden. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD099 (103099)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Diagnose - Ursache: falscher oder fehlender &SYS-PAR4.

IDHD120 (103120)

Die Version des Formates &SYS-PAR1 kann nicht verarbeitet werden. Informieren Sie den Anwendungsentwickler!

IDHD121 (103121)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Es gibt einen Widerspruch zwischen der ermittelten und der eingetragenen Zeilenanzahl der Maske. Erzeugen Sie das Format neu!

IDHD122 (103122)

Bei der Verarbeitung des Formates &SYS-PAR1 tritt ein Überlauf in internen Tabellen (&SYS-PAR4) auf. Versuchen Sie das Format mit weniger Feldern zu erzeugen.

IDHD123 (103123)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Information für die Diagnose: Falscher Typ im MDBE11.

IDHD124 (103124)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Information für die Diagnose: FDB-Kette für Pull-Down-Menü inkorrekt.

IDHD125 (103125)

Warnung! Im Format &SYS-PAR1 gibt es Felder, die keinen Namen haben. In diese Felder eingegebene Daten werden ignoriert.

IDHD126 (103126)

Im Format &SYS-PAR1 ist das Kommandoingabefeld oder ein Auswahleingabefeld ein geschütztes Feld (Feldname: &SYS-PAR2). Korrigieren Sie das Format!

IDHD127 (103127)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Information für die Diagnose: Der Arbeitsbereich der Maske ist zu klein.

IDHD128 (103128)

Das Format &SYS-PAR1 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Information für die Diagnose: Variable Felder im Hilfeformat.

IDHD129 (103129)

Das Format &SYS-PAR0 ist intern falsch aufgebaut! Diagnose: Für eine Liste fehlt die Eintragung "Anzahl der Spalten/Zeilen". Erzeugen Sie das Format neu!

IDHD130 (103130)

Das Format &SYS-PAR1 ist intern falsch aufgebaut! Diagnose: Widersprüche zwischen MDBE7 und der MDBE-Kette. Erzeugen Sie das Format neu!

IDHD131 (103131)

Interner Fehler! Diagnose: NULL oder NULL-1 als Formatadresse im Stapel. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD132 (103132)

Das Format &SYS-PAR0 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Diagnose: mindestens eine Zeile ist breiter als die Maske.

IDHD133 (103133)

Es soll ein Bild mit einer Box erzeugt werden (evtl. auch Pull-Down-Box). Dabei wird festgestellt, daß mehr als sieben unterschiedliche Zeichensätze benötigt werden. (Format: &SYS-PAR0)

IDHD201 (103201)

Das Format &SYS-PAR0 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Diagnose: Index für Editregel existiert im FDB, es gibt aber keinen MDBE2.

IDHD202 (103202)

Das Format &SYS-PAR0 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Diagnose: Angeforderte Edit- oder Prüfregel fehlt im MDBE2.

IDHD203 (103203)

Interner RC für Aufbereitungsfehler im Format &SYS-PAR0! Im Feld &SYS-PAR1 ist der nicht erwartete Wert "&SYS-PAR2" eingegeben worden. Diese Meldung wird normalerweise durch eine detaillierte Meldung ersetzt.

IDHD204 (103204)

Das Format &SYS-PAR0 ist falsch. Bei der Wertprüfung des Feldes &SYS-PAR1 wurde festgestellt, daß der Vergleichswert im Format nicht der geforderten Aufbereitungsregel genügt. Wert: "&SYS-PAR2", EDIT-RC=&SYS-PAR3 (siehe Hilfe)!

IDHD205 (103205)

Das Format &SYS-PAR0 kann nicht angezeigt werden, da der Wert der Dialogvariablen "&SYS-PAR1" nicht der geforderten Prüfung genügt. Wert: "&SYS-PAR2", EDIT-RC=&SYS-PAR3 (siehe Hilfe). Informieren Sie den Anwendungsentwickler!

IDHD206 (103206)

Das Format &SYS-PAR0 ist intern falsch aufgebaut. Erzeugen Sie das Format neu!
Diagnose: Fehler im MDBE2 für Feld &SYS-PAR1 .

IDHD208 (103208)

Im Format &SYS-PAR0 ist der CCS-Name &SYS-PAR3 angegeben. Dieser CCS-Name ist im System unbekannt.

IDHD209 (103209)

Im Format &SYS-PAR0 liefert ein XHCS-Systemruf den Rückkehrcode &SYS-PAR3..

IDHD210 (103210)

Das Format &SYS-PAR0 ist intern falsch aufgebaut. Erzeugen Sie das Format neu.
Diagnose: Fehler im MDBE11.

IDHD211 (103211)

Bei der Ausgabeverarbeitung des Formates &SYS-PAR0 wird festgestellt, daß XHCS nicht verfügbar ist (RC = &SYS-PAR3).

IDHD212 (103211)

Es soll der Wert "&SYS-PAR2" in die Dialogvariable &SYS-PAR1 geschrieben werden. Dabei ist der Fehler &SYS-PAR3 aufgetreten.

IDHD213 (103213)

Im Format &SYS-PAR0 wurde beim Lesen der Dialogvariablen &SYS-PAR1 der RC = &SYS-PAR3 geliefert.

IDHD214 (003214)

Im Format &SYS-PAR0 wurde der Wert einer Dialogvariablen für ein Ausgabefeld abgeschnitten.

IDHD215 (003215)

Der CCS-Name des Formates &SYS-PAR0 wird ignoriert.

IDHD217 (103217)

Interner Fehler! Diagnose: Im Format &SYS-PAR0 sind im FDV für das Feld &SYS-PAR1 falsche Informationen. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD218 (103218)

Bei einer Listenanzeige stimmt die ermittelte erste Zeile nicht mit der Angabe von &SYS-PAR4 überein. Index der ersten Zeile: &SYS-PAR5 Index von &SYS-PAR4: &SYS-PAR7

IDHD219 (103219)

Interner Fehler! Diagnose: Fehlerhafter interner Bildaufbau für eine Liste (Fehler im Listenkopf) . Bitte informieren Sie Ihren SNI-Betreuer!

IDHD220 (103220)

Interner Fehler! Diagnose: Fehlerhafter interner Bildaufbau für eine Liste (Fehler beim Verschieben). Bitte informieren Sie Ihren SNI-Betreuer!

IDHD221 (103221)

Das Format &SYS-PAR0 enthält eine Liste, die seit der vorhergehenden Anzeige verkürzt wurde (letzte angezeigte Zeile: &SYS-PAR5, Listenende jetzt: &SYS-PAR7) (Siehe auch Hilfe zur Meldung!)

IDHD222 (103222)

Interner Fehler! Diagnose: Falsches Bildstapelniveau oder zerstörtes Format beim Schreiben von Dialogvariablen. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD223 (103223)

Die Steuervariable &SYS-PAR1 ist negativ oder zu groß. Format: "&SYS-PAR0".

IDHD224 (103224)

Das Format &SYS-PAR0 ist fehlerhaft. Die Modellzeile für eine Liste enthält keine variablen Felder. Korrigieren Sie das Format!

IDHD225 (103225)

Interner Fehler! Diagnose: Der interne Dienst IDHHLST wurde mit falschen Parametern aufgerufen. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD226 (103226)

Interner Fehler! Diagnose: Falscher Wert für SCROLL-Lines in CPXCA. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD227 (103227)

Interner Fehler! Diagnose: Falscher Wert für SCROLL-Directions in CPXCA. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD228 (103228)

Warnung! Beim Schreiben der globalen Mark-Variablen des Formates "&SYS-PAR0" trat ein Fehler auf. Informationen zu markierten Feldern gelangen nicht zur Anwendung.

IDHD229 (103229)

Warnung! Die globale Mark-Variable des Formates &SYS-PAR0 ist nicht korrekt definiert. Informationen zu markierten Feldern gelangen nicht zur Anwendung.

IDHD230 (103230)

Interner Fehler! Diagnose: Beim Lesen der DV &SYS-PAR1 wird eine Länge geliefert (<0 oder >Maximum). Bitte informieren Sie Ihren SNI-Betreuer!

IDHD231 (103231)

Beim Ermitteln der Elementanzahl eines Arrays von Dialogvariablen trat ein Fehler auf (interner Rückkehrcode = &SYS-PAR3).

IDHD232 (103232)

Warnung! Der Wert der Dialogvariablen &SYS-PAR1 wurde bei der Ausgabe verkürzt.

IDHD233 (103233)

Interner Fehler! Diagnose: Versuchtes Schreiben nach Listenende!

IDHD234 (103234)

Das Format &SYS-PAR0 ist intern falsch aufgebaut! Diagnose: Es existieren im Format FDB's mit falscher Zeilen/Spaltenangabe. Korrigieren Sie das Format oder fragen Sie Ihren SNI-Betreuer.

IDHD241 (103241)

Interner Fehler! Diagnose: IDHRKEY Falsche Funktion. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD242 (103242)

Interner Fehler! Diagnose: IDHRKEY Kein Name einer Keylist angegeben. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD243 (103243)

Interner Fehler! Diagnose: IDHRKEY Falsche Tastennummer. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD244 (103244)

Interner Fehler! Diagnose: IDHRKEY Falsche Tastenkennzeichnung. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD246 (103246)

Interner RC! Für die Taste ist in der Tastenbeschreibung keine Eintragung vorhanden.

IDHD249 (103249)

Das Format &SYS-PAR9 ist als Teilformat erzeugt worden. Teilformate werden durch den FHS-Dialogmanager nicht unterstützt. Korrigieren Sie das Format!

IDHD250 (103250)

Das Standardformat &SYS-PAR9 ist intern falsch aufgebaut. Diagnose: Fehler wegen &SYS-PAR3. Erzeugen Sie das Format neu!

IDHD251 (103251)

Interner Fehler! Diagnose: IDHRKEY Fehler in der Fehlerbehandlung (&SYS-PAR9) Bitte informieren Sie Ihren SNI-Betreuer!

IDHD252 (103252)

Das Format &SYS-PAR0 kann nicht angezeigt werden, da das Standardformat für die Tastenbelegung IDHKEYA nicht geladen werden kann. Bitte benachrichtigen Sie den Anwendungsbetreuer!

IDHD253 (103253)

Interner Fehler! Diagnose: IDHRKEY der im STE eingetragene Boxentyp ist unbekannt. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD271 (103271)

Es soll das Format &SYS-PAR9 angezeigt werden. Der CCSNAME &SYS-PAR4 ist verschieden vom CCS-Namen des Vollbildes (&SYS-PAR5). Informieren Sie den Anwendungsentwickler!

IDHD272 (103272)

Interner Fehler! Diagnose: Dem Ladedienst wurde kein gültiger Modulname übergeben. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD273 (103273)

Interner Fehler! Diagnose: Falscher Parameter zum Ladedienst! Bitte informieren Sie Ihren SNI-Betreuer!

IDHD275 (103275)

Das Format &SYS-PAR9 wurde während der Anzeige durch eine neue Version ersetzt. (&SYS-PAR4 -> &SYS-PAR5).

IDHD276 (103276)

Im Format &SYS-PAR9 ist eine unbekannte Bildschirmgruppe eingetragen. Erzeugen Sie das Format neu!

IDHD277 (103277)

Das Format &SYS-PAR9 kann an dieser Datensichtstation nicht angezeigt werden. Es wurde für eine Bildschirmgruppe hergestellt, die die aktuelle Datensichtstation nicht enthält.

IDHD278 (103278)

Im Format &SYS-PAR9 ist eine Version angezeigt, die mit dem aktuellen FHS-DM nicht verarbeitet werden kann. Diagnose: Falsche Version im MDBE7 Bitte informieren Sie Ihren Anwendungsbetreuer!

IDHD279 (103279)

Das Format &SYS-PAR9 soll als Maskenbeschreibungsformat genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD280 (103280)

Das Format &SYS-PAR9 soll als Masken-Layout-Beschreibung genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD281 (103281)

Es soll ein Format zum Masken-Layout geladen werden (Dialogerweiterung, keine Hilfe). Das gefundene Format &SYS-PAR9 hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht nicht dem geforderten Typ. Informieren Sie den Anwendungsentwickler!

IDHD282 (103282)

Das Format &SYS-PAR9 soll als Hilfeformat genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD283 (103283)

Das Format &SYS-PAR9 soll als Tastenbeschreibungsformat genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD284 (103284)

Das Format &SYS-PAR9 soll als Meldungsformat genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD285 (103285)

Das Format &SYS-PAR9 soll als ICE-Zeichensatzformat genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD286 (103286)

Das Format &SYS-PAR9 soll als Rahmenbeschreibung genutzt werden. Das gefundene Format hat den Typ "&SYS-PAR4" (siehe Hilfe) und entspricht so nicht dem geforderten Typ. Bitte informieren Sie Ihren Anwendungsentwickler!

IDHD301 (103301)

Es wurde ein ADDPOP-Dienst mit POPLOC(<name>)-Operand aufgerufen. Es existiert aber kein Bild, auf das eine Box ausgegeben werden kann.

IDHD302 (103302)

Im Operanden &SYS-PAR2 wird ein Feldname angefordert. Im Format &SYS-PAR0, das das Feld enthalten müsste, existieren keine Felder mit Namen.

IDHD303 (103303)

Der Name &SYS-PAR1 im Operanden &SYS-PAR2 wurde im Format &SYS-PAR0 nicht gefunden.

IDHD304 (103304)

Es soll durch ein REMPOP (ohne ALL-Operand) ein Boxniveau entfernt werden. Es existiert aber keine Box. Hinweis, durch Fehler in einem vorhergegangenen DISPLAY-Dienst kann das Bild entfernt worden sein.

IDHD305 (103305)

Interner Fehler! Diagnose: DMOPEN liefert Fehler. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD306 (103306)

Interner Fehler! Diagnose: Kein SCTCA (Verständigungsbereich für einen Dialogabschnitt) gefunden. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD307 (103307)

Ein ADDPOP-Dienst ist in einer falschen Reihenfolge angefordert worden (z.B. zwei ADDPOP nacheinander oder ADDPOP mit POPLOC ohne vorhergehenden Aufruf des DISPLAY-Dienstes).

IDHD308 (103308)

Ein REMPOP-Dienst ist in einer falschen Reihenfolge angefordert worden (z.B. ein REMPOP nach einem ADDPOP ohne vorhergehenden Aufruf des DISPLAY-Dienstes).

IDHD309 (103309)

Es wurde der Dienst "CONTROL DISPLAY RESTORE" aufgerufen. Es existiert jedoch kein "gerettetes" Bild.

IDHD311 (103311)

Interner Fehler! Diagnose: Fehler bei der Cursorpositionierung nach der Listenverarbeitung. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD401 (103401)

Warnung! Bei der Meldungsverarbeitung trat ein Fehler auf. Dieser wurde ignoriert und eine Ersatzmeldung ausgegeben.

IDHD411 (103411)

Die Position der Box &SYS-PAR1 wird durch die Anwendung festgelegt. An die geforderte Position passt die Box nicht!

IDHD412 (103412)

Die Box &SYS-PAR1 ist grösser als das Bildschirmformat.

IDHD413 (103413)

Interner Fehler! Diagnose: Bei der Erzeugung einer Meldungsbox wurde kein MSGCB gefunden. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD414 (103414)

Interner Fehler! Diagnose: Interne Namenstabelle nicht vorhanden. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD415 (103415)

Interner Fehler! Diagnose: Name nicht gefunden. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD501 (103501)

Implementierungsbedingte Einschränkung! Die Attribute von zu vielen verschiedenen Feldern sollen dynamisch geändert werden.

IDHD502 (103502)

Implementierungsbedingte Einschränkung! Die Zeichenlänge der Feldnamen, deren Attribute dynamisch geändert werden sollen, passt nicht in die interne Namenstabelle.

IDHD800 (103800)

RESHOW interner Rückkehrcode.

IDHD801 (103801)

Die Eingabe von der Datensichtstation kann nicht verarbeitet werden. Auch nach einer Wiederholung der Eingabe trat der Fehler auf. Ein interner oder technischer Fehler ist nicht auszuschliessen. Informieren Sie den Anwendungsentwickler!

IDHD901 (103901)

Interner Fehler! Diagnose: Dem CONTROL-Dienst ist ein falscher Steuerblock übergeben worden. Bitte informieren Sie Ihren SNI-Betreuer.

IDHD916 (103916)

Interner Fehler! Diagnose: Bitte informieren Sie Ihren SNI-Betreuer!

IDHD917 (103917)

Interner Fehler! Diagnose: Der Anzeigedienst wurde mit einem falschen OP-Code aufgerufen. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD918 (103918)

Interner Fehler! Diagnose: Kein Speicherelement. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD919 (103919)

Implementierungsbedingte Einschränkung! Überlauf im internen Stapel. Informieren Sie den Anwendungsentwickler!

IDHD920 (103920)

Interner Fehler! Diagnose: Bitte informieren Sie Ihren SNI-Betreuer!

IDHD921 (103921)

Interner Fehler! Diagnose: Speicherüberlauf in &SYS-PAR4. Bitte informieren Sie Ihren SNI-Betreuer!

IDHD999 (103999)

Ein Fehler in der Fehlerbehandlung ist aufgetreten. Eine Weiterarbeit ist nicht möglich. Bitte informieren Sie den Anwendungsentwickler!

IDHF000

Fehler in der Fehlerbehandlung

IDHF001

soll die Meldung mit dem Meldungsschlüssel &SYS-PAR1 ausgegeben werden. Im Bibliothekselement, wo die Meldung erwartet wird, kann die Meldung nicht gefunden werden. Bitte informieren Sie den Anwendungsentwickler! Weiter mit ENTER!

IDHF002

Es soll die Meldung mit dem Meldungsschlüssel "&SYS-PAR1" ausgegeben werden. Der Meldungsschlüssel ist syntaktisch falsch. Informieren Sie den Anwendungsentwickler! Fortsetzung durch Eingabetaste.

IDHF003

Es soll die Meldung mit dem Schlüssel "&SYS-PAR1" ausgegeben werden. Das Bibliothekselement in dem die Meldung enthalten sein soll, kann nicht geladen werden. Bitte informieren Sie den Anwendungsentwickler! Weiter mit ENTER!

IDHF004

Die Angaben in der TOPINDEX-Variablen (&SYS-PAR5) und der Indexangabe in CURPOS (&SYS_PAR7) widersprechen sich. Bitte informieren Sie den Anwendungsentwickler! Weiter mit ENTER!

IDHF005

Bei einer impliziten Aktion im Anzeigedienst des FHS-DM ist ein Fehler aufgetreten. Interner Rückkehrcode: &SYS-PAR6, MSGID: &SYS-PAR5. Informieren Sie den Anwendungsentwickler! Weiter mit ENTER!

IDHF006

Bei einer impliziten Aktion im Anzeigedienst des FHS-DM ist ein Fehler aufgetreten. Interner Rückkehrcode: &SYS-PAR6. Informieren Sie den Anwendungsentwickler! Weiter mit ENTER!

IDHF007

Warnung! Beim Lesen der Steuervariablen &SYS-PAR1 für das Auswahlfeld trat der RC=&SYS-PAR3 auf. Der Fehler wird ignoriert. Bitte informieren Sie den Anwendungsentwickler!

IDHF010

Warnung! Es existieren in der Maske &SYS-PAR0 Felder, die nicht mit der Anwendung verbunden sind (Felder ohne Namen). Das erste Feld befindet sich an der Position &SYS-PAR3! Weitere Informationen erhalten Sie in der Hilfe zur Meldung!

IDHF011

Es liegt keine Meldung vor!

IDHF180

In einer Hilfe- oder Meldungsbox wurde ein unbekanntes Kommando eingegeben.

IDHF182

Das Kommando SYS ist in dieser Programmumgebung nicht unterstützt.

IDHF184

Das Kommando PANELID enthält einen Syntaxfehler.

IDHF185

Das Kommando KEYAREA enthält einen Syntaxfehler.

IDHF186

Das Kommando SETP enthält einen Syntaxfehler.

IDHF187

Das eingegebene Verschiebekommando ist nicht erlaubt.

IDHF188

Es existiert keine Hilfe.

IDHF191

Das Hilfeformat &SYS-PAR1 ist nicht vorhanden oder das Format ist nicht als Hilfeformat erstellt worden.

IDHF192

Der angegebene Wert ist für eine Auswahl nicht vorgesehen. Deshalb kann keine Hilfe für einen Eintrag gefunden werden. Löschen Sie das Eingabefeld oder tragen Sie einen gültigen Wert ein. Danach können Sie weitere Hilfe anfordern!

IDHF193

Die Schreibmarke befindet sich in der Menüleiste, sie steht jedoch auf keinem Menütitel. Positionieren Sie die Schreibmarke auf einen Menütitel oder auf eine geeignete Position außerhalb der Menüleiste.

IDHF194

Das durch Drücken einer Funktionstaste ausgelöste Kommando ist nicht erlaubt, wenn sich die Schreibmarke in der Menüleiste befindet.

IDHF198

Das INDEX-Format &SYS-PAR1 ist nicht vorhanden.

IDHP001 (106001)

Die Steuervariable ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP002 (106002)

Die Steuervariable "SERVICE" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP003 (106003)

Die Steuervariable "RESOURCE" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP004 (106004)

Die Steuervariable "MESSAGE-ID" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP005 (106005)

Die Steuervariable "MESSAGE-FIELD" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP006 (106006)

Der SDF-P-Name in der Steuervariablen "MESSAGE-FIELD" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP007 (106007)

Die Steuervariable "MSG-FIELD-IND" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP008 (106008)

Die Steuervariable "CURSOR-OUTPUT" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP009 (106009)

Der SDF-P-Name in der Steuervariablen "CURSOR-OUTPUT" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP010 (106010)

Die Steuervariable "CURSOR-OUTPUT-INDEX" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP011 (106011)

Die Steuervariable "CURSOR-OUTPUT-POS" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP012 (106012)

Die Steuervariable "LOCK" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP013 (106013)

Die Steuervariable "ALARM" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP014 (106014)

Die Steuervariable "AUTOTAB" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP015 (106015)

Die Steuervariable "MANDATORY" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP016 (106016)

Die Steuervariable "HARDCOPY" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP017 (106017)

Die Steuervariable "ROW" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP018 (106018)

Die Steuervariable "COLUMN" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP019 (106019)

Die Steuervariable "POP-LOCATION" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP020 (106020)

Der SDF-P-Name in der Steuervariablen "POP-LOCATION" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP021 (106021)

Die Steuervariable "POP-LOC-IND" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP022 (106022)

Die Steuervariable "ATTR.FIELD" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP023 (106023)

Die Steuervariable "ATTR.FIELD-INDEX" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP024 (106024)

Die Steuervariable "ATTR.TYPE" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP025 (106025)

Die Steuervariable "ATTR.HILITE" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP026 (106026)

Die Steuervariable "ATTR.INTENSITY" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP027 (106027)

Die Steuervariable "ATTR.COLOR" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP028 (106028)

Die Steuervariable "ATTR.OUTPUT" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP029 (106029)

Die Steuervariable "REFRESH" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP030 (106030)

Die Steuervariable "ACK" ist ungültig. Bitte informieren Sie den Anwendungsentwickler!

IDHP031 (106031)

Die FHSTABLE ist ungültig wegen eines internen Fehlers. Bitte informieren Sie den Systemverwalter!

IDHP032 (106032)

Im Server-Info-Parameter ist die Bibliothek falsch angegeben. Bitte korrigieren Sie das ASSIGN-Kommando!

IDHP033 (106033)

Die Information im Server-Info-Parameter ist ungültig. Bitte korrigieren Sie das ASSIGN-Kommando!

IDHP034 (106034)

Die Kontroll-Variable ATTR ist keine Liste. Bitte informieren Sie den Anwendungsentwickler!

IDHT000

Kein Fehler! Alles korrekt!

IDHT009

In dieses Feld muß eine Eingabe erfolgen.

IDHT010

Das Feld enthält mindestens ein falsches Zeichen.

IDHT011

Der eingegebene Wert ist kleiner als zulässig. Die Grenzwerte sind: "&SYS-PAR3"

IDHT012

Der eingegebene Wert ist grösser als zulässig. Die Grenzwerte sind: "&SYS-PAR3"

IDHT013

Es dürfen nur Ziffern und als letztes Zeichen ein Vorzeichen eingegeben werden.
Das Vorzeichen kann entfallen.

IDHT014

Für dieses Feld sind nur folgende Werte zulässig: "&SYS-PAR3"

IDHT015

Für dieses Feld sind folgende Werte nicht zulässig: "&SYS-PAR3"

IDHT017

Die eingegebene Zeichenkette enthält mindestens ein unzulässiges Zeichen. Für dieses Feld sind nur folgende Zeichen zulässig: "&SYS-PAR3"

IDHT018

Die eingegebene Zeichenkette enthält mindestens ein unzulässiges Zeichen und ist deshalb falsch. Folgende Zeichen dürfen nicht eingegeben werden: "&SYS-PAR3"

IDHT020

Das Feld enthält zuwenige relevante Zeichen.

IDHT030

Es tritt ein nicht näher bestimmbarer Fehler im Datum oder in der Uhrzeit auf, z.B. Trennzeichen am Feldende, zuwenig Zeichen eingegeben.

IDHT031

Das eingegebene Datum liegt außerhalb des zulässigen Bereiches, d.h. das Datum ist kleiner als 1582-10-15 oder grösser als 2099-12-31.

IDHT032

Im Datum ist der Tag falsch eingegeben. A

- IDHT033
Im Datum ist der Monat falsch eingegeben.
- IDHT034
Im Datum ist das Jahr falsch eingegeben.
- IDHT035
Das Trennzeichen im Datum /Uhrzeit ist falsch oder es fehlt. Erwartetes Trennzeichen:
"&SYS-PAR3"
- IDHT037
Die Stundenangabe in der Uhrzeit ist falsch.
- IDHT038
Die Minutenangabe in der Uhrzeit ist falsch.
- IDHT039
Die Sekundenangabe in der Uhrzeit ist falsch.
- IDHT040
Der eingegebene Wert ist zu groß. Bitte beachten Sie, daß je nach Definition des Feldes Platz für Dezimalstellen, Tausendertrennzeichen oder Vorzeichen benötigt wird.
Definierte Schablone: "&SYS-PAR3"
- IDHT050
In ein arithmetisches Feld wurden zu viele Dezimalstellen eingegeben. Definierte Schablone: "&SYS-PAR3"
- IDHT060
Das arithmetische Feld darf kein Vorzeichen enthalten.
- IDHT061
In ein arithmetisches Feld wurde mehr als ein Vorzeichen eingegeben.
- IDHT063
In das arithmetische Feld wurden nur Vorzeichen eingegeben.
- IDHT070
Das arithmetische Feld darf keine Zifferntrennzeichen enthalten. Definierte Schablone:
"&SYS-PAR3"
- IDHT071
In einem arithmetischen Feld tritt ein Fehler bei der Zifferngruppierung auf.
- IDHT080
In das arithmetische Feld darf kein Dezimaltrennzeichen eingegeben werden.
Definierte Schablone: "&SYS-PAR3"
- IDHT090
In dem arithmetischen Feld tritt ein nicht näher bestimmbarer Fehler auf.
Definierte Schablone: "&SYS-PAR3"

IDHT092

Zum gegenwärtigen Zeitpunkt ist die gewünschte Auswahl gesperrt.

IDHT093

Der angegebene Wert ist für eine Auswahl nicht vorgesehen.

IDHT094

Es wurde ein falsches Auswahlzeichen eingegeben. Gültige Auswahlzeichen:
"&SYS-PAR3"

IDHU001 (204001)

Der DMCOMM ist ungültig. Bitte rufen Sie den Dienst mit einem gültigen DMCOMM erneut auf!

IDHU002 (204002)

FHS läuft nur mit einem VTSU-B ab Version 11.0. Bitte informieren Sie den Systemverwalter, damit er die geeignete VTSU-B-Version installiert.

IDHU003 (204003)

FHS läuft nur auf einem BS2000 ab Version 10.0. Bitte benutzen Sie die entsprechende BS2000-Version.

IDHU004 (204004)

Während der Installation des Laufzeitsystems trat ein Fehler auf. Bitte informieren Sie den Systemverwalter!

IDHU005 (204005)

Die Anzahl der DMOPEN ist zu groß. Es sind nur 29 DMOPEN möglich.
Bitte korrigieren Sie Ihr Programm.

IDHU006 (204006)

Die Adresse der Parameterliste ist nicht auf Wortgrenze ausgerichtet.
Bitte korrigieren Sie Ihr Programm.

IDHU007 (204007)

Der Bereich, der die Länge des Operandenbereiches enthält, ist nicht auf Wortgrenze ausgerichtet.. Bitte korrigieren Sie Ihr Programm.

IDHU008 (204008)

Der Längenbereich beginnt nicht an Wortgrenze. Bitte korrigieren Sie Ihr Programm!

IDHU009 (004009)

FHS-DOORS ist nicht verfügbar. Es erfolgt keine Anzeige. Bitte stellen Sie den Modul FHS-DOORS bereit. (IDHIO).

IDHU010 (204010)

Der DMCOMM ist bereits geschlossen. Bitte rufen Sie die Funktion mit einem gültigen DMCOMM erneut auf.

IDHV001 (007001)

Das Profilelement &SYS-PAR17 wurde nicht in der Bibliothek mit dem Dateikettungs-namen IDHPROF gefunden. Es wird ein leerer Profilpool erstellt.

IDHV002 (107002)

Das Profilelement &SYS-PAR17 kann nicht aus der Bibliothek mit dem Dateikettungs-namen IDHPROF gelesen werden. Der DMOPEN-Dienst bricht mit einem Rückkehrcode ab. Bitte korrigieren oder löschen Sie das Profilelement.

IDHV003 (107003)

Das Profilelement kann nicht in die Bibliothek mit dem Dateikettungs-namen IDHPROF geschrieben werden. Bitte prüfen Sie, ob die Bibliothek und das Element geändert werden dürfen.

IDHV011 (107011)

Interner Fehler. ILAM-Initialisierungsfehler &SYS-PAR12. Bitte informieren Sie Ihre Systemverwaltung oder -wartung.

IDHV012 (107012)

Es wird eine Profilbibliothek mit dem Dateikettungs-namen IDHPROF benötigt. Der Dateikettungsname IDHPROF ist nicht angegeben oder verweist auf eine falsche Bibliothek. Bitte überprüfen Sie die Namen!

IDHV016 (107016)

Interner Fehler. Falscher Variablenbehälter. Es sind nur PROFILE, TASK oder PROCEDURE unterstützt. Bitte informieren Sie Ihre Systemverwaltung oder -wartung!

IDHV031 (107031)

Die Variable &SYS-PAR13 ist bereits als explizite Variable definiert. Sie kann nicht noch einmal definiert werden. Bitte informieren Sie den Anwendungsentwickler!

IDHV032 (107032)

Die Variable &SYS-PAR13 hat für dieses Format eine falsche Länge. Bitte informieren Sie den Anwendungsentwickler!

IDHV101 (107101)

Fehler bei der Konvertierung der Variablen &SYS-PAR13. Bitte prüfen Sie die Variablendefinition. Diagnose: Fehler bei der Konvertierung vom Typ &SYS-PAR14 in &SYS-PAR15; Wert &SYS-PAR16, von Länge &SYS-PAR17 in Länge &SYS-PAR18

IDHV102 (107102)

Die Variable &SYS-PAR13 ist eine implizite Variable. Sie darf deshalb kein Array sein. Bitte informieren Sie den Anwendungsentwickler!

IDHV103 (107103)

Die explizite Variable ist kein Array. Sie darf deshalb nicht mit einem Index angegeben werden. Bitte informieren Sie den Anwendungsentwickler!

IDHV104 (107104)

Interner Fehler bei der Arbeit eines Variablendienste (VAS). Beim Aufruf der VAS-Funktion &SYS-PAR11 trat der Rückkehrcode &SYS-PAR12 auf. Bitte informieren Sie Ihre Systemverwaltung oder -wartung!

IDHV106 (107106)

Die Systemvariable &SYS-PAR13 darf nicht geändert werden.

IDHV107 (107107)

Interner Fehler. Falsche Pool-Angabe bei einem Variablenzugriff. Bitte informieren Sie Ihre Systemverwaltung oder -wartung!

IDHV111 (107111)

Interner Fehler. Beim Zugriff auf die Variable &SYS-PAR13 trat ein Fehler auf, weil die Variable mit einem nicht unterstützten VAS-Typ übertragen werden soll. Bitte informieren Sie Ihre Systemverwaltung oder -wartung!

IDHV112 (107112)

Interner Fehler. Beim Zugriff auf die Variable &SYS-PAR13 trat ein Fehler auf, weil die Variable in einen nicht unterstützten VAS-Typ übertragen werden soll. Bitteinformieren Sie Ihre Systemverwaltung oder -wartung!

IDHV113 (107113)

Die Funktion kann nicht ausgeführt werden, da die Variable &SYS-PAR13 nicht existiert. Bitte informieren Sie den Anwendungsentwickler!

IDHV114 (107114)

Die Funktion kann nicht ausgeführt werden, da die S-Variable nicht existiert. Bitte informieren Sie den Anwendungsentwickler!

IDHV115 (107115)

Der Zugriff auf die S-Variable &SYS-PAR13 ist falsch, weil die Variable in einen nicht unterstützten S-Variablentyp übertragen werden soll. (Nur Zeichenketten und Ganzzahlen zulässig) Informieren Sie den Anwendungsentwickler!

IDHV117 (007117)

Die Dialogvariable &SYS-PAR13 existiert nicht. Bitte informieren Sie den Anwendungsentwickler!

IDHV116 (00711)

Interner Fehler. Falscher Variablenbehälter. Es werden nur PROFILE, TASK oder PROCEDURE unterstützt. Bitteinformieren Sie Ihre Systemverwaltung oder -wartung!

IDHV118 (007118)

Die SDF-P-Variable &SYS-PAR13 existiert nicht. Bitte informieren Sie den Anwendungsentwickler!

IDHV120 (007120)

Die Variable &SYS-PAR13 hat eine Dimension &SYS-PAR16. Der Zugriff ist mit einem Index, der grösser als dieser Wert ist, nicht möglich. Bitte informieren Sie den Anwendungsentwickler!

IDHV121 (007121)

Verkürzung der Daten bei der Konvertierung der Variablen &SYS-PAR13 vom Typ &SYS-PAR14 in &SYS-PAR15; Wert &SYS-PAR16, von Länge &SYS-PAR17 in Länge &SYS-PAR18. Bitte informieren Sie den Anwendungsentwickler!

IDHV126 (107126)

Die S-Variable &SYSPAR13 kann nicht geschrieben werden. PUTVAR-Fehler &SYSPAR15. Bitte informieren Sie den Anwendungsentwickler!

IDHV127 (107127)

Die S-Variable &SYSPAR13 kann nicht gelesen werden. GETVAR-Fehler &SYSPAR15. Bitte informieren Sie den Anwendungsentwickler!

IDHV131 (107131)

Um die Variable &SYS-PAR13 in einer Liste anzeigen zu können, muß sie durch SDF-P bereits als Liste erzeugt worden sein. Bitte informieren Sie den Anwendungsentwickler!

IDHV132 (107132)

Um die Variable &SYS-PAR13 anzeigen zu können, muß sie eine einfache S-Variable vom Typ Zeichenkette oder Ganzzahl sein. Bitte informieren Sie den Anwendungsentwickler!

IDHV133 (107133)

Interner Fehler. Die Variable &SYS-PAR13 soll mit einem Variablentyp gelesen werden, der nicht unterstützt ist. Bitte informieren Sie Ihre Systemverwaltung oder -wartung!

IDHV134 (107134)

Die Erweiterung einer Liste von S-Variablen ist in dieser Version des FHS nicht unterstützt. Bitte informieren Sie den Anwendungsentwickler!

IDHZ001 (101001)

Das Format &SYS-PAR18 kann nicht geladen werden. Bitte prüfen Sie die Hierarchie der Formatbibliotheken! (Rückkehrcode &SYS-PAR15 von BIND oder \$PBBND1)

IDHZ002 (001002)

Das Format &SYS-PAR18 kann nicht geladen werden. Bitte prüfen Sie die Hierarchie der Formatbibliotheken! (Rückkehrcode &SYS-PAR15 von UNBIND oder \$PBUNL1)

IDHZ003 (001003)

Interne Warnung. Das Format &SYS-PAR18 kann nicht geladen werden. Das Format wird in der internen Tabelle nicht gefunden.

IDHZ004 (101004)

Interner Fehler. Es wurde eine falsche Aktion vom Ladedienst angefordert.

IDHZ005 (101005)

Die maximale Anzahl von gleichzeitig benutzbaren BLS-Kontexten wurde erreicht. Hinweis: Jeder Dialogkomplex ("DMOPEN INIT") ist mit einem BLS-Kontext verbunden. Bitte informieren Sie den Anwendungsentwickler!

IDHZ006 (101007)

Interner Fehler. Fehler wurde vom Ladedienst angezeigt. Rückkehrcode &SYS-PAR15 von (\$)VSVI1 für das Format &SYS-PAR18.

IDHZ007 (101007)

Das Format &SYS-PAR18 wurde richtig geladen, aber in der Struktur des Formates wurde ein Fehler angezeigt. Bitte prüfen Sie das Format.

IDHZ011 (201011)

Es darf kein zusätzlicher Speicherplatz angefordert werden. Rückkehrcode &SYS-PAR18 vom Systemruf &SYS-PAR15. Sie haben bereits Ihren gesamten verfügbaren Speicherplatz angefordert. Bitte wenden Sie sich an den Systemverwalter!

IDHZ012 (201012)

Es darf kein zusätzlicher Speicherplatz angefordert werden. Rückkehrcode &SYS-PAR18 vom Systemaufruf &SYS-PAR15.

IDHZ013 (101013)

Interner Fehler. Ein Speicherplatz mit dieser ID wurde nie angefordert und kann deshalb auch nicht freigegeben werden.

IDHZ014 (101014)

Interner Fehler. Die Speicherplatzverwaltung wurde von einer unbekanntenen Aufgabe gerufen.

IDHZ015 (101015)

Interner Fehler. Fehler in der Fehlerbehandlung der Speicherverwaltung. Rückkehrcode &SYS-PAR18 vom Systemruf &SYS-PAR15.

IDHZ016 (001016)

Fehler bei der Freigabe von Speicherplatz. Rückkehrcode &SYS-PAR18 vom Systemruf &SYS-PAR15.

7.3 Übersicht: Systemvariable des Dialogmanagers

Alle Systemvariablen sind vom Zeichenkettenformat. Die Systemvariable sind in verschiedenen Pools enthalten. Es bedeuten:

- A (aktuell ermittelt)
- F - Funktionspool
- P - Profilpool
- S - Systempool (nur Lesen erlaubt).
- I - interne Diagnoseinformation

allgemeine Systemvariable (für Datum und Uhrzeit)

Name	Pool	Typ	Länge	Bedeutung
SYS-DATE	A	CHAR	12	aktuelles Datum in der ISO-Form YY-MM-DDiii b 1)
SYS-STDDATE	A	CHAR	14	aktuelles Datum in der ISO4-Form YYYY-MM-DDiii b 1)
SYS-TIME	A	CHAR	5	aktuelle Tageszeit einer 24-Stunden-Uhr im Format HH:MM 1)
SYS-STDTIME	A	CHAR	8	aktuelle Tageszeit einer 24-Stunden-Uhr im Format HH:MM:SS 1)
SYS-DAY	A	CHAR	2	Tag des Monats (2 Ziffern)
SYS-MONTH	A	CHAR	2	Monat (2 Ziffern)
SYS-YEAR	A	CHAR	2	Jahreszahl (2 Ziffern)
SYS-STDYEAR	A	CHAR	4	Jahreszahl (4 Ziffern)

- 1) Großbuchstaben in den beschriebenen Formaten sind jeweils ein Zeichen der folgenden Werte:

- Y Jahr
- M Monat in Datumsangabe
- D Tag
- H Stunde
- M Minute in Zeitangaben
- S Sekunde
- iii laufender Tag des Jahres
- b Leerzeichen

Systemvariable des Dialogmanagers

Name	Pool	Typ	Länge	Bedeutung
SYS-COMMAND	F	CHAR	255	Anwendungskommando, das vom Anwendungsprogramm verarbeitet werden kann
SYS-FHS-VERSION	S	CHAR	6	Version des Dialogmanagers
SYS-P-KEYS-SETTING	P	CHAR	24	Definition der P-Tasten Zuordnung. Der Wert kann mittels des Systemkommandos SETP geändert werden
SYS-KEY-AREA	P	CHAR	1	Art der Anzeige der Tastenbelegung - 0: Tastenbelegung anzeigen (Standard) - 1: Tastenbelegung nicht anzeigen. Der Wert kann mittels des Systemkommandos KEYAREA geändert werden
SYS-PANEL-ID	P	CHAR	1	Art der Anzeige des Formatnamens und des Meldungsschlüssels. - 0: Formatname und Meldungsschlüssel nicht anzeigen - 1: Formatname und Meldungsschlüssel anzeigen Der Wert kann mittels des Systemkommandos PANELID geändert werden

Die Profilvariablen werden beim DMOPEN mit INIT im Profilpool gesucht. Sind sie nicht vorhanden, wird ein Standard erzeugt. Beim DMCLOSE für den Dialogkomplex werden diese Variablen zurückgeschrieben. Der Inhalt dieser Variablen kann nur durch DM-Systemkommandos verändert werden.

Die Systemvariablen der folgenden Tabelle haben erst dann den angegebenen Wert, wenn die Anzeige des Formates beendet wird.

Name	Pool	Typ	Länge	Bedeutung
SYS-CURSOR-FIELD	F	CHAR	255	Name des Maskenfeldes, das den Cursor enthält oder "\$zzz#sss", falls sich der Cursor auf einem unbenannten Feld oder in einem Feldzwischenraum befindet
SYS-CURSOR-INDEX	F	FIXEDS	2/4	Index der Listenzeile einer angezeigten Liste, die den Cursor enthält. Der Wert ist nur dann gültig, wenn SYS-CURSOR-FIELD den Namen eines Listenfeldes enthält
SYS-CURSOR-POS	F	FIXEDS	2/4	Position des Cursors in dem Feld, dessen Name in SYS-CURSOR-FIELD steht
SYS-CCS-NAME	F	CHAR	8	CCSNAME, der durch den DISPLAY-Dienst für die Anzeige des letzten Formates verwendet wurde

Diagnosevariable des Dialogmanagers

Name	Pool	Typ	Länge	Bedeutung
SYS-PARxx	I	CHAR	<80	xx = 0 - 20 0 - 9: Diagnose in den Anzeigediensten 10 - 20: Diagnose in den Variablendiensten interne Diagnoseinformationen, die in Meldungen verwendet werden
SYS-PAR0	I	CHAR	8	Name des aktuellen expliziten Formates
SYS-PAR9	I	CHAR	8	Name des aktiven Formates

Bei Feldprüfungen werden folgende Informationen in den Info-Variablen abgelegt. Auf diese Angaben kann in privaten Meldungen, die bei IFG festgelegt werden können, zugegriffen werden.

- SYS-PAR1 Name des aktuellen Feldes
- SYS-PAR2 Inhalt des aktuellen Feldes
- SYS-PAR3 Zusatzinformation Editfehler: EDIT-Fehlernummer
Prüfung auf CCS-Name: CCS-Name

Aus Kompatibilitätsgründen zu FHS-UTM werden die Namen ZPAR1, ZPAR2 und ZPAR3 auf SYS-PAR1, SYS-PAR2 und SYS-PAR3 abgebildet. Diese Namen sollten nicht weiter verwendet werden.

Fachwörter

ADDPOP

Erzeugen von Dialogboxen.

Attribut

Eigenschaft eines Formates oder Feldes zur Darstellung, Aufbereitung oder Prüfung. Ein Attribut wird entweder bei der Formaterstellung mit dem IFG festgelegt (statisches Attribut) oder im Anwendungsprogramm über die Global- und Feldattribute (dynamische Attribute).

Ausgabefeld

Feld, in das vom Anwendungsprogramm Daten ausgegeben werden.

Ausschlußzeichen

Zeichen am Bildschirm, kennzeichnet eine gesperrte Auswahl eines Auswahlfeldes.

Bildschirmwiederanlauf

Erneute Ausgabe des letzten, vollständig formatierten Bildschirms nach einer Unterbrechung.

Box

Kurzform von Dialog-Box, siehe dort.

DE-Format

Format, das die Funktionen der Dialogerweiterung nutzen kann. Die Eigenschaft "DE-Format" muß beim Erzeugen mit dem IFG explizit angegeben werden.

Dialog-Box

Rechteckiger Rahmen am Bildschirm, der ein DE-Format enthält.

Dialogerweiterung

Bestandteil von FHS, mit dem Alpha-Styleguide konforme Formate am Bildschirm angezeigt werden können. Die Dialogerweiterung ermöglicht u.a. mehrstufige Zwischendialoge, Eingabe von Kommandos, erweiterte Prüfungen der Eingaben und anwendungsspezifische Hilfesystem und Meldungen.

Dialogvariable

Dialogvariable dienen dem Datenaustausch zwischen Anwendungsprogramm, FHS-DM und Dialogelementen. Dialogvariable übernehmen die Funktion eines Datenübergabebereiches

Differenzausgabe

Ausgabe eines Formates, bei der nur die Felder an der Datensichtstation neu ausgegeben werden, die vom Anwendungsprogramm verändert werden.

Eingabefeld

Feld, in das vom Datenstationsbediener Daten für das Anwendungsprogramm eingegeben werden.

Explizite Box

Dialog-Box, deren Ausgabe durch die Anwendung veranlaßt wird.

FHS-DM

Format Handling System - Dialog Manager

Format

Logische Datenstruktur die ein "Formular" beschreibt.

Formateinsatzdatei

Bibliothek, in der die Formatbeschreibungen abgelegt sind.

Globale Hilfe

Hilfe für Objekte eines DE-Formates, die aus mehreren Bestandteilen bestehen wie z.B. Einfach- oder Mehrfachauswahlfelder.

Grundformat

Format, das durch die Anwendung ausgegeben wird und das durch Dialog-Boxen überlagert werden kann.

Hilfe-Box

Dialog-Box mit Hilfeinformationen, wird durch FHS ausgegeben.

Implizite Box

Dialog-Box, die von FHS ausgegeben wird, z.B. für Meldungen oder Hilfeinformationen.

KEY-Format

Format, das die Belegung der Funktionstasten enthält.

Kontrollbereich

Speicherbereich, in dem Formatierungsparameter und Rückmeldungen abgelegt werden.

Modale Box

Dialog-Box, die eine Eingabe des Benutzers erwartet. Die darunterliegende Box bzw. das darunterliegende Format ist dabei inaktiv.

Nichtmodale Box

Dialog-Box, die keine unmittelbare Aktion des Benutzers erwartet. Die darunterliegende Box bzw. das darunterliegende Format kann weiter bearbeitet werden.

REMPOP

Entfernen von Boxen, der vorhergehende Untergrund wird wieder angezeigt.

S-Variable

Variable für SDF-P.

Textfeld

Feld, das festen Text enthält, der bei der Formaterstellung festgelegt wird.

Zeichensatzdatei

Datei die Zeichensätze enthält, die mit dem ICE erstellt wurden.

Literatur

FHS - Formatierungssystem für UTM, TIAM, DCAM (TRANSDATA)

Benutzerhandbuch

IFG für FHS (TRANSDATA)

Benutzerhandbuch

ICE (TRANSDATA) Interaktiver Zeichensatzeditor

Benutzerhandbuch

FHS-DOORS (SINIX, BS2000)

Benutzerhandbuch

SDF-P V2.0 (BS2000/OSD)

Benutzerhandbuch

RPG3 (BS2000)

RPG-Compiler

Benutzerhandbuch

TIAM (TRANSDATA, BS2000)

Benutzerhandbuch

BS2000/OSD-BC V1.0

Makroaufrufe an den Ablaufteil

Benutzerhandbuch

TRANSVIEW-NMA (PDN)

TRANSVIEW-NMAE (PDN)

Netzmanagement und Meßdatenerfassung im PDN (TRANSDATA, PDN)

Kommandos

Benutzerhandbuch

TRANSVIEW-NMA (PDN)

TRANSVIEW-NMAE (PDN)

Netzmanagement und Meßdatenerfassung im PDN (TRANSDATA, PDN)

Funktionen und Leistungen

Benutzerhandbuch

Netzmanagement im BS2000 (TRANSDATA)

Benutzerhandbuch

BS2000/OSD-BC V1.0

Benutzer-Kommandos (SDF-Format)

Benutzerhandbuch

COBOL85 (BS2000)

COBOL-Compiler

Benutzerhandbuch

TRANSIT-CD (TRANSDATA)

Benutzerhandbuch

Datensichtstationen 9749, 9750, 9752 (TRANSDATA)

Schnittstelle für Programmierer

Benutzerhandbuch

BS2000 Datenkommunikationssystem

Technische Beschreibung

Alpha-Styleguide

Richtlinien zur Gestaltung von zeichenorientierten Benutzeroberflächen

Benutzerhandbuch

Bestellen von Handbüchern

Die aufgeführten Handbücher finden Sie mit ihren Bestellnummern im *Druckschriftenverzeichnis* der Siemens Nixdorf Informationssysteme AG. Neu erschienene Titel finden Sie in den *Druckschriften-Neuerscheinungen*.

Beide Veröffentlichungen erhalten Sie regelmäßig, wenn Sie in den entsprechenden Verteiler aufgenommen sind. Wenden Sie sich bitte hierfür an Ihre zuständige Geschäftsstelle. Dort können Sie auch die Handbücher bestellen.

Stichwörter

*ADDDPOP-DISPLAY 178
*ADD-POPUP 178
*DISPLAY 178
*REMOVE-POPUP 178
*REMPOP-DISPLAY 178

A

Ablauf des DISPLAY-Dienstes 137
ABORT 67
ACTIONS 29, 67
ADDDPOP 96, 126
Alphabetische Zeichenkette 50
ändern
 Feldattribute 183
Anwendung beenden 67
Anwendungsabschnitt beenden 69
Anwendungskommandos 63
Anzeige beenden/abbrechen 68
Anzeigedienste 96
anzeigen einer Meldung 137
anzeigen eines Formates 137
Anzeigen Tastenbelegung 83
Anzeigen von Begriffen zu FHS-DM 72
Arbeit mit Boxen 14
Arbeitsbereich 12
Arithmetische Zeichenkette 50
ASSIGN-STREAM 173, 174
ATTR 96, 129
Aufbau DM-Format 11
Aufbau einer Meldungsbox 18
Aufbereitung des Meldungstextes 85
Aufruf der Listenanzeige 39
Ausführen von BS2000-Kommandos 76
Ausgabefeld
 Hilfe 89

Ausgabeort einer Meldung 84
Ausrichtung und Füllzeichen 46
Auswahlfeld 25
 Hilfe 89

B

Beispiel
 Dialoganwendung 114
 FHS-DM als Subsystem 207
 graphischer Bibliotheksmanager 219
 parallele Dialoganwendung 115
Beispiel-Programm
 ASSEMBLER 204
 COBOL 201
Beispiel-Programme 201
Beispielprozeduren für FHS in SDF-P 212
Bibliotheken zuweisen 170
Bildschirmbreite
 Dialog-Box 13
Bildschirminhalt ausdrucken 70
Blätterkommandos 76
Blättern in einer Liste 34
Box
 explizit 14
 implizit 14, 17
 modal 13
 nichtmodal 13

C

CANCEL 68
CCSNAME 200
CCS-Name 25
coded character set 25
CONTROL 97, 134
CSRPOS 39, 93
CURSOR 39, 93
Cursor in Menüleiste positionieren 67
CURSOR-Operand 39
Cursorpositionierung 93

D

Darstellung der Uhrzeit 45
Darstellung des Datums 45
Datenaufbereitungen 45
Datenstationen 7
Datumumwandlung
 beim DISPLAY-Dienst 61
Datumumwandlungen 150, 168
Definition eines Listenbereiches 32
DE-Format 11
DE-Meldungen 84
Dezimaltrenner 45
DIAGINFO 178
Dialogbox entfernen 146
Dialog-Boxen 13
Dialogdienste 95, 96
 aufrufen 116
 Beschreibung 125
Dialogelemente 95, 96
Dialogerweiterung TIAM 95
Dialogkomplex 113
Dialogmanager 95
Dialogvariable 98
Dienste 111
Differenzausgabe 94
DISPLAY 96, 137
DISPLAY-Dienst 103
DMCLOSE 97, 112, 142
DMOPEN 97, 112, 144
dynamische Attribute definieren 129

E

eigene Meldungen 86
Einfachauswahlfeld 25
 globale Hilfe 90
einfache Listenzeilen 34
Eingabe in Boxen 14
Eingabeeigenschaften 49
Eingabeformatierung 186
Eingabepflicht 49
Endezeile 32
Entfernen
 Dialogbox 146
 explizite Box 15

- implizite Box 15
- Erstellen einer Hilfe-Box 19
- Erweiterte Hilfe 89
 - anfordern 70
 - zum Format 89
- EXIT 69
- explizite Box 14
 - entfernen 15
- explizite Dialogvariablen 103
 - definieren 152
- explizite Meldungen 84
 - ausgeben 87
 - erstellen 84
- EXTHELP 70, 89

F

- Fehlerbehandlung 121
- Feldattribute ändern 183
- feldbezogene Hilfe 89
- feldbezogene Hilfe-Box 19
- FHS als Ausgabe-Server 173
- FHS-Dienste für SDF-P-Anwendungen 177
- FHS-DM als Subsystem 207
- FHS-PRIV 173
- FHS-Rückgabe-Werte 186
- Format 5
- Formateigenschaften bei der Ausgabe 181
- Formatname 12, 180
- Formatname anzeigen 73
- Formattitel 12
- Füllzeichenbehandlung 47
- Funktionspool 109
- Funktionspool freigeben 113
- Funktionstaste 63, 77

G

- geschachtelte S-Prozeduren 191
- geschachtelter DMOPEN 113
- globale Hilfen 90
- Größe einer Hilfe-Box 19

H
HARDCOPY 70
HELP 71, 89
HELPHelp 72, 93
Hilfe
 anfordern 71
 Einfachauswahlfeld 71
 für Ausgabefeld 89
 global 90
 zu FHS-Kommandos 92
 zu Meldungen 90
 zum Feld 89
 zum Format 89
 zum Hilfesystem 93
 zur Tastenbelegung 73, 92
Hilfe-Box
 erstellen 19
 feldbezogen 19
 Größe 19
Hilfeformate 88
Hilfesystem 88
 Übersicht 72

I
ICE 20
IDHKEYA 79, 80, 83
IDHKEYE 81
IDHKEYF 80
IDHKEYH 81
IDHKEYI 83
IDHKEYK 82
IDHKEYM 82
IDHKEYN 82
IDHKEYS 79
IDHMBDR 22
IDHTKHP 92
IDHx 91
IFG 2
implizite Box 14, 17
 entfernen 15
 Position 17
Implizite Dialogvariable 106
Implizite Meldungen 84, 86
INDEX 72, 91

Information über Anzeigedienste 178
INIT 112
Interaktiver Formatgenerator 2
Interaktiver Zeichensatzgenerator 20
Interface-Routine ISPCI 116
ISPCI 116

K

K3 74
KEYAREA 72, 83
KEY-Formate 77
KEYSHELP 73, 83, 92
Kommandobereich 12, 63
 globale Hilfe 90
Kommandofeld 12
Kommandos über Funktionstasten 64
Kommandozeile 12
Kompatibilität 171
Komponenten des Dialogmanagers 95
Kopieren von Dialogvariablen 162

L

Layout und Version 177
Listenanzeige 32
 beenden 41
Listenbereich 32
Listenfeld 32
Listentitel 32
Listenverarbeitung 187
Listenzeile 32
Löschen von Dialogvariablen
 im Funktionspool 159
 im Profilpool 161

M

Mehrfachauswahlfeld 25
 globale Hilfe 90
mehrfacher DMOPEN 113
mehrstufige Zwischendialoge 13
mehrzeilige Listenzeilen 34
Meldebereich 12
Meldungen 84, 180
 erstellen 84
Meldungsbox 18

Meldungskennzeichen anzeigen 73
Meldungsschlüssel 84
Meldungstyp 84
Menüleiste 12, 29
Menütitel 29
MESSAGE IDENTIFICATION 84
Minimale Eingabelänge 49
modale Box 13
Modellzeile 32
modifizierte Listenzeilen 40
MODINDEX-Variable 38

N

Namensbildung Dialogvariablen 101
Namenskonventionen für Listenverarbeitung 187
nichtmodale Box 13
NO AUTOTAB 89
NUMROW-Variable 36

O

Operandenbereich 122

P

PANELID 73
Poolhierarchie 109
Pools 259
Position einer Dialogbox definieren 126
Position impliziter Boxen 17
Positionierung des Cursors 93
Profilpool 109
Programmaufbau 112
Programmvariable 98
Prozedurvariable 98
Prüfen von Listenfeldern 40
Prüfung auf Eingabepflicht 40
Prüfungen 50
P-Tasten zuordnen 75

Q

Querverweise 91

R

Rahmen einer Box 20
Rahmenzeichen 22
Regeln für Dialogboxen 16
REMPOP 96, 146
RESHOW 74
Return codes 186
RMSG 74

S

SDF-P-Prozedur 177
SDF-P-Schnittstelle 173
SDF-P-Variable 98, 172
SERVER-INFORMATION 174
SETP 75
Simulation einer F-Taste zurücknehmen 75
Spaltentitel 32
S-Prozeduren 189
 Beispiele 214
 geschachtelt 191
Standardbelegung von F-Tasten 83
Standard-Header 197
Standard-Hilfeformate 91
Standard-Tastenbelegungstabellen 78
Statusbereich 12
Steuerung der Listenanzeige 36
Steuerungsdienste 96, 97
Steuervariablen 36
S-Variablen 98
 ausgeben 189
 erzeugen 189
 in FHS-TIAM-Programmen 189
SYS 76, 108
SYSFHS-CONTROL 176, 193
Systemkommandos 63
Systemvariable 108

T

Taste K3 74
Tastenbelegung
 anzeigen 83
 ein/ausschalten 72
Tastenbelegungstabelle 63, 77
Tastenzuordnungen 77
Teilformat, FHS-DE 11
Textfelder 32
Titel des Formats 12
TOPINDEX, Regeln 37
TOPINDEX-Variable 37
TRANSMIT-BY-STREAM 173, 175
Typ einer Meldung 84

U

Übersicht über das Hilfesystem anfordern 72
Unterschiede FHS-DM und FHS-PRIV 200
Update-Ausgabe 94

V

Variablendienste 96, 98
Variablenpool 99, 109
Variablensubstitution 123
VCOPY 96, 147
VCOPY-Dienst 106
VDEFINE 96, 152
VDEFINE-Dienst 103
VDELETE 96, 159
Verarbeitungsoptionen vereinbaren 134
VERASE 96, 161
verdeckte Felder 14
Verschiebekommandos 76
Verständigungsbereich 118
VGET 96, 162
Vollformat, FHS-DE 11
VPUT 96, 164
VREPLACE 96, 166
VREPLACE-Dienst 106

W

Wiederanzeige

einer Maske 74

einer Meldung 74

Z

Zeichenkette

alphabetisch 50

arithmetisch 50

Zeichensatz für Rahmen 20

Zifferntrenner 45

Zuordnen P-Tasten zu F-Tasten 75

zusammengesetzte Kommandos 64

Zwischendialog 13

Inhalt

1	Einleitung	1
1.1	Zielsetzung und Zielgruppen des Handbuchs	3
1.2	Wegweiser durch das Handbuch	3
2	Einführung in FHS	5
3	Einführung in die Dialogerweiterung	11
3.1	Aufbau von DE-Formaten für den FHS-Dialogmanager	11
3.2	Datenaustausch mit Dialogvariablen	13
3.3	Dialog-Boxen	13
3.3.1	Explizite Boxen	15
3.3.2	Implizite Boxen	17
3.3.3	Meldungsboxen	18
3.3.4	Hilfe-Boxen	19
3.3.5	Rahmen einer Box	20
3.4	Formate mit CCS-Namen	25
3.5	Auswahlfelder	25
3.5.1	Einfachauswahlfeld	25
3.5.2	Mehrfachauswahlfeld	27
3.6	Menüleiste und Pull-Down-Menüs	29
3.7	Listenverarbeitung	32
3.8	Datenaufbereitung und Prüfung	42
3.8.1	Codetabellen	44
3.8.2	Datenaufbereitungen	45
3.8.3	Eingabeeigenschaften	49
3.8.4	Prüfungen für den Inhalt von Maskenfeldern	50
3.9	Kommandos	63
3.9.1	Systemkommandos des Dialogmanagers	66
	ABORT - Anwendung beenden	67
	ACTIONS - Cursor in Menüleiste positionieren	67
	CANCEL - Anzeige beenden/abbrechen	68
	EXIT - Anwendungsabschnitt beenden	69
	EXTHELP - Erweiterte Hilfe anfordern	70
	HARDCOPY - Bildschirminhalt auf Drucker ausgeben	70
	HELP - Hilfe anfordern	71
	HELPHELP - Übersicht über das Hilfesystem anfordern	72

	INDEX - Anzeigen von Begriffen zu FHS-DM	72
	KEYAREA - Tastenbelegung ein/ausschalten	72
	KEYSHELP - Hilfe zur Tastenbelegung	73
	PANELID - Formatnamen und Meldungskennzeichen anzeigen	73
	RESHOW - Wiederanzeige einer Maske	74
	RMSG - Wiederanzeige einer Meldung	74
	SETP - P-Tasten zuordnen	75
	Blätterkommandos	76
	SYS - Ausführen von BS2000-Kommandos	76
3.10	Tastenbelegungstabellen	77
3.11	Meldungen ausgeben	84
3.12	Hilfesystem	88
3.12.1	Hilfen, die der Anwendungsentwickler erstellen kann	88
3.12.2	Hilfen, die FHS selbst anbietet	91
3.13	Positionierung des Cursors	93
3.14	Differenzausgabe	94
4	Programm-Schnittstelle für TIAM-Anwendungsprogramme	95
4.1	Variablen des Dialogmanagers	98
4.1.1	Dialogvariable	98
4.1.1.1	Explizite Dialogvariablen	103
4.1.1.2	Implizite Dialogvariable	106
4.1.1.3	Regeln für Dialogvariable	107
4.1.1.4	Systemvariable des Dialogmanagers	108
4.1.1.5	Konventionen für Dialogvariablen in der Modellzeile (Listenverarbeitung)	108
4.2	Variablenpool	109
4.3	Programmaufbau einer Dialoganwendung	112
4.4	Dialogdienste aufrufen	116
4.4.1	Interface-Routine ISPCI	116
4.4.2	Verständigungsbereich	118
4.4.3	Fehlerbehandlung durch den Dialogmanager	121
4.4.4	Struktur des Operandenbereiches	122
4.4.5	Variablensubstitution im Operandenbereich und in Meldungen	123
4.5	Beschreibung der Dialogdienste	125
4.5.1	ADDPOP - Definieren der Position einer Dialogbox	126
4.5.2	ATTR - Definieren dynamischer Attribute für Maskenfelder	129
4.5.3	CONTROL - Einstellen von Arbeitsmodi	134
4.5.4	DISPLAY - Anzeigen eines Formates und/oder einer Meldung	137
4.5.5	DMCLOSE - Ende der Nutzung von Dialogdiensten	142
4.5.6	DMOPEN - Beginn der Nutzung von Dialogdiensten	144
4.5.7	REMPOP - Entfernen der Definition einer Dialogbox	146
4.5.8	VCOPY - Kopieren von Dialogvariablen in das Anwendungsprogramm	147
4.5.9	VDEFINE - Definieren von expliziten Dialogvariablen	152
4.5.10	VDELETE - Löschen von Dialogvariablen im Funktionspool	159

4.5.11	VERASE - Löschen von Dialogvariablen im Profilpool	161
4.5.12	VGET - Kopieren von Variablen aus Profilpool in den Funktionspool	162
4.5.13	VPUT - Kopieren von Variablen aus Funktionspool in Profilpool	164
4.5.14	VREPLACE - Ersetzen von Dialogvariablen im Funktionspool	166
4.6	Bibliotheken zuweisen	170
4.7	Kompatibilität	171
4.8	SDF-P-Variable in FHS-TIAM-Programmen	172
5	SDF-P-Schnittstelle	173
5.1	FHS als Ausgabe-Server	173
5.1.1	ASSIGN-STREAM	174
5.1.2	TRANSMIT-BY-STREAM	175
5.2	FHS-Dienste für SDF-P-Anwendungen	177
5.2.1	Layout und Version	177
5.2.2	Information über Anzeigendienste	178
5.2.3	Angaben zu Formatnamen und Meldungen	180
5.2.4	Angaben zu Formateigenschaften bei der Ausgabe	181
5.2.5	Feldattribute (TU ATTR) ändern	183
5.2.6	Eingabeformatierung	186
5.2.7	FHS-Rückgabe-Werte (return codes)	186
5.3	Namenskonventionen für Listenverarbeitung	187
5.4	FHS-Anwendungen durch S-Prozeduren steuern	189
5.4.1	S-Variablen mit FHS 8.1 ausgeben	189
5.4.2	S-Variablen in FHS-TIAM-Programmen ausgeben und erzeugen	189
5.4.3	FHS-Anwendungen in geschachtelten S-Prozeduren steuern	191
5.5	SYSFHS-CONTROL - Struktur für Layout und Initialisierung	193
5.6	Unterschiede zwischen FHS-DM und FHS-PRIV	200
6	Beispiel-Programme	201
6.1	Aufruf von FHS-DM als Subsystem	207
6.2	Beispielprozeduren für FHS in SDF-P	212
6.3	Beispiele für das Arbeiten mit FHS mittels S-Prozeduren	214
6.4	Beispiel für die Verwendung von Feld-Attributen:	217
6.5	Beispiel: Erstellen eines graphischen Bibliotheksmanagers	219
7	Anhang	231
7.1	Übersicht der Rückkehrcodes	231
7.2	Liste der Meldungen	232
7.3	Übersicht: Systemvariable des Dialogmanagers	259
	Fachwörter	263
	Literatur	267
	Stichwörter	271

FHS V8.1A (BS2000/OSD, TRANSDATA)

Dialogerweiterung für TIAM und SDF-P

Zielgruppe

Anwendungsentwickler

Inhalt

Das Handbuch beschreibt die Programmschnittstelle für den Einsatz des FHS-Dialo-
managers in TIAM- und SDF-P-Anwendungen.

Ausgabe: Dezember 1994

Datei: FHS_DM.PDF

BS2000 und TRANSDATA sind eingetragene Warenzeichen der Siemens Nixdorf
Informationssysteme AG

Copyright © Siemens Nixdorf Informationssysteme AG, 1995.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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



Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at

[http://ts.fujitsu.com/...](http://ts.fujitsu.com/)

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter

[http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009