****

FORMS-2 Utility Manual

Version 1.3

for CIS or LEVEL II COBOL

UNIX Operating System Implementation

****

Document Release 1.3

June 24, 1983

Date of Printing: July 18, 1983

NOTICE

The following are registered trademarks:

FORMS-2 is a trademark of Micro Focus Ltd.

LEVEL II (L/II) COBOL is a trademark of Micro Focus Ltd.

CIS COBOL is a trademark of Micro Focus Ltd.

UNIX is a trademark of Bell Laboratories.

TABLE OF CONTENTS

TABLE OF CONTENTS

PREFACE

Overview

This manual describes your FORMS-2 package to design, create and edit interactive screen layouts for use with CIS or L/II COBOL application programs. It provides the necessary information for the use of the FORMS-2 package on the UNIX operating system.

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 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 7 are descriptions of FORMS-2 output file contents and relate directly to optional features that can be used. Read these chapters if you require the features described.

It is recommended that, in any case, you run the demonstration programs in the COBOL distribution before using FORMS-2.

Related Publications

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

CIS COBOL Operating Guide, UNIX O.S. Implementation

LEVEL II COBOL Operating Guide, UNIX O.S. Implementation

CIS COBOL Language Reference Manual

LEVEL II COBOL Language Reference Manual

Notation

Throughout this manual the following notational convention is observed:

> The "<<" symbol is used to indicate that the RETURN key is to be pressed.

CHAPTER 1. INTRODUCTION

A. General Description

1. 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:

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

2. 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:

1. 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.

2. 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.

3. Screen Image files of exact copies of the user defined forms. The files are generated as filename.Snn (Snn is 00 to 99).

4. 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 user's form. The file is generated as filename.GEN.

3. Phases

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

a. Initialization Phase

The Initialization Phase is passed through only once 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.

b. 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 the 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.

B. 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 appropriate 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.

1. 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 functions of the cursor control keys are summarized in the following table.

Table 1-1. Cursor Control Keys

KEY:	FUNCTION:
-->	Position cursor right one data character
<--	Position cursor left one data character
v	(Down) Position cursor at start of succeeding data field
^	(Up) Position cursor at start of preceding data field
HOME	Move cursor to start of first data field
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.

C. Getting Started

Table 1-2. FORMS-2 Files Provided

FILE:	CONTENTS:	EXPLAINED IN:
forms2.INT forms2.IS1 forms2.IS2	FORMS-2 Program (COBOL) and its overlays.	This manual
FORMS2.I01 FORMS2.I02	Initialization 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
FORMS2.CH1 FORMS2.CH2	Check-Out Program Components	Chapter 6
FORMS2.GN1 FORMS2.GN2	Index Program Components	Chapter 8

If the FORMS-2 program has not already been installed on your system, refer to Appendix A for instructions on how this is done. You can check to see if this is the case by listing the contents of /usr/lib/cobol where the preceding files must exist.

In addition to these files, the forms2 program needs to have been installed in /usr/bin. This program is in executable (a.out) format to allow FORMS-2 to be utilized from any directory in the UNIX file system. The source for this program (forms2.c) is provided with the distribution to allow for any changes that might be desired.

1. FORMS-2 Validation

1. Enter:

```
forms2 <<
```

2. The program will run, and will display the first screen, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3              INITIALISATION PHASE              SCREEN I01

FORMS2 PARAMETERS:

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

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

SPECIAL-NAMES clause:

     CURRENCY SIGN              [$]        (ANSI currency signs only)

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


Press RETURN when complete
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

(The spellings are British) A six-character base for file-names and data-names is requested followed by four other questions. The question on CRT size needs to be answered if your CRT is a non-standard size. Otherwise all the default replies are accepted and you need only key a name, e.g. "demo", followed by the RETURN key.

3. FORMS-2 then displays Screen I02 to request the output file option type and directory prefix, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

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

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

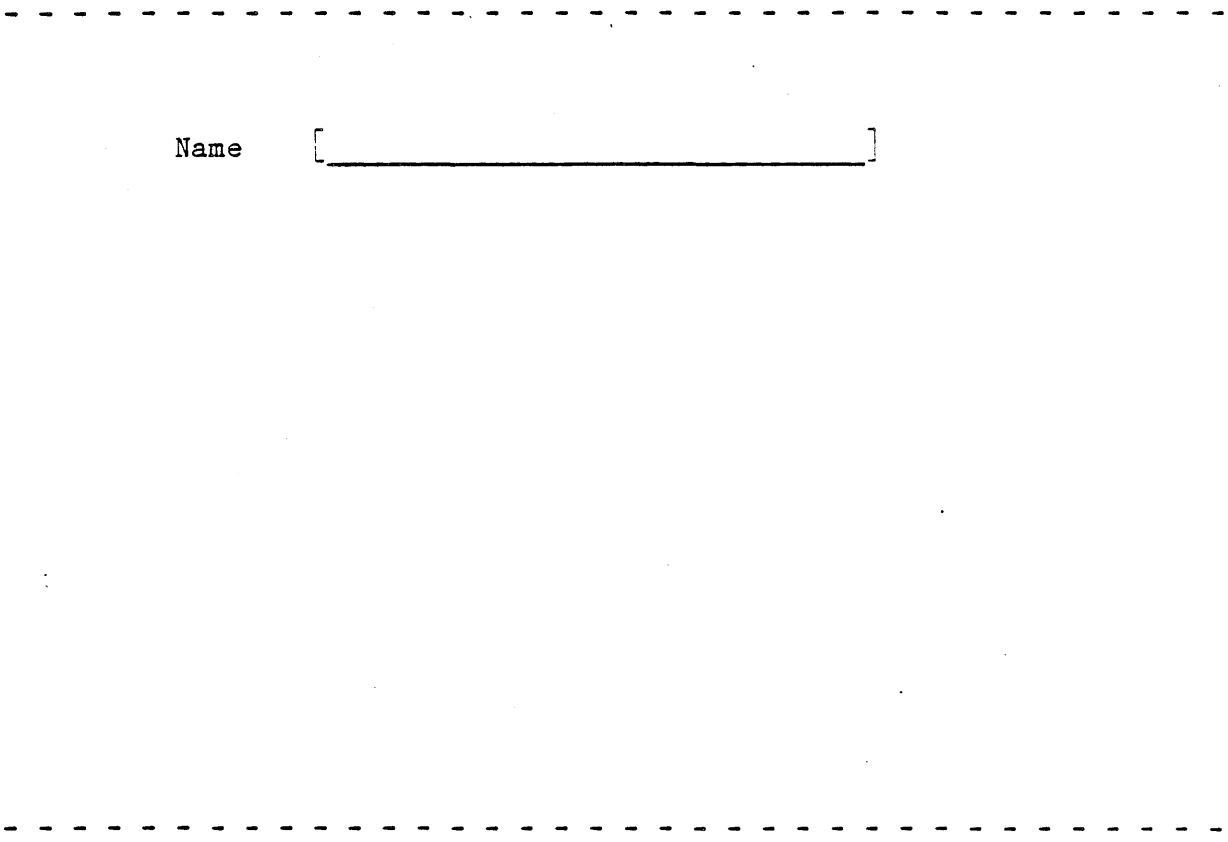
Key F for the output file selection then RETURN. If you wish the files output by FORMS-2 to be directed to a different directory than the current one, enter the pathname in the field marked DEVICE/DIRECTORY PREFIX. You must provide the final slash in the prefix.

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 = Fixed text on clear screen)
                               (B = Fixed text on last screen)
                               (C = Variable data redefines last screen)
                               (D = Variable data without redefinition)
                               (! = Complete this FORMS run)

    Fixed Text allows:         All characters

    Variable Data allows:      X or Y to define alphanumeric fields
                               9 or 8 to define numeric fields
                               edit chars to define numeric edit fields


Press RETURN when complete


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

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 are now in Command Mode. At this point you would usually proceed directly to step 16, however, enter ? and then press the RETURN key.

7. FORMS-2 displays screen HO1, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                 HELP SCREEN                   SCREEN HO1

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
                     O1 = Turn off automatic WORK screen preparation
                     Cn = Insert n spaces at cursor position
                     Dn = Delete n chars at cursor position
                     In = Insert n blank lines before cursor line
                     Kn = Delete n lines including cursor line
                     An = Overwrite n lines with data of cursor line
                     Un = Move cursor up n lines
                     Vn = Move cursor down n lines

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

Press RETURN when complete

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter ? then RETURN.

9. FORMS-2 displays screen H03, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

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

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter ? then RETURN.

10. FORMS-2 displays screen H04, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                  HELP SCREEN                    SCREEN H04

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


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

Press RETURN when complete

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Simply press RETURN.

11. FORMS-2 redisplays the fixed text that you keyed in at step 5. Press RETURN.

12. FORMS-2 puts "__" in the top left of the screen. Enter F then press RETURN.

13. FORMS-2 displays screen WO2, thus:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3              WORK PHASE                          SCREEN WO2

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

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter A then RETURN.

14. Again FORMS-2 redisplays 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 WO1 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    [XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX]














- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

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 everything is in place and ready to be used.

CHAPTER 2. INITIALIZATION PHASE

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

A. Initialization Screen I01

This screen is displayed and FORMS-2 is immediately.

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:

1. Data-name and the File-name Base
2. Lines per CRT Screen
3. Currency Sign
4. Decimal-Point Representation

1. Data-Name and File-Name Base

The file-name 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 data-names 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 CIS or L/II 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 3)

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

2. 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 or 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 or L/II COBOL Language Reference Manual for a list of valid characters.

3. Decimal-Point

This option allows the roles of the period or full-stop sign (.) and the comma sign (,) to be exchanged. If "," 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.

B. Initialization 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:

1. Types of files to be created
2. Directory into 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 Initialization Phase.

1. File Combinations

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

a. DDS - The user may generate CIS or L/II 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 or L/II COBOL program should consult the CIS or L/II COBOL Language Reference Manual (especially the sections on ACCEPT/DISPLAY, FILLER, REDEFINES).

b. CHK - 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.

c. 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.)

d. 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 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 was inadvertently pressed before completing screen I01. This command is not mentioned on the screen.

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).

A. Screen W01

1. 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 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 not redefine the fixed text data area in the applications program. This may sometimes be of assistance to the programmer even though it results in larger application programs.

2. 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 initialization 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).

B. <u>Work Screen</u>

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.

CHAPTER 3. WORK PHASE

1. 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 or D are chosen, the Foreground is overlayed 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 (see below) 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.

2. 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.

a. Fixed Text

In the design of the Fixed Text of a form (i.e. the fixed fields analogous 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.

b. Variable Data

In the design of the Variable Data fields of a form (i.e. the fields analogous 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:

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 alphanumerics 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 and 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 CIS or L/II COBOL program either directly or as specified during the Initialization Phase of the FORMS-2 run (see Chapter 2).

NOTE:

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

Variable data fields are checked for validity but only when a DDS file is being created. The user should consult the CIS or L/II COBOL Language Reference Manual (ACCEPT/DISPLAY Verbs) for detail of how screen handling works.

3. 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.

a. Commands

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

1. General Work Screen 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 summarized within HELP screens 1 and 2 (H01 and H02). Group 3 is summarized on HELP screens 3 and 4 (H03 and H04).

b. General Work Screen Commands

A description of each general command is given below.

? - Display Help Screens

Entry of the question mark then RETURN causes the first HELP screen to be displayed, which includes a summary of the General Commands. This screen remains displayed until the next input command.

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. If repeated, each HELP screen will be displayed 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.

_ - Return to Edit Mode (Underline)

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 text 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, though not mentioned there, and again its function is to return a step within the phase (in this case back to screen I01).

! - Terminate FORMS2 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, or the "!" command is given to terminate the session and abandon the work in progress.

c. 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 on to 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 WO2 above. The specified option is executed and control returned to Edit Mode without display of the WO2 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 O (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.

Cn - Insert Spaces

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

Dn - Delete Characters

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

In - Insert Blank Lines

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

Kn - Delete (Kill) Lines

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

An - Repeat Current Line

Repeat 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 right and left arrow keys.

Vertical cursor positioning is achieved by means of the up and down arrow keys after the cursor is moved to the first position of the current line.

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.

d. <u>Programming Commands</u>

G - The Dataname Structuring Command

The default recordname format generated by FORMS-2 for inclusion in the user's CIS or L/II 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 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.

Alternatively:

The G (or GO) 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 CIS or L/II 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 CIS and 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. 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 JO 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 designed 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 FILLER

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

S (or SO) - 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 option is not available if option C or B is specified at screen IO2.

P - Cursor Control 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 used within datanames, this command provides an easy alternative method of ascertaining the coordinates of any field.

Wn - Window Commands

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 <u>outside</u> the window. For example, if a 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 which begins below the top of the screen, 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 and L/II COBOL ACCEPT/DISPLAY AT coordinates facility.

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

W (or WO) positions the cursor at current window start. This is the equivalent of the HOME 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 to 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:

Delimiters do not corrupt Background/Foreground contents.

The output Screen Image will include the full Foreground part of the Work Screen without delimiters, irrespective of whether a window has been defined. This could be used to include annotation on the Screen Image which does not affect DDS generation.

One use of the window facility would be to display a form 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.

C. 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:

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 signaled 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 repeat the SPACE command after making any corrections.

FORMS-2 will allow editing characters but will not verify that the combinations of these are valid; CIS or L/II COBOL editing rules must therefore be obeyed to ensure error free code. Note that these fields should be separated by spaces.

Only Foreground data is output to the Screen Image file.

CHAPTER 4. DATA DESCRIPTIONS

This chapter describes the CIS and L/II COBOL Data Descriptions that FORMS-2 generates in the .DDS file. Some knowledge of CIS or L/II COBOL should be considered a prerequisite for reading it.

The CIS and L/II COBOL extensions to the ACCEPT and DISPLAY verbs allow comprehensive screen handling to be included in a user application. (See the appropriate 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 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 Initialization Phase, FORMS-2 automatically converts this input to the necessary CIS or L/II 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.

CHAPTER 4. DATA DESCRIPTIONS

A. Record-Name and Data-Name Generation

At Initialization 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.

1. Record Naming

The default record-name format generated by FORMS-2 for inclusion in the user's CIS or L/II 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 record-name generated will be:

```
bbbbbb-rr-ll
```

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

2. 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.

B. 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.

CHAPTER 4. DATA DESCRIPTIONS

1. 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 and 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.

2. 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 and L/II COBOL picture characters 9 and X but note the additional use of 8 and Y as alternatives to 9 and X.

Also note the exclusion of S, V and P as described under "Variable Data" in Chapter 3.

C. 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.

D. 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 and L/II COBOL. This is described in the CIS and L/II COBOL Language Reference Manuals.

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:

1. Validate the DDS file
2. Demonstrate the operation of the proposed application
3. Check the use of his forms for data entry
4. Check the use of his forms for data amendment

The Check-Out source code which is in CIS and L/II 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.

NOTE:

> If numeric edited fields are included in the Variable Data fields of a form, error free code is not guaranteed with Standard CIS and L/II COBOL. Compilation is necessary to achieve full validation of numeric edited fields. If numeric edited fields do cause compilation errors, use can be made of the FORMS-2 Screen Image facility to recall the offending screen and alter the Variable Text numeric edited fields as necessary.

CHAPTER 5. THE CHECK-OUT PROGRAM

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 F command and the D 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 users to check their forms for both initial data entry and data amendment.

A. Check-Out Program Generation

The Check-Out program is then compiled in the usual way by entering the command:

```
cobol basename.CHK<<
```

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

B. Check-Out Program Running

1. Loading

The program can be executed immediately after compilation by the command:

```
cbrun basename<<
```

2. 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.

a. 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.

1. Fixed text on a clear screen

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

2. 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.

b. 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 and L/II COBOL Language Reference Manuals under The ACCEPT Statement).

On other than the first pass through the sequence of screens the previously 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 normalization performed by the CIS and L/II COBOL Run-Time Systems. Note that the normalized fields are not automatically echoed to the CRT.

3. 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 and L/II COBOL data description statements described in Chapter 4. These files contain exact text images of the user described forms. These form images can be:

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

A. Screen Image File Generation

The facility is invoked by selection of an appropriate option at Screen I02 during the Initialization 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 in the
Initialization 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 field 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).

B. Forms Maintenance

CIS or L/II 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.

C. 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.

D. 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 CIS or L/II 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

In this example, you will use FORMS-2 to build the data entry form:

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

where NAME and ADDRESS are alphanumeric fields and TEL is a numeric field. You can then use the check-out program generated to experiment with data entry. Then you will change the ADDRESS field name to ABODE. Afterwards, the screen will appear as pictured below:

```
NAME     [                    ]
ABODE    [                    ]
         [                    ]
         [                    ]
TEL      [    -    ]
```

To accomplish this, you must follow these steps:

1. Invoke FORMS-2 with the command:

```
forms2<<
```

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 device/directory prefix. 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        ____-____
```

Note that the underline characters are treated as spaces when the form is displayed by an applications program, and are provided only for the convenience of the FORMS-2 user for indicating the fields to be redefined as variable fields.
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      [XXXXXXXXXXXXXXXXXXXX]
ADDRESS   [XXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXX]
TEL       [999-9999]
```

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 variable 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 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:

```
cobol -c demo1.CHK<<
```

The -c (COPYLIST) flag provides the text of the copied files in the program listing.

14. When the compilation finishes, the two screens can be checked out by typing:

```
cbrun demo1<<
```

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 accepted into the fields depends on whether X or 9 was specified. You may also test the effect of left fill character "."

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 FORMS-2 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 learned 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 20 onward 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

    ```
    forms2<<
    ```

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 directory; 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:

    ```
    demo1.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 WO1 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 demo1.SOO, then press RETURN.

27. FORMS-2 displays Screen WO2 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 overtype it with ABODE. Remember to overtype 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 you previous Screen Image in the file:

```
demo1.SOO already exists
overwrite? [N] (Y=Yes)
```

Enter 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 = demo1.SOO
```

Press RETURN to continue.

32. FORMS-2 displays screen WO1 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 follows to assist in defining the variable fields.

```
NAME      [______________________]
ABODE     [______________________]
          [______________________]
          [______________________]

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 presss 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.SO1 already exists
overwrite? [N] (Y=Yes)
```

As before, enter Y and press RETURN.

40. A message is displayed as follows:

```
File created = demo1.SO1
```

41. FORMS-2 displays screen WO1 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 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.

1. Select records by key field for display (Enquiry by key field)
2. Select records sequentially for display (Sequential Entry)
3. Amend existing records
4. Delete existing records
5. 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 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 8. INDEX PROGRAM

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

A. 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. Initialization

   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 3 are available <u>except</u>:

      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 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 terminated automatically without any need for the termination (!) command. In fact the ! command is only used to abandon the run when generating the Index Program.

B. Files Generated

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

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

C. Index Program Compilation

The Index program is then compiled in the usual way by entering the command:

```
cobol basename.GEN<<
```

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

After compilation, the user can run the generated program.

D. Index Program Running

1. Loading

The program can be executed immediately after compilation by the command:

```
cbrun basename<<
```

2. Data Processing Facilities

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

The form remains on the screen throughout the 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.

a. Enquiry by Key Field

Amend key fields only, 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.

b. 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.

c. Amend Displayed Record

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

d. Delete Displayed Record

Press the HOME key and press RETURN. The message RECORD DELETED is displayed.

e. 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, press HOME.

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 RECORD ALREADY EXISTS WITH THIS KEY is displayed. The facilities available on the subsequent input are as follows:

1. Force replacement of existing record:
   Press HOME 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.

f. 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.

3. 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:

a. Key and Data Fields Unchanged

The function performed depends upon cursor position as follows:

1. If an end-of-file condition has just been reported, a request to terminate the run is assumed irrespective of cursor position.

2. Otherwise if the cursor has been moved to the HOME position and a record is currently displayed, a delete request is assumed.

3. If neither of these conditions exists, a request to display the next record relative to the 'current' position in the file is assumed.

b. Key Changed and Data Unchanged

The function performed depends on cursor position as follows:

1. If cursor is moved to either the HOME position or the last data character position, and attempt to insert a record is assumed, and processing is as described under "Key and Data Changed".

2. Otherwise an enquiry with respect to this key is assumed, and either the record is displayed or its absence is reported.

c. Key Unchanged and Data Changed

This is a request to update the file, and either a new record is written or the existing record amended as appropriate.

d. 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 the HOME position (or the last data character position) insertion of the new record is forced, and the existing record is overwritten.
   >
   > If the cursor is at any other position a request to abandon the insertion attempt and display the existing record is assumed.

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

In this example you will 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      [   -    ]
```

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

If you then use the form for data entry and key in John Smith, 500 Chestnut St., Santa Cruz CA 95060, 425-7222, the form will appear as:

```
NAME     [John Smith            ]
ADDRESS  [500 Chestnut St.      ]
         [Santa Cruz            ]
         [CA  95060             ]
TEL      [425-7222]
```

To accomplish this, carry out the following steps:

1. The operator keys:

   ```
   forms2<<
   ```

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 directory. Key G then RETURN to accept 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 hand 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 displays on the CRT the 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 in 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      [XXXXXXXXXXXXXXXXXXXX]
ADDRESS   [XXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXX]
          [XXXXXXXXXXXXXXXXXXXX]
TEL       [999-9999]
```

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, and displays to the CRT the DDS source code as generated, followed by a redisplay of the variable 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:

```
cobol -c demo.GEN<<   (if you are using CIS COBOL) or
```

The -c (COPYLIST) option is used to instruct the compiler to include the text of any "COPY" files in the program listing.

14. When the compilation finishes, the generated Index program demo2 can be run by typing:

```
cbrun demo2<<   (CIS COBOL) or
```

15. The generated Index 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 the 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.

19. 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.

20. Change the town field and press RETURN. The message RECORD AMENDED is displayed.

21. Press RETURN and the third record is displayed. You could progress through a whole file in this way.

22. 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.

23. 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.

24. Press RETURN with end-of-file showing and termination occurs.

    You have now seen the record handling method demonstrated.

25. At this point the following files will exist in your directory:

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	-	Data file.

NOTE:

The two files demo2.idx and demo2 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.

APPENDIX A

FORMS-2 INSTALLATION

1. Contents

The following directories and files should exist on the distribution issue medium.

1. forms2/install - Installation shell script.
2. forms2/lib - Directory of files for the COBOL library.
3. forms2/bin - Directory of executable programs.
4. forms2/cmd - Directory of C source for command.

2. Instructions

Extract the contents of the issue tape or disk into a working directory using the UNIX "cpio" or "tar" utilities, depending on the format of your distribution (check your tape label). Your working directory should now contain the directories and files listed above.

The file "install" is a shell script which may be executed to perform the installation of the forms2 utility. Super-user permission will probably be required to allow the script to write in /usr/lib and /usr/bin.

The forms2.c program, which is the source for forms2, is provided in the cmd directory for documentation and to allow for any modifications which might be desired.

APPENDIX B

SCREENS

1. FORMS2.IO1

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3              INITIALISATION PHASE                   SCREEN IO1

FORMS2 PARAMETERS:

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

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

SPECIAL-NAMES clause:

     CURRENCY SIGN              [$]        (ANSI currency signs only)

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


Press RETURN when complete
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

2. FORMS2.I02

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

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

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

3. FORMS2.WO1

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                 WORK PHASE                      SCREEN WO1

WORK SCREEN SELECTION:

    SCREEN TYPE          [ ]     (A = Fixed text on clear screen)
                                 (B = Fixed text on last screen)
                                 (C = Variable data redefines last screen)
                                 (D = Variable data without redefinition)
                                 (! = Complete this FORMS run)

    Fixed Text allows:           All characters

    Variable Data allows:        X or Y to define alphanumeric fields
                                 9 or 8 to define numeric fields
                                 edit chars to define numeric edit fields


Press RETURN when complete


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

4. FORMS2.WO2

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

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


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

5. FORMS2.HO1

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                     HELP SCREEN                   SCREEN HO1

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


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

6. FORMS2.HO2

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                  HELP SCREEN                     SCREEN HO2

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

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

Press RETURN when complete

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

7. FORMS2.HO3

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                  HELP SCREEN                     SCREEN HO3

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


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

8. FORMS2.HO4

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FORMS2 V1.3                  HELP SCREEN                         SCREEN H04

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

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

Press RETURN when complete

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```