

**Chapter
2**

**SCREEN
DESIGN**

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C Characteristics of the 3270.
- C 3270 native-mode programming.
- C Creating a mapset.
- C Data formats.
- C Send maponly.
- C Screen design steps.
- C Types of online applications.
- C Types of operators.
- C Response-time requirements.
- C Terminal features.
- C Screen design guidelines.
- C Screen checklist.
- C Diagraming screen sequences.

Attributes:

Protected/Unprotected	Data may only be keyed into unprotected fields; the keyboard locks if data entry attempted in protected field.
Intensity	Bright Normal Dark
Autoskip	Provides automatic tab to first position of next unprotected field. Used to mark end of data entry field. Autoskip implies protected.
MDT	Can be turned on in attribute byte on output. For data entry field, turned on when data keyed into field by terminal operator. Data sent to CPU by 3270 only from fields with MDT on.

Cursor Control:

Insert cursor character causes 3270 to place cursor in current buffer position.

2 3270 Native-Mode Programming

Buffer Addressing

SBA	Indicates next two bytes contain buffer address; does not occupy screen position.
Buffer Address	Two-byte line number and column number in 3270 code; does not occupy screen position.

Field Definition

SF	Indicates next byte is an attribute byte; does not take up screen position.
Attribute byte	Defines characteristics of field up to next attribute byte; occupies screen position as protected blank.

```

+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
* +))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
* *
* *          [ ]ORDER ENTRY          * *
* *
* *          [ ]CUST: [ ]      [ ]      * *
* *
* *          [ ]ITEM: [ ]      [ ]      [ ]QUANTITY:[ ]      [ ]      * *
* *
* *          [ ]SHIP TO NAME:[ ]      [ ]      * *
* *
* *          [ ]SHIP TO ADDRESS:[ ]      [ ]      * *
* *
* *          [ ]ZIP:[ ]      [ ]      [ ]COUNTRY:[ ]      [ ]      * *
* *
* *          [ ]SHIPPING CODE:[ ]      [ ]      * *
* *
* *
* *          .))))))))))))))))))))))))))))))))))))))))))))))))))))- *
* *          .))))))))))))))))))))))))))))))))))))))))))))))))))))-

```

3 BMS Overview

Components of Basic Mapping Support

One or more mapsets, each mapset containing one or more maps.

BMS commands

Mapsets

Generated in batch mode

Contain descriptions of screen formats, field lengths and attributes, etc.

Provide device independence and format independence at execution time.

BMS Commands

Specify mapset, and map in the mapset to be used by BMS for information on field sizes and positions, attributes, and other control information.

BMS converts device-dependent input into standard-format device-independent 'record'

BMS converts standard-format, device-independent output into device-independent data stream.

BMS commands, with the ACCUM option, may be used to combine multiple maps into one screen image, or page.

4 Creating a Mapset

Mapsets are created using Assembler Language macros.

DFHMSD TYPE=MAP/DSECT	one per mapset.
DFHMDI	one per map within a mapset.
DFHMDF	one per field within a map.
DFHMSD TYPE=FINAL	one per mapset.

A mapset is assembled twice:

- 1) With TYPE=MAP, a physical map is generated, which is catalogued to the CICS core image/load library, and load by BMS at execution time.
- 2) With TYPE=DSECT, a symbolic map is generated and catalogued to the source statement library. This symbolic map is copied into the application program at execution time and provides symbolic field names for the device-independent input and output data formats received from and passed to BMS.

```

+))))))))) ,
*DFHMSD *
+2))))))))) , *TYPE=DSECT*
*DFHMD /)2))))))>* ASSEMBLER *
+2))))))))) , *
*DFHMD /)- *
+2))))))))) , *
*DFHMDI /)- . ))))O)))- . ))))O)))-
+2))))))))) , *
*DFHMSD /)- *
* *
. ))))O)))- *
* TYPE=MAP *
* *
W *
+))))))))) ,
* *
* ASSEMBLER *
* *
* *
. ))))O)))- *
* *
* *
W *
+))))))))) ,
* *
* PHYSICAL *
* MAP *
* *
* *
. ))))O)))- *
* *
* *
W *
?444444444444@ *
* *
* SOURCE *
* STATEMENT *
* LIBRARY *
* *
. ))))O)))- *
* *
* *
W *
?444444444444@ *
* *
* CORE IMAGE *
* OR *
* LOAD LIBRARY *
* *
. ))))O)))- *

```

4.1 Data Formats

Data may be passed to and received from BMS in three different formats:

Field Data Format

Data passed to and received from BMS as separate fields.

Each field given symbolic name which application program uses to refer to the field.

Each field consists of two-byte length area, one byte attribute area, and data area.

Map describes field position, data length, attributes, etc.

Block Data Format

Data consists of one or more line segments.

Fields are positioned within line segments and given symbolic names.

Each field consists of a one-byte attribute and a data area.

Fields separated by one or more blanks.

Map describes number and lengths of line segments, field positions, data length, etc.

Text Data Format

Output data passed to BMS as data stream divided into lines not longer than width of screen.

Lines never end within a word.

New Line characters may be inserted in data stream by application program.

Maps not used with text data format.

Sys-Ed MAP LAYOUT

Page: _____

System ID: _____ MAP: _____

Analyst: _____ Date: _____ Ver: _____



5 SEND MAPONLY

The first phase clears what was on the screen, then writes the prompts to the screen. The second routine read the data (no prompts) from the screen.

In addition, if our program is editing the data which comes in - as we do with a data entry screen - we must have a routine which "writes" error messages to the screen, and a fourth routine which tells the user that the entry is OK and invites the user to continue working.

In CICS, writing prompts to the screen is done with a SEND MAP MAPONLY command. Writing error messages is done with a SEND MAP DATAONLY command. Reading data from the screen is done with a RECEIVE MAP command.

We can show the relationship of these routines to the screen in two different ways: Using flow diagrams; using a procedural text format.

6 Introduction

Screen design has a significant effect on the success or failure of an online system; it is in this phase of the online design that the interface between terminal user and the computer system is defined.

The area of screen design is a difficult task because it is application - and user-oriented and involves many human factors which must be taken into consideration. Good screen design is not a precise science: is an art that must be acquired by the online designer.

The key to good screen design and a successful on line system is to service the online user. The system should be easy to use and provide the user with the data needed in the required format.

The objectives of the screen designer should be to achieve an effective online system by:

- C Designing to the user's requirements.
- C Maximizing operator productivity.
- C Eliminating errors on input.
- C Making the system easy to navigate.
- C Minimizing transmission line cost.
- C Minimizing CICS/VS resource utilization.



7 Screen Design Steps

The designer should follow a series of steps in putting an application online.

The designer should:

- C Clearly identify the objectives of the system.
- C Identify the users of the system and the location of the terminals.
- C Determine the application response time.
- C Clearly identify the nature of dialogue for the application.
- C Outline error-correction procedures.
- C Design the required screen formats for the application.
- C Determine security requirements.
- C Outline standards for use of 'PF' and 'PA' keys, codes, mnemonics, and special commands.
- C Establish entry for the application.
- C Develop a test plan.

8 Types of Online Applications

Inquiry	An inquiry application usually involves the interrogation of one or more data bases for information requested by an online user.
Data Base Update	A data base update application usually involves a real-time update of a data base record with information keyed by an online user.
Data Collection	This application is similar to a batch key punching operation. Data collection applications are processed in a batch system.

9 Elements to be Considered in Determining Screen Format Type

Input/Output Volumes

This area cannot be generalized because of application dependencies; therefore, it must be examined in conjunction with other considerations.

The following guidelines should be kept in mind when designing screen formats.

- C Provide an easy method of navigation for screen involving large volumes of text data.
- C Where possible, avoid the use of multiple low-volume screens.
- C Where possible, avoid including a large volume of small data fields on a single screen.

10 Source of Input

This area can be broken into four categories: verbal, formatted documents, unformatted documents, and machine readable documents.

- C Verbal input is most frequently associated with inquiry applications and includes input via telephone and/or in person.
- C Formatted documents are forms designed for specific applications and are normally associated with data base update and data collection.
- C Unformatted documents are input source documents containing data not written in a specified format.
- C Machine readable documents are simply cards (credit cards, ID cards) encoded with a magnetic strip which can be read by some terminals.

11 Types of Operators

Terminal operators fall into two categories:

- C Application Operators
- C Casual Operators

Application operators are dedicated terminal operators who are trained in an application area and can be compared to key punch operators. They are able to rapidly enter data into a formatted screen from source input documents associated with an application area.

Casual operators are not dedicated terminal operators; they use the terminal simply to assist them in their jobs.

12 Response-Time Requirements

When determining what response time is realistic, the designer must take into consideration the following factors:

- C Network configuration (local and/or remote).
- C Number of terminals and lines in the network.
- C Input volume.
- C Transmission rate of terminals.
- C Priority of the application.
- C Number of data base accesses.

13 Terminal Features

The terminal features which should be incorporated into screen design are listed below:

- C Cursor movement keys.
- C Field attributes such as:
 - Protected and unprotected
 - Autoskip
 - Selector pen detectable
 - Field display intensities (bright, dark, normal)
- C Block mode vs interactive dialogue mode.
- C Full screen addressing.
- C Character insert and delete keys.
- C Program function keys (PF1 - PF24)

14 Type of Screen Formats

The type of screen format which can be employed for an application are:

- C Fixed Format
- C Adjacent Format
- C Check or Change
- C Free Format with Delimiters
- C Free Format with Keywords
- C Selector Pen

14.1 Example of a Blank Fixed Screen Format

```
+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))-,  
*  
*          DATA BASE MAINTENANCE          *  
*  
*   DATE:                                     *  
*  
*   NO:           NAME:                       *  
*  
*   STREET ADDRESS:                          CITY: *  
*  
*   STATE:           ZIP CODE:                *  
*  
*   *MSG/INST:                                 *  
*  
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))-
```

14.2 Example of a Filled in Fixed Screen Format

```
+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))-,  
*  
*          DATA BASE MAINTENANCE          *  
*  
*   DATE:   05/18/86                        *  
*  
*   NO:     010      NAME: DAVID SHAPIRO     *  
*  
*   STREET ADDRESS: 35 W 35TH STREET  CITY: NEW YORK *  
*  
*   STATE: NY          ZIP CODE: 10001      *  
*  
*   *MSG/INST: *  
*  
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))-
```

14.3 Example of a Blank Adjacent Screen Format

```
+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))-,  
*  
*          DATA BASE MAINTENANCE          *  
*  
*   DATE: *  
*   NO: *  
*   NAME: *  
*   STREET ADDRESS: *  
*   CITY: *  
*   STATE: *  
*   ZIP CODE: *  
*  
*   *MSG/INST: *  
*  
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))-
```

14.4 Example of a Filled in Adjacent Screen Format

```
+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))-,  
*  
*          DATA BASE MAINTENANCE          *  
*  
*   DATE: 05/18/86                          *  
*   NO: 010                                  *  
*   NAME: DAVE SHAPIRO                       *  
*   STREET ADDRESS: 35 W 35TH STREET         *  
*   CITY: NEW YORK                           *  
*   STATE: NY                                *  
*   ZIP CODE: 10001                          *  
*  
*   *MSG/INST:                               *  
*  
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))-
```

14.5 Another Example of the Adjacent Format

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,  
*  
*          DATA BASE MAINTENANCE          *  
*  
*    DATE: 05/18//86                        *  
*  
*    NO: 010                                *  
*  
*    NAME: DAVE SHAPIRO                     *  
*  
*    STREET ADDRESS: 35 W 35TH STREET       *  
*  
*    CITY: NEW YORK                         *  
*  
*    STATE: NY                              *  
*  
*    ZIP CODE: 10001                        *  
*  
*    *MSG/INST:                             *  
*  
.))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```

14.6 Example of a Check or Change Screen Format

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*           PAYROLL MAINTENANCE           *
*
*   P/E DATE:                PROCESS DATE: *
*
*   EMPNO:                   DEPNO:       *
*
*   EMPNAME:
*
*   -----HOURS WORKED-----   -----PAY ADJ----- *
*   REG-HRS:      OV-HRS:         HRS:      DOL:         *
*
*   *MSG/INST:
*
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```

14.7 Example of a Filled in Check or Change Screen Format

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*                               PAYROLL MAINTENANCE                               *
*
*   P/E DATE: 05/07/86           PROCESS DATE: 05/18/86                         *
*
*   EMPNO: 0105      DEPNO: 01                                               *
*
*   EMPNAME: DAVE SHAPIRO                                                    *
*
*   -----HOURS WORKED-----      -----PAY ADJ-----                    *
*   REG-HRS: 35      OV-HRS: 5           HRS:      DOL: 50.00                 *
*
*   *MSG/INST:                                                                *
*
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```


14.9 Example of a Filled in Free Format Screen with Delimiters

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*                                ORDER ENTRY                                *
*
*   DATE:                                                                    *
*
*   CUSTOMER NO:                                                                *
*
*   ORD DATE;  ORD NO;  ORD QTY;  ORD DESC                                     *
*
*   050286;0010;15;NUTS                                                         *
*   050386;0010;10;BLUE NUTS                                                    *
*   050386;0100;5;BOLTS                                                         *
*   042686;0010;1000;TAPES 6400BPI                                             *
*
*
*
*
*   *MSG/INST:                                                                    *
*
* .)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```


14.11 Example of a Free Format Screen with Keywords (Filled In)

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*           ACCOUNTS PAYABLE INPUT
*
*   DATE: 05/18/86
*
*   VENDOR NO: 010
*
*   VENDOR NAME:  DAVE SHAPIRO
*
* -----
* (INV DATE:DATE=) (INV NO:INVNO=) (INV AMT:AMT=)
* -----
*   DATE=050186INVNO=86010025AMT=$1000.50
*   DATE=030486INVNO=86030101AMT=500.00
*
*   ---- O R ----
*   INVNO=86010025DATE=050186AMT=$1000.50
*
*   ---- O R ----
*   AMT=500.00DATE=030486INVNO=86030101
*
* *MSG/INST:
*
* .)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```


14.13 Example Menu to Select Customer

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*          ACCOUNTS RECEIVABLE          *
*          CUSTOMER SELECTION MENU      *
*
* ----- ENTER CUSTOMER NUMBER OR LAST NAME ----- *
*          NO:          LAST NAME:  SILVER          *
*
* ----- *
*  S  *   NO   *           NAME           *
* ----- *
*  ?  *   102  *   SILVER NUT             *
*  ?  *   876  *   SILVER BOLT            *
*  ?  *   010  *   SILVERBERG DAVE M      *
*  ?  *   608  *   SILVER DAVID           *
*  ?  *   999  *   SILVERBERG NUT        *
*
* ----- *
* *MSG/INST: *
*
*
*
* .)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```

14.14 Example of a Selected Customer Account

```
+))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
*
*
*           ACCOUNTS RECEIVABLE
*
*   DATE:   05/18/86
*
*   CUSTOMER NO:   010
*
*   CUSTOMER NAME:  DAVE M SHAPIRO
*
*-----*
*   PAYMENT AMOUNT: $1000.50 |   ON-ACCT AMOUNT:
*-----*
*   INV DATE   * P *   INV NO.       *   INV AMT
*-----*
*   12/15/85   * ? *   85120010       *   $650.00
*   01/15/86   * > *   86010125       *   $1000.00
*   02/20/86   * ? *   86020005       *   $375.00
*-----*
* * CALCULATOR:
*   -- PF1 = APPLY, PF2 = CALC AMT SELECTED
*
*           --AMT SELECTED:
*
*   PF3 = ADJ (+ OR - ACCT)  ADJ AMT:
*
*   PF4 = REFRESH SCREEN, PF5 = RETURN
*
*-----*
* .))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))-
```

15 Screen Design Guidelines

The key to good screen design is to service the terminal user. Screen formats must be easy to user and at the time, satisfy the requirements of both the terminal user and the online application.

To aid the designer in the task of screen design, the general screen design guidelines listed below are recommended:

C Always respond to the operator.

If the situation exists where the response will be slow because of processing requirements, the application should be programmed to immediately acknowledge receipt of the input.

C Provide the operator with an online help screens.

C Always give the operator a way out - provide a method to abort a session.

C Provide the operator with a method to:

- Suspend a session.
- Restarted a suspended session.
- In an application where multiple screens of data is entered or displayed, provide the operator with paging functions to scan backward and forward through the data.
- Where possible, input fields should be followed by a short protected field to indicate the field-data type and length.

```
+)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) ,
*
*           DATA BASE MAINTENANCE
*
*   DATE:
*
*   NO:   [N3]   NAME:
*
*   STREET ADDRESS:           [A15]   CITY:           [A15]
*
*   STATE:           [A10]           ZIP CODE:           [N5]
*
*   *MSG/INST:
*
.)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))) -
```

16 Screen Checklist

- C Where possible, maintain certain screen standards across applications:
 - type of dialogue.
 - use of codes and mnemonics.
 - similar screen formats.
- C Provide a procedure to allow the operator to request a refresh of a screen format or display.
- C Where possible, design the screen format to handle a single aspect of an application (do not try to perform an inquiry and a file update at the same time).
- C Make the dialogue clear to the operator. Leave no questions as to the information desired, yet be concise. Avoid symbols, abbreviations, and difficult words.
- C One idea per screen display - screen displays should be kept uncluttered and contain only meaning full information.
- C Operator use of the insert (ins) mode key and delete key (the operator should be instructed in the use of these keys).
- C Display input error fields in high intensity.
- C Place cursor at the first field in error (use symbolic cursor positioning).
- C Provide a method for retrieving previously entered data for correction.
- C Include a protect or autoskip stopper field at the end of each input field - for variable-length fields; use a protect stopper to assure that the field is not exceeded.

- C Where possible, define error messages in a non-display mode and change to high intensity when an error occurs.
- C Make use of the numeric shift and numeric lock feature.
- C Make use of the 'PF' keys. Use standard PF value thru the entire application system.
- C Develop operator training procedures.



17 Diagramming Screen Sequences

In most cases, the designer will find it helpful to make a block diagram showing screens in an application and the relationships between the screens.

