

CIS COBOL FORMS-2 Utility Manual

Version 1.3

CIS COBOL FORMS-2 Utility Manual: Version 1.3

Copyright © 1980, 1983 Micro Focus Ltd

Not to be copied without the consent of Micro Focus.

Micro Focus Limited
58 Acacia Road, St. Johns Wood
London
NW8 6AG

Table of Contents

WELCOME TO FORMS-2!	vii
NOTATION IN THIS MANUAL	vii
RELATED PUBLICATIONS	viii
1. Introduction	1
GENERAL DESCRIPTION	1
FACILITIES	1
OUTPUTS	1
PHASES	1
Initialisation Phase	1
Work Phase	1
OPERATOR INTERFACE	2
CURSOR MOVEMENT FACILITIES	2
GETTING STARTED	2
FORMS-2 VALIDATION	3
2. INITIALISATION PHASE	9
INITIALISATION SCREEN I01	9
DATA-NAME AND FILE-NAME BASE	9
LINES PER CRT SCREEN	9
CURRENCY SIGN	9
DECIMAL-POINT	10
INITIALISATION SCREEN I02	10
FILE COMBINATIONS	10
DEVICE/DIRECTORY PREFIX	11
3. WORK PHASE	13
SCREEN W01	13
SCREEN TYPE SELECTION	13
TERMINATING THE RUN	13
WORK SCREEN	13
BACKGROUND/FOREGROUND	14
EDIT MODE	14
Fixed Text	14
Variable Data	14
COMMAND MODE	15
Commands	15
General Work Screen Commands	15
Work Screen Manipulation Commands	17
Programming Commands	19
WORK PHASE COMPLETION	21
4. DATA DESCRIPTIONS	23
RECORD-NAME AND DATA-NAME GENERATION	23
RECORD NAMING	23
DATA NAMING	23
PICTURE GENERATION	24
FIXED TEXT	24
VARIABLE DATA FIELDS	24
EDITING THE DDS FILE	24
INCORPORATION OF DDS FILE CONTENTS	24
5. THE CHECK-OUT PROGRAM	25
CHECK-OUT PROGRAM GENERATION	25
CHECK-OUT PROGRAM COMPILATION	25
CHECK-OUT PROGRAM RUNNING	26
LOADING	26
CHECK-OUT PROCESSING	26
Fixed Text Screens	26
Variable Data Screens	26

CHECK-OUT COMPLETION	27
6. SCREEN IMAGE FILE	29
SCREEN IMAGE FILE GENERATION	29
FORMS-2 MAINTENANCE	29
PRINTED FORMS	30
FORM IMAGES IN THE DESIGN PROCESS	30
7. FORMS-2 USER SCREEN GENERATION EXAMPLE	31
8. INDEX PROGRAM	37
INDEX PROGRAM GENERATION	37
FILES GENERATED	38
INDEX PROGRAM COMPILATION	38
INDEX PROGRAM RUNNING	39
LOADING	39
DATA PROCESSING FACILITIES	39
Enquiry by Key Field	39
Sequential Enquiry	39
Amend Displayed Record	40
Delete Displayed Record	40
Insert New Record	40
Terminate Run	40
USE ON MULTI-USER SYSTEMS	40
Enquiry	40
Amend Displayed Record	41
Insert New Record	41
USER REQUIREMENT INTERPRETATION SUMMARY	41
Key and Data Fields Unchanged	41
Key Changed and Data Unchanged	41
Key Unchanged and Data Changed	42
Key and Data Changed	42
9. USER INDEX - PROGRAM EXAMPLE	43
A. INITIALISATION SCREENS	47
B. WORK SCREENS	49
C. HELP SCREENS	51
D. OPERATING FORMS-2 WITH CP/M	53
FORMS-2 LOADING	53
FORMS 2 CHECKOUT PROGRAM COMPILATION.	53
FORMS-2 CHECK-OUT PROGRAM RUNNING	53
FORMS-2 INDEX PROGRAM COMPILATION	53
FORMS-2 INDEX PROGRAM RUNNING	54

List of Tables

1.1. Cursor Control Keys	2
1.2. FORMS-2 Issue Disk Contents.	2

WELCOME TO FORMS-2!

This manual describes your FORMS-2 package to design, create and edit interactive screen layouts for use in CIS or L/II COBOL application programs.

As the product is compatible with both the Micro Focus base COBOL products CIS COBOL and LEVEL II COBOL, the common abbreviation "CIS or L/II COBOL" is used throughout this manual.

Most application programs will be written separately according to individual requirements but FORMS-2 can automatically generate a powerful user-oriented indexed data entry and filing system maintained by use of the screen layouts (forms).

If you want to generate an automatic indexed filing system Index program and then use it, carry out Chapter 1 procedures, read Chapters 2 and 3 briefly and then refer straight to Chapter 8 before performing the comprehensive sample run in Chapter 9.

If you wish to generate some screen layouts, carry out the procedures in Chapter 1, read briefly Chapters 2 and 3, and then perform the comprehensive sample run in Chapter 7.

To gain a further understanding of the FORMS-2 features and FORMS-2 operation read Chapters 1, 2 and 3 in detail. Chapters 4, 5, 6 and 8 are descriptions of FORMS-2 output file contents and relate directly to optional features that can be used. Read these Chapters only if you require the feature described.

It is recommended that, in any case, you run the sample programs before using FORMS-2 to generate your own forms or indexed files.

NOTATION IN THIS MANUAL

Throughout this manual, the following notation is used to describe the format data input or output:

1. All words printed in small letters are generic terras representing names which will be devised by the programmer.
2. The carriage return (CR) or equivalent data input terminator key is referred to throughout this manual as the RETURN key.
3. The symbol << in this manual indicates that the RETURN key must be pressed once.
4. The space bar or key is referred to throughout this manual as the SPACE key.

Headings are presented in this manual in the following order of importance:

CHAPTER N
 Chapter Heading
TITLE

ORDER ONE HEADING
ORDER TWO HEADING
Order Three Heading Text two lines down
Order Four Heading
Order Five Heading: Text on same line

Numbers one (1) to nine (9) are written in text as letters, e.g. one.

Numbers ten (10) upwards are written in text as numbers, e.g. 12.

RELATED PUBLICATIONS

For details of CIS or L/II COBOL operation and language respectively, refer to the documents:

CIS or Level II COBOL COBOL Operating Guide

CIS or Level II COBOL COBOL Language Reference Manual

Chapter 1. Introduction

GENERAL DESCRIPTION

FACILITIES

The FORMS-2 package is an extension to the CIS or L/II COBOL software development system which enables interactive creation and editing of data entry screens for applications programs at a CRT. The package provides four powerful facilities to aid the design and development of interactive applications written in CIS or L/II COBOL:

- Translation of user screen layouts into COBOL record descriptions for inclusion in COBOL applications programs.
- Verification of user screen layouts in a Check-Out program before their incorporation in an application program.
- Retention of exact screen images of the user screens in disk files' for subsequent editing and printing.
- Generation of an entire COBOL program to allow data capture, update and interrogation by means of application screens and an indexed sequential file.

OUTPUTS

The user can choose any valid combination of the above facilities and dependent on the options selected FORMS-2 will automatically produce the following four types of disk output file:-

- A source file of CIS or L/II COBOL Data Description Statements defining the user designed screens (forms). These statements can subsequently be included in a CIS or L/II COBOL application program using the COPY verb. The file is generated as filename.DDS.
- A source file of a Check-Out program incorporating the Data Description Statements defining the user's screens. After compilation the user is able to verify the data entry form prior to building the actual application. The file is generated as filename.CHK.
- Screen Image files of exact copies of the user defined forms. The files are generated as filename.Snn (nn is 00 to 99).
- A file of the source of an Index program based on the user screen. After compilation the generated program can be used for the storage, retrieval, updating and deletion of data entered via the users form. The file is generated as filename.GEN.

PHASES

FORMS-2 processing is divided between a number of logically distinct units but two main phases can be identified - Initialisation Phase and Work Phase:

Initialisation Phase

The Initialisation phase is passed through once only and establishes the characteristics of this particular run of the program. It is a series of screens containing self-explanatory prompts to which the user replies as necessary.

Work Phase

At least two Work Phases are passed through for each data entry screen required by the application.

The FORMS-2 screen is analogous to a paper form in which the printed fixed text is used as a guide to entering the variable data in the spaces provided. To the human eye it is obvious where the variable

data entry areas occur on a form but the computer needs to have these areas defined explicitly. There are, therefore, two types of Work Phase: one in which fixed text is specified and one in which variable data fields are specified.

OPERATOR INTERFACE

FORMS-2 is written in COBOL and uses the ACCEPT and DISPLAY extended CIS and L/II COBOL features. These two verbs are described in the *CIS or L/II COBOL Language Reference Manual*, as are the cursor control features.

Advantages of this CRT interface are:

- Corrections can be directly overtyped
- Numeric fields accept only numeric characters
- The full stop or period (.) when keyed in a numeric field automatically zero-fills the field from the left.

CURSOR MOVEMENT FACILITIES

The user has the ability to move the cursor quickly and easily about the screen using the standard CIS and L/II COBOL ACCEPT statement interactive facilities.

The general functions of the cursor control keys are summarised in the following table. The actual key may vary from that shown dependent on the particular CRT being used.

Table 1.1. Cursor Control Keys

KEY:	FUNCTION:
→	Position cursor right one data character
←	Position cursor left one data character
↓	Position cursor at start of succeeding data field
↑	Position cursor at start of preceding data field
ctrl ↑	Move cursor to start of first data field referred to as HOM or # throughout this manual.
TAB	Position to next tab stop

In Edit Mode the screen is split into fields 80 characters long. There are 22-24 fields per screen depending on the number of lines specified at Screen I01.

Corrections to text may be made by overtyping or by switching into Command Mode and using the editing commands.

GETTING STARTED

The FORMS-2 Issue Disk is a formatted data disk formatted to the requirements of CP/M.

Load the Operating System system disk and the FORMS-2 Issue Disk, bootstrap load the Operating System as usual, and obtain a directory of the FORMS-2 Issue Disk. All the files listed in Table 1-2 should be present on the disk. If they are not, or any problems arise, contact your Distributor.

Table 1.2. FORMS-2 Issue Disk Contents.

FILE	CONTENTS	DESCRIPTION
FORMS2.COM FORMS2.I51	FORMS2 program	The manual

FILE	CONTENTS	DESCRIPTION
FORMS2.I52 FORMS2.I53		
FORMS2.I01 FORMS2.I02	Initialisation Phase Screens	Chapter 2
FORMS2.W01 FORMS2.W02	Work Phase Screens	Chapter 3
FORMS2.H01 FORMS2.H02 FORMS2.H03 FORMS2.H04	Help Screens	Chapter 3 (? Command)
FORMS2.CH1 FORMS2.CH2	Check-Out Program Skeleton	Chapter 5
FORMS2.GN1 FORMS2.GN2	Index Program Skeleton	Chapter 8

Having validated that all the files are present copy all of them to a working disk using the standard copy program provided with your operating system, and store the Issue Disk as a back-up master.

FORMS-2 VALIDATION

Before using your copy of FORMS-2, you should first validate the main files on your disk by performing the simple run below:

1. Boot the system up, and load FORMS-2 by entering the command:

```
FORMS2<<
```

2. The program will run, and will come up with the first screen, thus:

```
FORMS2 V1.3                INITIALISATION PHASE                SCREEN I01

FORMS2 PARAMETERS:

      DATA-NAME & FILE-NAME    [      ]    (1-6 alphanumeric characters)
      CRT lines                  [23]        (22 or 23 or 24)

SPECIAL-NAMES clause:

      CURRENCY SIGN              [$]          (ANSI currency signs only)
      DECIMAL-POINT              [.]          ( "." or ",")

Press RETURN when complete
```

Press RETURN when complete.

A six-character base for file-names and data-names is requested followed by three other questions. You need only key DEMO followed by the RETURN key to accept the defaults displayed.

3. FORMS-2 displays Screen I02 to request the output file option type and device/directory prefix.

```

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

```

Key F then RETURN.

4. FORMS-2 displays Screen W01 to request the Screen Type option, thus:

```

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
                                   (CIS COBOL Standard only)
Press RETURN when complete

```

Note the default "A" and press the RETURN key.

5. FORMS-2 displays a blank screen. You are currently in Edit Mode, and should be able to position the cursor at any point on the screen. Use the cursor control keys and the normal character keys to set up the following text on the screen:

```

NAME [                ]

```

Finally press the RETURN key.

6. FORMS-2 puts "_" in the top left of the screen indicating that you now in Command Mode. Enter ? and then press the RETURN key.
7. FORMS-2 displays screen H01, thus:

```

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

```

Enter ? then RETURN.

8. FORMS-2 displays screen H02, thus:

```

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
        Ol    = Turn off automatic WORK screen preparation
        Cn    = Insert n spaces at cursor position
        Dn    = Delete n characters 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

```

Enter ? then RETURN.

9. FORMS-2 displays screen H03, thus:

```

FORMS2 V1.3                                HELP SCREEN                                SCREEN H03

```



```
(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
```

Enter A then RETURN.

14. Again FORMS-2 redisplay the fixed text entered at step 5. Press RETURN.

15. FORMS-2 puts "_" in the top left hand of the screen. Press the SPACE and then RETURN keys.

16. FORMS-2 displays screen W01 again to request the Screen Type option. Note the default is "C" and press RETURN.

17. FORMS-2 displays the fixed text screen. Use the cursor control keys and key in X's alone to set up the screen as follows:

```
NAME [XXXXXXXXXXXXXXXXXXXXXXXXXXXX]
```

Press RETURN.

18. FORMS-2 puts "_" in the top left of the screen. Press the SPACE key and then RETURN.

19. FORMS-2 displays screen W01 again. This time enter ! and press RETURN to complete the run.

20. FORMS-2 terminates with the message:

```
END OF FORMS2 RUN

FORMS2 COPYRIGHT (C) 1979, 1982 MICRO FOCUS LTD
```

You have now used all the FORMS-2 Screens and you can be sure that you have a usable product.

Chapter 2. INITIALISATION PHASE

The Initialisation Phase of the FORMS-2 Utility program immediately follows program load, and is only carried out once in any one run of FORMS-2. It consists of replying to questions asked on the two Initialisation screens I01 and I02.

INITIALISATION SCREEN I01

This screen is displayed immediately FORMS-2 is loaded.

Four items of information are requested. Note that the effect of pressing the RETURN key during this screen display immediately enters all responses so far made and any remaining defaults. Do not therefore press RETURN until all the required entries have been made, for the following:

- Data-name and the File-name Base
- Lines per CRT Screen
- Currency Sign
- Decimal-Point Representation

DATA-NAME AND FILE-NAME BASE

The record/filename base keyed in at this point is used in the following ways by FORMS-2:-

1. It is taken as the first part of all the data-names and record-names generated in this run. Uniqueness is achieved by adding a two-digit sequence number for new records and adding the sequential number of the field within the form for, datanames within records. Optionally by means of a Work Phase command, uniqueness may be achieved by adding the screen coordinates.
2. It is taken as the main filename for files generated. These can consist of:-

filename .DDS for COBOL Data Description Statements (See Chapter 4)

filename .CHK for Checkout program (See Chapter 5)

filename .Snn for Screen images (nn=00, 01, 02,...,99) (See Chapter 6)

filename .GEN for Index program (See Chapter 8)

Note that only one DDS file is output per FORMS-2 run, whereas a separate screen image file is output for each screen built.

LINES PER CRT SCREEN

FORMS-2 can be used with screens of 22, 23 or 24 lines. FORMS-2 defaults to 24 for this entry.

CURRENCY SIGN

This entry allows the default currency sign (\$) to be overridden. It will cause generation of an appropriate Special-Names entry in either the Checkout and Index program. The specified currency sign should be used when specifying numeric edited fields in the Work Phase, and will be used in the generated data description statements.

Note

The specified character is not validated. Users should refer to the *CIS COBOL or L/II COBOL Language Reference Manual* for a list of valid characters.

DECIMAL-POINT

This option allows the roles of the period or full-stop sign (.) and the comma sign (,) to be exchanged. If it is specified then a DECIMAL-POINT IS COMMA clause will be generated in the Checkout or Index Programs. The default is ".". The specified decimal point sign should be used when specifying numeric edited fields in the Work Phase, and will be used in the generated data description statements.

INITIALISATION SCREEN I02

Screen I02 is displayed immediately after screen I01 entries are terminated by pressing the RETURN key.

At this point the user specifies the following:

- Types of files to be created
- Device/Directory to which files are to be written.

Once screen I02 is released by pressing RETURN the Work Phase is entered and it is no longer possible to amend information specified during the Initialisation Phase.

FILE COMBINATIONS

FORMS-2 offers options for all valid combinations of these files, each identified by a unique file name extension as follows:

1. DDS — The user may generate COBOL source Data Description (Statements (DDS) corresponding to the screens he has created. These are output to a standard ASCII text file and may be subsequently compiled into any program using the standard COBOL COPY facility. In Particular they are used by the Check-Out and Index programs, (see below).

The reader who is unfamiliar with screen handling in a CIS COBOL or L/II program should consult the appropriate Language Reference Manual (especially the sections on ACCEPT/DISPLAY, FILLER, REDEFINES).

2. CMK — in addition to generating DDS, FORMS-2 can also generate a Checkout program. This consists of simply the Procedure Division statements (ACCEPT and DISPLAY) which correspond to the screens that have been created. These statements are contained in filename.CHK (see record/ filename prefix) and they are combined with the following COPY files:-

filename.DDS , FORMS2.CH1, FORMS2.CH2

The Checkout program allows the user to demonstrate on the screen exactly how the system will operate, by displaying successively the screens he has just created, and by allowing data to be entered just as it would be under actual operating conditions.

3. Snn — The user may also output the text of the screen just designed to a file on disk in the form of a screen image. This file can be retrieved later in this run or in subsequent FORMS-2 runs, for further amendment if required.

Alternatively they may be printed, and the hard copy simply used as a means of communicating between different individuals at different times (e.g. the end user and the programmer).

4. GEN — FORMS-2 can generate an Index program. This includes all code necessary to set up and maintain an indexed sequential file with records corresponding to the structure of the user's form. The code is output to filename.GEN and is combined with the following copy files:-

filename.DDS, FORMS2.GN1, FORMS2.GN2

Index program generation places constraints upon the user during the FORMS-2 run. The creation and operation of the Index program is discussed in detail in Chapter 8.

Note

If the Q command (see Chapter 3) is entered at this point, FORMS-2 will "quit" back to screen I01, allowing amendment of information given there. This can be useful if RETURN is inadvertently pressed before all options have been entered.

DEVICE/DIRECTORY PREFIX

All the files output by FORMS-2 are on the same device/directory. Specify the mnemonic device/directory prefix as required by your operating system. If this reply field is left blank, the files will be created on the default device/directory.

Chapter 3. WORK PHASE

The user defines the screen layouts (forms) to be used in a CIS or L/II COBOL application by entering text at the keyboard to produce model forms on the screen. The user may define as many forms as he wishes in a single FORMS-2 run. To define one form requires at least two distinct Work Phases: one to define the fixed text of the form, and another to define the variable data entry fields.

Most commonly the first Work Phase is used to specify the fixed text form and the subsequent Work Phase to specify the variable data fields within the form. However this need not always be the case and FORMS-2 needs to know which type of text is to be input in a particular phase. Therefore the Work Phase is introduced by a screen presenting the various options (W01).

SCREEN W01

SCREEN TYPE SELECTION

Fixed Text selections offered at this screen are as follows:

- A The CRT screen is cleared to spaces in preparation for the user to enter the fixed text for a new form.
- B The previous screen is redisplayed to assist the user in defining additional fixed text. Text from the previous screen is used only as a background in this case, and is *not* included in the record definition for the fixed text currently being keyed in. The user must therefore ensure that if any part of the previous screen is inadvertently overkeyed, the original characters must not be replaced but cleared to spaces.

Variable Data selections offered at this screen are as follows:

- C The previous screen is redisplayed to assist the user in the redefinition of the form to incorporate variable data field specifications. In the application the data is keyed into the fixed text form itself.
- D The previous screen is redisplayed to assist the user in the definition of variable data fields which will be kept separate from the fixed text within the applications program. This may sometimes be of assistance to the programmer even though it results in larger application programs.

TERMINATING THE RUN

Screen W01 is redisplayed after completion of each Work Phase, and is the screen used to terminate the program. This is done by entering the character ! and pressing the RETURN key.

Warning

Use of the ! command at any other time causes immediate abandonment of the run.

On termination, the DDS file is closed and an identification message is displayed. If the Check-out facility was specified during initialisation then output of the Check-out program to disk is completed and the CHK file is closed with an identification message displayed.

Termination occurs automatically after the second Work Phase if an Index program is being generated (see Chapter 8).

WORK SCREEN

After the screen type has been selected, the user is presented with the appropriate Work Screen for that text to be entered, i.e., if option A (fixed text on clear screen) is selected, a blank screen is displayed. For the other options, the previous screen is redisplayed to allow correct alignment of the current input.

BACKGROUND/FOREGROUND

In order to process only the data entered in this phase, FORMS-2 must keep this data separate from previously entered data which is displayed purely for alignment purposes. FORMS-2 does this by constructing the displayed Work Screen from two separate data areas, termed Background and Foreground. The Foreground holds the data entered during the current Work Phase. The Background holds previously entered data which has been retained for alignment of the data entered in the current Work Phase. At the end of each Work Phase FORMS-2 processes the Foreground data only.

When screen W01 is next reached if options B, C D or are chosen, the Foreground is overlaid on the current Background contents and then Foreground is cleared to spaces. If option A is selected both Background and Foreground are cleared to spaces.

In this way the new Work Screen is prepared automatically.

Note

It is possible to override this automatic Work Screen preparation for the next phase by means of a Work Screen command, described later, and leave both areas unchanged.

Generally data is entered into the Foreground via the keyboard, and is moved into the Background only from the Foreground. The F Work Screen command described later provide facilities for further manipulation of these areas. In particular it is possible to input a Screen Image file from a previous run into the Foreground, thus enabling amendment of existing forms.

During entry of data into the Work Screen (i.e. Foreground) two modes can be invoked as follows:

- | | |
|--------------|--|
| Edit Mode | The mode in which the user keys data to create the model form. The Initial mode is always Edit Mode. |
| Command Mode | Commands are available to assist in the creation of the edited Work Screen and in its processing. |

EDIT MODE

Edit Mode is identified to the user as the mode in which the cursor can be freely moved to any part of the screen by use of the cursor control keys. Entries may also be made into any part of the screen, in accordance with the screen type selected at the start of this Work Phase.

Fixed Text

In the design of the Fixed Text of a form (i.e. the fixed fields analagous to the pre-printed text on a paper form) any legible characters can be entered anywhere on the screen. This text will be displayed as "prompt" text during a data entry run of the application.

Variable Data

In the design of the Variable Data fields of a form (i.e. the fields analagous to the entry spaces on a pre-printed form) the characters X, Y and 8, 9 can be entered.

When variable data is being keyed in, X denotes an alphanumeric character and 9 denotes a numeric character. If it is required to have two alphanumeric fields contiguous with each other, Y's are placed in the character positions of the second field. Similarly, for contiguous numeric fields 8's are used.

Example 3.1.

INVOICE NO

Suppose in an application the operator must key in an Invoice Number. Then the fixed text in this example could be "INVOICE NO" One example value of an invoice number could be "CA3021". It is necessary to define the area and type of this variable data explicitly for the computer. Hence if

the. invoice number always had two alphanumeric followed by four numerics, the user of FORMS-2 would key in XX9999 at the point on the screen (the dots in this example) where he wishes the operator to key the actual invoice number when the application itself is running. (Note that CIS and L/II COBOL provides an automatic validation of numeric fields).

Additionally, special editing characters can be input to specify numeric edited fields. Note that these fields should be separated by spaces. Numeric Edited Fields are described in the *CIS or L/II COBOL Language Reference Manual*. The valid characters are:-

Z, *, +, -, CR, DB, .(period), ,(comma), B, /, 0(zero), \$

The \$ sign is the currency sign which may be replaced by another sign as specified in the SPECIAL-NAMES clause of the COBOL program either directly or as specified during the Initialisation Phase of the FORMS-2 run (See Chapter 2).

Note

The picture characters S, V, P are not allowed.

COMMAND MODE

To switch to Command Mode from Edit Mode, the user simply presses the RETURN key.

Command Mode is identified, to the user as the mode in which two underline characters initially bound the cursor, and the cursor is constrained to stay within these two characters.

Commands are invoked by keying the command always followed by pressing the RETURN key.

When execution of a command is complete, all commands (except SPACE, ! and Q) return to Edit Mode.

The default command is the underline character () and this causes immediate re-entry to Edit Mode.

Commands

The commands available to the user during the Work Phase fall into three main groups. All commands are entered by typing the command character/s followed by pressing the RETURN key.

1. General Commands

General commands perform such functions as releasing the Work Screen for processing.

2. Work Screen Manipulation Commands

Work Screen Manipulation commands assist in the preparation and editing of the Work Screen. It is recommended that all users become very familiar with these commands.

3. Programming Commands

Programming commands have been introduced mainly for the convenience of the COBOL programmer, and some of them will not be meaningful without an understanding of COBOL. They include commands to assist in producing efficient code, and to give more control over the files output.

Note

Groups 1 and 2 are summarised within HELP screens 1 and 2 (H01, H02) Group 3 is summarised on HELP screens 3 and 4 (H03, H04) . See Appendix C.

General Work Screen Commands

A description of each general command is given below.

- ? Display Help Screens
- Entry of the question mark then RETURN causes display of the first HELP screen, which includes a summary of the General Commands. This screen remains displayed until further input is made.
- Keying RETURN alone at this stage returns to EDIT MODE.
- Keying ? then RETURN again causes display of the next HELP screen, which is a summary of the Work Screen Manipulation commands, and if repeated, each HELP screen in sequence until the end of the series when the first screen is displayed again.
- The ? command is also available from the first (Select Text Type) screen W01.
- ?n Display Help Screen specified by n
- The entry of ? followed by a numeric digit then RETURN directly displays the nth screen in the HELP series.
- (Space) Terminate the Work Phase
- Entry of the space character terminates the current Work Phase and Initiates processing of the data just entered in that Work Screen.
- _ (Underline) Return to Edit Mode
- Entry of the underline character exits from Command Mode back to Edit Mode.
- All commands except SPACE, ! and Q return to Edit Mode. The default command was introduced for convenience in case Command Mode is entered inadvertently.
- Q Quit
- Entry of Q returns to Screen W01 (Select Type of Text).
- Whatever text type is then selected, the current Foreground/Background components of the Work Screen will be unchanged on re-entry to Edit Mode.
- The most likely use for this command is where the default option C (variable data fields) is taken at screen W01 and then the Work Screen is incorrectly used to set up a fixed form. Validation errors then occur with automatic re-entry to Edit Mode. To return to screen W01 and correct the selection without loss of the text just entered, key RETURN to enter Command Mode and enter the Q command. The screen type can now be corrected and the Work Screen has been preserved for reprocessing (as fixed text). The Q command is also available at screen I02, and again its function is to return a step within the phase (in this case back to screen I01).
- ! Terminate FORMS-2 run
- Entry of the ! character causes termination of the program. This command is available throughout the FORMS-2 program, but is normally used on automatic return to screen W01 when a work phase has been completed. When used elsewhere it abandons the run.
- X Reposition Command Area
- The standard command area is in columns 1 and 2 of line 1 as indicated by the two underline characters displayed on entry to Command Mode. FORMS-2 always attempts to restore any data in these positions upon return to Edit Mode.

However to enable these positions to be preserved intact at all times, the facility to reposition the command area is provided. To achieve this, the cursor is placed at the desired location prior to Command Mode being entered. Entry of the character X then causes the required change. The next time Command Mode is entered the prompting underline characters will appear at the new specified location.

* Define Key/Data split for Index Program

This command is particular to generation of the Index Program and is described in Chapter 8. Note however, that if the Index Program option G is taken at screen I02, at some stage prior to termination of the second Work Phase the cursor should be positioned at the first non-key variable data position and the * command entered.

If this is not correctly done, FORMS-2 will continually return to Edit Mode until it is.

Work Screen Manipulation Commands

The commands F and O are preparation commands.

F Invoke Foreground/Background Menu Screen (W02).

Screen W02 contains options to assist in setting up the Foreground component of the Work Screen. The concept of Foreground and Background is described earlier in this Chapter.

The options made available by the FORMS-2 W02 Screen are:-

- A Return to Edit Mode.
- B Clear Foreground to spaces.
- C Clear Background to spaces.
- D Overlay Background data on to Foreground.
- E Overlay Foreground data on to Background.
- F Overlay a Screen Image file onto Foreground. This option allows forms defined in previous runs (and also earlier in this run) to be amended. If selected the user is prompted for the identity of the required file.
- G Overlay a Screen Image file onto Background.
- H Show Foreground. This displays just the Foreground component of the Work Screen for examination. The full Work Screen will be restored on return to Edit Mode.
- I Show Background. This displays just the Background component of the Work Screen for examination.
- J Show a Screen Image file. The user is prompted for the identity and the specified file is displayed, but without corrupting the current contents of either Foreground or Background. This enables a check to be made prior to using option D.

Note

The options H, I and J cause a display which remains until RETURN is pressed.

Fx Specifies Required Foreground/Background Option

x is the option code as contained in screen WP02 above. The specified option is executed and control returned to Edit Mode without display of the WP02 screen.

O1 "Switch Off" Automatic Background/Foreground Preparation.

The Background/Foreground preparation sequence is described under BACKGROUND/FOREGROUND earlier in this Chapter. This command is used to prevent the current Foreground being merged into the Background or either area being cleared for the next phase.

The O1 command remains effective until the command 0 (described below) is entered.

O or OO Reset Background/Foreground Preparation.

The Background/Foreground preparation sequence is reset to automatic (starting at the beginning of the next Work Phase).

Note

The Q command has a similar effect (beginning at the next phase).

The commands C, D, I, K and A are editing commands and are controlled by the position of the cursor at the time Command Mode is entered (i.e. the current cursor position) and operate only on the Foreground data. Background data remains on the same position.

C*n* Insert Spaces

Inserts *n* (1-9) spaces prior to the character at the current cursor position. Only the current line is affected.

D*n* Delete Characters

Deletes *n* (1-9) characters including the character at the current cursor position. Only the current line is affected.

I*n* Insert Blank Lines

Inserts *n* (1-9) lines prior to the line containing the current cursor position, irrespective of the column. Only whole lines can be inserted.

K*n* Delete (Kill) Lines

Deletes *n* (1-9) lines including the line containing the current cursor position. Only whole lines can be deleted using this command.

A*n* Repeat Current Line

Repeats the line containing the current cursor position *n* (1-9) times.

Note

This does not act as an insert. Any Foreground data in the next *n* lines will be overwritten.

Commands U and V are cursor positioning commands.

Horizontal cursor positioning can be achieved by means of the → and ← keys.

Vertical cursor positioning is partially achieved by means of the ↓ and ↑ keys, but the cursor is usually placed at column 1.

Vertical tabulation within the same column can be required when setting up a form. Two "tabulation" commands are provided.

Un Move Cursor Up

Moves the cursor up *n* (1-9) lines from the current cursor position. Cursor position within the line is maintained.

Vn Move Cursor Down

Moves the cursor down *n* (1-9) lines from the current cursor position. Cursor position within the line is maintained.

Programming Commands

G The Dataname Structuring Command

This command is not available if option G (for Index Program) is specified at screen I02.

The default recordname format generated by FORMS-2 for inclusion in the user's COBOL source program for screen formatting is as follows:

```
bbbbbb-rr (01 level)
```

where: *bbbbbb* is the 1-6 character base specified at screen I01

rr is the record number, starting at 00 in the first Work Phase and increasing by one for each subsequent Work Phase.

Note

If the window commands are used to define a window starting in other than line 1 then the default recordname generated will be:

```
bbbbbb-rr-ll
```

where *ll* is the line number.

The default elementary dataname structure generated by FORMS-2 for inclusion in the COBOL source program for screen formatting is as follows:

```
bbbbbb-rr-nnnn
```

where: *nnnn* is the sequence of this field within the screen, starting at 0001.

Alternatively:

The G (or G0) command causes *nnnn* within the dataname to be the screen coordinates of the start of the field. This can sometimes be of use to the programmer as a reference guide when using COBOL facilities to set cursor position.

The G1 command restores the default dataname generation to using sequential field numbers.

The commands J and M are Multiple Spaces and FILLER Commands.

The COBOL interactive ACCEPT and DISPLAY verbs operate only on named fields; FILLER areas are left alone. The time taken to display a screen depends both on the size and also the number of constituent fields.

When processing fixed text screens, FORMS-2 by default generates FILLER wherever multiple spaces appear. On some forms this can result in many small fields separated by small FILLER fields. The problem may be alleviated by:

Jn Reset Multiple Spaces

The *Jn* command resets the number of contiguous spaces FORMS-2 will allow within the VALUE clause of a named field. This is initially set to 1.

n may be 0 - 9

Note

J or J0 will force FILLER's even for single spaces.

An alternative method of forcing spaces within named fields is by use of the underline which is designated for this purpose. Use of the underline character in a field results in an actual space in the corresponding position in the generated VALUE clause.

If it is required to change the designated character from underline to something else (presumably because there is a requirement to generate VALUE "_"), the command M is used as follows:

Mx Change Default "space"

The *Mx* command changes the default character (underline) to that specified by *x*.

Note

If the space character () itself is specified this will force generation of named fields for the entire screen without any FILLER's.

Sn File Output Control Command

This command is not available if Option G (for Index Program) is specified at screen I02.

S (or S0) cancels any other *Sn* commands in effect at the time.

S1 suppresses DDS (& CHK) text generation for this Work Screen. Generation of this text resumes for the next Work Screen unless the same command is repeated in the next phase.

S2 suppresses Screen Image (*Snn*) text generation for this Work Screen. Commonly used to suppress Screen Images of just variable data fields. Again the effect only lasts for the current phase.

S3 results in the user being given the opportunity to override the default Screen Image file identifier for the current Work Screen. Normally if a file already exists with the default identifier the user is given the option of overriding it. If he rejects this option he is prompted for an alternative file identifier. This command forces the alternative file identifier to be requested even when no file exists with the default identifier.

S9 causes FORMS-2 to halt after display of each line of code during DDS generation. FORMS-2 re-ACCEPTs the line before outputting it to the .DDS file. This provides the programmer with a limited editing capability, which may prove useful under special circumstances. This facility is not available if option C or G (for generated programs) is specified at screen I02.

P Cursor Coordinate Command

Causes the coordinates of the current cursor position to be displayed at the command area position. This display lasts a few seconds, after which the Work Screen contents at the command area are restored and control is returned automatically to EDIT MODE. Where sequential field numbers are being used within datanames, this command provides an easy alternative method of ascertaining the coordinates of any field.

Wn Window Commands

These commands are not available if option G (for Index Program) is specified at screen I02.

The "window" defines the area (full lines) to be processed by FORMS-2 when generating DDS text. By default the window is the full screen. Where window start or end is other than start or end of screen, a delimiting line of hyphens may optionally be displayed on the line just outside the window. e.g. if window starts in line 4, delimiters appear along the length of line 3.

The principal use of the window is to allow a form to be created in situ, but save memory by avoiding the description of blank lines at the top of the screen.

Note

Where window is used in this way the generated record name incorporates the start line number of the window which can then act as a guide to the programmer, using the CIS or L/II COBOL ACCEPT/DISPLAY AT coordinates facility.

The detailed commands give the user very comprehensive window formatting capability as follows:

- W (or W0) positions the cursor at current window start. This is the equivalent of the HOM (\\) key when the window facility is in use.
- W1 sets start of window to current line with delimiters on previous line.
- W2 sets end of window at end of current line with delimiters on next line.
- W3 sets start of window to current line without delimiters.
- W4 sets end of window at end of current line without delimiters.
- W5 displays delimiters preceding current window start.
- W6 displays delimiters following current window end.
- W7 erases start delimiters and restores any Work Screen data to display.
- W8 erases end delimiters and restores any Work Screen data to display.
- W9 positions the cursor at current window end.

Notes

1. Delimiters do not corrupt Background/Foreground contents.
2. The output Screen Image will include the full Foreground part of the Work Screen without delimiters, irrespective of whether window has been defined. This could be used to include annotation on the Screen Image which does not affect DDS generation.
3. A useful purpose for the window facility would be if a form is required to be displayed in two stages: the first 10 lines then the second 10 lines. This could be created as a single Screen Image including both sections of the form, and the programmer could 'window in' on the relevant portions as required when the DDS text is generated.

WORK PHASE COMPLETION

To complete the Work Phase of FORMS-2, the user selects Command Mode, keys the SPACE character and then the RETURN character. The SPACE character is the command to release the Work Screen for processing.

FORMS-2 completes the Work Phase (depending on the file selection at screen I02) as follows:

1. If this is a variable data field definition Work Phase (Option C or D at screen W01), validation then occurs with the message:

WORK SCREEN VALIDATION in progress - DO NOT press RETURN

2. If DDS file generation is selected at screen I02, the source code produced is echoed to the screen as it is written to disk. If the S9 command was specified processing stops after each line of code to enable changes to be made as required. This is recommended only if special requirements dictate its use.
3. If a screen image file was requested at screen I02, the screen image is echoed to the screen as it is written to the disk file. The identity of the created file is displayed and the user must press RETURN to continue.
4. Screen W01 is redisplayed so that the run can be terminated or continued.

Notes

1. During validation of variable data only those characters listed in the description of text-types are permitted (plus space). If any other character is encountered, an error is notified by the validation routine by alternately displaying "?" and the offending character to give a flashing effect. This error indication then ceases and FORMS-2 returns to Edit Mode with the cursor positioned under the erroneous character. The user must reissue the SPACE command after making any corrections.
2. FORMS-2 will allow editing characters but will not verify that the combinations of these are valid; COBOL editing rules must therefore be obeyed to ensure error free code. Note that these fields should be separated by spaces.
3. Only Foreground data is output to the Screen Image file.

Chapter 4. DATA DESCRIPTIONS

This Chapter describes the CIS or L/II COBOL Data Descriptions that FORMS-2 can generate in the file `filename.DDS`, and a knowledge of COBOL is a prerequisite for reading it.

The CIS COBOL and L/II extensions to the ACCEPT and DISPLAY verbs allow comprehensive screen handling to be included in a user application. (See *CIS or L/II COBOL Language Reference Manual*). Programming the necessary data description statements can be tedious and expensive in terms of programmer time, particularly since it is very prone to simple errors.

FORMS-2 simplifies the production of error-free data descriptions by allowing screen layouts (forms) to be specified in the most convenient way, namely by setting them up in situ on the screen as described in Chapters 2 and 3. If the facility is invoked by selection of an appropriate option at screen I02 during the Initialisation Phase, FORMS-2 automatically converts this Input to the necessary COBOL statements and outputs these to a data description (DDS) file. The user merely incorporates this code in his source application code by means of the CIS or L/II COBOL COPY verb and uses record-names and data-names consistent with those generated by FORMS-2.

RECORD-NAME AND DATA-NAME GENERATION

At Initialisation screen I01 a base name is requested from the user. This is a 6-character field into which the user enters any name of his choice consistent with COBOL data naming (See *CIS or L/II COBOL Language Reference Manual*). This base is then used to generate the COBOL data-names.

RECORD NAMING

The default record-name generated by FORMS-2 for inclusion in the user's COBOL source program for screen formatting is as follows:

```
bbbbbb-rr (01 level)
```

where: *bbbbbb* is the 1-6 character base specified at screen I01. *rr* is the record number, starting at 00 in the first Work Phase and increasing by one for each subsequent Work Phase.

Note

If the windows commands are used to define a windows starting in other than line 1 then the record-name generated will be:

```
bbbbbb-rr-ll
```

where *ll* is the line number. This serves as a useful reminder to the programmer when coding the appropriate ACCEPT/DISPLAY statements.

DATA NAMING

The elementary data-name structure generated by FORMS-2 for inclusion in the CIS or L/II COBOL source program for screen formatting is as follows:

```
bbbbbb-rr-nnnn
```

where: *nnnn* is the sequence of this field within the screen, starting at 0001.

Sometimes it may be more convenient to the programmer to have the screen coordinates incorporated in the data-name rather than a field sequence number. This can be achieved by use of the G command during the Work Phase.

PICTURE GENERATION

Generation of PICTURE clauses by FORMS-2 depends on the type of text selected at screen W01 at the start of each Work Phase. Note that FORMS-2 will force field boundaries at the end of each line in order to be compatible with certain types of CRT.

FIXED TEXT

At the end of a fixed text Work Phase FORMS-2 generates only FILLER areas or named alphanumeric fields with associated VALUE clauses.

The CIS or L/II COBOL interactive ACCEPT and DISPLAY verbs operate only on named fields; FILLER areas are left alone. The time taken to display a screen depends both on the size and also the number of constituent fields.

When processing fixed text screens, FORMS-2 by default generates FILLER wherever multiple spaces appear. This default can be altered by means of the J command described in Chapter 3. Alternatively (underline) can be used to force inclusion of spaces within a VALUE clause. The default character used for this purpose can be changed by means of the M command described in Chapter 3.

VARIABLE DATA FIELDS

At the end of a variable data Work Phase FORMS-2 generates alphanumeric, numeric or numeric edited fields depending on the actual characters keyed by the user (see Chapter 3). These are usually normal CIS COBOL picture characters 9 and X but note the additional use of 8 and Y as alternatives to 9 and X; and also the exclusion of S, V and P as described under *Variable Data* in Chapter 3.

EDITING THE DDS FILE

Normally the DDS output from FORMS-2 should be all that is required. Where special circumstances dictate the use of particular datanames or the disallowed picture characters, the S9 command (See Chapter 3) will allow DDS lines to be edited prior to output. Alternatively a conventional text editor can be used to edit the file. However note that this editing process must be repeated if ever the form is amended by means of FORMS-2.

It is also possible to suppress completely the DDS output for a particular Work Phase by means of the S1 command. Note that if this is used the record number incorporated in data-names will still be stepped up by 1 for the next Work Phase.

INCORPORATION OF DDS FILE CONTENTS

All that the user has to do to incorporate the generated data descriptions into the application program is to copy in the DDS file using the COPY statement available in CIS or L/II COBOL. This is described in the *CIS or L/II COBOL Language Reference Manual*.

The COPY statement to incorporate the DEMO1 sample forms designed in Chapter 7 would be:

```
000000 COPY "DEMO1.DDS" .
```

and would be coded within the Data Division.

This statement is included in all Check-Out or Index programs generated, and any of these can be referred to for an example.

Chapter 5. THE CHECK-OUT PROGRAM

This Chapter describes the Check-Out program that FORMS-2 can generate automatically while generating the created forms. The Check-Out program enables the user to:

- Validate the DDS file
- Demonstrate the operation of the proposed application
- Check the use of his forms for data entry
- Check the use of his forms for data amendment

The Check-Out source code which is in COBOL includes a COPY statement for the DDS file exactly as it would be coded in the user's application and is therefore a true validation of the DDS file when compiled.

The Check-Out program logic is a sequence of DISPLAY or ACCEPT statements for the screens defined in the FORMS-2 run, in the order in which they were created. Therefore by entering all required forms in a single FORMS-2 run, a demonstration program using all the forms can be simply and rapidly created with no programming necessary. For a complex application the best method might be to create each form in isolation, using screen image output only. FORMS-2 can then be run again to produce the required Check-Out program, using the facility to re-input screen images (the FF command or the F command and the F option in the subsequent screen display). Use of this facility would also enable a complex sequence of screens to be set up for demonstration purposes incorporating the same screen more than once.

After passing through the sequence of screens, Check-Out gives the option of repeating the whole sequence. On the second pass previously entered data is redisplayed, allowing the user to check the use of his forms for both initial data entry and data amendment.

CHECK-OUT PROGRAM GENERATION

The facility is invoked by selection of an appropriate option at screen I02 during the Initialisation Phase. Note the the default option results in generation of the Check-Out Program.

The source code of the program is written to a file named:

basename.CHK

where: basename is the name entered by the user at screen I01 during the Initialisation Phase.

CHECK-OUT PROGRAM COMPILATION

The program is then compiled from the CHK file. The following files must be present during compilation:

basename.DDS - the DDS file produced in the FORMS2 run. This must be on the device/directory selected at Screen I02, or the default drive if none was selected.

FORMS2.CH1, FORMS2.CH2 - the skeleton for the Check-Out program on the default device/directory

The Check-Out program is then compiled in the usual way by entering the standard CIS or L/II COBOL compile command for your Operating System with the source file:

basename.CHK

See Appendix D

Details of compilation using the CIS or L/II COBOL Compiler are given in the appropriate Operating Guide.

CHECK-OUT PROGRAM RUNNING

LOADING

The program can be loaded immediately after compilation by use of the standard O/S run command and the name of the intermediate code file:

basename.INT

However, to be able to load directly in subsequent use the = parameter of the run command should be used and the SAVE file renamed. (See Appendix D for Operating System specific commands). Thereafter the direct load command can be used.

CHECK-OUT PROCESSING

The basic function of the Check-Out program is to display the fixed text fields of the form and enable data to be entered into the variable data fields of the form in the sequence in which the screens were created.

However the detailed logic is slightly more sophisticated. The following notes make references to the options taken for screen type at Screen W01, and these are discussed in Chapter 3.

Fixed Text Screens

The fixed text of a form is displayed. If there are two consecutive fixed text forms, Check-Out pauses after the first display until the user presses RETURN.

Fixed text on clear screen

If option A was taken for the creation of the form, Check-Out clears the CRT before displaying the screen.

Fixed text on last screen

If option B was taken for the creation of the screen, any text displayed remains on the CRT except where it is overwritten by the text of the new screen.

Variable Data Screens

An ACCEPT statement is issued for a variable data screen, allowing the user to enter data in the unprotected areas, (i.e. the fields specified by means of X's and 9's etc.).

Users can check the extents of the fields. For numeric fields they can also check that only numeric characters may be entered, and the effect of entering the left zero fill character ".". (Use of the "." character is described in the *CIS or L/II COBOL Language Reference Manual* under the ACCEPT Statement).

On other than the first pass through the sequence of screens entered data is redisplayed before the ACCEPT is issued.

If the variable data screen includes numeric edited fields, the ACCEPT for the screen is followed by a corresponding DISPLAY to show the effect of the editing or normalisation performed by the CIS COBOL run time system. Note that the normalised fields are not automatically echoed to the CRT.

CHECK-OUT COMPLETION

After the entire sequence of screens has been passed, the Check-Out program displays:

```
CHECK-OUT completed  
Repeat? [N] (Y=Yes)
```

If it is required to repeat the sequence of screens. Key Y and press RETURN.

Otherwise simply press RETURN to take the default to terminate the program.

Chapter 6. SCREEN IMAGE FILE

This Chapter describes the Screen Image file that FORMS-2 can generate in addition to (or instead of) the CIS COBOL data description statements described in Chapter 4. These files contain exact text images of the user described forms. These form images can be:

- Used to provide the basis for amendments to the form
- Printed to yield printed copies of the form.
- Used as a means of communication between the system designer and the applications programmer.

SCREEN IMAGE FILE GENERATION

The facility is invoked by selection of an appropriate option at Screen I02 during the Initialisation Phase. Note that the default option will cause screen image output.

Screen images are output to files named:

basename.*Snn*

where: *basename* is the name entered by the user at screen I01 the Initialisation Phase, *nn* is a number 00 - 99

The default filename can be overridden by issuing the S3 command during the Work Phase. This causes FORMS-2 to request input of the required filename during processing of this Work Screen.

A separate file is created at the end of each Work Phase, the numeric part of the name (*nn*) incremented by 1 each time. A screen image file is structured as a standard line sequential file with a record for each line of the screen. Each screen image contains only text entered during the Work Phase in which it is generated (i.e. Foreground data - see Chapter 3). Thus for a variable data Work Phase the output screen image contains only X's, 9's, Y's and 8's.

It is possible to suppress the screen image output from any Work Phase by issuing the S2 command during that phase. If this command is used the numeric part of the filename extension will still be updated for the next phase to keep in line with the record numbering within the generated data descriptions (DDS)

FORMS-2 MAINTENANCE

The COBOL Data Description statements have been generated from a user-designated form by FORMS-2 in a DDS file. There is likely to be a continuing requirement to make corrections and adjustments to maintain the form. The DDS file can be maintained using a conventional text editor, but this involves the high risk of simple but expensive errors which FORMS-2 eliminates. Therefore the user's form is output to a screen image file as an exact image, and FORMS-2 provides the facility to read screen images back from disc to allow for further amendment. This is achieved by running FORMS-2 and issuing the FF command once the first Work Screen is reached (see the Fx Work Screen Manipulation command in Chapter 3). The user is then prompted for the identity of the screen image required. FORMS-2 reads the screen image file into the Foreground area of the Work Screen and then returns to Edit Mode. The form is then displayed as if it had just been keyed and any required amendments can be made before releasing the screen for processing by means of the SPACE command.

Note

When FORMS-2 is used for maintenance in this way it will overwrite the existing files, but only after issuing warnings that the files already exist, and receiving confirmation to proceed.

For screen image files FORMS-2 offers the facility of specifying an alternative file identity if the user wishes to retain the old version.

PRINTED FORMS

The screen image files are created as line sequential files in accordance with the conventions of the operating system. Standard software can therefore be used to print them, and the resultant hard copy will be an exact image of the user's form with no risk of transcription error.

FORM IMAGES IN THE DESIGN PROCESS

Form images can be used as a step within the applications design process, providing a valuable part of the designer/programmer interface.

For interactive applications, design of the user interface (i.e. the screen layouts or forms) may take place well in advance of the actual program being written, and the forms designer need not have any detailed knowledge of COBOL.

FORMS-2 enables a non-technical user to generate valid COBOL statements. An experienced COBOL programmer can make use of commands available to generate the most efficient code (e.g. by influencing the number of fields to be displayed).

Thus it may sometimes be advantageous to use screen image output alone as an intermediate stage in the design process, with the programmer using the image files as input to FORMS-2 to produce the final DDS file. If FORMS-2 is used in this way, both fixed text and variable areas could conveniently be indicated on a single fixed text screen. The programmer can easily then use this screen to generate the DDS file, and the form designer does not need to know any details of COBOL data field specifications.

Chapter 7. FORMS-2 USER SCREEN GENERATION EXAMPLE

It is required to build the data entry form:

```
NAME      [                ]
ADDRESS   [                ]
          [                ]
          [                ]
TEL       [                ]
```

where NAME and ADDRESS are alphanumeric fields and TEL is a numeric field. At data entry time after insertion of the name, address and telephone number P. Smith, 8 George Street, Plymouth, Devon, 88326, the form is required to appear as:

```
NAME      [SMITH. P        ]
ADDRESS   [8 George Street, ]
          [Plymouth       ]
          [Devon          ]
TEL       [88326         ]
```

It is assumed the system is booted, the issued files have been copied to the CIS or L/II COBOL O/S system disk so that the CHK file contents can be compiled (Step 13), and that FORMS-2 has been configured to your CRT.

The following steps must then be carried out:

1. The operator loads FORMS-2 by entering the load command for your O/S.

Note that the program name for load purposes is FORMS2.

See Appendix D for specific Operating System format for this command.

2. FORMS-2 displays Screen I01 requesting a six-character base for file-names and data-names followed by four other questions. If the CRT is standard (24 lines) no further questions need be answered for this screen. Key DEMO1 followed by the RETURN key if the default screen size (24) is correct.
3. FORMS-2 displays Screen I02 to request the output file option type and drive number. Key RETURN to accept the default values.
4. FORMS-2 displays Screen W01 to request the Screen Type option. Note the default "A" and press the RETURN key.
5. FORMS-2 displays a blank screen. Use the cursor control keys and the normal character keys to set up the following text on the screen:

```
NAME      [                ]
```

```
ADDRESS [          ]
        [          ]
        [          ]
TEL     [          ]
```

Press the RETURN key.

6. FORMS-2 puts "___" in the top left of the screen. Press the SPACE and RETURN keys
7. FORMS-2 processes the screen to create a fixed text form. This takes a short period and involves the following displays on the CRT:

```
DDS source code as generated, followed by a redisplay of the fixed
text as it is written to the Screen Image file.
```

A message is then displayed giving the name of the fixed text Screen Image file created. Press RETURN as requested.

8. FORMS-2 displays Screen W01 to request the Screen Type option. Note the default is "C" and press the RETURN key.
9. FORMS-2 displays the fixed text screen as background data. The operator then uses the cursor control keys and keys in X's and 9's alone to set up the screen as follows:

```
NAME      [XXXXXXXXXXXXXXXXXXXXX]
ADDRESS   [XXXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXXX]
TEL       [999999999]
```

Press the RETURN key.

10. FORMS-2 displays "___" in the top left hand of the screen; press the SPACE and RETURN keys. There is a short pause while FORMS-2 validates the screen content, during which the following message is displayed:

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

11. FORMS-2 processes the X's and 9's to create a variable data form, with the following displays to the CRT as it goes:

```
DDS source code as generated, followed by a redisplay of the fixed
text as it is written to the Screen Image file.
```

A message is then displayed giving the name of the variable data screen image file created. Press RETURN as requested.

12. FORMS-2 displays screen W01 again. Key ! followed by RETURN to terminate the run. FORMS-2 displays the names of the DDS and CHK files created and displays an END OF RUN message.
13. Compile the check-out program by typing the standard CIS COBOL compilation command for your Operating System with the directive COPYLIST and file name DEMO1 . CHK (See Appendix D).

14. When the compilation finishes, the two screens can be checked out by using the standard run command for your Operating System to load the intermediate code from file:

```
DEMO1 .INT
```

See Appendix D.

15. The Demonstration program will then run. The fixed data form is displayed on the screen. The variable data form is used to accept data.

Satisfy yourself that the cursor can only be placed in the variable fields, and that the data keyable into the fields depends on whether X or 9 was specified. The effect of left fill character "." may also be tested.

When satisfied, press RETURN to complete. A message is displayed as follows:

```
CHECK-OUT completed
Repeat? [N] (Y=Yes)
```

Press RETURN to accept the No default and complete.

16. The Check-Out program displays:

```
END OF FORMS2 CHECK-OUT
```

Note

The variable form is used in the demonstration for ACCEPTing data. In practice the form can be used for DISPLAYing data as well as ACCEPTing it. The demonstration shows the extent and type of each field which will be the same in DISPLAY as well as ACCEPT. A useful technique for clearing just the variable data fields on the screen is to move spaces to the ACCEPT record and then display it.

17. You can now examine the disk files:

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

to check the output from FORMS-2 during this use.

18. You have now learnt how to use FORMS-2 to create screens of fixed and variable data automatically for inclusion in your CIS or L/II COBOL program.

If you continue with steps 19 onwards you will learn to update both the fixed and variable data screens already created by moving them from background into foreground.

19. Reload FORMS-2 by typing the load command for your O/S. See Appendix D.

20. FORMS-2 displays Screen I01 requesting the six-character file- and data-name base as at step 2. Answer the questions as necessary at step 2 and press RETURN.

21. FORMS-2 displays Screen I02 requesting the output file option type and drive number; key RETURN to accept the default values.

22. FORMS-2 displays the message:

```
DEMO1.DDS already exists
```

```
overwrite? [N] (Y=Yes)
```

Key Y and press RETURN

Note

If the No default is entered here, the run is abandoned.

23.FORMS-2 displays the message:

```
DEM01.CHK already exists  
overwrite? [N] (Y=Yes)
```

Key Y and press RETURN

Note

If the No default is entered here, the run is abandoned.

24.FORMS-2 displays Screen W01 again. Press RETURN to accept the default option A.

25.FORMS-2 displays a blank screen in Edit Mode. Press RETURN to enter Command Mode, then F followed by RETURN to invoke the Foreground/Background selection screen. (We want to update our form so it must be in Foreground).

26.FORMS-2 displays the Foreground option screen. Enter option F then the filename DEM01 .S00, then press RETURN.

27.FORMS-2 displays screen W02 again. Select option A to return to Edit Mode.

28.FORMS-2 displays the fixed text screen (previously created at step 5). Move the cursor to the word ADDRESS and overwrite it with ABODE. Remember to overwrite the extra characters SS with spaces, and then press RETURN.

29.Enter the SPACE command, then RETURN.

30.FORMS-2 displays the following message reminding you that your altered fixed text Screen Image is about to overwrite your previous Screen Image in the file:

```
DEM01.S00 already exists  
overwrite? [N] (Y=Yes)
```

Key Y and press RETURN

Note

If the No default is entered here, a file identity for a new Screen Image would be requested.

31.FORMS-2 displays the screen image and then displays the file name as follows:

```
File created = DEM01.S00
```

Press RETURN to continue.

32.FORMS-2 displays screen W01 with option C as default to enable specification of variable data fields. Enter RETURN to accept the default.

33.FORMS-2 displays the altered fixed text as followed to assist in defining the variable fields.

```

NAME      [                ]
ADDRESS   [                ]
          [                ]
          [                ]
TEL       [                ]

```

Press RETURN to enter Command Mode then F then RETURN.

34.FORMS-2 displays the Foreground/Background Operations screen again. Enter the option F then the file name DEMO1 .S01, then press RETURN to retrieve your variable text created at step 9.

35.FORMS-2 displays Screen W02 with option H as default. If you press RETURN to accept this default, FORMS-2 displays the current Foreground contents. Note that this is only the X's and 9's that define the variable data fields (the fixed text is in the Background area). Press RETURN to re-invoke Screen W02.

36.FORMS-2 displays Screen W02 with A as default. Press RETURN to accept this default.

37.FORMS-2 displays the whole form again. (We could now alter the variable text fields if required).

You have now seen facilities to retrieve fixed text and variable text from previously created files. Note that with a small number of variable data fields such as in this example it would, in practice, be easier to re-key them.

38.Press RETURN then SPACE then RETURN to process the altered form. Again there is a pause while FORMS-2 validates the variable fields.

39.FORMS-2 produces the DDS file then displays the message:

```

DEMO1.S01 already exists
overwrite? [N] (Y=Yes)

```

40.A message is displayed as follows:

```

File created = DEMO1.S00

```

41.FORMS-2 displays screen W01 again. This time enter ! and press RETURN to complete the run.

42.Repeat steps 13 to 16 if you wish to run the Check-Out program again to verify the altered form.

Chapter 8. INDEX PROGRAM

FORMS-2 provides facilities for automatically generating a COBOL Index program to create and maintain an indexed sequential file. The input required to generate the Index program and use it to maintain files is supplied interactively by the Operator through the CRT.

The user designs a data entry screen using FORMS-2 by specifying the fields that will comprise the indexed sequential file records in the usual fixed text and variable text work phases described in Chapter 3.

The user interface to the generated Index program is simply the form designed by the user that reflects the desired record structure. Users need give no thought to setting up specific 'command' areas, but only to consider their data requirements.

It should be noted that the user must have access to the CIS or L/II COBOL software to compile the source Index program that FORMS-2 produces.

The generated Index program is written to the file `filename.GEN` and provides the following facilities required for the creation and maintenance of an indexed sequential file.

- Select records by key field for display (Enquiry by key field)
- Select records sequentially for display (Sequential Enquiry)
- Amend existing records
- Delete existing records
- Insert new records

The program has been developed so that it is not necessary for the user to explicitly state the facility to be invoked at any one time; the program is able to follow the logic from the way the actual data and cursor position are manipulated.

It can be seen that only the variable text data is written to the file and the fixed text data is merely a template to enable each field to be entered separately at data entry time.

A record in the indexed sequential file is constructed by concatenating the variable fields of the form. In the order in which they appear. The record must include a key area by which it can be uniquely accessed. the Index program logic requires that this key area must be at the beginning of the record i.e. must be the first integral field/s in the form, and must not exceed 32 characters in length.

This key area constitutes part of the record data. For convenience, the remaining fields are known as the data fields.

Chapter 9 shows the sample application used in Chapter 4 adapted to create and maintain a file of names, addresses and telephone numbers.

INDEX PROGRAM GENERATION

An Index program is generated using FORMS-2 as described in Chapters 2 and 3.

All existing FORMS-2 facilities are present, but logic is incorporated to prevent the use of inappropriate features if the Index Program option is taken. The steps involved are:-

1. Initialisation
 - a. Screen I01
Specify name-base etc. as normal.
 - b. Screen I02

Specify option G for Index program generation.

2. Work Phase One

a. Screen W01

Work Screen Selection - The program forces the default option 'A' for fixed text entry by refusing to accept anything else. (except ! to abandon the run or ? to display Help screens).

b. Fixed Text Work Screen

A blank work screen is then displayed for input of the fixed text form.

FORMS-2 commands as described in Chapter are available *except*:-

G The generated program relies on the default dataname structure. This command is rejected.

S It would be inappropriate to switch off DDS generation, and this command is rejected.

W This feature is not available to the user, and the command is rejected. However the program reserves the bottom line for potential use in the generated program for system messages ("RECORD NOT FOUND" etc.), and a delimiting line of hyphens marks this fact.

The screen is released for processing by the sequence 'RETURN SPACE RETURN', when the fixed text screen has been completely entered.

The Work Screen Selection screen is again displayed.

3. Work Phase Two

a. Screen W01

b. This time the program forces the default option 'C'.

Variable fields are specified as described in Chapter 3 i.e., X/Y/8/9 and editing characters. At some time before releasing this screen it is necessary to define the end-of-key/start-of-data bound within the record. This is done by positioning the cursor on the first data field, entering Command Mode and keying the '*' command (i.e. the sequence 'RETURN * RETURN').

Note

A key field cannot exceed 32 characters.

The screen is released by the usual 'RETURN-SPACE-RETURN' sequence. If the program is not satisfied with the specification of the key/data boundary it will return to Edit Mode.

Upon completion of the variable text screen FORMS-2 completes its processing and terminates automatically without any need for the termination (!) command. In fact the ! command is only used to abandon the run when generating the Index Program.

FILES GENERATED

The following files are written to disk by FORMS-2.

basename.S00	-	Screen image files
basename.S01		
basename.DDS	-	COBOL data description statement file
basename.GEN	-	Source file for the generated Index program.

INDEX PROGRAM COMPILATION

The Index program can now be compiled from the .GEN file in the usual way. The following files must be present during compilation:

basename.DDS	the DDS file produced in the run. This must be on the drive selected at screen I02, or the logged drive if none was specified.
FORMS2.GN1	the skeleton for the generated program on the logged drive.
FORMS2.GN2	

The Index program is then compiled in the usual way by entering the standard CIS and L/II COBOL compilation command for your Operating System (O/S) to load the Index program from file:

```
basename . GEN
```

See Appendix D

Details of compilation using the CIS and L/II COBOL compilers are given in the appropriate Operating Guides.

After compilation, the user can run the generated program.

INDEX PROGRAM RUNNING

LOADING

The program can be loaded immediately after compilation by using the standard run command for your Operating System to load the program from file:

```
basename . INT
```

However, to be able to load directly in subsequent use the = directive of the command must be used, and the SAVE file renamed to basename.COM. See Appendix D for the commands for your O/S and the CIS or L/II COBOL Operating Guide for fuller details of load directives. Thereafter the direct load command can be used.

DATA PROCESSING FACILITIES

Immediately the program is loaded, the user designed form is displayed.

The form remains on the screen throughout a run, processing being controlled by manipulation of the data in the variable fields.

A screen display reflects the structure of a single record. The required processing function is instigated by entering data and positioning the cursor as described below, and then pressing the RETURN key. Index program messages are displayed in an unused area of the screen as necessary.

The basic operator functions and Index program messages are described below, and will suffice in general use. Details of the Index program interpretation of data manipulation and cursor position follow this description.

Enquiry by Key Field

Amend key fields only to required key, and press RETURN. The required record is displayed. If the record is not found (i.e. key cannot be found) the message RECORD NOT FOUND is displayed.

Sequential Enquiry

Simply press RETURN to show next record. If the end of the file is reached, the message END OF FILE REACHED - RETURN WILL TERMINATE is displayed.

Amend Displayed Record

Amend data fields only and press RETURN. The message RECORD AMENDED is displayed.

Delete Displayed Record

Press the HOM or # key then press RETURN. The message RECORD DELETED is displayed and the data fields are blanked out.

Insert New Record

Amend key and data fields as required and press RETURN. If the data fields currently displayed do not need changing (i.e. it is required to enter the existing data fields under a new key) prior to pressing RETURN *either* press HOM (#) *OR* press ↑ repeatedly until end of the list data field is reached.

The message NEW RECORD WRITTEN is displayed if insertion takes place.

If a record already exists with the specified key the current display is retained and the warning message RECORD ALREADY EXISTS WITH THIS KEY is displayed. The facilities available on the subsequent input are restricted to three as follows:

1. Force replacement of existing record:

Either press HOM *or* press ↑ repeatedly until the cursor reaches the end of the last data field, then press RETURN.

The record is replaced and, the message RECORD REPLACED is displayed.

2. Amend key field and re-attempt the insertion:

Amend key fields and press RETURN (cursor position is immaterial).

3. Abandon insertion attempt and display existing record:

Press RETURN only.

Terminate Run

Enquire up to the end-of-file by means of continual sequential enquiry or a combination of enquiry by key to a specific record, then sequential enquiry.

When end-of-file is reached the message END OF FILE REACHED - RETURN WILL TERMINATE is displayed. Press RETURN to terminate the run.

USE ON MULTI-USER SYSTEMS

The Index program is compatible with the FILESHARE optional product for use with LEVEL II and CIS COBOL.

The Index program will lock any record that you access, thus preventing another user from updating that record whilst you are processing it. The lock is released when you move on to another record. Similarly it will not make available to you a record which another user has locked.

The effects on Index operation are as follows:-

Enquiry

If you attempt to read a record, either by sequential enquiry or by key, and that record is locked, the message RECORD LOCKED is displayed. The key of the locked record is displayed but the data fields are blanked out. On the subsequent input you may:

1. Retry the enquiry, Press RETURN only.
2. Abandon the enquiry and read next record; Move cursor away from default (start-of-data) position and press RETURN.
3. All other functions operate as normal: i.e., as though the enquiry had failed with RECORD NOT FOUND.

Amend Displayed Record

This will be unaffected because the preceding read will have locked the record, denying any other user access.

Insert New Record

If a record exists with the specified key then RECORD ALREADY EXISTS WITH THIS KEY will be returned irrespective of whether the record is locked or not. The subsequent input is as described earlier for that condition, except that:

1. If you attempt to force replacement and the record is locked, the message EXISTING RECORD LOCKED is displayed. The subsequent input is again as described for the RECORD ALREADY EXISTS condition.
2. If you simply try to display the existing record and it is locked, this will be handled exactly as described under ENQUIRY.

Note

Immediately after update operations (Amend/Insert), Index attempts to read the record again to re-establish the lock:

Under exceptional circumstances it is possible for another user to read the record, thus locking it for his own purposes, before the lock can be re-established. In this case the normal message confirming the update will be displayed, immediately followed by the RECORD LOCKED message and blanking of the data fields. You will then be in the same position as if an enquiry had failed because of a locked record, and the subsequent input is as described under Enquiry.

USER REQUIREMENT INTERPRETATION SUMMARY

The Index program interprets the user's requirements depending on the change status of key and data fields and the cursor position as follows:

Key and Data Fields Unchanged

The function performed depends upon cursor position as follows:

- If an end-of-file condition has just been reported, a request to terminate the run is assumed irrespective of cursor position.
- Otherwise if the cursor has been moved to the HOM position and a record is currently displayed, a delete request is assumed.
- If neither of these conditions exists, a request to display the next record relative to the 'current' position in the file is assumed.

Key Changed and Data Unchanged

The function performed depends on cursor position as follows:

- If cursor is moved to either the HOM position or the last data character position, an attempt to Insert a record is assumed, and processing is as described under Key and Data Changed
- Otherwise an enquiry with respect to this key is assumed, and either the record is displayed or its absence is reported, or a lock condition is reported.

Key Unchanged and Data Changed

This is a request to update the file. If a lock has previously been established against this record it will be amended. Otherwise an insert attempt is assumed, and processing is as described under Key and Data Changed.

Key and Data Changed

This is a request to insert a new record. However, it is assumed that the user should not overwrite a record without at least being informed of its presence. Therefore if a record exists with the specified key, a warning message is displayed, and the subsequent three functions can be performed depending on the change status of key and data fields and the cursor position:

1. Key and Data Unchanged

The function required depends on cursor position as follows:

- If the cursor has been moved to either the HOM position or the last data character position, a request to overwrite the existing record is assumed. Either the existing record is overwritten, or an existing-record-locked condition is reported in which case the subsequent input is processed in the same way as the current input.
- If the cursor is at any other position a request to abandon the insertion attempt and display the existing record is assumed, and processing is as described under Key Changed and Data Unchanged.

2. Data Unchanged and Key Changed

An attempt is made to insert the data under the new key irrespective of cursor position. If necessary the warning message will be repeated.

3. Key and Data Changed

A normal insert request as described above is assumed.

Chapter 9. USER INDEX - PROGRAM EXAMPLE

It is required to generate an indexed sequential file that contains records of names, addresses and telephone numbers with name as key field, and process these records using the form as used in Chapter 7:

```
NAME      [                ]
ADDRESS   [                ]
          [                ]
          [                ]
TEL       [                ]
```

NAME and ADDRESS are alphanumeric fields and TEL is a numeric field.

At data entry time after insertion of the name, address and telephone number P. Smith, 8 George Street, Plymouth, Devon, 88326, the form is required to appear as:

```
NAME      [ SMITH. P      ]
ADDRESS   [ 8 George Street, ]
          [ Plymouth     ]
          [ Devon        ]
TEL       [ 88326       ]
```

It is assumed the system is booted, the issued files have been copied to the CIS or L/II COBOL O/S system disk so that the Index program can be compiled (Step 13), and that FORMS-2 has been configured from your CRT.

The following steps must then be carried out:

1. The operator loads FORMS-2 by entering the load command for your O/S.
See Appendix D for specific Operating System format for this command.
2. FORMS-2 displays Screen I01 requesting a six-character base for file-names and data-names followed by four other questions. If the CRT is standard (24 lines), no further questions need be answered for this screen. Key DEMO2 followed by the RETURN key if the default screen size (24) is correct.
3. FORMS-2 displays Screen I02 to request the output file option type and drive number. Key G then RETURN to select the option for the Index program.
4. FORMS-2 displays Screen W01 to request the Screen Type option. Note the default "A" and press the RETURN key.
5. FORMS-2 displays a blank screen with the end of window one line up from the bottom of the screen and delimiters in the bottom line. Use the cursor control keys and the normal character keys to set up the following text on the screen:

```

NAME      [          ]
ADDRESS   [          ]
          [          ]
          [          ]
TEL       [          ]
    
```

Press the RETURN key.

6. FORMS-2 puts "___" in the top left of the screen. Press the SPACE and RETURN keys.
7. FORMS-2 processes the screen to create a fixed text form. This takes a short period and involves the following displays on the CRT:

```

DDS source code as generated, followed by a redisplay of the fixed
text as it is written to the Screen Image file.
    
```

A message is then displayed giving the name of the fixed text Screen Image file created. Press RETURN as requested.

8. FORMS-2 displays Screen W01 to request the Screen Type option. Note the default is "C" and press the RETURN key.
9. FORMS-2 displays the fixed text screen as background data; now use the cursor control keys and key in X's to fill the NAME variable data field. Move cursor to the first character position in the address variable data field and then press RETURN to enter Command Mode. Enter * to set the first character position in the ADDRESS variable data field as the start of data position and then press RETURN. Continue to enter X's and 9's to fill the data fields as shown below.

```

NAME      [XXXXXXXXXXXXXXXXXXXXXXX]
ADDRESS   [XXXXXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXXXXX]
TEL       [999999999]
    
```

Note that you have now specified the NAME variable data field as the key field.

10. FORMS-2 displays "___" in the top left hand of the screen; press the SPACE and RETURN keys. A message is displayed showing validation in progress.
11. FORMS-2 processes the X's and 9's to create a variable data form, with the following displays to the CRT as it goes:

```

DDS source code as generated, followed by a redisplay of the fixed
text as it is written to the Screen Image file.
    
```

12. FORMS-2 terminates automatically after displaying the end of run screen:

```

Files created DEMO2.DDS
              DEMO2.GEN
END OF FORMS2 RUN
    
```

13. Compile the Index program by typing the standard CIS or L/II COBOL compilation command for your Operating System (O/S) using the Index program source file name:

```
DEMO2.GEN
```

and the COPYLIST directive of this command.

14. When the compilation finishes, the generated Index program DEMO2 can be run by use of the standard run command for your O/S to load the program intermediate code from file:

```
DEMO2.INT
```

The compilation and run commands for your O/S are described in Appendix D.

15. The generated Index DEMO2 program will then run. Your screen as designed in step 9 is displayed. The fixed text form is displayed on the screen. The variable data fields are used to accept data.
16. You are now ready to practice all the file maintenance commands. The next steps show all of these in practice but you can vary the sequence or add any steps to these once you have gained confidence.
17. To insert the first record into the new indexed sequential file, simply key names and addresses into the screen format terminating each record by RETURN key. (Remember to enter surname first before initials to keep the application feasible).
18. Enter two more complete records overkeying all data from the previous record, because all displayed data is written to the file.

When three records have been inserted, you can amend the second record as follows:

Enter the name field as for the second record added followed by RETURN. The whole record is displayed because the name is the key which finds that record. You have now seen the enquiry facility operated. All records can be recalled as easily as that.

19. Change the town field and press RETURN. The message RECORD AMENDED is displayed.
20. Press RETURN and the third record is displayed. You could progress through a whole file in this way.
21. To delete the third record entered move the cursor to HOME position and press RETURN. The fields clear showing deletion of that record, and a message RECORD DELETED is displayed.
22. Press RETURN. The Index program attempts to show the next record but one does not exist so an end-of-file message is shown: END OF FILE REACHED - RETURN WILL TERMINATE.
23. Press RETURN with end-of-file showing and termination occurs.

You have now seen the record handling method demonstrated and can, if you wish, generate a more ambitious Index program or reload as at step 14 and familiarise further with record manipulation.

24. Before you do this, however, you can examine the files on the disk.

```
DEMO2.DDS - Data Description Statements for form (COBOL source)
DEMO2.GEN - Source code of Index program DEMO2
DEMO2.INT - Intermediate code of Index program DEMO2
DEMO2.LST - List file code of Index program DEMO2
DEMO2.IDX - Index file
DEMO2.IDX - Indexed Sequential file
DEMO2.DAT - Sequential data file
```

Notes

1. The two files DEMO2.IDX and DEMO2.DAT constitute the Indexed Sequential file created by the generated Index program, and in any further runs of this program these two files will be used.

2. This example shows the demonstration program run in a _ single-user environment. The program does however include facilities for use in a file-sharing environment with the Micro Focus FILESHARE package and appropriate error handling and notification is provided if used in this environment and locked files or records are encountered.

Appendix A. INITIALISATION SCREENS

The Initialisation screens I01 and I02 shown below are displayed automatically at Initialisation Phase (See Chapter 2).

```
FORMS2 V1.3                INITIALISATION PHASE                SCREEN I01

FORMS2 PARAMETERS:

      DATA-NAME & FILE-NAME    [      ]    (1-6 alphanumeric characters)

      CRT lines                  [23]        (22 or 23 or 24)

SPECIAL-NAMES clause:

      CURRENCY SIGN             [$]         (ANSI currency signs only)

      DECIMAL-POINT             [.]        (". " or ", ")

Press RETURN when complete
```

```
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
```

Appendix B. WORK SCREENS

The Work Screen W01 shown below is displayed at the start of the Work Phase. Work screen W02 can be summoned by the F command (See Chapter 3).

```
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
                               (CIS COBOL Standard only)
Press RETURN when complete
```

```
FORMS2 V1.3                                WORK PHASE                                SCREEN W02
FOREGROUND/BACKGROUND OPERATIONS:
      OPTION           [ ]      (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
```

Appendix C. HELP SCREENS

The four screens contained in this Appendix (H01 to H04) can be summoned for display by the Operator keying ? from Command Mode at any time. They are intended for advisory purposes only. The command ?n (where n is the number 1 to 4 corresponding to the screen number H0n) summons a particular Help Screen.

```
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
```

```
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 characters 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
```

```
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

FORMS2 V1.3

HELP SCREEN

SCREEN H04

PROGRAMMING COMMAND SUMMARY:

W = Position cursor to current window start
W1 = Start window at cursor line
W2 = End window at cursor line
W3 = Start window et 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

Appendix D. OPERATING FORMS-2 WITH CP/M

To clarify the sequences, the CP/M prompts are included before the commands in this Appendix. The symbol << indicates that the RETURN key or equivalent should be pressed once.

It is assumed that CP/M is loaded, and that the issued files have been copied to the CIS COBOL CP/M system disk.

Note

FORMS-2 is preconfigured for your CRT. It can be reconfigured if necessary for any other special CRT.

FORMS-2 LOADING

To load FORMS-2 as issued, the command is:

```
B>FORMS2<<
```

FORMS 2 CHECKOUT PROGRAM COMPILATION.

To compile the Check-Out program that enables you to check your fixed text and variable data fields, the general command is:

```
B>COBOL basename.CHK COPYLIST<<
```

In the sample runs in Chapters 2, 7 and 9, basename is, of course, DEMO1 and DEMO2 respectively.

FORMS-2 CHECK-OUT PROGRAM RUNNING

The Check-Out program that enables you to check your fixed text and variable data fields can be loaded immediately after compilation by the general command:

```
B>RUN basename.INT<<
```

In the sample runs in Chapters 2, 7 and 9, basename is, of course, DEMO1 and DEMO2 respectively.

To be able to load the Check-Out program directly in subsequent use the following general command is entered:

```
B>RUN = basename.INT<<
```

Thereafter the general command following can be used to load the Check-Out program:

```
B>basename<<
```

FORMS-2 INDEX PROGRAM COMPILATION

To enable the Index program that processes an indexed sequential data file from your FORMS-2 screens to be compiled the following general command is entered:

```
B>COBOL basename.GEN COPYLIST<<
```

FORMS-2 INDEX PROGRAM RUNNING

The Index program that processes an Indexed sequential data file from your FORMS-2 screens can be loaded immediately after compilation by the general command:

```
B>RUN basename.INT<<
```

In the sample run in Chapter 9 basename is, of course, DEMO2.

To be able to load the Index program directly in subsequent use the following general command is entered:

```
B>RUN = basename.INT<<
```

Thereafter the general command following can be used to load the Index program:

```
B>basename<<
```

Colophon

This book was reconstructed into DocBook format from a scanned PDF found on the Internet. The PDF file already had OCR performed and the text was embedded in the file.

Source version: 1.0
