

**SIEMENS**

**FORMS-2™**

Maskengenerator für LEVEL II COBOL™

**SINIX**

für Siemens PC



---

**FORMS-2™ (SINIX)**  
**Maskengenerator**  
**für LEVEL II COBOL™**

**Ausgabe Mai 1985 (FORMS-2 Version 1.0B)**

---

Bestell-Nr. U2223-J-Z95-1  
Printed in the Federal Republic of Germany  
1920 AG 6852. (2400)

Level II Cobol und Forms-2 sind Warenzeichen  
von Micro Focus

Alle Rechte vorbehalten.

Vervielfältigung dieser Unterlage  
sowie Verwertung ihres Inhalts unzulässig,  
soweit nicht ausdrücklich zugestanden.

Im Laufe der Entwicklung des Produktes  
können aus technischen oder wirtschaftlichen  
Gründen Leistungsmerkmale hinzugefügt bzw.  
geändert werden oder entfallen. Entspre-  
chendes gilt für andere Angaben in dieser  
Druckschrift.

Siemens Aktiengesellschaft

---

## Vorwort

Dieses Handbuch beschreibt **FORMS-2** im Rahmen des Programmierpakets

- Übersetzer LEVEL II COBOL
- Laufzeitsystem LZS
- FORMS-2
- ANIMATOR

### Was enthält dieses Handbuch?

Mit FORMS-2 können Sie im Dialog Ihre Bildschirmmasken für LEVEL II COBOL

- entwerfen
- erzeugen (als Datei)
- ändern und
- testen.

Meistens schreibt man ja sein Anwenderprogramm unabhängig von den Bildschirmmasken, mit denen man Daten ein- und ausgibt. FORMS-2 dient auch dazu, daß Sie beim Erstellen der Masken diese gleich ohne Ihr Programm testen können.

Dabei kann Ihnen FORMS-2 auch gleich ein Daten- Ein- Ausgabeprogramm erzeugen, das mit ISAM sequentiell auf eine Datei zugreift.

### Welche Handbücher gibt es noch für das LEVEL II COBOL Entwicklungssystem?

Neben der vorliegenden FORMS-2-Bedienungsanleitung sind im Rahmen des LEVEL II COBOL Entwicklungssystems für die SINIX-Rechner von Siemens noch folgende Handbücher erhältlich:

- LEVEL II COBOL Bedienungsanleitung
- LEVEL II COBOL Sprachbeschreibung
- ANIMATOR

---

## **An wen wendet sich dies Handbuch?**

An LEVEL II COBOL-Programmierer, die mit Bildschirmmasken zur Daten-Ein/Ausgabe arbeiten wollen.

Aber auch wenn Sie Anwender sind und von LII COBOL wenige oder keine Kenntnisse haben, können Sie mit FORMS-2 Masken erstellen und austesten.

## **Wie ist dies Handbuch aufgebaut?**

Das Kapitel 1 zeigt Ihnen allgemein die Vorgehensweise von FORMS-2, indem es Ihnen einen Ablauf vorstellt, bei dem Sie alle FORMS-2-Masken benutzen und damit kurz kennenlernen.

Ausführliche Beispiele sind in den Kapiteln 7 und 9 beschrieben.

- |               |   |
|---------------|---|
| Kapitel 2     | stellt Ihnen die FORMS-2-Einrichtungsphase vor,   |
| Kapitel 3     | beschreibt die Arbeitsphase.  |
| Kapitel 4     | zeigt auf, wie FORMS-2 die Daten beschreibt und ausgibt, die Sie als Masken erzeugt haben.                                |
| Kapitel 5     | erläutert Ihnen das FORMS-2-interne Check-Out-Programm, mit dem Sie die erstellten Datenbeschreibungen überprüfen können. |
| Kapitel 6     | beschreibt die Dateien, die FORMS-2 von den Bildschirmmasken anlegt, und was Sie alles damit tun können.                  |
| Kapitel 7 + 9 | enthalten die Beispiel-Programme und  |
| Kapitel 8     | zeigt, wie Sie vorgehen, wenn Sie ein Index-Programm erzeugen wollen.   |

Wenn Sie ein Index-Programm mit ISAM-Zugriff erzeugen und benutzen wollen, so führen Sie die Tätigkeiten des Kapitel 1 durch, lesen kurz Kapitel 2 und 3 und arbeiten anschließend die Kapitel 8 und 9 durch.

---

Wenn Sie Masken oder Formulare generieren wollen, arbeiten Sie bitte die Kapitel in der Reihenfolge 1, 2, 3 und 7 durch.

Wenn Sie mehr über die Möglichkeiten und Arbeitsweise von FORMS-2 wissen wollen, lesen Sie bitte die Kapitel 1, 2 und 3 genau.

Die Kapitel 4 und 6 sind Beschreibungen der FORMS-2-Ausgabedateien und der Angaben, die dabei gemacht werden können. Sie brauchen diese Kapitel nur zu lesen, wenn Sie die speziellen Angaben benutzen wollen.

Sie sollten auf jeden Fall die Beispiele ausprobieren, bevor Sie FORMS-2 benutzen.

### **Wie finden Sie sich in diesem Handbuch zurecht?**

Oben auf jeder Seite steht ein Hinweis darauf, was in dem Abschnitt jeweils behandelt wird. Am Ende des Handbuchs gibt es ausklappbare Seiten, auf denen die FORMS-2-Menübildschirme mit den deutschen Erklärungen abgebildet sind. Wenn Sie die Beispiele durcharbeiten, klappen Sie die Seiten zur Übersicht mit aus!

### **Was können Sie tun, damit das Handbuch besser wird?**

Eine ganze Menge! Schreiben Sie uns bitte, an welchen Stellen das Handbuch falsch ist oder unübersichtlich - kurz, wo Sie im Handbuch gestolpert sind.

Oder schreiben Sie uns auch, wenn Sie mit dem Handbuch zufrieden sind, das freut uns dann ganz besonders!

Auf jeden Fall - dafür schon jetzt vielen Dank!

### **Manualredaktion K D ST PM 2**

Otto-Hahn-Ring 6, 8 München 83

---

# Inhalt

<b>1</b>	<b>Allgemeine Beschreibung</b>	<b>1-1</b>
1.1	Möglichkeiten von FORMS-2	1-1
1.2	Von FORMS-2 erzeugte Dateien	1-2
1.3	Programmunterteilung	1-3
	Einrichtungsphase	1-3
	Arbeitsphase	1-4
1.4	Bedienoberfläche	1-6
1.5	Cursor-Steuerung	1-7
1.6	Dateien beim FORMS-2-Betrieb	1-8
1.7	Überprüfen von FORMS-2	1-9
<b>2</b>	<b>Einrichtungsphase</b>	<b>2-1</b>
2.1	Initialisierungsbildschirm I01	2-1
	Datennamen- und Dateinamen-Basis	2-1
	Zeilen pro Bildschirm	2-2
	Währungszeichen	2-3
	Dezimalpunkt-Stelle	2-3
2.2	Initialisierungsbildschirm I02	2-4
	Dateikombinationen	2-4
	Gleiches oder anderes Dateiverzeichnis?	2-6
<b>3</b>	<b>Arbeitsphase</b>	<b>3-1</b>
3.1	Bildschirm W01	3-2
	Auswahl des Bildschirmtyps	3-2
	FORMS-2-Lauf beenden	3-3
	Arbeitsbildschirm	3-3
	Hintergrund und Vordergrund	3-4
	EDIT-Modus	3-6
	Fester Text	3-6
	Variable Daten	3-7
	Kommando-Modus	3-10
3.2	Kommandos	3-11
3.2.1	Allgemeine Kommandos am Arbeitsbildschirm	3-12
3.2.2	Arbeitsmasken-Bearbeitungs-Kommandos	3-15
3.2.3	Programmier-Kommandos	3-20
3.2.4	Beenden der Arbeitsphase	3-30

<b>4</b>	<b>Datenbeschreibungen</b>	<b>4-1</b>
4.1	Generieren von Datensatz- und Datennamen	4-1
4.2	Datensätze	4-2
	Benennen der Datennamen	4-2
	PICTURE-Klauseln	4-3
	Fester Text	4-3
	Felder für variable Daten	4-3
	Bearbeiten der DDS-Datei	4-4
	Aufruf einer DDS-Datei	4-4
<b>5</b>	<b>Das CHECK-OUT-Programm</b>	<b>5-1</b>
5.1	Allgemeines	5-1
5.2	Erzeugen des CHECK-OUT-Programms	5-2
5.3	Übersetzung des CHECK-OUT-Programms	5-2
5.4	Ablauf des CHECK-OUT-Programms	5-3
	Laden	5-3
	Ausführen	5-3
	Masken mit festem Text	5-3
	Festgelegter Text auf leerem Bildschirm	5-3
	Festgelegter Text auf dem vorhergehenden Bildschirm	5-3
	Eingabefelder für Daten (variabler Teil der Maske)	5-4
	Beenden	5-4
<b>6</b>	<b>Bildschirm-Masken-Datei</b>	<b>6-1</b>
6.1	Allgemeines	6-1
6.2	Generieren der Bildschirm-Masken	6-1
6.3	FORMS-2 Handhabung	6-3
6.4	Ausdruck der Formulare	6-4
6.5	Masken für die Programmentwicklung	6-4
<b>7</b>	<b>Beispiel einer FORMS-2-Maskengenerierung</b>	<b>7-1</b>

<b>8</b>	<b>Index-Programm . . . . .</b>	<b>8-1</b>
8.1	Allgemeines . . . . .	8-1
8.2	Generieren eines Index-Programms . . . . .	8-3
8.3	Erzeugte Dateien . . . . .	8-6
8.4	Übersetzen des Index-Programms . . . . .	8-6
8.5	Ablauf des Index-Programms . . . . .	8-7
	Laden . . . . .	8-7
	Datenverarbeitungsmöglichkeiten . . . . .	8-7
	Abfrage mit Schlüssel . . . . .	8-7
	Sequentielle Abfrage . . . . .	8-8
	Änderung des angezeigten Satzes . . . . .	8-8
	Löschen des angezeigten Satzes . . . . .	8-8
	Einfügen eines neuen Satzes . . . . .	8-8
	Programm beenden . . . . .	8-9
8.6	Satzsperrung . . . . .	8-10
	Was passiert, wenn ein Satz gesperrt ist? . . . . .	8-10
8.7	Zusammenfassung der Anwendungsmöglichkeiten . . . . .	8-12
<b>9</b>	<b>Beispiel eines Anwender-Indexprogramms . . . . .</b>	<b>9-1</b>
<b>A</b>	<b>Anhang Bildschirme</b>	

**Stichwörter**



# 1 Allgemeine Beschreibung

## 1.1 Möglichkeiten von FORMS-2

FORMS-2 ist ein Bestandteil des LEVEL II COBOL Software-Entwicklungssystems. Es ermöglicht eine dialogorientierte Erstellung und Änderung von Bildschirmmasken für LII COBOL-Programme. Das Paket bietet Ihnen hauptsächlich in vier Punkten Unterstützung:

- Umsetzung von Benutzermasken in COBOL-Satzbeschreibungen, die in COBOL-Anwenderprogramme übernommen werden können.
- Überprüfung von Benutzermasken in einem Testprogramm, vor der Übernahme in das Anwenderprogramm.
- Speicherung von genauen Bildschirmabbildern der Benutzermasken in Dateien für eventuell nachfolgende Wiederverwendung und Ausdruck.
- Generierung eines kompletten COBOL-Programms, das Datenerfassung, Datenveränderung und Abfragen ermöglicht. Dies geschieht mit Hilfe der erstellten Anwenderbildschirmmasken und einer index-sequentiellen (ISAM-)Datei.

### 1.2 Von FORMS-2 erzeugte Dateien

Sie können zwischen jeder gültigen Kombination der oben genannten Möglichkeiten wählen. Abhängig von den ausgewählten Optionen werden von FORMS-2 automatisch die folgenden vier Typen von Dateien erstellt:

- **basisname.DDS**

Die von Ihnen erstellte Maske (Formular) wird in einer Quelldatei mit LEVEL II COBOL-Datenbeschreibungen mit dem Namen **basisname.DDS** abgelegt. Diese COBOL-PICTURE-Klauseln können Sie dann mit der COPY-Anweisung in Ihr COBOL-Anwenderprogramm übernehmen.

- **basisname.CHK**

Unter diesem Namen wird ein Testprogramm abgelegt, das die Datendefinitionen (= **basisname.DDS**) von der Datei enthält, die die Benutzermaske definiert. Nach der Übersetzung dieses Programms können Sie die Dateneingabe mit dieser Maske überprüfen, bevor Sie sie in der eigentlichen Anwendung benutzen.

- **basisname.Snn**

Dies sind Dateien, die abdruckbare Bilder der definierten Masken (Formulare) enthalten. Die Dateien werden mit **basisname.Snn** generiert, wobei nn die Zahlen von 00 bis 99 bedeuten.

- **basisname.GEN**

Dies ist die Datei eines Index-Quellprogramms, das den Benutzerbildschirm verwendet. Nach Übersetzung kann das generierte Programm über das Benutzerformular Daten

eingeben

abfragen

ändern

löschen.

### 1.3 Programmunterteilung

Den FORMS-2-Ablauf kann man in eine Reihe von logisch unterscheidbare Einheiten unterteilen. Es gibt aber zwei Hauptteile:

- die Einrichtungsphase und
- die Arbeitsphase

#### Einrichtungsphase

Die Einrichtungsphase wird nur einmal durchlaufen und setzt die Parameter für diesen speziellen FORMS-2-Programmdurchlauf.

Am Bildschirm werden eine Reihe selbst erklärender Masken ausgegeben, die Sie entsprechend ausfüllen müssen.

Dazu stehen Ihnen die beiden Bildschirme I01 und I02 zur Verfügung.

- Bildschirm I01

Hier können Sie z.B. den Basis-Namen der Dateien angeben, die FORMS-2 erzeugen soll.

- Bildschirm I02

Mit diesem Bildschirm können Sie die Dateien angeben, die FORMS-2 erzeugen soll und wenn Sie **G** angeben, daß ein Index-Programm erzeugt werden soll.

### Arbeitsphase

Wenn Sie eine Benutzermaske erstellen, werden zwei Arbeitsphasen durchlaufen. Die FORMS-2-Bildschirmanzeige ist wie ein Papierformular; Sie haben einen festgelegten vorgegebenen Text, und in die freien Bereiche können Sie Text eingeben. Der feste Text dient Ihnen dabei als Führungshilfe. Beide Arten von Feldern, die festen Teile und die, in die Sie Daten eingeben können, müssen getrennt definiert werden.

Es gibt deshalb zwei verschiedene Arbeitsphasen; dort wird erzeugt:  
der **unveränderliche** Text (eigentliche Maske)  
der **variable** Teil der Maske (Felder, in die man eingeben kann).

### Arbeitsphase 1

- Bildschirm W01

Auswahl des Arbeitsbildschirms - Sie können hier nur  eingeben und damit den festen Text definieren. Außer, Sie wollen den Lauf hier beenden, dann drücken Sie  oder für die Anzeige des Hilfsbildschirms .

Es wird ein leerer Arbeitsbildschirm für den festen Textteil der Maske ausgegeben. Sie können außer    alle im Kapitel 3 beschriebenen Kommandos benutzen.

Wenn Sie den festen Teil der Maske fertig eingegeben haben, übergeben Sie den Bildschirm mit der Folge   zum weiteren verarbeiten.

Der Bildschirm W01 wird dann wieder angezeigt.

## Arbeitsphase 2

- Bildschirm W01

Jetzt will FORMS-2 die voreingestellte Option  C.

Die variablen Felder werden nun definiert, wie es in Kapitel 3 beschrieben ist, d.h. die Zeichen  X  Y  8  9 sowie die Druckaufbereitungszeichen sind zugelassen.

Wenn Sie ein LII COBOL-Programm für indizierte Dateizugriffe generieren, müssen Sie, bevor Sie den Bildschirm verarbeiten, das Ende des Schlüsselfeldes bzw. den Anfang der Daten innerhalb des Satzes definieren. Das tun Sie, indem Sie den Cursor auf den Anfang des ersten Datenfeldes setzen und eingeben:

↓  \*  ↓.

### 1.4 Bedieneroberfläche

FORMS-2 ist in COBOL geschrieben und verwendet die erweiterten Möglichkeiten des ACCEPT und DISPLAY des LEVEL II COBOL. ACCEPT und DISPLAY werden im Handbuch LEVEL II COBOL Sprachbeschreibung näher beschrieben, die Funktionen des Cursors im Handbuch LEVEL II COBOL Bedienungsanleitung.

Die Vorteile dieser erweiterten Bildschirm-Ein/Ausgabe sind:

- Korrekturen können direkt durch Überschreiben gemacht werden
- In numerische Felder können nur numerische Zeichen eingegeben werden.
- Die Taste  füllt in einem numerischen Feld das Feld von links mit Nullen.

## 1.5 Cursor-Steuerung

In der folgenden Tabelle sind die Funktionen der Cursor-Steuertasten zusammengestellt.

---

Funktion	Taste
Setzt den Cursor um ein Zeichen nach rechts	→
Setzt den Cursor um ein Zeichen nach links	←
Setzt den Cursor auf den Beginn des folgenden Datenfeldes	↓
Setzt den Cursor auf den Beginn des vorhergehenden Datenfeldes	↑
Setzt den Cursor auf den Beginn des ersten Datenfeldes	↖
Der Cursor geht an den nächsten Tabulator-Stop	→

---

Im Edit-Modus wird der Bildschirm in 80 Zeichen lange Datenfelder geteilt. Es gibt 22 - 24 Felder pro Bildschirm, je nach Anzahl der Zeilen, die im Bildschirm I01 angegeben wurde.

Sie können den Text korrigieren, indem Sie ihn überschreiben oder Sie wechseln in den Kommando-Modus und verwenden die Edit-Kommandos (s. Kapitel 3.1).

## 1.6 Dateien beim FORMS-2-Betrieb

Voraussetzung für einen FORMS-2-Lauf ist ein ordnungsgemäß installierter LEVEL II COBOL Compiler.

Installieren Sie Ihr FORMS-2-System wie gewöhnlich unter der Kennung "admin" mit der Menü-Auswahl "Installation von Softwareprodukten". Nach erfolgreicher Installation finden Sie folgende Dateien in Ihrem System vor:

D A T E I	Inhalt	Beschreibung
/usr/lib/cobol/FORMS-2.INT	FORMS2-Programm	
FORMS-2.I01 FORMS-2.I02	Initialisierungs Bildschirme	Kapitel 2
FORMS-2.W01 FORMS-2.W02		Kapitel 3
FORMS-2.H01 FORMS-2.H02 FORMS-2.H03 FORMS-2.H04	Hilfsbildschirm " " "	Kapitel 3
FORMS-2.CH1 FORMS-2.CH2	Gerüst für Check-Programm	Kapitel 5
FORMS-2.GN1 FORMS-2.GN2	Gerüst für Index-Programm	Kapitel 8
/usr/bin/forms2	FORMS-2-Aufruf	

Wenn alle oben beschriebenen Dateien vorliegen, können Sie mit FORMS-2 beginnen.

## 1.7 Überprüfen von FORMS-2

Überprüfen Sie nun bitte die Funktionsfähigkeit und Vollständigkeit Ihres FORMS-2-Paketes, indem Sie den nächsten einfachen Testlauf mal probieren.

1. Rufen Sie FORMS-2 auf. Das geht ganz einfach:

forms2

Sollten Sie mit SINIX-Menüs arbeiten, wählen Sie bitte, vom Hauptmenü ausgehend, folgendermaßen aus:

e – Entwicklungssysteme

I – Level II COBOL-Entwicklungssystem

f – FORMS-2 Programm erstellen

## Testlauf

---

2. Das Programm FORMS-2 wird gestartet. Sie bekommen folgenden Bildschirm:

FORMS2 V1.3	INITIALISATION PHASE	SCREEN I01
FORMS2 PARAMETERS:		
DATA-NAME & FILE-NAME	[ _ ]	(1-6 alphanumeric characters)
CRT lines	[24]	(22 or 23 or 24)
SPECIAL-NAMES clause:		
CURRENCY SIGN	[\$]	(ANSI currency signs only)
DECIMAL-POINT	[.]	("." or ",")
Press RETURN when complete		

Die Grundlage für die Dateinamen und die Datenfeldnamen bildet ein maximal sechsstelliger Name, den Sie dort eingeben, wo der Cursor steht. Für die restlichen Eingabefelder kann die Voreinstellung übernommen werden. Die Eingabe wird mit  beendet.

3. FORMS-2 gibt nun den Bildschirm I02 aus und will wissen, welche Ausgabedateien erzeugt werden sollen und in welches Dateiverzeichnis sie eingetragen werden sollen.

```
FORMS2 V1.3                INITIALISATION PHASE                SCREEN I02

FILES TO BE CREATED:

      FILE COMBINATIONS      [C]      (A = DDS)
                                   (B = DDS & CHK)
                                   (C = DDS & CHK & Snn)
                                   (D = DDS & Snn)
                                   (E = Snn)
                                   (F = No files output)
                                   (G = DDS & Snn & GEN)

DEVICE/DIRECTORY PREFIX (0-40 Chars) [                ]

Press RETURN when complete
```

Geben Sie erst  (keine Ausgabe-Dateien) ein und dann .

## Testlauf

---

4. FORMS-2 zeigt den Bildschirm W01 und will wissen, welchen Maskenteil Sie erstellen wollen.

FORMS2 V1.3	WORK PHASE	SCREEN W01
WORK SCREEN SELECTION:		
SCREEN TYPE	[A]	(A = Fixed text on clear screen) (B = Fixed text on last screen) (C = Variable data redefines last screen) (D = Variable data without redefinition) (! = Complete this FORMS run)
Fixed Text allows:	All characters	
Variable Data allows:	X or Y to define alphanumeric fields 9 or 8 to define numeric fields edit chars to define numeric edit fields	
Press RETURN when complete		

Übernehmen Sie die Voreinstellung A (fester Text auf leerem Bildschirm) und geben Sie  ein.

5. FORMS-2 gibt einen leeren Bildschirm aus. Sie befinden sich im Edit-Modus. Sie können den Cursor an jeden beliebigen Punkt des Bildschirms setzen. Verwenden Sie die Cursor-Kontrolltasten und die normalen Buchstabentasten, um den folgenden Text auf den Bildschirm zu bringen:

---

NAME [ ]

---

Drücken Sie abschließend .

6. FORMS-2 setzt " \_ " in die linke obere Ecke, d.h., Sie befinden sich nun im Kommando-Modus. Geben Sie  (nächster Bildschirm) ein und dann .

## Testlauf

---

### 7. FORMS-2 zeigt den Bildschirm H01:

```
FORMS2 V1.3                HELP SCREEN                SCREEN H01

GENERAL COMMAND SUMMARY:
SPACE = Process the work screen
_     = Re-enter EDIT mode
?     = Display the next HELP screen
?n    = Display the nth HELP screen
Q     = Re-enter WORK PHASE screen selection
!     = Terminate FORMS run immediately
X     = Position commands at EDIT mode cursor
*     = Indicate Index Form's data area start

NOTE:   SPACE is the command to process the EDIT mode screen

HELP option  [?]  (_ = Re-enter EDIT mode)
              [?]  (? = Display next HELP screen)
              [!]  (! = Abandon FORMS2 run immediately)

Press RETURN when complete
```

Geben Sie  (nächster Bildschirm) ein und dann .

8. FORMS-2 gibt den Bildschirm H02 aus:

```

FORMS2 V1.3                HELP SCREEN                SCREEN H02

MANIPULATION COMMAND SUMMARY:
F = Invoke FOREGROUND/BACKGROUND manipulation
Fx = Invoke FOREGROUND/BACKGROUND option "x"
O = Turn on automatic WORK screen preparation
O1 = Turn off automatic WORK screen preparation
Cn = Insert n spaces at cursor position
Dn = Delete n chars at cursor position
In = Insert n blank lines before cursor line
Kn = Delete n lines including cursor line
An = Overwrite n lines with data of cursor line
Un = Move cursor up n lines
Vn = Move cursor down n lines

HELP option    [ _ ]    ( _ = Re-enter EDIT mode)
                 [ ? ]    (? = Display next HELP screen)
                 [ ! ]    (! = Abandon FORMS2 run immediately)

Press RETURN when complete
    
```

Geben Sie [?] (nächster Bildschirm) ein und dann [↵].

## Testlauf

---

### 9. FORMS-2 erstellt den Bildschirm H03:

```
FORMS2 V1.3                HELP SCREEN                SCREEN H03

PROGRAMMING COMMAND SUMMARY:
G = Give datanames screen coordinates suffix
G1 = Give datanames sequential number suffix
Jn = Allow up to n consec. spaces in fixed text
Mx = Interpret "x" as "space"
S = Cancel previous Sn command
S1 = Inhibit DDS & CHK output at next processing
S2 = Inhibit Snn output at next processing
S3 = Prompt for Snn file-name at next processing
S9 = Line edit DDS output at next processing
P = Display cursor position coordinates

HELP option    [-]    (_ = Re-enter EDIT mode)
                  (? = Display next HELP screen)
                  (! = Abandon FORMS2 run immediately)

Press RETURN when complete
```

Geben Sie  (nächster Bildschirm) ein und dann .

10. FORMS-2 zeigt den Bildschirm H04:

```

FORMS2 V1.3                                HELP SCREEN                                SCREEN H04

WINDOW COMMAND SUMMARY:

W = Position cursor to current window start
W1 = Start window at cursor line
W2 = End window at cursor line
W3 = Start window at cursor line, no delim's
W4 = End window at cursor line, no delim's
W5 = Display start window delimiters
W6 = Display end window delimiters
W7 = Re-display data overwritten by start delim's
W8 = Re-display data overwritten by end delim's
W9 = Position cursor to current window end

HELP option  [_]  (. = Re-enter EDIT mode)
              (? = Display next HELP screen)
              (! = Abandon FORMS2 run immediately)

Press RETURN when complete
    
```

Drücken Sie nur .

11. FORMS-2 zeigt den festen Teil des Textes an, den Sie unter Punkt 5 eingeben haben. Drücken Sie .

12. FORMS-2 schreibt "\_\_" in die linke obere Ecke des Bildschirms. Geben Sie  (Vordergrund/Hintergrund) ein und dann .

## Testlauf

---

### 13. FORMS-2 erstellt den Bildschirm W02:

FORMS2 V1.3	WORK PHASE	SCREEN W02
FOREGROUND/BACKGROUND OPERATIONS:		
OPTION	[H]	(A = Re-enter EDIT MODE) (B = Clear FOREGROUND) (C = Clear BACKGROUND) (D = Merge BACKGROUND into FOREGROUND) (E = Merge FOREGROUND into BACKGROUND) (F = Merge screen image into FOREGROUND) (G = Merge screen image into BACKGROUND) (H = Display FOREGROUND) (I = Display BACKGROUND) (J = Display screen image)
	NOTE:	(H & I & J display until RETURN pressed)
FILE-NAME [		]
		(F & G & J only)
Press RETURN when complete		

Geben Sie erst A (Ausgabe-Modus) ein und dann J.

14. Wieder zeigt FORMS-2 den festgelegten Text, den Sie unter Punkt 5 eingegeben haben. Drücken Sie J.
15. FORMS-2 schreibt " \_ \_ " in die linke obere Ecke des Bildschirms. Drücken Sie die Leertaste (zum Verarbeiten des Bildschirms) und dann J.

16. FORMS-2 zeigt wieder den Bildschirm W01 und will wissen, welchen Maskenteil Sie erstellen wollen. Diesmal ist C vorgegeben, drücken Sie .

Jetzt kommt der Bildschirm wieder, den Sie vorher erstellt haben, aber Ihre eingegebenen Daten stehen nun im Hintergrund; im Vordergrund können Sie jetzt Ihre variablen Daten eingeben.

17. Verwenden Sie die Cursor-Kontrolltasten und geben 'X' für variable Daten im Namen-Feld ein.

---

NAME [XXXXXXXXXXXXXXXXXXXXX]

---

Drücken Sie .

18. FORMS-2 schreibt " \_ " in die linke obere Ecke des Bildschirms. Drücken Sie die Leertaste (zum Verarbeiten) und dann .

19. Wieder zeigt FORMS-2 den Bildschirm W01. Geben Sie diesmal  ein und beenden Sie den Lauf. Drücken Sie .

## Testlauf

---

20. FORMS-2 verabschiedet sich mit der Anzeige:

---

```
END OF FORMS-2 RUN
FORMS-2 COPYRIGHT (C) 1979, 1982 MICRO FOCUS LTD
```

---

Sie haben nun alle FORMS-2 Bildschirme benutzt und vielleicht schon ein bißchen Gefühl für FORMS-2 bekommen.

Wir wiederholen noch einmal kurz:

Die Bildschirme I01 und I02 sind zur Initialisierung, d.h., hier werden Angaben über Dateinamen und gewünschte Ausgaben gemacht.

W01 und W02 sind Arbeitsbildschirme, mit denen geben Sie an, welche Maske Sie im Vordergrund (also die Daten ändern) oder im Hintergrund (als Bild der schon eingegebenen Daten) haben wollen.

Die Hilfsbildschirme H01 bis H04 können jederzeit vom Kommando-Modus aus mit  aufgerufen werden und weisen auf Einzelfunktionen hin.

## 2 Einrichtungsphase

Die Einrichtungsphase des FORMS-2-Programms folgt unmittelbar nach dem Laden des Programms und wird in jedem FORMS-2-Lauf nur einmal durchlaufen. Sie besteht darin, daß Sie auf Fragen der beiden Einrichtungsbildschirme I01 und I02 antworten.

Die Abbildungen und Erklärungen können Sie am Ende des Handbuchs herausklappen.

### 2.1 Initialisierungsbildschirm I01

Diese Bildschirmanzeige wird sofort nach dem Laden von FORMS-2 ausgegeben.

Sie können vier Informationen eingeben. Beachten Sie, daß während dieser Bildschirmanzeige sofort alle bis dahin gegebenen Antworten und die noch verbleibenden Voreinstellungen abgeschickt werden, wenn Sie die Taste  drücken. Machen Sie also erst die entsprechenden Angaben bei folgenden Fragen:

- Datennamen- und Dateinamenbasis
- Zeilen pro Bildschirm (CRT)
- Währungszeichen
- Dezimalpunkt-Stelle

#### Datennamen- und Dateinamen-Basis

Die an dieser Stelle eingegebene Datensatz/Dateinamen-Basis wird von FORMS-2 folgendermaßen benutzt:

1. Sie wird als erster Teil aller in diesem Lauf generierten Datenfeldnamen und Datensatznamen verwendet. Die Namen enthalten
  - eine zweistellige aufsteigende Zahl, beginnend bei 00, für jeden neuen Satz und
  - eine 4-stellige aufsteigende Folgezahl, beginnend bei 0000, für die einzelnen Felder jeder Maske.

### Beispiel

Sie geben bei Basis an:

beispl

Dann erhalten die ersten 3 Felder die Namen:

```
beispl-00-0000  PIC X(0011)  VALUE "Das ist das".
beispl-00-0001  PIC X(0007)  VALUE " erste ".
beispl-00-0002  PIC X(0006)  VALUE " Feld " .
```

Der nächste Satz, z.B. die variablen Daten, sehen dann so aus:

```
beispl-01-0000  PIC X(0003) .
beispl-01-0001  PIC 9(0002) .
beispl-01-0002  PIC 9(0005) .
```

Wahlweise können durch das Kommando **G** in der Arbeitsphase die letzten 4 Stellen des Namens durch die Koordinaten des jeweiligen Feldes in der Maske gebildet werden.

2. Sie wird als Haupt-Bestandteil für die Namen der generierten Dateien verwendet. Diese können sein:

basisname.DDS für COBOL-Datenbeschreibungsklauseln  
(Picture-Klauseln, siehe Kapitel 4)

basisname.CHK für das Checkout(= Kontroll)-Programm  
(s. Kapitel 5)

basisname.Snn für Bildschirmmasken (nn = 00, 01, 02, ..., 99  
s. Kapitel 6)

basisname.GEN für das Index-Programm (s. Kapitel 8)

Beachten Sie, daß bei jedem FORMS-2 Lauf nur eine DDS-Datei ausgegeben wird, während für jede angezeigte Bildschirmmaske eine eigene Datei (basisname.Snn) aufgebaut wird.

### Zeilen pro Bildschirm

FORMS-2 kann mit Bildschirmen, die 22, 23 oder 24 Zeilen anzeigen, benutzt werden. FORMS-2 hat die Voreinstellung 24.

### **Währungszeichen**

Mit dieser Eingabe wird das voreingestellte Währungszeichen (\$) überschrieben. Damit wird im Checkout (Kontroll)- und im Index-Programm eine passende Sonderzeicheneingabe generiert.

### **Dezimalpunkt-Stelle**

Mit dieser Wahlmöglichkeit können Sie die Funktion des Punktes "." und des Kommas "," austauschen. Wenn Sie "," angeben, dann wird eine DECIMAL-POINT IS COMMA-Klausel in den Programmen Checkout und Index generiert. Die Voreinstellung ist hier ".".

Das entsprechende Dezimalpunkt-Zeichen kann verwendet werden, wenn Sie in der Arbeitsphase numerisch aufbereitete Felder angeben.

### 2.2 Initialisierungsbildschirm I02

Diese Bildschirmanzeige (I02) wird ausgegeben, wenn Sie die am Bildschirm I01 nötigen Eingaben mit  abgeschlossen haben.

Mit dieser Maske können Sie festlegen:

- Welche Dateitypen sollen erstellt werden?
- In welches Dateiverzeichnis sollen die Dateien geschrieben werden?

Haben Sie den Bildschirm I02 mit  weggeschickt, dann beginnt die Arbeitsphase und Sie können in dieser Phase die Informationen zur Initialisierung nicht mehr ändern.

#### *Bitte beachten Sie*

Bei der Option  findet keine Überprüfung statt, ob die zulässigen Zeichen (8 9 X Y) für die Felder der variablen Eingabe verwendet wurden.

dient also nur zur Dokumentation.

#### **Dateikombinationen**

FORMS-2 bietet Ihnen Wahlmöglichkeiten für alle gültigen Kombinationen dieser Dateien, von denen jede durch eine eindeutige Erweiterung des Dateinamens gekennzeichnet wird.

**DDS** Sie können COBOL Datenerklärungs-Befehle (DDS = Data Description Statements) generieren, entsprechend der Bildschirmmaske, die Sie erstellt haben. Die Ausgabe erfolgt in einer Standard-ASCII-Textdatei und kann anschließend in jedem Programm übersetzt werden, das diese Datei mit der Standard-COBOL-Anweisung COPY aufruft. Diese Dateien werden im besonderen auch im Checkout- und Index-Programm verwendet (siehe unten).

Sollten Sie mit der Bildschirmbehandlung in einem LEVEL II COBOL-Programm nicht vertraut sein, so verweisen wir Sie auf die LEVEL II COBOL Bedienungsanleitung und hier besonders auf das Kapitel 4 Dialog.

**CHK** Zusätzlich zu der Datei `basisname.DDS` kann FORMS-2 auch ein Checkout (Kontroll)-Programm generieren. Es besteht einfach aus den PROCEDURE DIVISION-Befehlen ACCEPT und DISPLAY, die sich auf die erstellten Bildschirmmasken beziehen. Diese Befehle werden unter `basisname.CHK` abgelegt. In diesem kleinen COBOL-Programm werden folgende COPY-Dateien verwendet:

`basisname.DDS`

`FORMS2.CH1`

`FORMS2.CH2`

Mit dem Checkout (Kontroll)-Programm können Sie nacheinander die gerade erstellten Bildschirmmasken ausgeben. Sie können Daten eingeben, genau wie sonst auch.

**Snn** Sie können sich auch den Text einer gerade erstellten Bildschirmmaske auf eine Platten-Datei in Form eines Bildschirm-Abbildes abspeichern lassen. Diese Datei kann später in diesem Lauf oder in nachfolgenden FORMS-2-Läufen wieder verwendet und auch verändert werden.

Der Text dieser Datei kann darüberhinaus auch ausgedruckt werden und so als Vorlage für andere Anwender und Programmierer dienen.

GEN FORMS-2 kann ein Index-Programm generieren. Es beinhaltet den notwendigen Code, um eine indizierte Datei mit sequentiellem Zugriff aufzubauen und zu pflegen, die der Struktur des Benutzer-Formulars (= Maske) entspricht. Der Code wird in der Datei basisname.GEN abgespeichert und beinhaltet folgende COPY-Elemente:

basisname.DDS

FORMS2.GN1

FORMS2.GN2

Wenn Sie ein Index-Programm generieren, schränkt das die Funktionen von FORMS-2 ein. Im Kapitel 8 werden diese Einschränkungen, der Aufbau und die Bearbeitung eines Index-Programms eingehend beschrieben.

### *Achtung*

Wenn Sie beim Bildschirm I01 unabsichtlich die Taste  gedrückt haben, ehe Sie alle gewünschten Daten eingegeben haben, können Sie an dieser Stelle das -Kommando (s. Kapitel 3) eingeben und FORMS-2 geht zurück zum Bildschirm I01 und ermöglicht Ihnen, die Informationen nochmal zu verändern.

### **Gleiches oder anderes Dateiverzeichnis?**

Alle Dateien die von FORMS-2 erzeugt werden, befinden sich im gleichen Dateiverzeichnis. Wollen Sie die Dateien in einem anderen Dateiverzeichnis haben, dann geben Sie den Namen dieses Dateiverzeichnis' an. Machen Sie hierzu keine anderen Angaben, dann stehen die erzeugten Dateien alle im aktuellen Dateiverzeichnis.

### 3 Arbeitsphase

Sie wollen nun in Ihrem LII COBOL-Programm Formulare oder Masken verwenden. Mit FORMS-2 können Sie beliebig viele Formularentwürfe erzeugen, indem Sie den Entwurf als Text am Bildschirm eingeben.

Um ein Formular oder eine Maske zu definieren, sind mindestens zwei unterschiedliche Arbeitsphasen nötig:

eine, um den Text des Formulars oder der Maske zu definieren,  
und eine, um die variablen Dateneingabe-Felder zu definieren.

Die erste Arbeitsphase wird dazu verwendet, den Text und das Format des Formulars zu entwerfen, und die darauffolgende Arbeitsphase, um innerhalb des Formulars die variablen Dateneingabe-Felder anzugeben. Dann muß FORMS-2 noch wissen, was Sie in einer bestimmten Phase eingeben wollen. Deshalb fängt die Arbeitsphase mit dem Bildschirm W01 an, auf dem die verschiedenen Wahlmöglichkeiten angezeigt sind.

#### **Allgemeines zum Darstellungsformat**

Die einzugebenden Kommandos werden als Tasten dargestellt.

### 3.1 Bildschirm W01

#### Auswahl des Bildschirmtyps

Sie wollen feste Werte (Text) eingeben:

Drücken Sie die angegebene Taste und dann .

- A Der Bildschirm ist gelöscht. Sie können den festen Text für ein neues Formular eingeben.
- B Die vorhergehende Bildschirmanzeige wird noch einmal ausgegeben, damit Sie zusätzlich einen festen Text eingeben können. Der Text der vorhergehenden Bildschirmanzeige wird in diesem Fall nur als **Hintergrund** verwendet, und wird **nicht** in der Datensatzdefinition gespeichert, die für den gegenwärtig eingegebenen festen Text gültig ist.

#### *Beachten Sie*

Sollten Sie versehentlich irgendeinen Teil der vorhergehenden Bildschirmanzeige überschrieben haben, dann dürfen Sie nicht durch die ursprünglichen Zeichen ersetzt werden, sondern Sie löschen die falschen Zeichen mit der Leertaste.

Sie wollen variable Daten (Eingabedaten) eingeben:

Drücken Sie die angegebene Taste und dann .

- C Die vorhergehende Bildschirmanzeige wird noch einmal ausgegeben, um Ihnen bei der Definition der variablen Dateneingabefelder zu helfen. Dort stehen später die Daten, die Sie in die Maske eingeben können.
- D Die vorhergehende Bildschirmanzeige wird wieder ausgegeben, damit Sie die variablen Dateneingabe-Felder definieren können. Die Datendefinitionen der variablen Felder werden **getrennt** von den Daten des festen Teils der Maske abgelegt (d.h. nicht durch Redefinition). Bei größeren Programmen kann es sinnvoll sein, auf die Definitionen getrennt zuzugreifen, z.B. wenn Sie Ihr Programm auf mehrere Unterprogramme aufteilen.

### FORMS-2-Lauf beenden

Nach jeder Arbeitsphase wird wieder der Bildschirm W01 angezeigt. Mit diesem Bildschirm können Sie das Programm beenden. Drücken Sie  und dann .

#### *Achtung*

Verwenden Sie das  Kommando zu irgendeinem anderen Zeitpunkt, wird der FORMS-2-Lauf abgebrochen.

Wenn das Programm beendet wird, wird die DDS-Datei geschlossen und eine Nachricht ausgegeben. Wenn während der Einrichtungsphase CHK gewählt wurde, dann wird das Check-Out-Programm generiert und in eine Datei geschrieben. Die Datei "basisname.CHK" wird geschlossen und eine Schlußzeile mit den Namen der erzeugten Dateien ausgegeben.

Wurde ein Index-Programm generiert (siehe Kapitel 8), so wird der Lauf automatisch nach der zweiten Arbeitsphase beendet.

### Arbeitsbildschirm

Nachdem die Art der Bildschirmbearbeitung gewählt wurde (in der Maske W01), können Sie in den Arbeitsbildschirm Text eingeben. Bei  (Text in leeren Bildschirm) wird ein leerer Bildschirm angezeigt. Bei den anderen Optionen wird der vorhergehende Bildschirm angezeigt, damit Sie Ihre neue Eingabe an die richtige Stelle setzen können.

### Hintergrund und Vordergrund

Um nur die in dieser Phase eingegebenen Daten zu verarbeiten, muß FORMS-2 diese Daten von den vorher eingegebenen Daten getrennt speichern, die nur als Anhaltspunkte angezeigt werden. FORMS-2 erreicht dies durch Aufbau des angezeigten Bildschirms aus zwei getrennten Datenbereichen mit den Namen Hintergrund und Vordergrund.

Unter **Vordergrund** werden die während der gegenwärtigen Arbeitsphase eingegebenen Daten gespeichert. Unter **Hintergrund** sind die am vorhergehenden Bildschirm eingegebenen Daten abgelegt.

Also: Am Ende jeder Arbeitsphase bearbeitet FORMS-2 nur die Vordergrund-Daten.

Wenn der Bildschirm W01 das nächste Mal angezeigt wird, dann wird bei den Angaben  B  C  D der Hintergrundinhalt mit dem Inhalt des letzten Vordergrundes überlagert und der Vordergrund wird gelöscht. Wurde Option A gewählt, werden sowohl Hintergrund als auch Vordergrund gelöscht.

Auf diese Weise wird automatisch ein neuer Arbeitsbildschirm erstellt.

#### *Achtung*

Es ist möglich, diese automatische Bildschirmvorbereitung für die nächste Phase zu übergehen, indem man das Arbeitsbildschirm-Kommando  0  1 verwendet, das später beschrieben wird, und das beide Bereiche unverändert läßt.

Die Daten werden in den Vordergrund eingegeben und nur vom Vordergrund aus in den Hintergrund gebracht. Das Kommando **[F]**, das später beschrieben wird, gibt Ihnen die Möglichkeit zur weiteren Bearbeitung dieser Bereiche. Damit ist es möglich, eine Bildschirm-Bild-Datei (name.Snn) eines vorhergehenden Durchlaufs in den Vordergrund (oder auch in den Hintergrund) zu kopieren. Dadurch werden Änderungen von bereits bestehenden Formularen oder Masken möglich.

Während der Dateneingabe in den Arbeitsbildschirm (d.h. in den Vordergrund) kann in folgenden zwei Modi gearbeitet werden:

- **Edit-Modus** Der Modus, in dem Sie Daten eingeben, um eine Maske oder deren Eingabefelder zu definieren. Der Anfangsmodus ist immer der Edit-Modus.
- **Kommando-Modus** In diesem Modus können Kommandos abgesetzt werden, die Ihnen den gerade in Bearbeitung befindlichen Bildschirm erstellen und ihn bearbeiten.

### **EDIT-Modus**

Sie können in diesem Modus den Cursor mit Hilfe der Cursortasten zu jeder beliebigen Stelle auf dem Bildschirm führen. Auf dem gesamten Bildschirm können Daten zum Aufbau der Maske eingegeben werden, in Übereinstimmung mit dem Bildschirm-Typ (fester oder variabler Teil), der zu Beginn der Arbeitsphase ausgewählt wurde.

### **Fester Text**

Zur Erstellung des festgelegten Textes eines Formulars oder Maske (d.h. die festgelegten Felder, analog denen eines vorgedruckten Papier-Formulars) können Sie alle abdruckbaren Zeichen überall auf dem Bildschirm eingeben. Dieser Text wird dann als "Masken-Text" zur Dateneingabe während eines Programmablaufes angezeigt.

### Variable Daten

Bei der Erstellung der Felder für variable Daten eines Formulars (d.h. die die Felder analog den Leerfeldern für die Eingabe eines vorgedruckten Formulars) können Sie die Zeichen  X  Y  8  9 eingeben. Wenn Sie die variablen Datenfelder definieren, dann kennzeichnet X ein alphanumerisches Zeichen und 9 ein numerisches Zeichen. Sollten Sie zwei unmittelbar aufeinanderfolgende alphanumerische Felder benötigen und diese getrennt ansprechen wollen, dann geben Sie im zweiten Feld an Stelle des Zeichens X das Zeichen Y ein. In gleicher Weise wird für unmittelbar aufeinanderfolgende numerische Felder das Zeichen 9 durch das Zeichen 8 ersetzt.

## Beispiel

---

### *Beispiel*

Die Maske hat folgendes Aussehen:

Rechnungsnummer: ..... :

Nehmen wir an, daß der Anwender während des Programmablaufs eine Rechnungsnummer eingeben muß. In unserem Beispiel lautet der festgelegte Text "Rechnungsnummer: ..... :".

Die einzugebende Rechnungsnummer soll im Aufbau folgendermaßen aussehen: AB1234.

Sie müssen nun das Eingabefeld und den Typ dieses variablen Feldes für das Programm genau definieren. Wenn also die Rechnungsnummer immer zwei alphanumerische Zeichen, gefolgt von vier numerischen Zeichen lang ist, dann müssen Sie an der Stelle des Bildschirms (in diesem Beispiel die gepunktete Linie) XX9999 eingeben.

Wollen Sie aber in einem Feld Betrag mit 7 aufeinanderfolgenden Ziffern

5 Stellen für DM und

2 Stellen für Pfennig

getrennt ausgewiesen haben, dann definieren Sie auf dem Bildschirm

9999988

Die ersten fünf Stellen und die zwei Stellen rechts werden bei der Datenbeschreibung getrennt angegeben, obwohl sieben numerische Stellen **hintereinander** stehen.

Zusätzlich können spezielle Aufbereitungszeichen eingegeben werden, um numerisch aufbereitete Felder anzugeben.

### *Anmerkung*

Diese Felder sollten durch Zwischenräume getrennt sein. Numerisch aufbereitete Felder sind in der LEVEL II COBOL Sprachbeschreibung beschrieben. Die gültigen Zeichen sind:

Z, \*, +, -, CR, DB, . (Punkt), , (Komma), B, /, 0 (Null), \$

Das \$-Zeichen ist das Währungszeichen und kann durch ein anderes Zeichen, das in der SPECIAL-NAMES-Klausel des LEVEL II COBOL-Programms angegeben wurde, entweder direkt oder indirekt während der Einrichtungs-Phase des FORMS-2-Laufs ersetzt werden (s. Kapitel 2).

### *Anmerkung*

Die Zeichen S,V und P sind nicht erlaubt.

## Kommando-Modus

---

### Kommando-Modus

Um vom Edit-Modus in den Kommando-Modus zu schalten, drücken Sie ↓.

Den Kommando-Modus erkennen Sie durch Unterstreichungs-Zeichen (\_\_) in der linken oberen Ecke. Der Cursor kann nur innerhalb dieser beiden Zeichen bewegt werden.

Geben Sie ein Kommando ein und dann ↓.

Wenn die Ausführung eines Kommandos abgeschlossen ist, führen alle Kommandos außer Leertaste  und  in den Edit-Modus zurück, d.h. der Cursor geht auf die ursprüngliche Position in der Maske.

Das durch Voreinstellung gesetzte Kommando ist das Unterstreichungs-Zeichen (\_\_) und dies führt in den Edit-Modus.

### 3.2 Kommandos

Die Kommandos, die während der Arbeitsphase zur Verfügung stehen, lassen sich in drei Hauptgruppen einteilen.

Alle Kommandos werden eingegeben, indem Sie die entsprechende Taste drücken und dann mit  abschließen.

#### 1. Allgemeine Kommandos

Allgemeine Kommandos führen Funktionen aus, wie z.B. die Freigabe des Arbeitsbildschirms für die weitere Verarbeitung.

#### 2. Arbeitsbildschirm-Bearbeitungs-Kommandos

Diese Kommandos helfen bei der Vorbereitung und beim Aufbereiten des Arbeitsbildschirms. Machen Sie sich mit diesen Kommandos besonders vertraut!

#### 3. Programmier-Kommandos

Programmier-Kommandos wurden vor allem als Hilfe für COBOL-Programmierer eingeführt und geben ohne COBOL-Kennntnis keinen Sinn. Sie beinhalten auch Kommandos, die Ihnen helfen sollen, einen wirkungsvollen Code zu erstellen und die Ihnen eine größere Kontrolle über die Datei-Ausgabe ermöglichen.

#### *Anmerkung*

Die Kommandos der Gruppe 1 und 2 sind in den HELP-Bildschirmen 1 und 2 (H01, H02) zusammengefaßt, die Kommandos der Gruppe 3 auf den HELP-Bildschirmen 3 und 4 (H03, H04).

### 3.2.1 Allgemeine Kommandos am Arbeitsbildschirm

? □

#### Anzeige des Hilfs-Bildschirms

Nach Eingabe von ? gefolgt von ↵ wird der erste HELP-Bildschirm angezeigt, der eine Zusammenfassung der allgemeinen Kommandos beinhaltet. Dieser Bildschirm bleibt bestehen, bis weitere Eingaben gemacht werden. Geben Sie dann nur ↵ ein, und Sie kehren in den Edit-Modus zurück.

Die Eingabe von ? gefolgt von ↵ bewirkt die Anzeige des nächsten HELP-Bildschirms, der eine Zusammenfassung der Arbeitsbildschirm-Bearbeitungs-Kommandos enthält. Wird die Eingabe des Kommandos ? und ↵ wiederholt, so erscheinen nacheinander die HELP-Bildschirme H03 und H04. Anschließend kommt wieder der erste HELP-Bildschirm.

Das ?-Kommando ist auch vom ersten Bildschirm W01 (Textauswahl-Typus) aus möglich.

? □ n

Gibt den durch n angegebenen HELP-Bildschirm aus.

#### Beispiel

? 3 ↵ erzeugt den Bildschirm H03.

□

(Leertaste) beendet die Arbeitsphase.

Die Eingabe der Leertaste □ beendet die aktuelle Arbeitsphase und startet die Bearbeitung der gerade in diesen Arbeitsbildschirm eingegebenen Daten.

□

(Unterstrich) Rückkehr in den Edit-Modus

Mit Eingabe von □ verlassen Sie den Kommando-Modus und kehren in den Edit-Modus zurück.

Alle Kommandos außer □ ! □ ermöglichen die Rückkehr in den Edit-Modus.

Q

### Rückkehr zum Bildschirm W01

Mit der Eingabe von Q kehren Sie zum Bildschirm W01 (Auswahl des Text-Typ) zurück. Welchen Text-Typ Sie auch wählen, die aktuellen Vordergrund/Hintergrund-Komponenten des Arbeitsbildschirms werden bei Rückkehr in den Edit-Modus nicht verändert.

Sie können dieses Kommando verwenden, wenn Sie irrtümlich die voreingestellte Wahlmöglichkeit C (variable Datenfelder) am Bildschirm W01 ausgewählt haben und nun am Bildschirm den festen Teil einer Maske aufgebaut haben. Bei der anschließenden Verarbeitung kehrt FORMS-2 automatisch wieder in den Edit-Modus zurück, da er einen Fehler bei der Definition von variablen Datenfeldern annimmt.

Mit dem Kommando Q können Sie nun zum Menü W01 zurückkehren und die Wahl des Bildschirmtyps korrigieren, ohne daß der Inhalt des eben editierten Bildschirms verloren geht. Dann können sie die eingegebenen Daten als festen Teil einer Maske verarbeiten. Sie können Q auch von Bildschirm I02 aus eingeben, dann erscheint wieder der Bildschirm I01.

I

### Beenden des FORMS-2-Laufs

Wenn Sie I drücken, beendet sich das Programm. Dieses Kommando steht Ihnen während des ganzen FORMS-2-Programmlaufs zur Verfügung, aber es wird nur nach einer Arbeitsphase zur Rückkehr zur Maske W01 verwendet.

Bei sonstiger Verwendung bricht es den Lauf ab.

X

### Änderung des Kommandobereichs

Der Standard-Kommandobereich ist in den ersten beiden Spalten der ersten Zeile, und er wird durch zwei Unterstreichungs-Zeichen gekennzeichnet. FORMS-2 zeigt die Daten, die an dieser Stelle eingegeben wurden, bei der Rückkehr in den Edit-Modus wieder an.

Wollen Sie jedoch **Daten** an dieser Stelle stehen haben, können Sie den Kommandobereich versetzen. Sie setzen nur den Cursor an die gewünschte Stelle, ehe Sie zum Kommando-Modus zurückkehren. Geben Sie nun  ein und wenn Sie das nächste Mal den Kommando-Modus erreichen, stehen die beiden Unterstreichungs-Zeichen an der neuen Stelle.

#### *Beachten Sie*

Der Kommandobereich kann nur bis zur Stelle 77/78 nach rechts verschoben werden.

\*

### Definition der Trennung zwischen Satzschlussel und Daten im Eingabesatz

Dieses Kommando gilt nur bei der Generierung des Index-Programms und wird in Kapitel 8 beschrieben.

#### *Anmerkung*

Wenn in der Maske I02 die Programmoption  gewählt wurde, dann sollte vor Beendigung der zweiten Arbeitsphase, der Cursor auf die erste Datenposition der variablen Daten gebracht werden, die nicht mehr Bestandteil des Satzschlussels ist und dann das Kommando  und  eingegeben werden.

Wenn diese Eingaben nicht korrekt erfolgen, dann kehrt FORMS-2 solange in den Edit-Modus zurück, bis die Anweisungen richtig eingegeben wurden.

### 3.2.2 Arbeitsmasken-Bearbeitungs-Kommandos

Die Kommandos **F** und **O** sind vorbereitende Kommandos.

- F** Ruft die Maske W02 für die Vordergrund/Hintergrund-Bearbeitung auf. Mit dem Bildschirm W02 baut man den Vordergrund des Arbeitsbildschirms auf. Das Konzept Vordergrund/Hintergrund ist am Anfang dieses Kapitels beschrieben worden.

Sie haben folgende Auswahlmöglichkeiten im Bildschirm W02:

- A** Rückkehr in den Edit-Modus
- B** Löschen des Vordergrundes
- C** Löschen des Hintergrundes
- D** Überlagern der Hintergrunddaten in den Vordergrund
- E** Überlagern der Vordergrunddaten in den Hintergrund

**Beachten Sie**

Bei den beiden Kommandos **D** und **E** ist der Quelldatenbereich nach Ausführung des Kommandos immer gelöscht. Wollen Sie die Datenbereiche löschen, verwenden Sie die Kommandos **F B** und **F C**.

- F** Überlagern des Vordergrunds mit einer bestehenden Maske. Hiermit können Sie bestehende Masken (auch von früheren FORMS-2-Läufen) ändern. Sie müssen dann noch den Namen der gewünschten Datei eingeben.
- G** Überlagern einer bestehenden Maske in den Hintergrund
- H** Anzeigen des Vordergrundes. Dieses Kommando zeigt nur die Vordergrundkomponente des Arbeitsbildschirms zur Überprüfung an. Bei Rückkehr in den Edit-Modus werden wieder Vordergrund und Hintergrund angezeigt.

**I** Anzeigen des Hintergrundes.  
Dieses Kommando zeigt nur die Hintergrundkomponente des Arbeitsbildschirms zur Überprüfung an. Bei Rückkehr in den Edit-Modus werden wieder Vordergrund und Hintergrund angezeigt.

**J** Anzeigen einer bestehenden Maske.  
Sie müssen den Dateinamen eingeben. Die angegebene Datei (Maske) wird am Bildschirm angezeigt, ohne jedoch den Vordergrund oder Hintergrund zu verändern. Dieses Kommando ermöglicht eine Überprüfung der Maske, ehe sie mit **D** in den Arbeitsbildschirm kopiert wird.

*Anmerkung*

Bei Auswahl von **H** **I** und **J** bleibt die Bildschirmanzeige solange bestehen, bis Sie **↓** drücken.

**F** **x** Aufrufen der oben beschriebenen Möglichkeiten des Bildschirms W02.

x ist der Options-Code, wie er in Bildschirm W02 enthalten ist. Die ausgewählte Option wird ausgeführt, die Kontrolle geht an den Edit-Modus zurück, ohne daß der Bildschirm W02 zwischenzeitlich angezeigt wird.

**O** **1** Ausschalten automatischer Hintergrund/Vordergrund-Vorbereitung.  
Die Hintergrund/Vordergrund-Vorbereitung ist am Anfang dieses Kapitels unter Vordergrund/Hintergrund beschrieben worden. Dieses Kommando verhindert, daß der Vordergrund automatisch in den Hintergrund gemischt oder beide Bereiche als Vorbereitung für die nächste Maske gelöscht werden.

Das Kommando **O** **1** bleibt gültig, bis das Kommando **O** oder **O** **0** (siehe unten) eingegeben wird.

oder

Wiedereinschalten der automatischen  
Vordergrund/Hintergrund-Vorbereitung

Die Vorbereitungssequenz für Hintergrund/Vorder-  
grund wird wieder automatisiert (sie beginnt mit der  
nächsten Arbeitsphase).

Die Kommandos **C** **D** **I** **K** und **A** sind Kommandos, mit denen Zeilen- und Spaltenbereiche verändert werden können. Sie werden durch die Stellung des Cursors kontrolliert, sobald der Kommando-Modus eingeschaltet ist.

**C** **n**

Zwischenräume einfügen

Es werden n (1 - 9) Zwischenräume vor das Zeichen gesetzt, an dem der Cursor gerade steht. Es wird nur diese Zeile bearbeitet.

**D** **n**

Löschen von Zeichen

Es werden n (1 - 9) Zeichen gelöscht, einschließlich des Zeichens an der bestehenden Cursorposition. Es wird nur diese Zeile bearbeitet.

**I** **n**

Leerzeilen einfügen

Es werden n (1 - 9) Zeilen vor der Zeile eingefügt, in der der Cursor gerade steht, ungeachtet der Spaltennummer. Es können nur ganze Zeilen eingefügt werden.

**K** **n**

Zeilen löschen

Es werden n (1 - 9) Zeilen gelöscht, einschließlich der Zeile, in der der Cursor gerade steht. Es können nur ganze Zeilen gelöscht werden, wenn Sie dieses Kommando verwenden.

**A** **n**

Wiederholung der Zeile

Die Zeile, in der der Cursor steht, wird n (1 - 9) Mal wiederholt.

**Anmerkung**

Mit diesem Kommando können Sie nichts einfügen. Jede Vordergrund-Information der nächsten n Zeilen wird überschrieben.

Mit den Kommandos U und V können Sie den Cursor positionieren.

Horizontale (waagerechte) Verschiebung des Cursors erreichen Sie mit den Tasten → und ←.

Vertikale (senkrechte) Verschiebung ist teilweise möglich durch die Tasten ↓ und ↑, aber der Cursor steht anschließend immer in Spalte 1.

Vertikale Cursorverschiebung innerhalb einer Spalte kann bei Erstellung eines Formulars nötig werden. Dafür stehen zwei "Tabulator"-Kommandos zur Verfügung:

U n

Cursor nach oben

Bewegt den Cursor um n (1 - 9) Zeilen von der bestehenden Cursorposition aus nach oben. Die Cursorposition bleibt innerhalb der Zeilen erhalten.

V n

Cursor nach unten

Bewegt den Cursor um n (1 - 9) Zeilen von der bestehenden Cursorposition aus nach unten. Die Cursorposition bleibt innerhalb der Zeilen erhalten.

### 3.2.3 Programmier-Kommandos

G

Aufbau des Datennamens verändern

Dieses Kommando ist dann nicht möglich, wenn Sie am Bildschirm I02 die Option G (Index-Programm) gewählt haben.

Das vorgegebene Format des Datensatznamens, das von FORMS-2 generiert wird (zur späteren Übernahme in das COBOL-Quellprogramm), hat folgenden Aufbau:

bbbbbb-rr (01 level)

dabei ist:

bbbbbb

der bis zu sechs Zeichen lange Name, der im Bildschirm I01 definiert wurde.

rr

die Satznummer, beginnend bei 00 im ersten Arbeitsschritt. Sie wird jeweils um 1 für jeden folgenden Arbeitsschritt hochgezählt.

#### *Anmerkung*

Wenn die Fenster-Kommandos dazu verwendet wurden, um ein Fenster zu definieren, das an anderer Stelle als in Zeile 1 beginnen soll, dann heißt der generierte voreingestellte Datensatzname:

bbbbbb-rr-ii

dabei ist ii die Zeilennummer.

FORMS-2 generiert eine Struktur mit den elementaren Datennamen. Diese Datennamen können Sie in Ihr COBOL-Quellprogramm kopieren und für eine formatierte Masken-Ein/Ausgabe übernehmen.

Die Struktur sieht folgendermaßen aus:

bbbbbb-rr-nnnn

dabei ist:

nnnn

die Stellung innerhalb der Reihenfolge der Felder dieser Maske, beginnend bei 0001.

### *Anmerkung*

G (oder G 0) bewirkt, daß nnnn die Koordinaten des Feldes innerhalb der Maske darstellt. Das kann für den Programmierer manchmal von Nutzen sein, wenn er die COBOL-Möglichkeiten nutzt, um den Cursor auf eine bestimmte Position zu setzen.

G 1

schaltet die voreingestellte Funktion 'Durchnumerieren der Felder' wieder ein.

Die Kommandos **J** und **M** verändern die Zwischenraumbehandlung.

Mit den LII COBOL-Befehlen ACCEPT und DISPLAY werden nur Felder angesprochen, die nicht mit FILLER definiert sind. FILLER-Bereiche werden nicht angenommen oder ausgegeben.

Die Zeit, die benötigt wird, um eine Maske auszugeben, hängt sowohl von der Größe als auch von der Anzahl der auszugebenden Felder ab.

Wenn der festgelegte Textteil von Masken bearbeitet wird, erzeugt FORMS-2 in der Voreinstellung FILLER, dort wo mehr als ein Zwischenraum auftritt. Bei manchen Masken bewirkt dies natürlich viele kleine Felder, die durch FILLER-Felder getrennt sind. Dieses Problem lösen wir so:

**J n**

#### Zusammenfassen mehrerer Zwischenräume

Das Kommando **J n** verkleinert die Anzahl der generierten Felder durch Zusammenfassen mehrerer Zwischenräume. FORMS-2 generiert erst dann ein neues Feld, wenn im Maskentext mehr als n Zwischenräume hintereinander auftreten. Diese Anzahl ist zu Beginn auf 1 festgesetzt. n kann eine Zahl zwischen 0 und 9 sein.

#### Anmerkung

Für J oder J0 gilt ebenfalls  $n = 1$ .

Eine weitere Möglichkeit, um Zwischenräume innerhalb von Feldern zuzulassen, ist die Verwendung des Unterstreichungs-Zeichens. Bei Angabe des Unterstreichungs-Zeichens wird in der zugehörigen VALUE-Klausel ein Zwischenraum an der gleichen Stelle generiert.

Wenn Sie das Unterstreichungs-Zeichen durch irgend ein anderes Zeichen ersetzen wollen (z.B. weil das Unterstrich-Zeichen im Text vorkommt), benutzen Sie das **M** Kommando.

M x

### Verändern des Unterstreichungszeichens

Das M x Kommando tauscht das Unterstreichungszeichen mit dem unter x eingegebenen Zeichen aus.

#### *Anmerkung*

Wenn das Zwischenraum-Zeichen   selbst für x angegeben wird, werden für den gesamten Bildschirm Datenfelder erzeugt (ohne FILLER).

S n

## Dateiausgabe Kontroll-Kommando

Dieses Kommando steht Ihnen dann nicht zur Verfügung, wenn Sie in Bildschirm I02 die Option G (Index-Programm) eingegeben haben.

S oder

S 0

hebt alle anderen Sn-Kommando auf.

S 1

unterdrückt die Generierung der Dateien name.DDS und name.CHK für den aktuellen Arbeitsbildschirm. Für die nächste Maske werden die beiden Dateien wieder erzeugt, wenn das gleiche Kommando nicht noch einmal gegeben wird.

S 2

unterdrückt die Generierung des Bildschirm-Masken-Textes (Snn) für diesen Arbeits-Bildschirm. Normalerweise wird dieses Kommando dazu verwendet, um die Datei name.Snn zu unterdrücken, wenn man die variablen Eingabefelder definiert. Die Wirkung des Kommandos bezieht sich nur auf die gegenwärtige Phase.

S 3

Mit diesem Kommando können Sie den Namen der Datei, in die der aktuelle Arbeits-Bildschirm abdruckbar abgelegt wird (Snn), verändern.

Besteht bereits eine Datei mit dem vorgegebenen Namen, können Sie wählen, ob Sie die Datei überschreiben wollen. Sollten Sie dies nicht wünschen, werden Sie aufgefordert, einen anderen Dateinamen einzugeben.

S 3 fordert Sie auf, einen neuen Dateinamen einzugeben, auch dann, wenn keine Datei mit dem vorgegebenen Namen besteht.

S 9

Mit diesem Kommando erreichen Sie, daß FORMS-2 nach Anzeige jeder einzelnen Zeile des generierten Codes anhält, während FORMS-2 die Datei DDS generiert. FORMS-2 verarbeitet jede Zeile wieder, ehe sie in der .DDS-Datei abgelegt wird. Sie können damit die angezeigte Zeile ausgeben, was von Nutzen sein kann.

Dieses Kommando ist jedoch nicht möglich, wenn Sie Option C oder G (für generierte Programme) auf Bildschirm I02 angegeben haben.

P

### Cursor-Koordinaten-Kommando

Mit diesem Kommando werden die Koordinaten der gegenwärtigen Cursorposition im Kommandobereich (linke obere Ecke) angezeigt. Diese Anzeige bleibt für einige Sekunden stehen. Danach wird der Arbeits-Bildschirm im Kommandofeld wieder hergestellt und das Programm geht automatisch in den Edit-Modus zurück. Verwenden Sie im Aufbau der Datenfeld-Namen eine durchlaufende Numerierung, so haben Sie mit diesem Kommando eine einfache Methode, sich der Koordinaten jedes einzelnen Feldes zu vergewissern.

W

Fenster-Kommandos

Diese Kommandos stehen Ihnen dann nicht zur Verfügung, wenn Sie Option **G** (für das Index-Programm) auf Bildschirm I02 ausgewählt haben.

Das "Fenster" definiert den Bereich (ganze Zeilen), der bearbeitet werden soll, wenn FORMS-2 den DDS-Text generiert. Normalerweise ist das "Fenster" der ganze Bildschirm. Wo der Anfang oder das Ende des Fensters nicht mit dem Anfang oder Ende des Bildschirms übereinstimmt, kann wahlweise eine gestrichelte Begrenzungslinie genau eine Zeile **außerhalb** des Fensters angezeigt werden, z.B. wenn das Fenster in Zeile 4 beginnt, dann erscheint die Begrenzung entlang der Zeile 3.

Das Fenster-Kommando dient dazu, eine Maske in einem beliebigen Zeilenbereich aufzubauen, und gleichzeitig insofern Speicherplatz zu sparen, da Sie am oberen Bildschirmrand keine Leerzeilen definieren.

*Anmerkung*

Wenn das Fenster in dieser Weise benutzt wird, beinhaltet der generierte Datensatz die Zeilennummer, an der das Fenster beginnt. Damit weiß man dann, wie man die ACCEPT/DISPLAY AT-Koordinaten vergeben muß.

Mit folgenden Kommandos können Sie das Format des Fensters wählen:

**W** oder  
**W** **0**

Setzt den Cursor an den Beginn des aktuellen Fensters. Dieses Kommando ist der **↵** oder **⏏** Taste gleichgesetzt, wenn Sie die Fenster-Möglichkeiten nutzen.

**W** **1**

setzt den Beginn des Fensters auf die gegenwärtige Zeile, Begrenzungslinie ist eine Zeile vorher.

- W 2 setzt das Ende des Fensters auf die gegenwärtige Zeile mit der Begrenzungslinie in der nächsten Zeile.
- W 3 setzt den Beginn des Fenster auf die gegenwärtige Zeile ohne Begrenzungslinie.
- W 4 setzt das Ende des Fensters auf die gegenwärtige Zeile ohne Begrenzungslinie.
- W 5 gibt die Begrenzung aus, die dem aktuellen Beginn des Fensters vorausgeht.
- W 6 gibt die Begrenzung aus, die dem aktuellen Beginn des Fensters folgt.
- W 7 löscht die Anfangs-Begrenzung und zeigt wieder die Daten des Arbeits-Bildschirms.
- W 8 löscht die Ende-Begrenzung und zeigt wieder die Daten des Arbeits-Bildschirms.
- W 9 bringt den Cursor an das Ende des aktuellen Fensters.

### *Anmerkung*

1. Die Begrenzungen beeinträchtigen nicht den Inhalt des Hintergrundes bzw. des Vordergrundes.
2. Die ausgegebene Bildschirm-Maske beinhaltet den kompletten Vordergrund des Arbeits-Bildschirms ohne Begrenzungen, ganz unabhängig davon, ob ein Fenster definiert wurde oder nicht. Man kann deshalb Anmerkungen auf der Bildschirm-Maske einfügen, ohne daß dies das Generieren von DDS beeinträchtigt.
3. Die Verwendung des Fensters hat einen weiteren Vorteil. Wenn es nötig ist, können Sie sich das Formular (Maske) in zwei Abschnitten anzeigen lassen: die ersten 10 Zeilen und dann die zweiten 10 Zeilen. Diese können als eine einzelne Bildschirm-Maske, die beide Teile des Formulars (Maske) enthält, aufgebaut werden. Sie können so, wenn der DDS-Text generiert wird, in die entsprechenden Abschnitte hineinsehen.

### 3.2.4 Beenden der Arbeitsphase

Zum Beenden der Arbeitsphase von FORMS-2 wählen Sie den Kommando-Modus, drücken zunächst die Leertaste und dann . Das Kommando  verarbeitet den Arbeits-Bildschirm.

FORMS-2 beendet die Arbeitsphase (abhängig von der Auswahl der Datei auf Bildschirm I02) folgendermaßen:

1. Definiert diese Arbeitsphase variable Datenfelder, ( oder  auf dem Bildschirm W01), dann wird die beginnende Verarbeitung wie folgt angezeigt:

```
WORK SCREEN VALIDATION in progress - DO NOT press  
<RETURN >
```

2. Haben Sie auf Bildschirm I02 das Generieren einer DDS-Datei ausgewählt, wird der produzierte Quellcode am Bildschirm so angezeigt, wie er geschrieben wird. Wurde das Kommando   gegeben, dann hält die Verarbeitung nach jeder Code-Zeile an, und ermöglicht Ihnen, daß Sie die Zeile noch ändern. Diese Verwendung empfehlen wir Ihnen nur, wenn besondere Umstände es nötig machen.
3. Haben Sie auf Bildschirm I02 eine Bildschirm-Masken-Datei (Snn) gewählt, dann wird die Maske am Bildschirm angezeigt, so wie sie in die Datei geschrieben wird. Der Name der erzeugten Datei wird angezeigt. Sie müssen die Taste  drücken, um den FORMS-2-Lauf fortzusetzen.
4. Bildschirm W01 wird wieder angezeigt, so daß Sie den Ablauf entweder beenden oder fortführen können.

### Anmerkung

1. Während der Verarbeitung von variablen Daten sind nur jene Zeichen erlaubt, die in der Beschreibung der Text-Typen angegeben wurden.  
Wird ein anderes Zeichen verwendet, wird von der Verarbeitungsroutine ein Fehler gemeldet, indem abwechselnd "?" und das falsche Zeichen angezeigt werden. Diese Fehleranzeige hört dann auf, FORMS-2 kehrt in den Edit-Modus zurück und der Cursor steht unter dem falsch eingegebenen Zeichen. Sie müssen dann nach jeder Korrektur wieder die Kommandos   eingeben.
2. FORMS-2 erlaubt Aufbereitungs-Zeichen, aber überprüft nicht, ob die Kombination dieser Zeichen gültig ist. Deshalb müssen die Regeln des LII COBOL beachtet werden, damit Sie einen fehlerfreien Code bekommen. Beachten Sie, daß diese Felder durch Zwischenräume getrennt werden sollten.
3. Es werden nur Daten des Vordergrundes in der Masken-Datei abgelegt.



# 4 Datenbeschreibungen

In diesem Kapitel werden die COBOL Datenbeschreibungen erläutert, die FORMS-2 in der Datei Dateiname.DDS generiert. Damit Sie dieses Kapitel gut verstehen, sollten Sie COBOL-Kenntnisse haben.

Die Erweiterungen zu den ACCEPT- und DISPLAY-Befehlen erlauben Ihnen eine umfassende Ein-Ausgabe am Bildschirm in einer Benutzer-Anwendung (siehe LEVEL II COBOL Bedienungsanleitung). Das Programmieren der notwendigen Datenerklärungen ist langweilig und teuer bezüglich der Programmiererzeit, auch deshalb, weil es für einfache Fehler sehr anfällig ist.

FORMS-2 vereinfacht das Erstellen von fehlerfreien Datenbeschreibungen, indem Sie Bildschirmanzeigen (Formulare) in bequemer Weise angeben können: Sie können sie direkt auf dem Bildschirm aufbauen, so wie es in Kapitel 2 und 3 beschrieben wurde. Wenn Sie diese Möglichkeit durch die Auswahl einer entsprechenden Option auf Bildschirm I02 während der Einrichtungs-Phase aufgerufen haben, dann wandelt FORMS-2 die Eingabe in die notwendigen COBOL-Statements um und legt diese in einer Datendefinitions-Datei (DDS) ab. Sie rufen in Ihrem Programm mit COPY nur noch diesen Code auf und verwenden die Datensatznamen und Datennamen, die von FORMS-2 generiert wurden.

## 4.1 Generieren von Datensatz- und Datennamen

Beim Bildschirm I01 wird ein Basis-Name verlangt.

Dies ist ein 6-Zeichen langes Feld, in das Sie einen beliebigen Namen eintragen können.

Diese Basis wird dann dazu verwendet, die COBOL-Datennamen zu generieren.

### 4.2 Datensätze

Das vorgegebenen Format des von FORMS-2 generierten Datensatznamens, ist folgendermaßen:

bbbbbb-rr (01 level)

dabei ist

bbbbbb

die durch 1 - 6 Zeichen auf Bildschirm I01 angegebene Basis, und

rr

die Datensatznummer, die in der ersten Arbeitsphase bei 00 beginnt, und in jeder weiteren Arbeitsphase um eins hochgezählt wird.

#### *Anmerkung*

Werden die Fenster-Kommandos benutzt, um ein Fenster zu definieren, das in einer anderen Zeile als Zeile 1 beginnt, dann lautet der generierte Datensatzname:

bbbbbb-rr-ii

wobei ii die Zeilen-Nummer ist. Achten Sie dann bitte darauf, wenn Sie die entsprechenden ACCEPT/DISPLAY-Befehle codieren.

#### **Benennen der Datennamen**

Die Struktur der Datennamen, die FORMS-2 erzeugt, ist:

bbbbbb-rr-nnnn

dabei ist

nnnn

die Sequenznummer dieses Feldes innerhalb des Bildschirms, sie beginnt bei 0001.

Manchmal kann es günstiger sein, wenn die Bildschirmkoordinaten statt der Feldnummern in den Datennamen stehen. Dies geschieht durch die Verwendung von  während der Arbeitsphase.

### PICTURE-Klauseln

Das Generieren von PICTURE-Klauseln durch FORMS-2 hängt davon ab, welchen Text-Typ Sie auf Bildschirm W01 zu Beginn jeder Arbeitsphase auswählen. Beachten Sie dabei, daß FORMS-2 Feldbegrenzungen in jeder Zeile anbringt.

### Fester Text

Am Ende einer Arbeitsphase für festgelegten Text generiert FORMS-2 nur FILLER-Bereiche oder benannte alphanumerische Felder mit zugehörigen VALUE-Klauseln.

Die LEVEL II COBOL interaktiven ACCEPT- und DISPLAY-Befehle arbeiten nur mit Feldern mit Namen; FILLER-Bereiche werden nicht angezeigt oder ausgegeben. Die Zeit, die benötigt wird, um eine Bildschirmanzeige zu erstellen, hängt sowohl von der Größe als auch von der Anzahl der gewählten Felder ab.

Werden Masken mit festgelegtem Text verarbeitet, generiert FORMS-2 durch Voreinstellung FILLER, dort wo mehrere Leerstellen hintereinander erscheinen. Diese Voreinstellung kann durch Eingabe von  J, wie in Kapitel 3 beschrieben, geändert werden.

Alternativ kann der Unterstrich ( \_ ) dazu verwendet werden, um Zwischenräume in VALUE-Klauseln einzufügen. Das voreingestellte Zeichen, das für diesen Zweck verwendet wird, kann mit  M in ein anderes Zeichen geändert werden, wie in Kapitel 3 beschrieben.

### Felder für variable Daten

Am Ende einer Arbeitsphase für variable Daten generiert FORMS-2 alphanumerische, numerische oder numerisch druckaufbereitete Felder, abhängig von den Zeichen, die Sie eingeben. Dies sind gewöhnlich die normalen LEVEL II COBOL-PICTURE-Zeichen 9 und X. Sie können zusätzlich 8 und Y als Alternativen zu 9 und X verwenden. Beachten Sie auch, daß Sie die Zeichen S, V und P nicht verwenden können, wie dies im Kapitel 3 unter "Variable Daten" beschrieben ist. Dort, unter 3.1, ist auch ein Beispiel aufgeführt.

### Bearbeiten der DDS-Datei

Normalerweise genügt es, die DDS-Datei von FORMS-2 ausgeben zu lassen. Will man jedoch spezielle Datennamen verwenden oder nicht erlaubte PICTURE-Zeichen, so können mit dem `S 9`-Kommando (siehe Kapitel 3) DDS-Zeilen vor der Ausgabe bearbeitet werden. Ein herkömmlicher Text-Editor kann auch dazu verwendet werden, die Datei zu verändern. Beachten Sie jedoch, daß diese Bearbeitung wiederholt werden muß, wenn das Formular durch FORMS-2 verändert wurde.

Mit dem `S 1`-Kommando ist es auch möglich, die Ausgabe in die Datei `name.DDS` für eine spezielle Arbeitsphase vollständig zu unterdrücken. Wenn Sie diese Möglichkeit gewählt haben, sollten Sie bedenken, daß die in die Datennamen eingefügte Satznummer für die nächste Arbeitsphase dennoch um 1 hochgezählt wird.

### Aufruf einer DDS-Datei

Wollen Sie die generierte Datenbeschreibung in Ihr Anwenderprogramm einbringen, brauchen Sie nur die DDS-Datei mit der COPY-Anweisung in Ihr Programm hineinkopieren.

Die COPY-Anweisung, die das DEMO1-Formular-Beispiel von Kapitel 7 ins Programm kopiert, lautet:

```
COPY "DEMO1.DDS".
```

Diese Anweisung wird in der DATA DIVISION angegeben.

## 5 Das CHECK-OUT-Programm

### 5.1 Allgemeines

Dieses Kapitel beschreibt das Check-Out-Programm, das FORMS-2 automatisch generieren kann, während die erstellten Masken generiert werden. Mit dem Check-Out-Programm können Sie:

- die DDS-Datei überprüfen,
- sehen, wie die geplante Anwendung funktioniert,
- das Formular für die Dateneingabe kontrollieren,
- das Formular bei Datenänderung testen.

Das in COBOL geschriebene Check-Out-Quellprogramm ruft die DDS-Datei auch mit einer COPY-Anweisung auf, genau so, wie Sie es in einem Programm tun.

Die Logik des Check-Out-Programms ist eine Folge von DISPLAY- und ACCEPT-Befehlen für die in diesem FORMS-2-Lauf definierten Bildschirmmasken, und zwar in der Reihenfolge, in der sie erstellt wurden.

Damit können Sie mit einem Demonstrationsprogramm alle erzeugten Masken einfach und schnell prüfen, ohne daß Sie selbst ein Ein/Ausgabeprogramm erstellen.

Trotzdem könnte die beste Methode für umfangreiche Anwendungen sein, jede Maske einzeln zu erstellen, indem Sie allein die Bildschirmmasken-Ausgabe (Snn) verwenden.

FORMS-2 kann dann noch einmal durchlaufen werden, um das gewünschte Check-Out-Programm zu produzieren. Sie können dann die bereits erzeugten Masken noch einmal mit dem Kommando   auf den Bildschirm bringen. Diese Möglichkeit erlaubt es auch, viele Masken zu Demonstrationszwecken zusammenzustellen, wobei Sie die gleiche Maske mehrmals einbauen können.

Nachdem die Maskenfolge durchlaufen ist, fragt das Check-Out-Programm, ob die ganze Folge von Masken wiederholt werden soll. Im zweiten Durchlauf werden die vorher eingegebenen Daten wieder angezeigt, und erlauben Ihnen so, die Anwendung Ihrer Masken sowohl für eine Daten~~er~~steingabe als auch für eine Datenänderung zu testen.

### 5.2 Erzeugen des CHECK-OUT-Programms

Das Programm erzeugen Sie im Rahmen der Einrichtung mit der Maske I02. Dort ist schon als Vorgabe  eingetragen, das Programm wird also automatisch generiert.

Der Quell-Code des Programms steht in der Datei

basisname.CHK

dabei ist basisname der Name, den Sie auf dem Bildschirm I01 während der Einrichtungsphase eingegeben haben.

### 5.3 Übersetzung des CHECK-OUT-Programms

Das Programm basisname.CHK muß nun übersetzt werden. Die folgenden Dateien müssen während der Übersetzung zur Verfügung stehen:

basisname.DDS      die im FORMS-2-Lauf produzierte DDS-Datei.  
Sie muß in dem Dateiverzeichnis stehen, das in der Maske I02 angegeben wurde oder im aktuellen Dateiverzeichnis, falls keine anderen Angaben gemacht wurden.

FORMS-2.CH1      }  
FORMS-2.CH2      }      das Check-Out-Programm in dem  
Dateiverzeichnis /usr/lib/cobol.

Das Check-Out-Programm wird dann wie gewöhnlich übersetzt, indem Sie das Übersetzungskommando mit dem Namen Ihrer Quelldatei angeben:

```
cobol basisname.CHK 
```

Einzelheiten zur Übersetzung mit dem COBOL-Übersetzer finden Sie in Ihrer LEVEL II COBOL Bedienungsanleitung.

## 5.4 Ablauf des CHECK-OUT-Programms

### Laden

Das Programm kann dann nach der Übersetzung geladen werden:

```
cbrun name.INT 
```

### Ausführen

Die Grundfunktion des Check-Out-Programms ist die Anzeige des festen Teils der Masken bzw der Formulare. In den variablen Feldern der Maske können die Daten eingegeben werden. Die Reihenfolge der Maskenausgabe und damit die Dateneingabe ist so, wie sie ursprünglich generiert wurde.

Die Programmlogik dieses Testprogramms im einzelnen ist noch etwas komplizierter. Die folgenden Anmerkungen beziehen sich auf Optionen für den auf Bildschirm W01 gewählten Bildschirm-Typ, und diese sind in Kapitel 3 näher beschrieben.

### Masken mit festem Text

Der festgelegte Text eines Formulars wird angezeigt. Gibt es zwei aufeinanderfolgende Masken mit festem Text, hält das Check-Out-Programm nach der ersten Anzeige solange an, bis man die Taste  drückt.

### Festgelegter Text auf leerem Bildschirm

Wurde  zur Erstellung des Formulars angegeben, dann löscht das Check-Out-Programm den Bildschirm, bevor die Maske angezeigt wird.

### Festgelegter Text auf dem vorhergehenden Bildschirm

Wurde  zur Erstellung des Bildschirms verwendet, dann bleibt jede angezeigte Maske auf dem Bildschirm stehen, nur die Stellen werden überschrieben, die vom neuen Bildschirm benutzt werden, d.h., der Bildschirm wird vor der Ausgabe der neuen Maske nicht gelöscht.

## CHECK-OUT-Programm beenden

---

### Eingabefelder für Daten (variabler Teil der Maske)

Um Daten entgegenzunehmen, verwendet das Testprogramm den ACCEPT-Befehl. Dabei ist es Ihnen nur möglich, in den vorher definierten Feldern Eingaben zu machen (d.h. in den mit 9 bzw. X definierten Bereichen).

Sie können auch die Größe der Felder überprüfen. Bei numerischen Feldern können Sie auch kontrollieren, ob nur numerische Zeichen eingegeben werden können, ebenso die Auswirkung der Eingabe des Zeichens "." (bei Dezimalstellen werden von links Nullen eingefügt).

Außer beim ersten Durchgang durch die Maskenfolge werden die vorher eingegebenen Daten angezeigt, bevor der ACCEPT durchlaufen wird.

Enthält die Maske für variable Daten numerisch aufbereitete Felder, folgt dem ACCEPT für den Bildschirm ein entsprechender DISPLAY, um die Wirkung der Druckaufbereitung zu zeigen.

Beachten Sie bitte, daß die druckaufbereiteten Felder nicht automatisch am Bildschirm angezeigt werden.

### Beenden

Nachdem die vollständige Maskenfolge durchlaufen ist, zeigt das Check-Out-Programm an:

CHECK-OUT completed

Repeat? [N] (Y = Yes)

Wollen Sie die Maskenfolge wiederholen, dann geben Sie erst  Y ein und dann .

Andernfalls drücken Sie nur  und beenden damit das Programm.

## 6 Bildschirm-Masken-Datei

### 6.1 Allgemeines

Dieses Kapitel beschreibt die Bildschirm-Masken-Datei, die FORMS-2 zusätzlich zu (oder anstatt) der COBOL Datenerklärungs-Befehle generieren kann. Diese Dateien beinhalten genaue Text-Abbilder der vorher definierten Formulare. Diese Formular-Abbilder können

- als Basis verwendet werden, um das Formular zu verändern
- ausgedruckt werden, um gedruckte Exemplare des Formulars zu erhalten
- als Vorlage des Masken-Erstellers für den Anwender-Programmierer dienen.

### 6.2 Generieren der Bildschirm-Masken

Bildschirmmasken werden generiert, wenn Sie während der Einrichtungsphase bei der Maske I02 eine der Angaben  E,  C oder  D  G machen. Beachten Sie, daß die vorgegebene Option bereits eine Bildschirmmasken-Ausgabe bewirkt.

Bildschirm-Masken werden auf Dateien ausgegeben mit dem Namen:

basisname.Snn

dabei ist

basisname

der von Ihnen auf Bildschirm I01 während der Einrichtungsphase eingegebene Name und

nn

ist eine Zahl zwischen 00 und 99.

Der voreingestellte Dateiname kann mit dem Kommando  S  3 während der Arbeitsphase überschrieben werden. Dies bewirkt, daß FORMS-2 die Eingabe eines neuen Dateinamens während der Verarbeitung dieses Arbeitsbildschirms verlangt.

## Bildschirm-Masken-Datei

---

Am Ende jeder Arbeitsphase wird eine separate Datei erstellt, wobei die Ziffernfolge im Namen (nn) jedes Mal um 1 erhöht wird. Eine Bildschirm-Maskendatei ist wie eine normale sequentielle Datei aufgebaut, d.h. mit einem Datensatz für jede Zeile des Bildschirms.

Die Datei enthält nur den Text, der während **der** Arbeitsphase eingegeben wurde, in der sie aufgebaut wurde (d.h. als Vordergrund-Daten - siehe Kapitel 3). Deshalb enthält die Bildschirmmaske, in die in einer Arbeitsphase variable Daten eingegeben wurden, nur die Zeichen X, Y, 9 und 8.

Sie können die Maskenanzeige von jeder Arbeitsphase aus unterdrücken, indem Sie   eingeben.

Wenn Sie dieses Kommando verwenden, wird der Ziffernteil der Dateinamen-Benennung (Snn) für die nächste Phase dennoch aktualisiert, damit er mit der Datensatznumerierung innerhalb der generierten Datenerklärungen (DDS) übereinstimmt.

### 6.3 FORMS-2 Handhabung

FORMS-2 erzeugt in einer DDS-Datei COBOL-Datenerklärungsbefehle, die dem von Ihnen erzeugten Formular entsprechen.

Darüberhinaus müssen Sie wohl immer wieder Korrekturen und Veränderungen anbringen, um das Formular zu aktualisieren. Sie können diese Datei natürlich auch mit dem CED bearbeiten, haben dann jedoch das Risiko, Fehler zu machen, die FORMS-2 ausschließt.

Mit FORMS-2 haben Sie die Möglichkeit, diese Masken aus der Datei zurückzulesen und Änderungen vorzunehmen. Sie erreichen dies durch einen FORMS-2-Lauf und mit dem Kommando   , wenn der Arbeitsbildschirm W01 erreicht ist (vgl.   Arbeitsbildschirm-Bearbeitungs-Kommando in Kapitel 3).

Sie müssen dann die Kennung des gewünschten Formulars angeben. FORMS-2 liest die Maskendatei in den Vordergrund-Bereich des Arbeitsbildschirms und kehrt dann in den Edit-Modus zurück. Das Formular wird dann so ausgegeben, als wenn es gerade eingegeben worden wäre, und Sie können am Bildschirm ändern, bevor der Bildschirm durch Drücken der Leertaste verarbeitet wird.

#### *Anmerkung*

Damit überschreibt FORMS-2 bestehende Dateien, jedoch erst nachdem das Programm eine Warnung ausgegeben hat, daß die Dateien bereits bestehen und Sie das Überschreiben bestätigen. Für Masken-Dateien bietet FORMS-2 die Möglichkeit, einen alternativen Dateinamen anzugeben, wenn Sie die alte Datei-Version erhalten wollen.

### 6.4 Ausdruck der Formulare

Die Masken-Dateien werden als normale Textdateien abgelegt. Mit Ihrem Drucker können Sie Ihre Masken problemlos ausdrucken.

### 6.5 Masken für die Programmentwicklung

Masken sind innerhalb der Planung eines Anwenderprogramms ein eigenständiger Teil und wohl auch ein wichtiges Kommunikationsmittel zwischen Planer und Programmierer.

Bei dialogorientierten Anwendungen kann die Planung der Anwenderschnittstelle (d.h. das Aussehen der Masken oder Formulare) schon lange stattfinden, bevor das eigentliche Programm geschrieben wird. Dabei muß der Planer der Masken keine detaillierten Kenntnisse von COBOL haben.

Mit FORMS-2 kann jeder Nicht-Fachmann gültige COBOL-Befehle erzeugen. Ein erfahrener COBOL-Programmierer kann die zur Verfügung stehenden Kommandos benutzen, um einen möglichst effektiven Code zu erzeugen (d.h. durch Begrenzen der Anzahl der Felder, die ausgegeben werden).

So kann es manchmal von Vorteil sein, die Maskenausgabe-Dateien nur als Zwischenstufe in der Programmentwicklung zu verwenden. Der Programmierer braucht dann jedoch diese Dateien als Eingabe für FORMS-2, um die endgültigen .DDS-Dateien zu erzeugen.

Wenn FORMS-2 in dieser Weise verwendet wird, können sowohl der feste Teil der Maske als auch die variablen Felder auf einem einzigen Bildschirm mit festem Textteil angezeigt werden. Man kann dann diese Maske leicht verwenden, um die .DDS-Datei zu erzeugen, wobei der Masken-Planer keine detaillierten Kenntnisse von COBOL-Datenfelddefinitionen haben muß.

## 7 Beispiel einer FORMS-2-Maskengenerierung

Bei dem folgenden Beispiel ist es sinnvoll, die Masken mit ihren Erklärungen am Ende des Handbuchs auszuklappen und jeweils die Eingaben des Beispiels zu verfolgen.

Angenommen, Sie wollen folgendes Formular erstellen:

---

```
Name..... [          ]
Adresse..... [          ]
           [          ]
           [          ]
Telefon..... [          ]
```

---

wobei:

"Name" und "Adresse" alphanumerische Felder und "Telefon" ein numerisches Feld sein sollen. Nach der Eingabe von Name, Adresse und Telefonnummer wie z.B. Schmidt P., Georgenstr. 8, München, Bayern, 88326, soll die Maske folgendes Aussehen haben:

---

```
Name..... [Schmidt P.  ]
Adresse..... [Georgenstr.8 ]
           [München    ]
           [Bayern     ]
Telefon..... [88326     ]
```

---

Gehen Sie nun folgendermaßen vor:

- Sie laden FORMS-2 mit dem Kommando:  
forms2
- FORMS-2 zeigt den Bildschirm I01 an und fordert einen sechststelligen Basisnamen für die Datei- und Datennamen. Bei den anderen Fragen verwenden Sie die Voreinstellung.  
Geben Sie DEMO1 als Namen ein und beenden Sie die Eingabe mit .
- FORMS-2 zeigt den Bildschirm I02 an, um zu erfragen, welche Dateien erzeugt werden sollen. Sie wollen die Voreinstellung und geben  ein.

## Beispiel

---

- FORMS-2 zeigt den Bildschirm W01 an, für die Wahl der Maskenart. Sie wählen die Voreinstellung A und drücken .
- FORMS-2 zeigt einen leeren Bildschirm. Geben Sie nun den Text ein wie im Beispiel oben. Sie können dabei auch die Tasten    und  benutzen. Schließen Sie Ihre Eingabe mit  ab.
- FORMS-2 gibt in der linken oberen Ecke " \_\_ " aus. Drücken Sie die Leertaste und .
- FORMS-2 verarbeitet nun den Bildschirm, um eine Maske für den festen Text zu erzeugen. Das dauert eine kurze Zeit. Sie bekommen folgende Bildschirmausgabe:  
Den erzeugten Quell-Code, so wie er generiert wird (DEMO1.DDS), gefolgt von einer nochmaligen Ausgabe des festen Textes, so wie er in die Masken-Datei (DEMO1.S00) geschrieben wird.  
  
Abschließend wird eine Meldung ausgegeben, die den Namen der erzeugten Masken-Datei angibt. Drücken Sie dann .
- FORMS-2 zeigt wiederum Bildschirm W01 für die Auswahl der Maskenart an. Die Voreinstellung ist nun "C". Geben Sie einfach  ein.
- FORMS-2 zeigt nun den festen Teil der Maske, den Sie eben eingegeben haben, als **Hintergrund** an. Geben Sie nun  für variable alphanumerische Zeichen und  für die numerischen Felder ein.

---

```
Name..... [XXXXXXXXXXXXXXXXX]
Adresse..... [XXXXXXXXXXXXXXXXX]
              [XXXXXXXXXXXXXXXXX]
              [XXXXXXXXXXXXXXXXX]
Telefon..... [9999999]
```

---

Beenden Sie die Eingabe mit .

- FORMS-2 zeigt in der linken oberen Ecke wieder " \_\_ " an. Drücken Sie die Leertaste und dann .
- Es entsteht nun eine kleine Pause, in der FORMS-2 den neu eingegebenen Inhalt überprüft und folgende Meldung ausgibt:

```
WORK SCREEN VALIDATION in progress - DO NOT press
<RETURN>
```

- FORMS-2 verarbeitet die X- und 9-Zeichen, um die variablen Felder der Maske zu erzeugen.  
Folgende Bildschirmausgabe erscheint während der Verarbeitung:  
Der erzeugte Quell-Code, so wie er generiert wird (DEMO1.DDS)  
gefolgt von einer nochmaligen Ausgabe des variablen Textes, so wie er in die Masken-Datei (DEMO1.S01) geschrieben wird.  
Abschließend wird eine Meldung ausgegeben, die den Namen der erzeugten Masken-Datei angibt. Drücken Sie .
- FORMS-2 zeigt wieder den Bildschirm W01 an. Drücken Sie  und dann , um diesen Lauf zu beenden.  
FORMS-2 gibt die Namen der erzeugten Dateien und die Meldung  
END OF FORMS-2 RUN  
aus.
- Übersetzen Sie nun das Check-Out-Programm mit folgendem Kommando  
cobol DEMO1.CHK COPYLIST
- Wenn die Übersetzung beendet ist, können Sie Ihre Maske überprüfen, indem Sie das erzeugte Programm aufrufen  
cbrun DEMO1.INT
- Das Programm DEMO1 wird geladen. Der feste Teil der Maske wird am Bildschirm angezeigt. Der variable Teil wird verwendet, um Daten entgegenzunehmen.  
Überzeugen Sie sich, daß der Cursor nur innerhalb der variablen Felder positioniert werden kann und ob die Plausibilität auf alphanumerisch und numerisch funktioniert. Testen Sie auch die Wirkung der Eingabe von "." in einem numerischen Feld (rechtsbündiges Ausrichten).  
Wenn Sie die Eingabe mit  abgeschlossen haben, erscheint folgende Meldung:  
CHECK-OUT completed  
Repeat? [N] (Y = Yes)  
Mit der Eingabe  wählen Sie die Voreinstellung N und beenden den Programmablauf.

## Beispiel

---

- Das Check-Out-Programm verabschiedet sich mit der Meldung:  
END OF FORMS-2 CHECK-OUT

### *Anmerkung*

Der variable Teil der Maske wird in dem oben beschriebenen Programm für die Eingabe der Daten (ACCEPT) verwendet. In der Praxis kann dieser Teil der Maske sowohl für die Ausgabe der Daten (DISPLAY) als auch für die Eingabe verwendet werden. Das Beispiel zeigt die Länge und Art eines jeden Feldes, die für DISPLAY und ACCEPT jedoch in gleicher Weise gelten. Eine Möglichkeit wäre, um nur die Eingabefelder zu löschen, mit dem Befehl: "move spaces to eingabe" den Eingabesatz zu löschen und ihn dann auszugeben.

- Sehen Sie sich nun die erzeugten Dateien noch einmal mit dem Kommando an:

```
ls -l DEMO1*
```

Sie finden folgende Dateien vor:

```
DEMO1.DDS  
DEMO1.CHK  
DEMO1.S00  
DEMO1.S01  
DEMO1.INT  
DEMO1.LST
```

- Sie haben nun gelernt, wie Sie FORMS-2 anwenden müssen, um automatisch Masken mit festem und variablem Teil zu erzeugen, die Sie in Ihr COBOL-Programm übernehmen können.

Nun werden Sie lernen, den festen bzw. variablen Teil von bereits erzeugten Masken zu verändern, indem Sie sie vom Hintergrund in den Vordergrund spielen.

- Laden Sie FORMS-2 erneut.
- FORMS-2 zeigt Bildschirm I01 und fordert die Eingabe eines Basisnamens, wie bereits oben beschrieben. Geben Sie wieder DEMO1 ein und beenden Sie die Eingabe mit .
- FORMS-2 zeigt den Bildschirm I02, um zu erfragen, welche Dateien erzeugt werden sollen. Geben Sie  ein, da Sie die Voreinstellung übernehmen.

- FORMS-2 gibt die Meldung aus:

DEMO1.DDS already exists  
overwrite? [N] (Y = Yes)

Geben Sie  Y ein und .

*Anmerkung*

Wenn Sie hier die Voreinstellung  N eingeben, wird der Lauf beendet.

- FORMS-2 bringt folgende Meldung:

DEMO1.CHK already exists  
overwrite? [N] (Y = Yes)

Geben Sie  Y ein und .

*Anmerkung*

Wenn Sie hier die Voreinstellung  N eingeben, wird der Lauf beendet.

- FORMS-2 gibt wieder den Bildschirm W01 aus. Geben Sie nur  ein, da Sie die Voreinstellung A übernehmen.
- FORMS-2 gibt im Edit-Modus einen leeren Bildschirm aus. Geben Sie  ein, um in den Kommando-Modus zu kommen, anschließend  F und nochmals  ein, um den Bildschirm für die Vordergrund/Hintergrund-Behandlung aufzurufen. (Wir wollen das Formular abändern, daher muß es in den Vordergrund)
- FORMS-2 zeigt den gewünschten Bildschirm an. Geben Sie  F ein und anschließend den Dateinamen DEMO1.S00, dann .
- FORMS-2 gibt wieder den Bildschirm W02 aus. Geben Sie  A ein, um in den Edit-Modus zurückzukehren.
- FORMS-2 zeigt den festen Teil der Maske (so wie er vorhin erzeugt wurde). Bewegen Sie den Cursor auf das Wort "Adresse" und überschreiben Sie es mit "Wohnsitz".
- Geben Sie ein:  Leertaste .
- Sie müssen nun FORMS-2 bestätigen, daß Sie die bestehende Maske mit der gerade geänderten Maske überschreiben wollen.



- FORMS-2 zeigt Ihnen den Bildschirm W02 mit der Voreinstellung A, die Sie übernehmen. Drücken Sie .
- FORMS-2 zeigt wieder die gesamte Maske an (Sie könnten nun die variablen Text-Felder verändern).  
Sie haben nun die Möglichkeiten gesehen, wie Sie feste und variable Teile von bereits erzeugten Masken wieder abrufen können. Natürlich ist es in der Praxis bei einer kleinen Anzahl von variablen Datenfeldern (wie in unserem Beispiel) leichter, wenn man sie neu eingibt.
- Geben Sie ein:  Leertaste , um die veränderte Maske zu verarbeiten. Es entsteht wieder eine kleine Pause, während der FORMS-2 den Inhalt überprüft.
- FORMS-2 erzeugt die DDS-Datei und gibt dann die Meldung aus:  
DEMO1.S01 already exists  
overwrite? [N] (Y = Yes)  
Geben Sie  ein.
- Folgende Meldung wird ausgegeben:  
File created = DEMO1.S01
- FORMS-2 zeigt erneut den Bildschirm W01 an. Dieses Mal drücken Sie  und , um Ihren FORMS-2-Lauf zu beenden.
- Wenn Sie mit dem Check-Out-Programm die veränderte Maske überprüfen wollen, übersetzen Sie das Programm noch einmal.



## 8 Index-Programm

### 8.1 Allgemeines

FORMS-2 bietet Ihnen die Möglichkeit, automatisch ein COBOL-Programm für indizierten Dateizugriff zu generieren, mit dem Sie eine index-sequentielle Datei erzeugen und bearbeiten können. Die Eingaben, die nötig sind, um ein Index-Programm zu erstellen und zu benutzen und um die Dateien zu pflegen, werden durch einen Bildschirm-Dialog zwischen Ihnen und dem Programm unterstützt.

Man entwirft zuerst eine Dateneingabemaske mit Hilfe von FORMS-2. Dazu verwendet man die normalen Arbeitsphasen für den festen Textteil und für die variablen Felder, aus denen sich die index-sequentuellen Dateisätze zusammensetzen.

Die Verbindung zwischen Ihnen und dem erzeugten Index-Programm stellt ganz einfach die Maske dar, die die gewünschte Satzstruktur widerspiegelt. Sie brauchen sich über Befehle und Kommandobereiche keine Gedanken zu machen, sondern nur den Datenbedarf zu berücksichtigen.

#### *Beachten Sie*

Damit Sie die von FORMS-2 erzeugten Index-Programme auch übersetzen können, müssen Sie den COBOL-Übersetzer verfügbar haben.

In der Arbeitsphase von FORMS-2 ist die letzte Zeile des Formats für Meldungen des Index-Programms reserviert.

Auch wenn Sie diesen Bereich selbst nicht benutzen, werden die dort liegenden Felder dann immer durch das Meldungsfeld überdeckt.

## Index-Programm

---

Das erzeugte Index-Programm wird in die Datei basisname.GEN geschrieben.

Sie können zur Erzeugung und Bearbeitung einer index-sequentiellen Datei folgende Tätigkeiten ausführen:

- Suchen eines Satzes mit Hilfe eines Schlüssels und Anzeigen dieses Satzes (Direktzugriff)
- Sequentielles Suchen eines Satzes und Anzeigen dieses Satzes (sequentieller Zugriff)
- Ändern bestehender Sätze
- Löschen bestehender Sätze
- Einfügen neuer Sätze

Dabei brauchen Sie nicht die Funktion anzugeben, die Sie jeweils ausführen wollen, sondern das Programm erkennt allein aus der Änderung der Daten und aus der Cursorstellung, welche die gewünschte Funktion ist.

Zuerst mal werden nur die variablen Text-Daten in die Datei geschrieben und die Festtext-Daten sind nur eine Schablone, die Sie bei der Eingabe der Daten führt und die einzelnen Eingabefelder voneinander trennt.

Der einzelne Satz der index-sequentiellen Datei wird durch Aneinanderhängen der variablen Felder der Maske in der Reihenfolge ihres Auftretens gebildet. Der Satz muß ein Schlüsselfeld beinhalten, durch das eindeutig auf ihn zugegriffen werden kann. Die Logik eines Index-Programms erfordert es, daß dieses Schlüsselfeld am Anfang des Satzes steht, d.h. es muß das erste Eingabefeld der Maske sein. Das Schlüsselfeld darf nicht länger als 32 Zeichen sein. Dieses Schlüsselfeld ist ein Teil des Datensatzes. Die verbleibenden Felder bezeichnen wir einfach als Datenfelder.

In Kapitel 9 wird das Beispiel der Anwendung aus Kapitel 4 übernommen, um eine Datei mit Namen, Adressen und Telefonnummern zu bearbeiten.

### 8.2 Generieren eines Index-Programms

Ein Index-Programm wird genauso mit FORMS-2 erzeugt, wie es in Kapitel 2 und 3 beschrieben wird.

Alle beschriebenen FORMS-2-Möglichkeiten stehen zur Verfügung. Sie können jedoch, wenn Sie ein Index-Programm generieren, nur passende Optionen auswählen.

Folgende Schritte müssen durchlaufen werden:

#### 1. Einrichtungsphase:

- Bildschirm I01

Definieren des Basisnamens usw., wie normal

- Bildschirm I02

Auswahl der Option G für das Generieren eines Index-Programms

### 2. Arbeitsphase 1

#### – Bildschirm W01

Auswahl des Arbeitsbildschirms

Das Programm erzwingt die voreingestellte Option "A" zur Dateneingabe des festen Textes. Sie können hier nur  A  I oder  ? eingeben.

#### – Arbeitsbildschirm für den festen Textteil der Maske

Ein leerer Arbeitsbildschirm wird zur Eingabe des festen Textteils der Maske ausgegeben.

Alle FORMS-2-Kommandos stehen zur Verfügung, wie sie in Kapitel 3 beschrieben sind, mit folgenden **Ausnahmen**:

- G Da sich das erzeugte Programm auf die voreingestellte Struktur der Datennamen bezieht, wird dieses Kommando zurückgewiesen.
- S Das Kommando würde das Generieren der DDS-Datei ausschalten und wird deshalb zurückgewiesen.
- W Dieses Kommando wird zurückgewiesen, da die Funktion hier nicht zur Verfügung steht. Dennoch reserviert das Programm die Grundlinie, um in ihr in dem generierten Programm Systemmeldungen auszugeben ("RECORD NOT FOUND" etc.). Eine gestrichelte Trennungslinie weist am Ende des Bildschirms darauf hin.

Wenn Sie den festen Teil der Maske fertig eingegeben haben, übergeben Sie den Bildschirm mit der Folge  Leertaste  der Weiterverarbeitung.

Der Bildschirm W01 wird wieder angezeigt.

### 3. Arbeitsphase 2

#### — Bildschirm W01

Jetzt erzwingt FORMS-2 die voreingestellte Option C. Die variablen Felder werden nun definiert, wie es in Kapitel 3 beschrieben ist, d.h. die Zeichen X, Y, 9 und 8 sowie die Druckaufbereitungszeichen sind zugelassen.

Bevor Sie den Inhalt des Bildschirms verarbeiten, müssen Sie das Ende des Schlüsselfeldes bzw. den Anfang der Daten innerhalb des Satzes definieren. Dies erreichen Sie durch das Positionieren des Cursors auf den Anfang des ersten Datenfeldes und der Eingabe   .

Der Bildschirm wird mit der üblichen Folge  Leerteaste  verlassen. Wenn FORMS-2 mit den Angaben von Schlüssel und Daten nicht zufrieden ist, kehrt er in den Edit-Modus zurück.

Nach Beendigung des variablen Textteils schließt FORMS-2 die Verarbeitung ab und beendet sich, ohne daß man  eingeben muß. Das  Kommando wird nur dann benutzt, wenn Sie den Generierlauf des Index-Programms abbrechen wollen.

### 8.3 Erzeugte Dateien

Folgende Dateien werden von FORMS-2 erzeugt:

basisname.S00	}	Bildschirmmasken-Dateien
basisname.S01		
basisname.DDS		COBOL Datenerklärungs-Teil-Datei
basisname.GEN		Quelldatei des erzeugten Index-Programms

### 8.4 Übersetzen des Index-Programms

Das erzeugte Index-Programm (basisname.GEN) kann in der üblichen Weise übersetzt werden. Folgende Dateien müssen während der Übersetzung vorhanden sein:

basisname.DDS	Die DDS-Datei, die im vorangegangenen FORMS-2 Lauf erzeugt wurde. Sie muß in dem im Bildschirm I02 angegebenen Dateiverzeichnis oder, wenn die Voreinstellung verwendet wurde, im aktuellen Dateiverzeichnis vorhanden sein.
FORMS-2.GN1	Ein Gerüst für das generierte Programm.
FORMS-2.GN2	Diese beiden Dateien befinden sich im Dateiverzeichnis /usr/lib/cobol.

Das Index-Programm wird mit dem üblichen Übersetzungskommando übersetzt:

```
cobol basisname.GEN
```

Einzelheiten zur Verwendung des COBOL-Übersetzers finden Sie in der LEVEL II COBOL Bedienungsanleitung.

Nach der Übersetzung können Sie das generierte Programm starten.

### 8.5 Ablauf des Index-Programms

#### Laden

Das Programm kann gleich nach dem Übersetzen geladen werden. Verwenden Sie dazu den Aufruf des COBOL-Laufzeitsystems:

```
cbrun basisname.INT
```

#### Datenverarbeitungsmöglichkeiten

Wenn das Programm geladen ist, wird die vorher entworfene Maske angezeigt. Die Maske bleibt während des ganzen Laufs am Bildschirm stehen. Der Ablauf des Programms wird über die Veränderungen der Dateneingabefelder gesteuert. Die Bildschirmanzeige gibt den Inhalt immer eines einzigen Satzes wieder. Die gewünschte Programmfunktion wird durch Eingabe von Daten und durch Positionieren des Cursors gesteuert (s.u.). Jede Eingabe wird durch  beendet. Meldungen des Programms werden in einem unbenutzten Bereich des Bildschirms ausgegeben.

Die Grundfunktionen und die Meldungen des Index-Programms werden nachfolgend beschrieben. Sie reichen für normale Benutzung aus. Weitere Einzelheiten zur Auswertung von Datenbearbeitung durch das Index-Programm und der Cursor-Position folgen anschließend.

#### Abfrage mit Schlüssel

Tragen Sie nur im Schlüsselfeld den gesuchten Schlüssel ein und drücken Sie . Der gesuchte Satz wird angezeigt. Wenn der Satz nicht gefunden wird (d.h. wenn der Schlüssel nicht vorhanden ist) wird die Meldung

```
RECORD NOT FOUND
```

ausgegeben.

### Sequentielle Abfrage

Geben Sie einfach  ein und der nächste Satz wird angezeigt. Ist das Ende der Datei erreicht, wird die Meldung

END OF FILE - <RETURN> WILL TERMINATE

ausgegeben.

### Änderung des angezeigten Satzes

Ändern Sie nur den Inhalt der Datenfelder (d.h. das Schlüsselfeld unverändert lassen!) und geben Sie  ein.

Der am Bildschirm angezeigte Satz wird mit den veränderten Daten zurückgeschrieben. Es wird die Meldung

RECORD AMENDED

ausgegeben.

### Löschen des angezeigten Satzes

Möchten Sie einen eingelesenen Satz löschen, drücken Sie  und anschließend . Die Meldung

RECORD DELETED

wird ausgegeben, und die Datenfelder werden gelöscht.

### Einfügen eines neuen Satzes

Verändern Sie die Schlüssel- und Datenfelder wie verlangt, und geben Sie  ein. Wenn die gerade angezeigten Datenfelder nicht geändert werden müssen (d.h. wenn Sie die bestehenden Datenfelder unter einem neuen Schlüssel eingeben wollen), dann drücken Sie **entweder**  **oder** mehrfach , bis das Ende des letzten Datenfeldes erreicht ist und dann .

Die Meldung NEW RECORD WRITTEN wird ausgegeben, wenn der neue Satz eingefügt ist.

Wenn bereits ein Satz mit dem angegebenen Schlüssel besteht, wird die gegenwärtige Anzeige aufgehoben und die Warnung

**RECORD ALREADY EXISTS WITH THIS KEY**

wird ausgegeben.

Sie können jetzt 3 Dinge tun:

1. Sie wollen den Austausch des bestehenden Satzes erzwingen:

Drücken Sie entweder die  Taste oder mehrfach die  Taste, bis der Cursor am Ende des letzten Datenfeldes steht, und beenden Sie die Eingabe mit .

Der Satz wird ersetzt und die Meldung **RECORD REPLACED** wird angezeigt.

2. Satz einfügen mit neuem Schlüssel:

Verändern Sie den Inhalt des Schlüsselfeldes und fügen Sie mit  den Satz unter dem neuen Schlüssel ein (die Cursorposition spielt hierbei keine Rolle).

3. Abbrechen

Wenn Sie den Versuch, einen Satz einzufügen, abbrechen wollen, geben Sie nur  ein. Der bestehende Satz wird angezeigt.

### Programm beenden

Suchen Sie die Datei sequentiell bis zum Ende durch. Nach dem letzten Satz beendet sich das Programm. Eine andere Möglichkeit, das Programm zu verlassen, besteht darin, nach dem höchsten Schlüssel zu suchen und dann sequentiell das direkt darauffolgende Dateiende zu erreichen. Hierzu ist es sinnvoll, beim Einrichten der Datei einen Satz mit höchstem Schlüssel, z.B 9999, einzutragen.

Ist das Dateiende erreicht, wird die Meldung

**END OF FILE - <RETURN> WILL TERMINATE**

ausgegeben. Mit  beenden Sie das Programm.

### 8.6 Satzsperrung

Das Index-Programm unterstützt die in LEVEL II COBOL enthaltene Funktion einer mehrfach benutzbaren Datei.

Das Index-Programm sperrt jeden Satz, auf den Sie gerade zugreifen. Es ist keinem anderen Benutzer möglich, diesen Satz zu verändern, während Sie ihn verarbeiten. Die Sperre wird aufgehoben, wenn Sie auf einen anderen Satz zugreifen. In gleicher Weise können Sie nicht auf einen Satz zugreifen, der gerade von einem anderen Anwender bearbeitet wird.

#### Was passiert, wenn ein Satz gesperrt ist?

##### Abfragen

Wenn Sie versuchen, einen gesperrten Satz zu lesen, entweder sequentiell oder mit Hilfe eines Schlüssels, so wird die Meldung "RECORD LOCKED" ausgegeben. Der Schlüssel des gesperrten Satzes wird angezeigt, die Datenfelder werden gelöscht. Sie können jetzt folgendermaßen reagieren:

1. Sie können die Abfrage wiederholen, indem Sie  eingeben.
2. Sie können die Abfrage abbrechen und den nächsten Satz einlesen. Bewegen Sie hierzu den Cursor von seiner aktuellen Position (Beginn der Daten) weg und geben Sie  ein.
3. Sie können eine andere Funktion benutzen, so als ob die Meldung "RECORD NOT FOUND" gekommen wäre.

##### Ändern des angezeigten Satzes

Diese Funktion wird durch die Sperrfunktion nicht beeinträchtigt, da das vorangehende Lesen den Satz für andere Anwender bereits gesperrt hat.

### **Einfügen eines neuen Satzes**

Besteht bereits ein Satz mit dem angegebenen Schlüssel, so wird die Meldung "RECORD ALREADY EXISTS WITH THIS KEY" ausgegeben, unabhängig davon, ob der Satz gesperrt ist oder nicht. Die nachfolgende Eingabe entspricht den für diesen Fall schon beschriebenen Möglichkeiten mit folgenden Ausnahmen:

1. Wenn Sie den Satz auf jeden Fall ändern wollen und der Satz ist gesperrt, so wird die Meldung "EXISTING RECORD LOCKED" ausgegeben. Sie haben nun die gleichen Eingabemöglichkeiten wie nach der Meldung "RECORD ALREADY EXISTS".
2. Wenn Sie nun versuchen, sich den bestehenden Satz anzeigen zu lassen, und er ist gesperrt, wird Ihr Programm so reagieren, wie es unter "Abfragen" beschrieben ist.

### *Anmerkung*

Sofort nach der Update-Aktion (Ändern/Einfügen), versucht das Index-Programm, den Satz wieder zu lesen, um die Sperre aufrecht zu erhalten. In dieser kurzen Zeit ist es nun für einen anderen Anwender möglich, den Satz zu lesen und ihn seinerseits für sich zu sperren, bevor Ihre Sperre wieder wirksam wird. In diesem Fall wird die normale Meldung ausgegeben, die den Update bestätigt, unmittelbar gefolgt von der Meldung "RECORD LOCKED" und dem Löschen der Datenfelder. Sie befinden sich dann in der gleichen Lage, als ob eine Abfrage wegen eines gesperrten Satzes zurückgewiesen worden wäre. Die weiteren Eingaben erfolgen wie unter "Abfragen" beschrieben.

### 8.7 Zusammenfassung der Anwendungsmöglichkeiten

Das Index-Programm interpretiert die Anforderungen des Anwenders, die durch die Änderungen des Schlüssels und der Daten und durch die Position des Cursors ausgedrückt werden, folgendermaßen:

#### Schlüssel und Datenfelder unverändert

Die ausgeführte Aktion hängt von der Cursor-Position folgendermaßen ab:

- Wenn die Dateiende-Bedingung erreicht wird, wird unabhängig von der Cursorstellung angenommen, daß das Programm beendet werden soll.
- Ist die Dateiendebedingung nicht erfüllt, der Cursor wurde in die HOME-Position gebracht () und am Bildschirm wird gerade ein Satz angezeigt, dann wird angenommen, daß der Satz gelöscht werden soll.
- Steht der Cursor am Anfang der Daten (hier steht der Cursor immer wenn ein Satz ausgegeben wurde) und eine "Satz gesperrt"-Bedingung war aufgetreten, so wird angenommen, daß ein neuer Versuch unternommen werden soll, diesen Satz zu lesen.
- Wenn keine dieser Bedingungen besteht, wird angenommen, daß relativ zu dem gerade angezeigten Satz der nächste gelesen und angezeigt werden soll. Der Satz wird dann entweder tatsächlich angezeigt oder aber es wird eine Satzsperrung oder Dateiendebedingung gemeldet.

### **Schlüssel verändert und Datenfelder unverändert**

Die ausgeführte Aktion hängt folgendermaßen von der Cursorstellung ab:

- Wurde der Cursor auf die erste oder die letzte Stelle des Satzes positioniert, wird versucht, den Satz einzufügen (weitere Behandlung siehe unter "Schlüssel und Daten verändert").
- In allen anderen Fällen wird angenommen, den Satz mit dem aktuellen Schlüssel zu suchen. Der Satz wird daraufhin entweder angezeigt oder aber es wird gemeldet, daß er nicht vorhanden ist oder daß eine Satzsperrung eingetreten ist.

### **Schlüssel unverändert und Daten verändert**

- In diesem Fall wird immer angenommen, die Datei zu verändern. Wenn vorher eine Sperrung für diesen Satz eintrat (beim Lesen des Satzes), so wird sie nun gelöst. Im anderen Fall wird der Versuch angenommen, einen Satz einzufügen. Die Verarbeitung ist in diesem Fall so wie es unter "Schlüssel und Daten verändert" beschrieben ist.

### Schlüssel und Daten verändert

Dies ist eine Anforderung, einen neuen Satz einzufügen. Für den Fall, daß ein Satz mit diesem Schlüssel bereits besteht, soll der Satz nicht überschrieben werden können, ohne daß man es ausdrücklich will. Dazu wird eine Warnmeldung ausgegeben und folgende drei Aktionen können durchgeführt werden, abhängig vom Status des Schlüsselfeldes der Datenfelder und von der Cursorposition:

#### 1. Schlüssel und Daten unverändert

Die ausgeführte Aktion hängt wie folgt von der Cursorposition ab:

- Wurde der Cursor auf die erste oder die letzte Stelle des Eingabesatzes positioniert, dann wird angenommen, daß der bestehende Satz überschrieben werden soll. Entweder der Satz wird tatsächlich überschrieben oder aber eine Satzsperrung wird gemeldet. In diesem Fall wird die nächste Eingabe genauso behandelt, wie die gerade laufende.
- Ist der Cursor an einer beliebigen anderen Stelle, dann wird angenommen, daß der Satz nicht eingefügt und statt dessen der bereits bestehende Satz angezeigt werden soll. Die weitere Verarbeitung erfolgt wie unter "Schlüssel verändert und Daten unverändert" beschrieben.

#### 2. Schlüssel verändert und Daten unverändert

Es wird der Versuch gemacht die Daten unter dem neuen Schlüssel einzufügen, unabhängig von der Cursorposition. Wenn nötig wird die Warnung (RECORD ALREADY EXISTS) wiederholt.

#### 3. Schlüssel und Daten verändert

Ein normales Satzeinfügen wird angenommen, wie es oben beschrieben ist.

## 9 Beispiel eines Anwender-Indexprogramms

Angenommen, Sie wollen eine index-sequentielle Datei erzeugen, die Datensätze von Namen, Adressen und Telefonnummern enthält, wobei "Name" das Schlüsselfeld ist, und Sie wollen diese Datensätze verarbeiten, indem Sie das Formular verwenden, so wie in Kapitel 7 beschrieben.

---

```

NAME..... [          ]
ADRESSE..... [          ]
..... [          ]
..... [          ]
TELEFON..... [          ]
    
```

---

NAME und ADRESSE sind alphanumerische Felder und TELEFON ist ein numerische Feld.

Nach dem Eintragen von Name, Adresse und Telefonnummer (Schmidt P., Georgenstr. 8, München, Bayern, 88326) soll die Maske dann folgendes Aussehen haben:

---

```

NAME..... [Schmidt P.   ]
ADRESSE..... [Georgenstr.8   ]
..... [München       ]
..... [Bayern        ]
TELEFON..... [88326    ]
    
```

---

Nun müssen folgende Schritte ausgeführt werden:

- Sie laden FORMS-2 mit dem Kommando:

```
forms2 
```

- FORMS-2 zeigt den Bildschirm I01 an und fordert einen maximal sechsstelligen Basisnamen für die Datei- und Datennamen. Bei den anderen Angaben übernehmen Sie die Voreinstellung.

Geben Sie DEMO2 als Namen ein und beenden Sie die Eingabe mit .

## Beispiel für Index-Programm

---

- FORMS-2 zeigt den Bildschirm I02 an, um zu erfragen, welche Dateien erzeugt werden sollen. Geben Sie **G** und anschließend **↓** ein, damit generieren Sie das Index-Programm.
- FORMS-2 zeigt den Bildschirm W01 an, für die Wahl der Maskenart. Sie wählen die Voreinstellung A und drücken **↓**.
- FORMS-2 zeigt einen leeren Bildschirm. Den Abschluß bildet in Zeile 24 eine gestrichelte Zeile. Benutzen Sie die Cursor-Tasten und die normale Tastatur, um den Text der Maske zu erzeugen. Schließen Sie Ihre Eingabe mit **↓** ab.
- FORMS-2 gibt in der linken oberen Ecke zwei Unterstreichungszeichen aus. Drücken Sie die Leertaste, dann **↓**.
- FORMS-2 verarbeitet den Bildschirm, um eine Maske für den festen Text zu erzeugen. Das dauert eine kurze Zeit und beinhaltet folgende Bildschirmausgabe:  
Der erzeugte Quell-Code, so wie er generiert wird (DEMO2.DDS), gefolgt von einer nochmaligen Ausgabe des festen Textes, so wie er in die Masken-Datei (DEMO2.S00) geschrieben wird.  
  
Abschließend wird eine Meldung ausgegeben, die den Namen der erzeugten Masken-Datei angibt. Drücken Sie **↓**, wie verlangt.
- FORMS-2 zeigt wiederum Bildschirm W01 für die Auswahl der Maskenart an. Die Voreinstellung ist nun "C". Geben Sie einfach **↓** ein.

- FORMS-2 zeigt nun den festen Teil der Maske, den Sie eben eingegeben haben, als Hintergrund an. Sie geben nun die Zeichen **X** und **9** ein, wobei Sie die Cursor-Tasten verwenden dürfen. Haben Sie alle Felder definiert, so positionieren Sie den Cursor auf das erste Zeichen X des Adreßfeldes.

Mit der Eingabe    machen Sie FORMS-2 bekannt, daß das Namensfeld als Schlüssel verwendet werden soll, und mit dem Anfang des Adressenfeldes die "Daten" beginnen (natürlich ist das Schlüsselfeld auch Bestandteil der Daten). Ihr Bildschirm sieht dann folgendermaßen aus:

---

Name.....	[XXXXXXXXXXXXXXXXXX]
Adresse.....	[XXXXXXXXXXXXXXXXXX]
	[XXXXXXXXXXXXXXXXXX]
	[XXXXXXXXXXXXXXXXXX]
Telefon.....	[9999999]

---

Beenden Sie die Eingabe mit .

- FORMS-2 zeigt in der linken oberen Ecke wieder " \_ \_ " an. Drücken Sie die Leertaste und dann . Es entsteht nun eine kleine Pause, in der FORMS-2 den neu eingegebenen Inhalt überprüft und folgende Meldung ausgibt:

```
WORK SCREEN VALIDATION in progress - DO NOT press  
<RETURN>
```

- FORMS-2 verarbeitet die X- und 9-Zeichen, um die variablen Felder der Maske zu erzeugen, mit folgender Bildschirmausgabe während der Verarbeitung:  
Der erzeugte Quell-Code, so wie er generiert wird (DEMO2.DDS), gefolgt von einer nochmaligen Ausgabe des variablen Textes, so wie er in die Masken-Datei (DEMO2.S01) geschrieben wird.
- FORMS-2 beendet sich nach Verarbeitung der Maske automatisch mit folgenden Meldungen:

```
Files created DEMO2.DDS  
DEMO2.GEN  
END OF FORMS-2 RUN
```

## Beispiel für Index-Programm

---

- Übersetzen Sie nun das Index-Programm mit dem normalen Übersetzungskommando:

```
cobol DEMO1.GEN copylist 
```

- Wenn die Übersetzung beendet ist können Sie das Programm sofort mit dem COBOL-Laufzeitsystem starten:

```
cbrun DEMO2.INT
```

- Das generierte Index-Programm DEMO2 läuft nun. Die Maske, wie Sie sie vorhin definiert haben, wird angezeigt, und zwar der feste Teil der Maske; der variable Teil wird für die Eingabe der Daten (ACCEPT) verwendet.
- Sie können nun alle Dateibearbeitungs-Kommandos durchführen. Die nächsten Schritte zeigen dies alles in der Praxis, Sie können jedoch die Folge der einzelnen Schritte verändern oder andere Schritte hinzufügen, wenn Sie sich ein wenig vertraut gemacht haben.
- Um den ersten Satz in die index-sequentielle Datei einzufügen, geben Sie einfach einen Namen, Adresse und Telefonnummer ein und beenden die Eingabe mit .
- Geben Sie noch zwei weitere komplette Sätze ein und überschreiben Sie alle vorhandenen Zeichen, da die Sätze, so wie sie am Bildschirm zu sehen sind, in die Datei eingefügt werden.
- Wenn Sie drei Sätze geschrieben haben, können sie den zweiten Satz verändern:  
Geben Sie den Namen für den zweiten Satz ein und beenden Sie die Eingabe mit . Das Index-Programm sucht mit dem eingegebenen Schlüssel (Namen) den entsprechenden Satz und zeigt ihn am Bildschirm an. Sie haben damit die Möglichkeit der Dateiabfrage gesehen. Alle Sätze der Datei können auf diese Weise eingelesen werden.
- Ändern Sie nun den Wohnort und geben Sie  ein. Es wird die Meldung "RECORD AMENDED" angezeigt.
- Geben Sie  ein, und der dritte Satz wird Ihnen angezeigt. Sie können auf diese Weise die ganze Datei durcharbeiten.

- Wenn Sie nun den dritten Satz löschen wollen, drücken Sie  und .  
Die Felder werden gelöscht und zeigen somit an, daß auch der Satz in der Datei gelöscht wurde. Zusätzlich wird die Meldung "RECORD DELETED" ausgegeben.
- Geben Sie  ein. Das Index-Programm versucht den nächsten Satz anzuzeigen, aber da das Ende der Datei erreicht ist, wird folgende Meldung ausgegeben:  
  
END OF FILE REACHED - < RETURN > WILL TERMINATE
- Geben Sie  ein und das Programm beendet sich. Sie kennen nun die Methode der Satzbehandlung einer Datei und Sie können, wenn Sie wollen, sich nun ein umfangreicheres Programm erzeugen, oder das bestehende Programm nochmal mit cbrun aufrufen und sich mit der Satzbehandlung noch vertrauter machen.
- Bevor Sie das jedoch tun, überprüfen Sie doch einmal die inzwischen entstandenen Dateien:  
  
DEMO2.DDS - COBOL Source mit Datendefinition  
DEMO2.GEN - COBOL Source des Index-Programms  
DEMO2.INT - Zwischencode des Index-Programms  
DEMO2.LST - Übersetzungslisting  
DEMO2.IDX - Index-Datei  
(Index-sequentielle Datei)  
DEMO2.DAT - Sequentielle Datendatei

### Anmerkung

1. Die zwei Dateien DEMO2.IDX und DEMO2.DAT bilden zusammen die index-sequentielle Datei, die von dem Index-Programm erzeugt wurde. Bei jedem Lauf dieses Programms werden diese Dateien verwendet.
2. Das Beispiel lief in einer Single-User-Umgebung ab. Das Programm beinhaltet aber auch die Möglichkeiten, um bei gleichzeitigem Zugriff mehrerer Anwender auf die gleiche Datei den Satzschutz zu gewährleisten und die entsprechende Fehlerbehandlung durchzuführen (siehe Kapitel 8).



---

## **Stichwörter**

**Allgemeine Kommandos** 3-11, 3-12  
**Arbeitsbildschirm** 3-3  
**Arbeitsbildschirm-Kommandos** 3-11, 3-15  
**Arbeitsphase** 1-3, 3-1  
**Ausdruck der Formulare** 6-4

**Basis-Name** 1-3  
**Benutzermaske** 1-4  
**Bildschirm- Ein-Ausgabe** 1-6  
**Bildschirm-Masken-Datei** 6-1  
**Bildschirmtyp** 3-2

**CHECK-OUT-Programm** 5-1  
**Cursor positionieren** 3-19  
**Cursor-Steuerung** 1-7

**Dateien** 1-3, 1-8, 8-6  
**Dateiausgabe** 3-24  
**Dateinamen-Basis** 2-1  
**Dateiverzeichnis** 2-6  
**Dateinamen** 4-2  
**Dateinamen-Basis** 2-1  
**Datensätze** 4-2  
**Dezimalpunktstelle** 2-3

**Edit-Modus** 1-7, 3-5  
**Einrichtungsphase** 1-3, 2-1

**Fenster(-Kommandos)** 3-27  
**Fester Text** 3-7, 4-3

**Hintergrund** 3-4

**Index-Programm** 8-1

**Kommando-Modus** 3-5

---

**Parameter für FORMS-2** 1-3  
**PICTURE-Klauseln** 4-3  
**Programmentwicklung** 6-4  
**Programmier-Kommandos** 3-11, 3-20

**Satzsperrung** 8-10  
**Schlüssel** 8-7

**Unterstreichungszeichen** 3-23  
**Unveränderlicher Text** 1-4

**Variable Daten (Text)** 3-7  
**Vordergrund** 3-4

**Währungszeichen** 2-3

**Zwischenraumbehandlung** 3-22

# Anhang Bildschirme

*[Faint, illegible text block]*

*[Faint, illegible text block]*

## ALLGEMEINE KOMMANDOZUSAMMENFASSUNG:

ZWISCHENRAUM	=	Verarbeiten des Arbeitsbildschirms
_	=	Zurück in den EDIT-Modus
?	=	Anzeigen des nächsten Hilfsbildschirms
?n	=	Anzeigen des n-ten Hilfsbildschirms
0	=	Zurück zur Maskenauswahl der ARBEITSPHASE
!	=	Beendet den FORMS-Lauf sofort
X	=	Positioniert das Kommandoeingabefeld im Edit-Modus an aktueller Cursorposition
*	=	Anzeigen des Beginns der Datenfelder

ANMERKUNG: ZWISCHENRAUM ist das Kommando, um den Bildschirm aus dem EDIT-Modus heraus zu verarbeiten

HILFS-Option	[_]	{_ = Zurück in den EDIT-Modus)
	{?	= Anzeige des nächsten Hilfsschirms)
	{!	= Beendet den FORMS-2-Lauf sofort)

Zur Fortsetzung <RETURN> drücken

## ZUSAMMENFASSUNG DER BEARBEITUNGS-KOMMANDOS:

F	=	VORDERGRUND/HINTERGRUND-Verarbeitung einleiten
Fx	=	VORDERGRUND/HINTERGRUND-Verarbeitung für Option "x" einleiten
0	=	Automatische Vorbereitung des Arbeitsbildschirms einschalten
01	=	Automatische Vorbereitung des Arbeitsbildschirms ausschalten
Cn	=	An der Cursorposition n Zwischenräume einfügen
Dn	=	An der Cursorposition n Zeichen löschen
In	=	Vor der Cursorzeile n Zeilen einfügen
Kn	=	n Zeilen löschen, einschließlich der Cursorzeile
An	=	n Zeilen mit den Daten der Cursorzeile überschreiben
Un	=	Den Cursor n Zeilen aufwärts setzen
Vn	=	Den Cursor n Zeilen abwärts setzen

HILFS-Option	[_]	{_ = Zurück in den EDIT-Modus)
	{?	= Anzeige des nächsten Hilfsschirms)
	{!	= Beendet den FORMS-2-Lauf sofort)

Zur Fortsetzung <RETURN> drücken

## ZUSAMMENFASSUNG DER PROGRAMMIERKOMMANDOS:

- G = Datennamen mit Koordinaten ergänzen
- G1 = Datennamen mit Folgenummern ergänzen
- Jn = Erlaubt bis zu n aufeinanderfolgende Zwischenräume in einem Datennamen
- Mx = Interpretiert "x" als "Zwischenraum"
- S = Löschen des vorhergehenden Sn-Kommandos
- S1 = Im nächsten Verarbeitungsschritt wird die Ausgabe von DDS & CHK unterdrückt
- S2 = Im nächsten Verarbeitungsschritt wird die Ausgabe der Datei Snn unterdrückt
- S3 = Für den nächsten Verarbeitungsschritt wird ein Dateiname für die Snn-Datei angefordert
- S9 = Zur Generierung der Datei DDS wird jede Zeile vor der Verarbeitung am Bildschirm angezeigt
- P = Anzeige der aktuellen Cursorkoordinaten

HILFS-Option [\_] { \_ = Zurück in den EDIT-Modus)  
 (? = Anzeige des nächsten Hilfsschirms)  
 (! = Beendet den FORMS-2-Lauf sofort)

Zur Fortsetzung <RETURN> drücken

## ZUSAMMENFASSUNG DER FENSTERKOMMANDOS:

- W = Setzt den Cursor auf den Beginn des aktuellen Fensters
- W1 = Fensterbeginn in Cursorzeile
- W2 = Fensterende in Cursorzeile
- W3 = Fensterbeginn in Cursorzeile, ohne Trennungslinie
- W4 = Fensterende in Cursorzeile, ohne Trennungslinie
- W5 = Anzeige der Trennungslinie am Fensterbeginn
- W6 = Anzeige der Trennungslinie am Fensterende
- W7 = Wiederanzeige der durch die Anfangs-Trennungslinie überschriebenen Daten
- W8 = Wiederanzeige der durch die End-Trennungslinie überschriebenen Daten
- W9 = Setzt den Cursor an das aktuelle Fensterende

HILFS-Option [\_] { \_ = Zurück in den EDIT-Modus)  
 (? = Anzeige des nächsten Hilfsschirms)  
 (! = Beendet FORMS-2-Lauf sofort)

Zur Fortsetzung <RETURN> drücken

## FORMS2 PARAMETER:

DATEN/ DATEINAME [ ] (1-6 alphanumerische Zeichen)

Bildschirmzeilen [24] (22 oder 23 oder 24)

## SPECIAL-NAMES Modifikationen:

WÄHRUNGSZEICHEN [\$] (nur ANSI Währungszeichen)

DEZIMALPUNKT [.] (". " oder ",")

Zur Fortsetzung <RETURN> drücken

## ZU ERZEUGENDE DATEIEN:

DATEI-KOMBINATIONEN [C] (A = Von Ihrem erstellten Formular wird Datenbeschreibung erzeugt)  
 (B = Datenbeschreibung und Erzeugen des Testprogramms)  
 (C = Datenbeschreibung und Erzeugen des Testprogramms und einer druckbaren Masken-Datei)  
 (D = Datenbeschreibung und druckbare Maskendatei)  
 (E = Druckbare Masken-Datei)  
 (F = keine Dateiausgabe)  
 (G = Datenbeschreibung und druckbare Maskendatei und Index-Programm werden erzeugt)

DATEIVERZEICHNIS (0-40 Zeichen) [ ]

Zur Fortsetzung <RETURN> drücken

## ARBEITSBILDSCHIRM-AUSWAHL:

MASKEN-TYP [A] (A = Fester Text auf leerem Schirm)  
 (B = Fester Text auf letztem Schirm)  
 (C = Variable Daten mit Redefinition  
 des letzten Bildschirms)  
 (D = Variable Daten ohne Redefinition)  
 (! = Beenden dieses FORMS-2-Laufs)

Fester Text erlaubt: Alle Zeichen

Variable Daten erlauben: X oder Y, um alphanumerische  
 Felder zu definieren

9 oder 8, um numerische Felder  
 zu definieren

COBOL-Sonderzeichen, um numerisch  
 aufbereitete Felder zu definieren

Zur Fortsetzung <RETURN> drücken

## VORDERGRUND/HINTERGRUND-BEHANDLUNG:

OPTION [H] (A = Zurück in den EDIT-MODUS)  
 (B = Leerer VORDERGRUND)  
 (C = Leerer HINTERGRUND)  
 (D = HINTERGRUND in den VORDERGRUND mischen)  
 (E = VORDERGRUND in den HINTERGRUND mischen)  
 (F = Bildschirmmaske in den VORDERGRUND)  
 (G = Bildschirmmaske in den HINTERGRUND)  
 (H = nur VORDERGRUND anzeigen)  
 (I = nur HINTERGRUND anzeigen)  
 (J = Masken-Anzeige)

## Anmerkung:

(Bei H, I und J wird der Bildschirm  
 solange angezeigt, bis <RETURN> gedrückt wird)

DATEI-NAME [ ]

(nur F & G & J)

Zur Fortsetzung <RETURN> drücken