

GC27-6999-3

File No. S360/S370/S3/4300/8100-30

Systems

**Introduction to
Programming the
IBM 3270
Information Display System**

IBM

Fourth Edition (January 1980)

This is a major revision of, and obsoletes, GC27-6999-2. This revision incorporates the changes made by TNL GN32-0755. This revision incorporates changes to the 3270 data stream for the Color, Extended Highlighting, and Programmed Symbols enhancements. These changes are covered in Chapter 1. Other, minor technical changes and corrections have been made throughout the book. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest IBM System/360 or System/370 SRL Newsletter for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Examples appear in this manual that include the names and addresses of individuals. All these names and addresses are fictitious, and any similarity to the names and addresses used by actual individuals is entirely coincidental.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. Address additional comments to IBM Corporation, Department 520, Neighborhood Road, Kingston, New York 12401. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

BEFORE YOU USE THIS BOOK

Who This Book Is For

This book is for those who need to know what is involved in programming the IBM 3270 Information Display System.

This book is for those programmers who plan and code the panels seen on 3270 displays. It should be used in conjunction with the IBM 3270 Information Display System: Component Description, GA27-2749.

For those programmers who write the access method macro instructions or other I/O instructions, this manual is to be used in conjunction with the appropriate access method or IBM Program Product publications.

How This Book Is Organized

This manual is divided into six chapters:

Chapter 1, The 3270 Data Stream and Extensions to It, introduces and explains the 3270 data stream. It shows examples of the 3270 data stream and the 3270 extended data stream. It discusses the enhancements to the data stream, such as Color, Extended Highlighting, and Programmed Symbol sets.

Chapter 2, Screen Design, introduces important 3270 concepts. It shows an example of what a 3270 display message might look like, what coding elements are required to write this message in your program, and how terminal operator input might be handled.

Chapter 3, Screen Management, suggests macro definitions and programming routines that might be written to encode and decode messages to and from the display.

Chapter 4, BTAM Support, suggests including I/O operations (reading, writing, error recovery) in a module separate from message formatting. It contains descriptions and flowcharts to aid in writing error recovery routines for use with BTAM. It discusses sense/status analysis.

Chapter 5, TCAM Support, suggests handling messages by means of two modules for the user's application

program. It describes the TCAM macro instructions that affect the 3270. It suggests how to handle remote printers.

Chapter 6, VTAM Support, summarizes the VTAM information for the 3270. It describes using VTAM with SNA and non-SNA 3270s. It suggests guidelines for making non-SNA and SNA 3270s compatible in the same network.

Other Books You May Need

As a general Introduction to the 3270:

An Introduction to the IBM 3270 Information Display System, GA27-2739

To understand how the terminal operator sees the 3270:

Operator's Guide for IBM 3270 Information Display System, GA27-2742

IBM 3270 Information Display System Problem Determination Guide, GA27-2750

A Guide to Using the Test Request Feature on IBM 3270 Information Display Systems, GA27-2774

Operator's Guide for the IBM 3270 Information Display System Katakana Feature, GA18-1016

IBM 3270 Information Display System: IBM 3279 Color Display Station Operator's Guide, GA33-3087

As a reference on how the 3270 (including the printers) works:

IBM 3270 Information Display System: Color and Programmed Symbols, GA33-3056

IBM 3270 Information Display System: Component Description, GA27-2749

IBM 3270 Information Display System: Katakana Feature Component Description, GA18-1017

Suggested programming tools:

A green booklet: IBM 3270 Information Display System Reference Summary, GX20-1878

Panel layout sheets: IBM 3270 Information Display System Layout Sheet, GX27-2951

If you are using BTAM:

IBM 2260 BTAM and 2260 GAM to IBM 3270 BTAM Conversion Guide, GC27-6975

IBM System/360 Disk Operating System Basic Telecommunications Access Method, GC30-5001

DOS Programming Supplement for the 3270 Information Display System, GC27-6977 (applicable to DOS Release 26 only)

DOS Version 4 BTAM, GC27-6978

IBM System/360 Operating System Basic Telecommunications Access Method, GC30-2C04

BTAM Extended Support (BTAM-ES), SC38-0293

DOS/VS BTAM, GC27-6989

OS/VS BTAM, GC27-6980

If you are using TCAM:

Planning for TCAM with the IBM 3270 Information Display System, GC30-2021

OS TCAM Programmer's Guide and Reference Manual, GC30-2024

OS TCAM User's Guide, GC30-2025

OS/VS TCAM Programmer's Guide, GC30-2034

If you are using VTAM, or ACF/VTAM:

VTAM Concepts and Planning, GC27-6998

VTAM Macro Language Reference, GC27-6995

VTAM Macro Language Guide, GC27-6994

VTAM System Programmer's Guide (DOS/VS, GC27-6957; OS/VS1, GC27-6996)

OS/VS2 System Programming Library: VTAM, GC28-0688

ACF/VTAM Concepts and Planning, GC38-0282

ACF/VTAM System Programmer's Guide (refer to Concepts and Planning for the appropriate form number)

ACF/VTAM Macro Language Reference, SC38-0261

ACF/VTAM Macro Language Guide, SC38-0256

ACF/VTAM General Information Concepts, GC27-0463

ACF/VTAM Programming, SC27-0449

ACF/VTAM Installation (DOS/VSE), SC27-0464

ACF/VTAM Installation (OS/VS), SC27-0468

Advanced Communications Function for VTAM Entry (ACF/VTAME) General Information: Introduction, GC27-0438

CONTENTS

CHAPTER 1. THE 3270 DATA STREAM AND EXTENSIONS TO IT	1
Introduction	1
The 3270 Data Stream	1
How Fields Are Defined	2
Base Color	4
Extensions to the 3270 Data Stream	5
Attributes	5
3270 Field Attributes	5
Extended Field Attributes	6
Character Attributes	6
Start Field Extended (SFE) Order (X'29')	6
Modify Field (MF) (X'2C')	7
Set Attribute (SA) Order (X'28')	8
Attribute Processing	9
Structured Fields	11
Defining a Character	17
Triple Planes	19
Read Partition Query (Outbound) and Query Reply (Inbound) Structured Fields	20
CHAPTER 2. SCREEN DESIGN	27
Field Concept	27
What Attributes May Be Assigned to a Field	27
Example of Field Definition	31
Panel Design	34
An Example of a Sequence of 3270 Panels	34
Planning a Sequence of Panels	39
Defining the Purpose of Each Panel	39
Using the Panel Layout Sheet	39
An Example of Laying Out a Panel	40
Data Stream Coding	42
Orders	42
Adding Orders to the Panel Layout Sheet	43
Coding the Panel	47
Repeat to Address Order	51
Write Control Character (WCC)	52
Panel Design with the Extended 3270 Data Stream Attributes	54
Field 1: "SIGN-ON PROCEDURE"	54
Field 2: "PLEASE ENTER.... INFORMATION"	54
Field 3: "NAME"	54
Field 4: The Area Following "NAME"	54
Field 5: "LOCATION"	55
Analyzing Input Data	56
The Operator's Response	56
Attention Identifier (AID)	56
Input Data	58
SBA Codes	58
Program Attention Keys	59
Program Access (PA) Keys	59
Program Function (PF) Keys	59
Selector Pen and Cursor Select Input and Output	59
Selector Field Format	59
Designator Characters	60
The Relationship of One Data Stream to Another	62
Modifying Existing Panels	62
Write Control Character (WCC)	65
Erase Unprotected to Address	66
Erase All Unprotected Command	68
Repetitive Output	69
Program Tab	70
CHAPTER 3. SCREEN MANAGEMENT	73
Decoding and Generating Data Streams	74
Decoding Read Modified Input Data Stream	74
Nonselector Pen or Non-Cursor Select Data Streams	75
Immediate Selector Pen or Cursor Select Data Stream	79
Mixed Read Modified Input Data Streams	81
Building Output Data Streams	81
Static Data Streams	81
Semi-Dynamic Output Streams	84
Dynamic Output Streams	84
Large Screen Size	85
Copy Function for the 3271, 3272, 3274 and 3276 Copy Function	86
The Print Authorization Matrix for the 3274 and 3276	87
The Matrix Structure	88
Defining the Matrix	89
Local Operation	91
Host-Initiated Copy in Shared Mode for the 3274 and 3276	92
Using Katakana Character Set Codes	92
CHAPTER 4. BTAM SUPPORT	93
Telecommunications Management with BTAM	93
Write Structured Field (WSF)	93
Prepare to Read Operations (PTR)	94
Techniques for Managing Devices	96
The Advantages of a Terminal Control Program	96
The Advantages of a Master Terminal Program	96
Techniques for Keeping Track of Device Status	97
Reliability and Error Recovery	99
Remote Leased Line Event Completion Analysis	99
Transparency	107
Remote Dial Event Completion Analysis	113
Local Event Completion Analysis	126
Sense/Status Analysis	136
CHAPTER 5. TCAM SUPPORT	137
Defining the 3270 Network	137
TERMINAL Macro Instruction	139
INVLIST Macro Instruction	141
IEDRH Macro Instruction	141
Group and DCB Macro Instructions	141

Defining Local Non-SNA 3270 Clusters	141
Sample Resource Definitions	143
Remote LU Type 2	143
Remote LU Type 0	144
Non-NCP Remote BSC	144
Non-SNA Local	144
SNA Local	145
NCP Remote BSC	145
INTRO Macro Instruction	145
Controlling the 3270 Network	145
Message Handling	146
MSGFORM Macro Instruction	146
SCREEN Macro Instruction	147
MSGLIMIT Macro Instruction	147
Handling Remote Printers	147
Handling Other Types of Printers	147
Handling the AID Byte	148
Handling Sense/Status Conditions	148
TCAM Facilities for Error Handling	149
CHAPTER 6. VTAM SUPPORT	151
VTAM with BTAM and TCAM	151
Appearance of the 3270 in VTAM Record Mode	152
Defining the 3270	152
Defining the Local Non-SNA 3270	152
Defining the BSC 3270	154
Logon Requests (BSC and Local Attachment)	156
Defining an SNA 3270	157
PU Options	157
LU Options	157

Managing the 3270	158
Data Transfer Modes	158
Data Transfer Using Record Mode	158
Using SEND/RECEIVE	158
Input Considerations for Non-SNA and PU.T1 3270s (LU.T0)	159
Output Considerations for Non-SNA and PU.T1 3270s (LU.T0)	160
Copy Considerations for Non-SNA and PU.T1 3270s	161
Network Solicitor Considerations	161
Sense and Status Information	161
Data Transfer Using Basic Mode	162
Input Considerations	162
Output Considerations	163
Other Basic Mode Considerations	163
Sense and Status Information	164
Using VTAM with SNA 3270s	164
Logon Requests	164
SNA Logoff Requests	165
Non-SNA Logoff Requests	165
Migration from Non-SNA to SNA 3270	165
Switched Line Support for SNA (Switched PU Support)	165

GLOSSARY 169

INDEX 175

FIGURES

- Figure 1-1. 3270 Mapped Buffer 3
Figure 1-2. Mapping of Attribute and Graphic Characters to the Display Screen 3
Figure 1-3. A Sample 3270 Data Stream 4
Figure 1-4. Using the Start Field Extended Order 7
Figure 1-5. Using the Modify Field Order 7
Figure 1-6. Using the Set Attribute Order 8
Figure 1-7. Model for Character and Extended Field Attributes 9
Figure 1-8. Relationship of Character and Extended Field Attributes 10
Figure 1-9. Character Attribute Override 10
Figure 1-10. Structured Fields 11
Figure 1-11. Format of the WSF for Load PS (2 parts) 13
Figure 1-12. Programmed Symbol Sets 15
Figure 1-13. Character Definition for a 9x16 Display Matrix 18
Figure 1-14. Character Definition for a 10x8 Printer Matrix 18
Figure 1-15. Structured Field Data Stream to Load a Box "A" from a 9x16 Display Matrix 19
Figure 1-16. Structured Field Data Stream to Load a Box "A" from a 10x8 Printer Matrix 19
Figure 1-17. Character Definition of a Multicolor Symbol 20
Figure 2-1. Example of Four Fields and Attribute Bytes 28
Figure 2-2. Results of Keyboard and Field Combinations 29
Figure 2-3. Example of Attribute Specification 31
Figure 2-4. An Example of a Panel 34
Figure 2-5. Another Example of a Panel 35
Figure 2-6. Panel 1 of an Accounts Receivable Application 35
Figure 2-7. Panel 2, Showing the Results of a Search on a Customer Name 36
Figure 2-8. Panel 3, Showing the Customer's Open Invoices 36
Figure 2-9. Panel 4, Showing Use of the Calculator 37
Figure 2-10. Panel 5, Showing Selection of Invoices after Using the Calculator 38
Figure 2-11. Panel 6, Showing New Balance after Posting 38
Figure 2-12. Sign-On Panel Block Diagram 39
Figure 2-13. Block-Diagramming a Sequence of Panels 40
Figure 2-14. Sign-On Panel As Written Out on Layout Sheet 41
Figure 2-15. Panel Layout, Including Attribute and Cursor Positions 41
Figure 2-16. Laying Out Field Attributes 42
Figure 2-17. Text Items on Panel Layout Sheet 43
Figure 2-18. Field Attributes 44
Figure 2-19. Attribute Default Values 45
Figure 2-20. Completed Order and Attribute Information 46
Figure 2-21. Buffer Control Orders and Order Codes 48
Figure 2-22. Sign-On Procedure Panel Orders and Attributes 49
Figure 2-23. Attribute Character Combinations in Hexadecimal 50
Figure 2-24. Assembler Language Statements for Sign-On Panel 51
Figure 2-25. Example of RA Order 52
Figure 2-26. WCC Hexadecimal Codes 53
Figure 2-27. Layout Sheet for Extended Data Stream Attributes 55
Figure 2-28. Sign-On Panel with Operator's Input 56
Figure 2-29. Input Data Sequence 57
Figure 2-30. Attention Identifiers (AIDs) in Hexadecimal Codes 57
Figure 2-31. Definition of Field for Selector Pen Operation 60
Figure 2-32. Sample Panel for Selector Pen or Cursor Select Detection 61
Figure 2-33. Modifying an Existing Panel--Basic Panel 63
Figure 2-34. Existing Panel with Error Message 63
Figure 2-35. Panel Layout Changes for Error Message (Keyed to Text) 64
Figure 2-36. Error Message Panel with Serial Number Field Erased 66
Figure 2-37. Example of EUA Use 67
Figure 2-38. Sign-On Panel with Three Erased Fields 67
Figure 2-39. Erasing Multiple Fields with EUA 67
Figure 2-40. Example of Data Entry Panel 68
Figure 2-41. Data Entry Panel with Entered Data 69
Figure 2-42. Employee Data Panel 70
Figure 2-43. Panel Defined with Program Tab 71
Figure 3-1. Relationship of Screen Management to Telecommunications Management and Application Programs 73
Figure 3-2. Table of Requirements 77
Figure 3-3. Example of Selector Pen Panel 80
Figure 3-4. Sample Mapping Table 80
Figure 4-1. Table of Control Unit and Terminal Information 98
Figure 4-2. Example of a User-Built DECB Extension 99

- Figure 4-3. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Nonswitched Line Read Completion Analysis 100
- Figure 4-4. OS BTAM and OS/VS BTAM, Remote Nonswitched Line Read Completion Analysis 101
- Figure 4-5. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Nonswitched Line Write Completion Analysis 108
- Figure 4-6. OS BTAM and OS/VS BTAM, Remote Nonswitched Line Write Completion Analysis 109
- Figure 4-7. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Dial Read Completion Analysis 114
- Figure 4-8. OS BTAM and OS/VS BTAM, Remote Dial Read Completion Analysis 115
- Figure 4-9. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Dial Write Completion Analysis 120
- Figure 4-10. OS BTAM and OS/VS BTAM, Remote Dial Write Completion Analysis 121
- Figure 4-11. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Local Read Completion Analysis 127
- Figure 4-12. OS BTAM, OS/VS BTAM, and DOS/VSE BTAM-ES, Local Read Completion Analysis 128
- Figure 4-13. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Local Write Completion Analysis 133
- Figure 4-14. OS BTAM and OS/VS BTAM, Local Write Completion Analysis 134
- Figure 5-1. TCAM Release Support of 3270 Attachment Modes 138
- Figure 5-2. TCAM Release Support under OS/VS1 and OS/VS2 138
- Figure 5-3. TCAM Release Support of Application Program Interface Types 138
- Figure 5-4. TCAM Release Support of Cross-Domain Networking for 3270 Attachment Modes 139
- Figure 5-5. TCAM Macros Used to Define the 3270 Device According to Attachment Mode 139
- Figure 5-6. TERMINAL Macro Operands Used to Define 3270 Devices According to Attachment Mode 140
- Figure 5-7. DCB Macro Operands Used to Define 3270 Devices According to Attachment Mode 142
- Figure 5-8. GROUP Macro Operands Used to Define 3270 Devices According to Attachment Mode 143
- Figure 5-9. 3270 Attachment Modes Supported According to INTRO LINETYP Operand 145
- Figure 6-1. VTAM Support Summary 153
- Figure 6-2. Grouping Locally Attached 3270s into Logical Sets 155
- Figure 6-3. Processing a Terminal-Initiated Logon with the Network Solicitor 156
- Figure 6-4. LU Type Differences 166

SUMMARY OF AMENDMENTS (February 28, 1980)
TC GC27-6999-2 BY REVISION GC27-6999-3

This edition provides new programming information for the following 3270 components:

- 3274 Control Unit
- 3276 Control Unit Display Station
- 3278 Display Station

- 3279 Color Display Station
- 3287 Printer
- 3289 Line Printer

This edition covers the enhancements made to the 3270 data stream for Color, Extended Highlighting, and Programmed Symbols. It also corrects several technical inaccuracies.

CHAPTER 1. THE 3270 DATA STREAM AND EXTENSIONS TO IT

INTRODUCTION

This chapter introduces the 3270 data stream and the extensions that have been made to this data stream. These extensions, which include Color, Extended Highlighting (reverse video, blink, and underscore), and Programmed Symbols (user-defined symbol sets), enhance the presentation of data at the 3270 device. Chapters 2 and 3 explain how to lay out and code a panel for display.

For you to use whatever you have created for display and/or printing, your information must be communicated to a 3270 device or control unit. This information is communicated to the 3270 device by use of the 3270 data stream, which is made up of structured fields, or commands, control characters, orders, attributes, and/or data.

Structured fields provide another format for sending information to a display or printer by use of the extended 3270 data stream. (Structured fields are discussed later in this chapter.)

Commands control such things as whether you write to or read from a display and whether the screen is erased before new data is written.

Control characters are used with certain commands to perform such functions as sounding the audible alarm, formatting the printer, and restoring or enabling the keyboard.

Orders are instructions written to the 3270 to tell the display unit how to format your panel.

Attributes determine the characteristics of the fields and/or characters within a field.

Data is the information you are displaying or printing.

THE 3270 DATA STREAM

The 3270 data stream is a formatted data stream consisting of a command and, for write type commands, of a write control character (WCC) followed by orders, attributes, and/or character data. The 3270 command byte defines the function to be performed on the character buffer. Functions that may be specified include:

- WRITE to the character buffer
- ERASE and then WRITE to the character buffer
- ERASE WRITE/ALTERNATE to the character buffer and change the effect of the buffer size
- READ the full character buffer

- READ only the modified data from the character buffer
- ERASE ALL the unprotected data from the character buffer

The 3270 data stream is based upon the presence of a mapped character buffer in the device. There is a fixed one-to-one relationship between each character storage location in the buffer and each character position on the display. For instance, consider a display (Figure 1-1) for which the display screen is composed of 12 rows of 80 columns each. Row 1 maps to the first 80 character storage positions in the character buffer, row 2 maps to the second 80 character storage positions, and so on. This sequence is the same whether the display size is 12 rows of 80 columns or up to 27 rows of 132 columns. When the ERASE/WRITE command is transmitted to the device, the character buffer is first cleared to nulls (X'00') and subsequent text is written into the character buffer sequentially.

The character buffer may contain codes for graphic characters or field attribute bytes, as shown in Figure 1-2. Since each storage location in the character buffer is mapped to a screen position on the display, field attribute bytes take up a character position on the display. The 3270 field attribute defines a field as the subsequent field attribute byte position plus the character positions up to, but not including, the next field attribute byte in the character buffer.

In the 3270 data stream, the field attribute appears as a valid graphic character (between X'4C' and X'FE') in the data stream. In the 3270 data stream, following a write type command, there is a Write Control Character (WCC). The WCC, like the field attribute, may also be a valid graphic character. The coding and functions are defined in Figure 2-26 (Chapter 2). The data to be written to the character buffer follows the WCC. The character data may be intermixed with the control codes or orders to provide formatting functions in the character buffer. For a description and definition of the commands and orders, refer to the IBM 3270 Information Display System: Component Description, GA27-2749.

HCW FIELDS ARE DEFINED

With this introduction to the 3270 data stream, consider the example shown in Figure 1-3. In this example, first, the character buffer is erased by the ERASE/WRITE command; then, the WCC is processed; finally, the orders, attributes, and data are interpreted in sequential order.

The first order in the data stream is generally a Set Buffer Address (SBA) order. This order includes two address bytes and determines the position in the character buffer where the data or attribute bytes will be stored. The address bytes are generally followed by a Start Field (SF) order and a field attribute. The field attribute is stored at the buffer location specified in the SBA order.

Each data field is established by writing a field attribute as the first position of the field. The placement of attribute bytes defines the field lengths, and the content of the attribute byte defines the other field characteristics. All the characters in a field, except the attribute byte itself, assume identical characteristics based on the specifications within the attribute byte.

Field attributes can be modified or removed by a 3270 program, and changing the content of the field attribute alters the characteristics of the field.

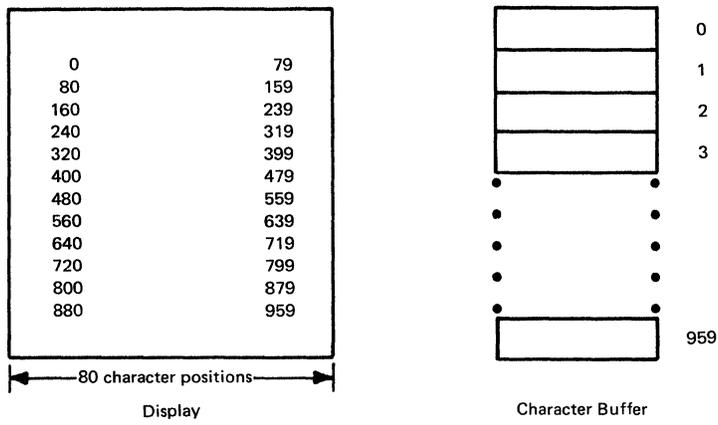


Figure 1-1. 3270 Mapped Buffer

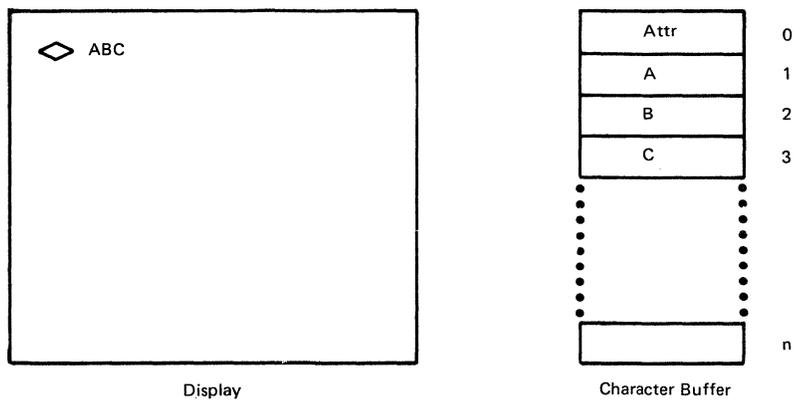


Figure 1-2. Mapping of Attribute and Graphic Characters to the Display Screen

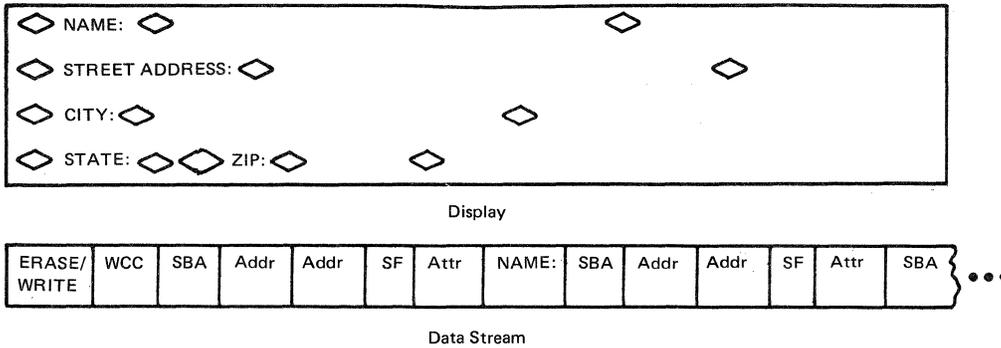


Figure 1-3. A Sample 3270 Data Stream

Following the field attribute, the character string "NAME" is stored in sequential character locations in the buffer. The next SBA order sets the current buffer address to a new location, and the process continues until the entire data stream has been interpreted.

EASE COLOR

To facilitate entry into the use of color, base color produces color displays from existing 3270 application programs with little or no reprogramming. Fields can be displayed in red, blue, green, or white. Field color is determined by the four combinations of the field protection and intensity attributes.

Ease color allows all programs that have been written for 3270 display stations and printers to execute in color on the equivalent 3279 or 3287 Model 1C or 2C, provided these programs use the field attributes of protection and intensity. Color will be produced as follows:

<u>Protection</u>	<u>Intensity</u>	<u>Resulting Color</u> 3287 1C and 2C	<u>Resulting Color 3279</u>	
			<u>Switch in</u> <u>Color Mode</u>	<u>Switch in</u> <u>Monochrome Mode</u>
Protected	Intensified	Black or green ¹	White	White
Unprotected	Intensified	Red	Red	White
Protected	Normal	Blue	Blue	Green
Unprotected	Normal	Green or black ¹	Green	Green

¹Depending on the Ease Color Specify feature ordered with the printer

When field attributes are not used (unformatted layouts--no fields have been defined), the 3279 will display all characters in green; the 3287 Models 1C and 2C will print in black.

In Base Color mode, the protected and unprotected field attributes still determine whether the input data format is fixed or variable; color is only a secondary effect of the attribute characteristics. Consequently, programming in these colors may be restricted by the way in which the logic of an application program needs to use the protection and intensify attributes.

Because the field attributes govern base color, the colors apply only to whole fields. It is not possible with this system to give different colors to separate characters within a field or to use color in unformatted layouts.

EXTENSIONS TO THE 3270 DATA STREAM

Extensions to the 3270 data stream have been made to enhance the presentation of data at the device. These enhancements provide the capability for expressing the Extended Color, Programmed Symbol set, and/or Extended Highlighting of the data to be displayed or printed at any terminal capable of interpreting these extensions to the 3270 data stream. To accomplish this, three new orders and one command have been defined to provide new functions in a device-independent manner, and the attribute concept has been expanded.

ATTRIBUTES

Remember that the 3270 field attribute is encoded to appear as a graphic character in the data stream and takes up a character position in the character buffer and on the screen. For the extended 3270 data stream functions, the concept of an attribute is different. The attribute has to be general in that it can apply to a character as well as to a field.

The attribute structure used in the extended 3270 data stream defines all attributes as an attribute type followed by an attribute value, in much the same way that an attribute is expressed in the English language; for example, "The COLOR of this car is BLUE." Each attribute type has an assigned 1-byte code. Most attribute types may be applied equally well to a character or to a field. For the extended 3270 data stream, each attribute type has associated with it a set of valid attribute values:

<u>Attribute Type</u>	<u>Attribute Value</u>
Color	Red, blue, yellow, turquoise, green, pink, white
Extended Highlighting	Blink, reverse video, and underscore
Programmed Symbol set	Nonloadable character set or Loadable Programmed Symbol set (see Programmed Symbol sets)

3270 Field Attributes

There are three basic forms of attributes for extended attribute terminals. The first was defined under the heading "3270 Data Stream" for normal 3270 field definition. This attribute defines the beginning of a field as described in the 3270 data stream and associates the following attributes with the field:

- Protected/unprotected fields
- Alphameric/numeric fields

- Display/nondisplay fields
- Intensity of fields (When high intensity is used on 3279 Color Displays, the character or field is displayed in white.)
- Selector-pen-detectable fields
- Tab stop positions (unprotected fields)

(See Figure 2-23 for hexadecimal values.) Each of these field attributes occupies a character position in the character buffer, and, during a display or printout, its character location appears as a space.

Extended Field Attributes

The second form of attribute is the extended field attribute. This attribute provides additional field attribute definition beyond that provided by the 3270 field attribute. It defines field characteristics such as field color, field highlighting (blink, reverse video, or underscore), and Programmed Symbol set of the field.

Extended field attributes are always associated with a field attribute. An extended field attribute does not occupy a position in the character buffer and does not display or print.

Character Attributes: The third form of attribute for extended attribute terminals provides definition of character attributes. These attributes are associated with individual characters to define characteristics such as Programmed Symbol set, character color, and character highlighting.

The field attributes of any single character are superseded by the character attributes associated with it. A character attribute does not occupy a position in the character buffer and does not display or print.

Start Field Extended (SFE) Order (X'29'): The first of the new data stream orders defined to deal with the extended field and character attributes, as well as the 3270 field attributes, is the Start Field Extended (SFE) order. Its syntax is:

SFE	n	First attribute	First attribute	nth attribute	nth attribute
X'29'		type	value		type	value

In the data stream, the byte count following the SFE order code indicates the number of "attribute type and attribute value" pairs that follow; the specified number follows the count. The use of this order is illustrated in Figure 1-4. Note that only the required attributes need to be specified. The unspecified attributes in the current buffer are set to nulls, and the current buffer address is incremented by 1. In this example all the unprotected fields are defined as underscored. Thus, the fields that the operator can enter will stand out from the rest of the fields on the panel. If an invalid attribute type or value is specified, processing of the outbound data stream stops, and an error code is returned.

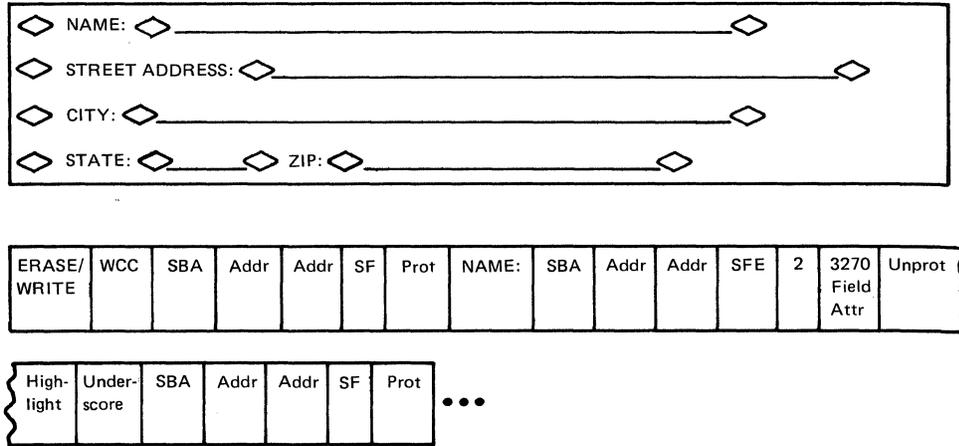
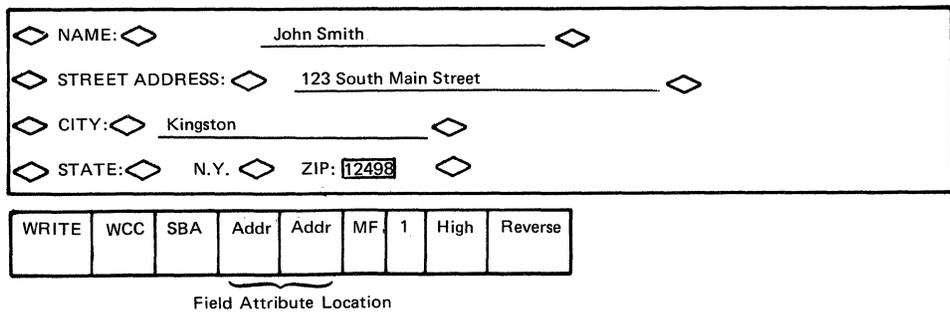


Figure 1-4. Using the Start Field Extended Order

Modify Field (MF) Order (X'2C'): The second order, the Modify Field (MF) order, is identical with the Start Field Extended order:



The Modify Field order allows specified field and extended field attributes to be modified without having to respecify all the attributes of the field. See Figure 1-5, where the application program wishes to highlight the zipcode field by changing its highlight attribute to reverse video. This is done with the Modify Field order without having to respecify the entire string of attributes previously associated with the field. The values of the attribute types that are selected in the order are changed to the new values specified, and the current buffer address is incremented by 1. The values of the attribute types not selected in the order remain unchanged.



Note: indicates reverse video.

Figure 1-5. Using the Modify Field Order

If the current buffer address is not a field attribute when a Modify Field order is processed, an error occurs and an error code is returned. If an invalid attribute type or value is specified, processing of the outbound data stream stops, and an error code is returned.

Set Attribute (SA) Order (X'28'): The Set Attribute order provides the ability to associate attributes with individual characters or strings of characters rather than with fields. The syntax of the Set Attribute order is:

SA	Attribute type	Attribute Value
X'28'		

For this order, only a single attribute "type and value" pair follows the order.

Every data character that follows a Set Attribute order in the outbound data stream is stored in the buffer with an associated character attribute having the type and value specified in the order. The specified attribute is associated with all subsequent data characters in the outbound data stream (even when they are directed to nonsequential addresses) until either (1) another write-type command is transmitted, (2) a new Set Attribute order changes the attribute, (3) the Clear key is depressed, or (4) the device is powered off.

The character attribute value determines how each associated data character is displayed (for example, in reverse video). If a character attribute value is X'00', then the highlighting attributes applied to the associated character are specified by the extended field attribute.

If an invalid attribute type or value is specified, processing of the outbound data stream stops, and an error code is returned.

In the example in Figure 1-6, the initial SA order indicates that all the characters following in the data stream are to be displayed with the reverse video attribute. Thus, the input data, John Smith, etc., which follows in the data stream will be displayed in reverse video characters independently of the highlight specified in the extended field attribute of the fields in which the characters are stored.

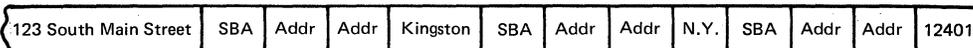
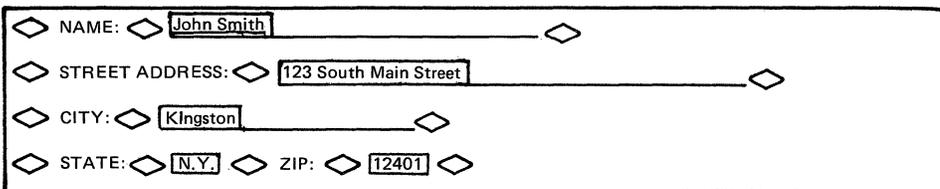


Figure 1-6. Using the Set Attribute Order

Attribute Processing

The model for attribute processing is shown in Figure 1-7. For each character position in the character buffer, there is an associated position in the Extended Attribute Buffer (EAB). Where the character in the character buffer is a field attribute, the associated position in the EAB is interpreted by the hardware as an extended field attribute. Where it is a graphic character, the associated position in the EAB is interpreted as a character attribute. The extended attributes do not occupy a position in the character buffer, but instead are in a related position in an EAB (Figure 1-7).

Now, if we were to use an extended field attribute, such as underscore, then the relation between character and extended field attributes would be that shown in Figure 1-8. Remember that the position in the EAB associated with a field attribute character is interpreted as an extended field attribute, and those positions associated with the graphic characters are interpreted as character attributes.

When the character attributes associated with characters in a field are not explicitly defined (NULL), the characters are displayed with the attributes specified for the field. As seen in Figure 1-9, when the character attributes are non-null, the characters are displayed with the attributes specified for the character, overriding the field attribute specification.

Attribute Types and Values: The following attribute types and values are defined for the Set Attribute, Start Field Extended, and Modify Field orders. All other attribute types and values are rejected.

Attribute Types

<u>Type Code</u>	<u>Type</u>	
X'00'	Character Attribute Reset	(used only with SA order)
X'41'	Extended Highlighting	
X'42'	Color	
X'43'	Programmed Symbols (PS)	
X'C0'	3270 Field Attribute	(not used with SA order)

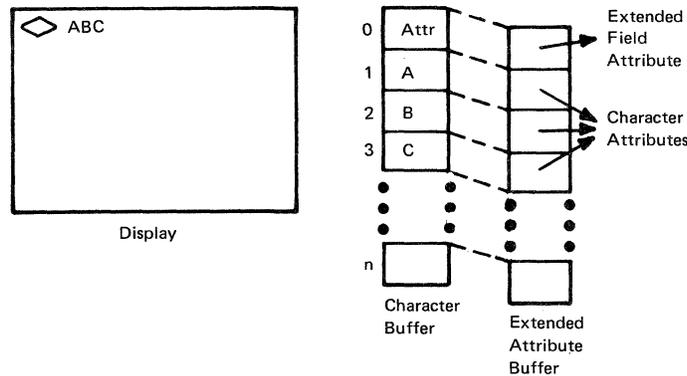


Figure 1-7. Model for Character and Extended Field Attributes

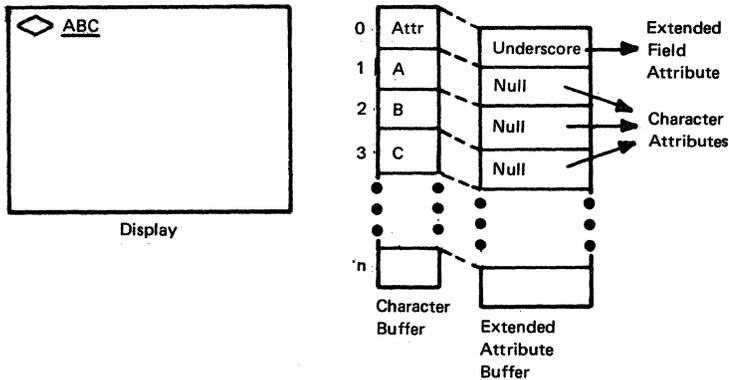


Figure 1-8. Relationship of Character and Extended Field Attributes

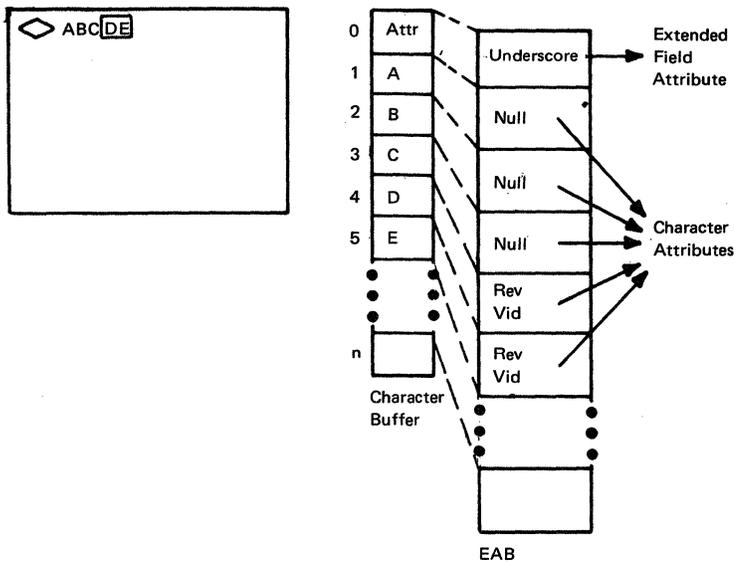


Figure 1-9. Character Attribute Override

Attribute Values: The attribute values for the above attribute types are defined below except for those for type X'C0', which remain as defined in the 3270 data stream. Any character defined with a character attribute value of X'C0' will assume the characteristics defined by the extended field attribute.

- Character attributes

X'00' - (Sets all character attributes specifiably in the SA order to their default value if type equals X'00'. This attribute type/value pair (X'00' X'00') code may only appear in the SA order.)

- Color

X'00' - The color will be green on a 3279 and black on a 3287.
 X'F1' - Blue

X'F2' - Red
 X'F3' - Pink
 X'F4' - Green
 X'F5' - Turquoise
 X'F6' - Yellow
 X'F7' - White for 3279, black for 3287

• Extended Highlighting

X'00' - It will assume the characteristics of the defined field attribute.
 X'F1' - Blink
 X'F2' - Reverse Video
 X'F4' - Underscore

• Programmed Symbol set

X'00' - The nonloadable character set shipped with the device is used.
 X'40'-X'EF' - Coded Graphic Character Set Local ID (CGLI) for loadable Programmed Symbol sets
 X'F1' - APL/Text character set (nonloadable symbol set, only for SA order)

STRUCTURED FIELDS

As pointed out earlier, the 3270 data stream is a formatted data stream. To provide additional controls and transmit various data types other than character, it was necessary to define a new data structure in the data stream. This is termed "structured fields."

In the 3270 data stream, structured fields are introduced with a new command, the Write Structured Field (WSF) command. This new command does not contain explicit control information as the other 3270 commands do. It simply means, "Here is data in a structured field format."

Following the WSF command, all data in the transmission must be in structured field format. A structured field transmission has the form shown in Figure 1-10.

The structured field syntax permits variable-length data and controls to be encoded in such a way that a device that is processing the data stream sequentially can decompose a sequence of fields into its component fields without having to scan every byte. Each structured field contains a 2-byte-length field. This indicates the length of the field (including the length bytes), in effect, pointing to the next structured field in the transmission. Next follows a 1-byte type field, and then parameters and/or data in the format defined by the type code. If the value specified in the length field is zero, this structured field is treated as the last structured field in the transmission. The type field in the structured field identifies the purpose of the field.

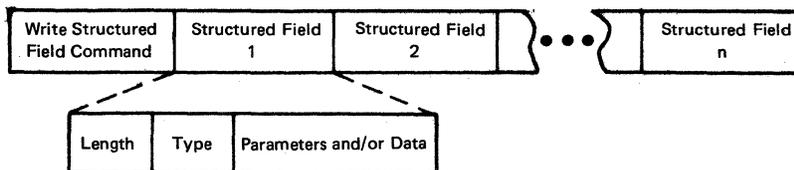


Figure 1-10. Structured Fields

The types of structured fields are as follows:

<u>Structured Field</u>	<u>Type Code</u>
Outbound Function	
Load Programmed Symbols (Load PSs)	X'06'
Read Partition Query	X'01'
Set Reply Mode	X'09'
Inbound Function	
Query Reply	X'81'
Under Query Reply, the following structured fields are used:	
Usable Area	X'81'
Symbol Sets	X'85'
Color	X'86'
Highlighting	X'87'
Reply Mode	X'88'

The structured field, Load Programmed Symbols, is used to load the Programmed Symbol sets into the buffers. This command and query will be discussed here. For a description of the other types of structured fields, refer to the IBM 3270 Information Display System: Component Description, GA27-2749.

Load Programmed Symbol Set (Load PS) X'06': The Load Programmed Symbol set is one type of structured field defined as an outbound control function. It is used to load character definition data into the device. A loadable character set contains 190 contiguous addressable character slots, which are associated with data stream code points X'41' through X'FE'. The slot associated with X'40' can be addressed, but cannot be loaded and contains a space. This structured field causes characters and/or symbols to be loaded into contiguous addressable slots. If no data is transmitted with this structured field, the operation is executed as specified in the parameters (see Figure 1-11) and no data is loaded. This means that characteristics of the symbol set may be changed without altering the contents of the symbol set.

As an example of structured field usage, consider the loading of Programmed Symbol (PS) sets. The PS attribute operates as shown in Figure 1-12. The display looks at the PS attribute for each character as it prepares to display it. If the character set specified for the character is the nonloadable character set (PS0), the dot pattern for the character is extracted from a nonloadable character set installed in the device. If the character set specified is a loadable Programmed Symbol set, the display extracts the dot patterns for the character or symbol from a user-defined symbol set (PSF here). The Programmed Symbol set is specified by the PS attribute. The EBCDIC character code defines which character in the PS is to be displayed. To follow the example further, the format for the WSF to load a PS set is given in Figure 1-11. This command is then used to create the box "A" in PSF in Figure 1-12.

Byte	Bit	Content	Meaning
0-1		Length	Length of structured field.
2		X'06'	Load PS.
3	0	Basic/Extended	Basic or extended form.
		b'0'	Basic form.
		b'1'	Extended form.
	1	Clear	Clear PS storage.
		b'0'	Do not clear PS storage.
		b'1'	Clear the PS (plane) specified before loading the character data in this structured field.
	2	Skip	Skip suppress.
		b'0'	Suppression off.
		b'1'	Suppression on.
	3-7		Type
3		b'0'	Display unique type: 18-byte form: the first 2 bytes contain a 16-bit vertical slice; the following 16 bytes contain 8-bit horizontal slices. Type 1 compressed. Printer unique type: Column loading (from left to right). Type 5 compressed.
4-7	X'1' (Type 1)		
	X'2' (Type 2)		
	X'5' (Type 5) X'6' (Type 6)		
4		CGLI (alias)	Coded Graphic Character Set Local ID: X'40' through X'EF' X'FF' indicates that the storage associated with this CGLI is free.
5		CHAR	Beginning EBCDIC Code Point (X'41' through X'FE') inclusive
6		Storage	PS set storage number (X'02 through X'07')

Figure 1-11 (Part 1 of 2). Format of the WSF for Load PS

Bytes 7 through 12 are parameters for the extended form.

Byte	Bit	Content	Meaning
7		p-length	Length of parameters for the extended form. This includes the length parameter itself. The parameters defined below may be progressively included by specifying the appropriate length. Omitted parameters are equated to X'00' per missing byte. Action is the same as receiving a byte containing X'00'.
8	0	APA b'0' b'1'	All points available. All points available. Not all points available.
	1	CB b'0' b'1'	CGLI compare No CGLI compare
	2	OB b'0' b'1'	PS set is keyboard-selectable. PS set is not keyboard-selectable.
	3-7	Reserved	Must be set to zeros.
9		LW	Number of X-units in block matrix.
10		LH	Number of Y-units in block matrix.
11		X'00'	One-byte codes.
12	0-4	Reserved	Must be zeros.
	5-7	Color b'000' b'001' b'010' b'100'	Color planes. Single or all planes Blue Red Green

This is the end of the parameters for the extended form.

M-N		Data	PS character block matrix definitions.
-----	--	------	--

Figure 1-11 (Part 2 of 2). Format of the WSF for Load PS

The description of the byte function of Figure 1-11 is as follows:

Byte 0 and 1 Give the length of the structured field.

Byte 2 Gives the function to be performed.

Byte 3 Bit 0 indicates whether byte 7 is included.

Bit 1 - When bit 1 is set to 1, the entire PS storage selected for loading is cleared of any previous character data. This can be done by overwriting with character data or nulls. If the bit is set to 0, the selected PS storage is not cleared; thus, characters can be added to an existing character set. For a triple-plane set, only the plane(s) indicated in byte 12 is cleared.

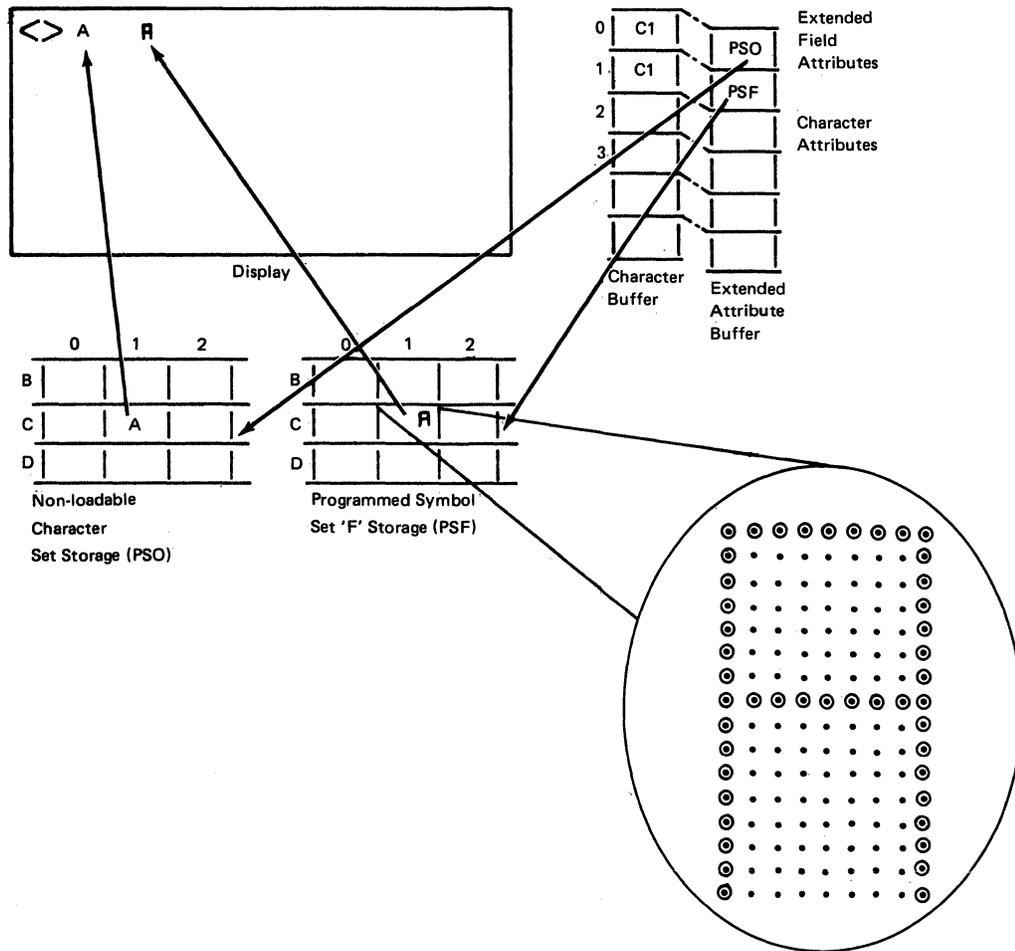


Figure 1-12. Programmed Symbol Sets

Bit 2 controls the vertical positioning of a row of characters. The next row is positioned vertically adjacent to the current row if the current row contains a character from the PS set having SKIP SUPPRESS on. If SKIP SUPPRESS is off, the row positioning is unaffected by the occurrence of a character from this PS storage.

Bits 3-7 specify the type of PS format. For type 1 or 2, the same data stream format is present for both the 9x12 and 9x16 block matrix. The data stream always defines a full 9x16 matrix. When a device uses the 9x12 only, the last four bits of decoded data of the second byte and the last four bytes of the decoded data for each block matrix are not displayed. For types 5 or 6, the decoded 10 bytes are loaded in the character matrix, starting from the left side of the character matrix.

Byte 4 A CGLI of X'FF' indicates that this PS storage is free. A PS set of X'FF' cannot be selected by the operator.

Byte 5 The value CHAR indicates the first slot in this storage to be loaded. CHAR must be in the range X'41' through X'FE'.

Byte 6 The storage number indicates the physical storage to be loaded. Each storage number is related to an attribute selection key defined for PS. These relations are storage numbers 02 through 07, which equate to attribute selection keys PSA through PSF, respectively.

Byte 7 Indicates the parameters specified. If byte 7 is included, it must not contain the value X'00' or a value greater than X'06'.

Byte 8 Bit 0 - When set to 1, implies that fewer than all points may be displayed or printed to allow a performance gain for a specific device.

Bit 1 - When the CB bit is set to 0 and a local copy is initiated, the CGLI of this symbol set is compared with symbol set CGLI in the printer to determine whether there is a match. If the CGLIs match, the copy is performed using the corresponding CGLI in the printer. If the CGLIs do not match, characters from the base character set of the printer are used. When the CB is set to 1, there is no compare and the characters are printed from the base character set of the printer.

Bit 2 - When set to 1, it signifies that this PS is intended for output only. The PS attribute select key normally associated with the storage containing this character set cannot be selected by the operator.

Bits 3-7 must be set to zero.

Byte 9 Must be set to ten for printers and nine for displays.

Byte 10 Must be eight for printers and sixteen for displays.

(If bytes 9 and 10 are not specified or are set to zero, a device default block matrix size of 10x8 for printers and 9x16 for displays is used to determine the dot matrix size of characters in the load PS data. The origin of each block matrix is chosen as the upper-left corner.)

Byte 11 Only 1-byte addressing is used; must be X'00'.

Byte 12 For multiple-plane PS, if color (bits 5-7) is specified with a value b'000', then, for each code point, the character definition is loaded into each plane of the PS.

 For a multiple-plane PS, if color is specified with one of the other values, then the PS data is loaded into the specified PS plane.

Bytes M-N These bytes contain the block matrix definition of the data to be loaded.

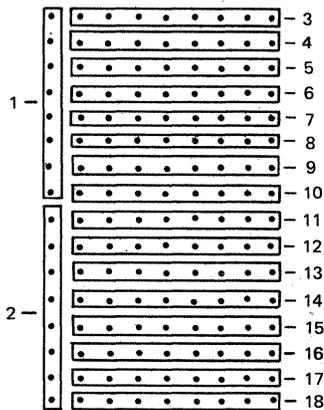
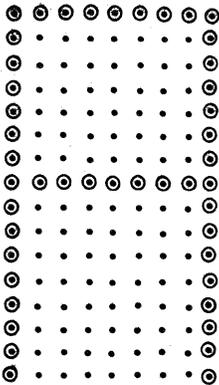
Defining a Character

To define a character, the byte value for the block matrix must be put into the data stream by use of bytes M-N of the structured field. A 10x8 block matrix is composed of 80 dots and a 9x16 block matrix has 144 dots. By slicing the block matrix into bytes, 8 dots to a byte, characters or symbols can be formed. If each dot is equal to a bit, then turning it on (1) allows this dot to display or print, as the case may be. By considering the slices as bit strings, turning each dot on or off gives a hex representation which determines what is printed or displayed. Consider the example of box "A" in Figure 1-12. Figure 1-13 shows the slicing of the block matrix and its bit string hexadecimal representation for a display 9x16 block matrix. Figure 1-14 shows the same for a 10x8 printer block matrix. Figures 1-15 and 1-16 give the structured field representation for Figures 1-13 and 1-14, respectively.

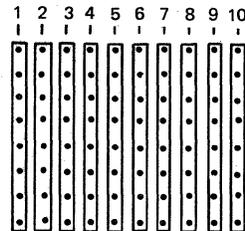
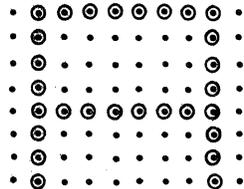
Programmed Symbol Sets: All 3270 display stations and printers have a nonloadable character set installed in the device. A second nonloadable set is present if the machine has the APL/Text capability. Application programs use the characters or symbols in these sets as they are; they cannot alter or add to them.

On the 3278 and 3279 display stations, each character is represented by a pattern of dots selected from a block matrix. The 3278 Models 2 and 3 have matrixes of 9 dots horizontally by 16 dots vertically, while the 3278 Model 4 and all 3279 models have matrixes of 9 dots horizontally by 12 dots vertically. On the 3287 printers, each character is represented by a pattern of dots selected from a block matrix of 10 dots horizontally by 8 dots vertically. Characters are normally represented by predefined patterns accessed by character codes in the data stream that is sent to the terminal.

Along with the nonloadable character sets found in the 3270, the extended 3270 data stream allows for the storage and accessing of up to six Programmed Symbol sets, referred to as PSA through PSF. These Programmed Symbols sets are user-defined symbols and can be altered whenever required. Each Programmed Symbol set can have a maximum of 190 graphic symbols. They are loaded and accessed under program control and may be used by the display operator. The user may use the full block matrix to design a character or symbol, allowing for a high degree of flexibility in the choice of graphic symbols, which can be used for the display and/or printing of diagrams, curves, bar graphs, and special fonts, such as bold face and mathematical symbols, in addition to other special symbols.



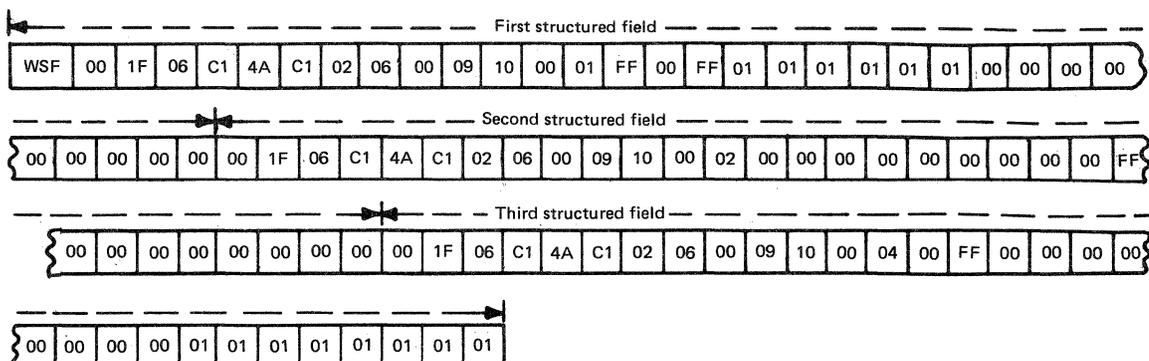
1. 11111111 = X'FF'
2. 11111111 = X'FF'
3. 11111111 = X'FF'
4. 00000001 = X'01'
5. 00000001 = X'01'
6. 00000001 = X'01'
7. 00000001 = X'01'
8. 00000001 = X'01'
9. 00000001 = X'01'
10. 11111111 = X'FF'
11. 00000001 = X'01'
12. 00000001 = X'01'
13. 00000001 = X'01'
14. 00000001 = X'01'
15. 00000001 = X'01'
16. 00000001 = X'01'
17. 00000001 = X'01'
18. 00000001 = X'01'



1. 00000000 = X'00''
2. 11111111 = X'FF'
3. 10001000 = X'88'
4. 10001000 = X'88'
5. 10001000 = X'88'
6. 10001000 = X'88'
7. 10001000 = X'88'
8. 10001000 = X'88'
9. 11111111 = X'FF'
10. 00000000 = X'00''

Figure 1-14. Character Definition for a 10x8 Printer Matrix

Figure 1-13. Character Definition for a 9x16 Display Matrix



This data stream would result in the character's being displayed as shown below.

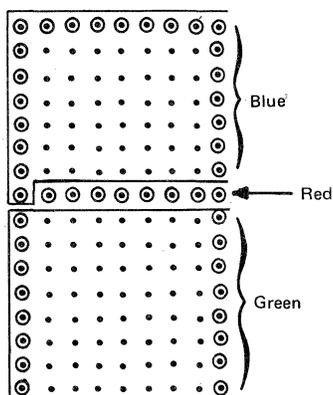


Figure 1-17. Character Definition of a Multicolor Symbol

- If a symbol defined in a triple-plane set is displayed or printed with an explicit color attribute other than X'F7' (for example, red), the whole symbol as defined by combining all three planes is displayed in the specified color (red).
- If a symbol defined in a triple-plane set is displayed or printed without any color attribute, the whole symbol appears in monochrome in the same way as any character.

Unless the user loads the three planes in a triple-plane symbol set separately, the system loads the same pattern into all three planes. Thereafter, the triple-plane set behaves (for any character that has the same pattern in all three planes) exactly as if it were a single-plane set. Consequently, a triple-plane set can always be used as if it were a single-plane set.

Read Partition Query (Outbound) and Query Reply (Inbound) Structured Fields

The Read Partition Query and Query Reply structured fields provide the mechanism for a host application program to request and receive information regarding a device's color, highlighting, usable area, reply modes, and Programmed Symbol set characteristics.

Read Partition Query Structured Field: The Read Partition Query structured field is valid only in outbound data streams and must be the last structured field in a WSF transmission.

The format of the Read Partition Query structured field is:

<u>Byte</u>	<u>Bit</u>	<u>Content</u>	<u>Meaning</u>
C-1	-	X'0000' or X'0005'	Length
2	-	X'01'	Structured field type
3	-	X'FF'	
4	-	X'02'	Identifies this structured field as a Read Partition Query

If bytes 3 and 4 do not exist, or byte 3 does not contain X'FF', or bytes exist after byte 4, an Cp Chk or error code is returned. If the outbound chain (SNA) does not contain CD, the chain is rejected.

The response to the Read Partition Query is the transmission of a series of structured fields that describe the characteristics of the addressed terminal. Response is immediate when SCLC/SNA protocols are being used; response is given when the terminal is polled if BSC protocol is being used and, in the case of the 3274 Model 1D, when a Read Modified CCW is received (see following).

3274 Model 1D Read Partition Query Response: When Read Partition Query is received by the 3274 1D as the last structured field in a WSF transmission, the 3274 1D returns a status of DE and terminates the operation. The Time indicator in the Operator Information Area is turned on, and the keycard is locked to prevent operator interference with the query reply.

If Read Partition Query is not the last structured field in the WSF transmission, or if there is an error in the WSF data, the 3274 1D terminates the operation with status of DE, UC, and sets the operation check bit in the sense byte.

Following acceptance of the Terminal Query structured field, the 3274 1D generates an asynchronous status of attention, requesting the host to issue a Read Modified command to obtain the query reply.

Query Reply Structured Fields: The response to the Read Partition Query function is the transmission inbound of a series of structured fields indicating the field and character attributes, the screen or page size and characteristics, the Programmed Symbol sets, and the reply modes available on the addressed terminal. Any or all of the structured fields may be transmitted, each defining a supported feature characteristic: color, highlighting, usable area, reply modes, and Programmed Symbol sets. Since each structured field contains its own unique identification, the order in which the fields are transmitted is not defined.

Query Reply Inbound Data Stream: A Query Reply inbound data stream consists of an AID byte (X'88') defining what follows as an inbound structured field data stream, followed by the structured fields. Each structured field is of the general format: length - type - data.

Query Reply (Color) Structured Field: This Query Reply structured field defines the color attribute values accepted by the addressed terminal (and may be returned in an inbound data stream if the terminal supports read commands). Eight pairs of bytes, one pair for each of the possible color attribute values, are returned to the host. The first byte of a pair contains the color attribute value; the second byte contains the same value if that color attribute is supported by the terminal or the default color attribute value (X'00') if it is not. There is one exception: the second byte of the pair defining the default color attribute support indicates the default color that is supported.

The structure is:

Byte	Bit	Content	Meaning
0-1		X'0016'	Length of structured field
2		X'81'	Query Reply structured field type
3		X'86'	Identifies this query reply as color
4	0	E'0'	Reserved
	1	E'0'	Printer only - black ribbon not loaded
		E'1'	Printer only - black ribbon loaded
2-7		E'CCCCC0'	Reserved
5		X'08'	Number of color pairs
6-21		(terminal-dependent. <u>Note</u> : Possible color attribute values are X'00' - default, X'F1' - blue, X'F2' - red, X'F3' - pink, X'F4' - green, X'F5' - turquoise, X'F6' - yellow, X'F7' - black for printers, white for displays)	

Byte	Possible Attribute					
	Value	3278	3287-1,-2	3287-1C,-2C	3279-2B,-3E	
6-7	X'00'	X'F4'	X'F7'	X'F7'	X'F4'	
8-9	X'F1'	X'00'	X'00'	X'F1'	X'F1'	
10-11	X'F2'	X'00'	X'00'	X'F2'	X'F2'	
12-13	X'F3'	X'00'	X'00'	X'00'	X'F3'	
14-15	X'F4'	X'00'	X'00'	X'F4'	X'F4'	
16-17	X'F5'	X'00'	X'00'	X'00'	X'F5'	
18-19	X'F6'	X'00'	X'00'	X'00'	X'F6'	
20-21	X'F7'	X'00'	X'00'	X'F7'	X'F7'	

As an example, the following Query Reply (Color) structured field might be transmitted for a 3287-1C,-2C printer:

X'00168186C00800F7F1F1F2F2F300F4F4F500F600F7F7'

Query Reply (Highlighting) Structured Field: This Query Reply structured field defines the highlighting attribute values accepted by the addressed terminal (and may be returned in an inbound data stream if the terminal supports read commands). Four pairs of bytes, one pair for each of the possible highlighting attribute values, are returned to the host. The first byte of a pair contains the highlighting attribute value; the second byte contains the same value if that highlighting attribute is supported by the terminal, or the default highlighting attribute value (X'00') if it is not. There is one exception: the second byte of the pair defining the default highlighting attribute support indicates the default highlighting that is supported.

The structure is:

<u>Byte</u>	<u>Content</u>	<u>Meaning</u>
0-1	X'0C0D'	Length of structure
2	X'81'	Query Reply structured field type
3	X'87'	Identifies this query reply as highlighting
4	X'04'	Number of highlighting pairs
5-12	(terminal-dependent. <u>Note</u> : Possible highlighting attribute values are X'00' - default, X'F1' - Blink, X'F2' - Reverse Video, X'F4' - Underscore)	

<u>Byte</u>	<u>Value</u>	<u>Possible Attribute</u>	<u>3278</u>	<u>3287-1,-2,-1C,-2C</u>	<u>3279</u>
5-6	X'00'		X'00'	X'00'	X'00'
7-8	X'F1'		X'F1'	X'00'	X'F1'
9-10	X'F2'		X'F2'	X'00'	X'F2'
11-12	X'F4'		X'F4'	X'F4'	X'F4'

Query Reply (Usable Area) Structured Field: This Query Reply structured field indicates the size and characteristics of the screen or page of the addressed terminal. Screen or page size is expressed as maximum width of usable area in characters (columns or print positions) and maximum depth of usable area in characters (rows or print lines). (For a printer, the values returned correspond to the maximum print position (MPP) and maximum print line supported by the hardware, not the current settings if operator-specifiable).

The default size of the block matrix within which a character is presented is also defined.

The structure is:

<u>Byte</u>	<u>Bit</u>	<u>Content</u>	<u>Meaning</u>
0-1	-	X'17'	Length of this structure
2	-	X'81'	Query Reply structured field type
3	-	X'81'	Identifies this query reply as "usable area."
4	0-2	B'000'	Reserved
	3	B'0'	Not a hard-copy device
		B'1'	A hard-copy device
	4-7	B'0C01'	14-bit addressing allowed
5	-	X'00'	Reserved
6-7	-	-	Width of usable area in characters block matrix
		X'50'	3278-2,-3,-4, 3279-2,-3
		X'84'	3278-5, 3287

<u>Byte</u>	<u>Bit</u>	<u>Content</u>	<u>Meaning</u>
8-9	-	-	Depth of usable area in characters block matrix
		X'18'	3278-2, 3279-2
		X'20'	3278-3, 3279-3
		X'2E'	3278-4
		X'1E'	3278-5
		X'66'	3287
10	-	X'00'	Unit of measure is the inch for distance between dots given for x and y directions in bytes 11-14 and 15-18.
11-14	-	-	Dot spacing in the X (horizontal) direction, expressed as a fraction; 2-byte numerator/2-byte denominator; and measured in the units defined in byte 10.
		X'00020089'	3278-2,-3,-4
		X'000A02E5'	3279-2,-3
		X'00010064'	3287
15-18	-	-	Dot spacing in the Y (vertical) direction, expressed as a fraction; 2-byte numerator/2-byte denominator; and measured in the units defined in byte 10.
		X'00020085'	3278-2,-3,-4
		X'0002006F'	3279-2,-3
		X'00020085'	3287
19	-	-	Default width of block matrix, in dots
		X'09'	3278-2,-3,-4, 3279-2,-3
		X'0A'	3287
20	-	-	Default depth of block matrix, in dots
		X'10'	3278-2,-3
		X'0C'	3278-4, 3279-2,-3
		X'08'	3287
21-22	-	-	Character buffer size, in bytes
		X'0780'	3278-2 (1920), 3279-2 (1920)
		X'0A00'	3278-3 (2560), 3279-3 (2560)
		X'0E70'	3278-4 (3440)
		X'nnnn'	3287 - Dependent on installed buffer size (2K or 4K). Equivalent to display sizes except when byte 24 of an LU type 3 BIND command is set to X'00'. Wrap points for the physical buffer are then given as follows:
			2K buffer - X'07E0' (1968)
			4K buffer - X'0EB0' (3760) with PS feature installed
			4K buffer - X'0FB0' (4016) no PS feature

Query Reply (Reply Mode) Structured Field: This Query Reply structured field indicates the format of inbound data stream that the addressed terminal supports.

The structure is:

<u>Byte</u>	<u>Bit</u>	<u>Content</u>	<u>Meaning</u>
0-1	-	X'CC07'	Length
2	-	X'81'	Query Reply identifier
3	-	X'88	Identifies this Query Reply as "reply mode."
4	-	X'00'	Indicates that the terminal supports 3270 inbound data streams.
5	-	X'01'	Indicates that the terminal supports extended 3270 data streams.
6	-	X'02'	Indicates that the terminal supports character mode extended 3270 data streams.

Query Reply (Symbol Sets) Structured Field: This Query Reply structured field indicates the number and kind of symbol sets (both user-defined Programmed Symbol sets and IEM-defined sets) present in the terminal. The terminal storage ID is given as well as an indication of whether it is associated with a symbol set. The structured field consists of a 12-byte base and up to eight 3-byte storage descriptors, one for each storage area present in the terminal.

The structure is:

<u>Byte</u>	<u>Bit</u>	<u>Content</u>	<u>Meaning</u>
0-1	-	X'nnnn'	Length - includes any 3-byte symbol set descriptors present
2	-	X'81'	Query Reply Identifier
3	-	X'85'	Identifies this query reply as "symbol sets."
4	0	E'1'	Graphic escape supported
	1	E'0'	Reserved
	2	E'0'	Load Programmed Symbols structured field <u>not</u> supported
		E'1'	Load Programmed Symbols structured field supported
3	E'0'	Load Programmed Symbols structured field extension <u>not</u> supported	
	E'1'	Load Programmed Symbols structured field extension supported	
4-7	-	E'CCCC'	Reserved
5	-	X'00'	Reserved
6	-	X'0A'	Default block matrix width - printer
	-	X'09'	Default block matrix width - display

7	-	X'08'	Default block matrix depth - printer
		X'10'	Default block matrix depth - display
8-11	-	X'4CCCC000'	Display supports load PS data format type 1 (will be X'60000000' if the 3274 has been customized to support decompression).
		X'04CCCC000'	Printer supports load PS data format type 5 (will be X'06CCCC000' if the 3274 has been customized to support decompression). Any other values for bytes 8-11 are rejected.
12	-	X'03'	Length of each symbol set descriptor that follows.

Descriptors (One or more 3-byte descriptors follow byte 12; a descriptor defines one terminal storage and symbol-set characteristics.)

0	-		Terminal storage identification: X'00' to X'07'.
		X'00'	Read-only storage containing I/O interface code symbol set.
		X'01'	Read-only storage containing AFL/Text set if feature present.
		X'02'	Host-loadable terminal storages for Programmed Symbol sets. These IDs are specified in the load PS structured field.
		X'07'	
1	0	B'0'	Read-only storage.
		B'1'	Loadable terminal storage.
	1	B'0'	Single-plane storage.
		B'1'	Triple-plane storage.
	2	B'0'	Symbols are accessed using a 1-byte code.
	3	B'0'	Comparison of the symbol set ID of the symbol set loaded in this storage with the symbol set ID(s) of sets loaded in the printer is allowed (copy operations).
		B'1'	Comparison is not allowed.
	4-7	B'CCCC'	Reserved.
2		X'nn'	Symbol set ID. The ID currently associated with the terminal storage ID contained in byte 0. Value range X'40' through X'EF' for valid symbol ID. A value of X'FF' indicates that the storage is not associated with any symbol set.

CHAPTER 2. SCREEN DESIGN

FIELD CONCEPT

People dealing with information see it as a collection of individual elements. For example, what we know about John Smith's employment may be a collection of individual elements: his name, serial number, location, and date of hire. The size of the element is the amount of data required to convey useful information. You do not think of "J" and "C" and "H" and "N" as useful individually, but collectively, as the name JOHN. You do not think of JOHNSMITH963981BOSTON070262 as being useful collectively, but see the elements individually: name: JOHN SMITH, serial number: 963981, location: BOSTON, date of hire: 07/02/62.

Each data element has its own characteristics. In this example, the serial number is 6 numeric digits and varies from employee to employee. The word "NAME" is 4 characters, is alphabetic, is all uppercase, and does not change. When people record these elements of data on paper they take on such additional characteristics as position (where on the sheet of paper the item is written), color (what ink or medium is used), size of the letters, and writing style.

In the past, when information was handled by a data processing device it was generally handled as an artificial entity called a record. The contents and characteristics of a record were primarily determined by device requirements, and little or no attention was given to the individual information elements. Data processing users had to adjust their thought pattern to conform to the machine requirements.

The IBM 3270 Information Display System recognizes that people deal with individual units of information. The system has been designed to conform to human needs and requirements, and it enables you to deal with data by individual elements or "fields," each with its own individual characteristics.

You may describe data to the 3270 on a field or character basis and specify the characteristics or "attributes" of each individual field or character. The 3270 then provides program and data control based on your individual field and character definitions.

What Attributes May Be Assigned to a Field

Besides length, which is controlled by the position of field attributes, you may specify these additional characteristics with the attribute byte:

Protection: A field is either protected or unprotected. When it is protected, the operator cannot enter or modify data in any location within that field.

In an unprotected field, the operator can enter characters or can delete or modify characters that are already there. Headings, labels, titles, and formats are commonly specified as protected. Any field in which the 3270 operator should enter or modify data must be specified as unprotected.

In Figure 2-1, NAME: would most likely be specified as protected. JOHN B DOE would be specified as protected if it was written by the computer and is to remain unchanged. If JOHN B DOE is to be entered or modified by the operator, the attribute 2 must specify unprotected.

Color: If the device has the capability of displaying or printing in color, then the fields or individual characters may be defined in four or seven colors, depending on whether the device has base-color (see "Base Color" in Chapter 1) or extended-color capability. With extended-color capability, the device can display fields in one of seven colors, individual characters in one of seven colors, or individual characters in multicolors. (See "Triple Planes" in Chapter 1.) Extended Color devices can display the colors white, red, blue, green, pink, yellow, and turquoise.

The base-color printer prints fields in one of four colors determined by the existing field-protect and field-intensify attributes. (See Chapter 1.) When the printer has the extended-color capability, color attributes can be applied to individual characters or to entire fields. The extended-color attributes are interpreted by the printers as follows:

Color Displayed On 3279	Color Printed on 3287 Models 1C and 2C
White	Black or green
Red	Red
Blue	Blue
Green	Green or black
Pink	Black
Yellow	Black
Turquoise	Black

Note: The printing of black or green, as shown, depends upon the Base Color (black or green) specify feature selected.

Extended Highlighting: If the display has the capability of interpreting the extended 3270 data stream, then Extended Highlighting attributes (blink, reverse video, and underscore) can be applied to entire fields or to individual characters. In Figure 2-1, the unprotected fields could be underscored to highlight them as operator input areas.

For printers with the ability to interpret the extended 3270 data stream, only the underscore attribute can be applied to whole fields or to individual characters.

Character Content: A field is either alphanumeric, numeric, or user-defined symbols. An operator can enter alphanumeric, numeric, special characters, or user-defined symbols in an alphanumeric field.

The numeric attribute is more complex; it depends upon whether the numeric Lock feature is present and which keyboard is attached to the display. Figure 2-2 shows what characters may be entered with various combinations of keyboards and field types.

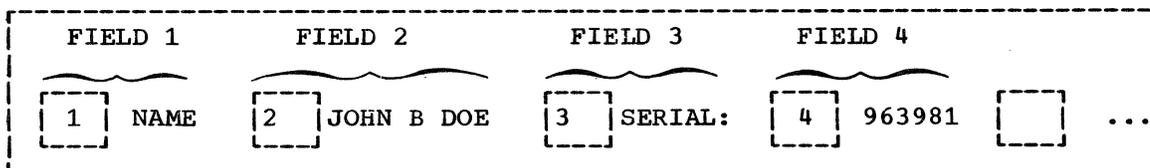


Figure 2-1. Example of Four Fields and Attribute Bytes

Keyboard Type	Keyboard Numeric Lock	Shift Key Pressed	Field Type	Protected	Resulting Characters		
					In Buffer	Displayed On Screen	Read Into Storage
Typewriter	No	No	Alpha or Numeric	No	Lowercase	Uppercase	Lowercase
Typewriter	No	Yes	Alpha or Numeric	No	Uppercase	Uppercase	Uppercase
Typewriter	Yes	No	Alpha	No	Lowercase	Uppercase	Lowercase
Typewriter	Yes	Yes	Alpha	No	Uppercase	Uppercase	Uppercase
Typewriter	Yes	No	Numeric	No	Can only enter 0-9, period, and minus sign; any other characters lock keyboard.		
Typewriter	Yes	Yes	Numeric	No	Can only press dup key; any other action locks keyboard.		
Data Entry	No	--	Alpha	No	Alpha keys produce uppercase alpha characters. Numeric shift key produces numeric characters. Alpha shift key has no effect.		
Data Entry	No	--	Numeric	No	Numeric shift key has no effect. Alpha shift key overrides numeric specification and allows alpha character entry.		
Data Entry	Yes	--	Alpha	No	Alpha keys produce uppercase alpha characters. Numeric shift allows numeric character entry. Alpha shift key has no effect.		
Data Entry	Yes	--	Numeric	No	Can only enter 0-9, period, dup, and minus sign. Any other characters lock all keys except for RESET key. Numeric shift key allows numeric character entry, alpha shift key allows alpha character entry.		

Figure 2-2. Results of Keyboard and Field Combinations

Visibility and Detectability: A field is either displayable or nondisplayable. When it is displayable and contains characters, those characters are displayed. When it is nondisplayable, any characters within that field are not displayed. The nondisplayable attribute is useful for entering classified or security information at a display unit that is in public view. Nondisplayable data is accepted by the 3270, but it is not visible on the screen.

If your 3270 system includes 3274s or 3276s, you must take certain precautions since these control units, unlike the 3271, 3272, and 3275, update screen images on a partial basis without removing previous images

from the screen (there is no screen "blink"). To maintain security, make sure that programs

- Send a nondisplay attribute byte prior to sending the intended new nondisplayable data to preclude its momentary appearance on the screen.
- Do not overwrite a nondisplay attribute byte of the currently displayed image when partially changing field formats.

All characters within a displayable field can be displayed at regular brightness, at a high intensity, in color, or with Extended Highlighting (blink, reverse video, underscore), so that they stand out among regular display fields. Any of these (Extended Highlighting, Color, or high intensity) can be used to call attention to error conditions or to highlight protected or format fields. High intensity, when used on a color display, results in the field's being displayed in white. Throughout this document, in discussions on highlighting fields for better operator recognition and performance, remember that, if the device has the capability of interpreting the extended 3270 data stream functions, then color, reverse video, blink, or underscore may be used to highlight fields or characters for display. Color and/or underscore can be used to highlight printer output. On 3278 displays, only Extended Highlighting and high intensity can be used. Normal intensity or underscore may be used for all input fields, so the terminal operator can tell at a glance which fields require operator action. You should not specify unprotected fields as high intensity since such fields may become selector-pen-detectable (if this feature is installed) if the operator enters a question mark or blank as the first input character. Fields are specified as either detectable or nondetectable. When a field is detectable, it can be used for selector-pen operations. A nondetectable field location cannot be detected by the selector pen or cursor select. You are urged to designate all detectable fields as protected to prevent the operator's changing the content of the sensitive field.

Transmission: The most common operation of the 3270 (Read Modified) sends to the computer only those fields that have been entered, deleted, or changed by the operator. The 3270 keeps track of such modifications and uses that information to select data to send to the computer. If you wish to pass a field into the computer regardless of modification, you may assign the "modified" or "modified data tag (MDT)" attribute. However, you should note that the operator can change the MDT attribute unless you also assign the protected attribute.

You can decide which combination of attributes you want within the limitations specified in the IBM 3270 Component Description. Certain attribute combinations produce additional characteristics. For example, the numeric (limiting keyboard use) and protected (eliminating keyboard use) attributes seem contradictory, but, when specified together, automatically skip the cursor past the field.

You should also be aware that the computer is not limited by attributes. The computer can, for example, place alphabetic information in a field defined as numeric, or protected, or both. The operator does not have such liberty.

If you do not specify any combination of attributes, a field is assumed to have the following attributes:

- Alphameric
- Unprotected
- Displayable (at regular brightness)
- Nondetectable by the selector pen or cursor select

- Not modified
- For extended 3270 data stream
 - Color - Green
 - PS -Nonloadable character set
 - Highlighting - 3270 field attribute defaults

The attribute character for each 3270 data stream except the extended 3270 data stream field uses a single nondisplayed and protected character position on the screen and serves as a visual separation between successive fields.

Example of Field Definition

A typical sign-on procedure illustrates how you might define fields. Figure 2-3 illustrates a simple procedure in which the computer requests the operator to provide his name, location, and serial number.

FIELD 1: "SIGN-ON PROCEDURE"

This field is a heading which the operator should not be able to alter. It is unnecessary for the words "SIGN-CN PROCEDURE" to be returned to the computer when the ENTER key is pressed. This field should be protected, alphameric, displayed at normal intensity, not detectable by the selector pen or cursor select, and not modified. All default attributes can be assumed, except that you must specify this field as protected.

FIELD 2: "PLEASE ENTER ... INFORMATION"

You should specify this field as protected. Remember that the characteristics of a field are determined by the attribute byte at the beginning of the field. Field 1 and field 2 have identical attributes and are adjacent to each other. You may choose to define them separately and use two attribute bytes, or you may choose to omit the attribute byte at the beginning of field 2. In the latter case the two headings combine to become a single field of greater length.

FIELD 3: "NAME:"

This field should be protected, alphameric, not modified, and not detectable by the selector pen. The heading could be displayed at high intensity. Specify the protected and high-intensity attributes (the two deviations from the default attributes).

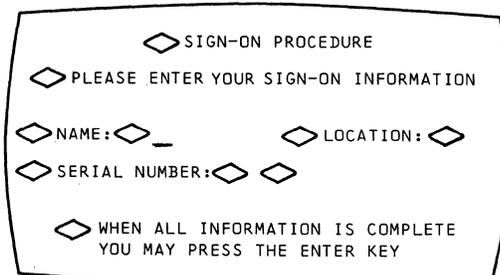


Figure 2-3. Example of Attribute Specification

FIELD 4: The area following "NAME:"

The null area following NAME: is an input area for the operator and must therefore be unprotected. The 3270 marks this field as modified if anything is entered into it, so you should not specify the modified attribute. The default attributes (alphanumeric, unprotected, displayable at normal intensity, not detectable by the selector pen or cursor select, and not modified) apply. Use a default attribute at the beginning of this field.

The maximum number of characters the operator can enter is determined by the length of this field. The length is equivalent to the number of nulls, or available positions on the screen, between the attribute character for field 4 and the attribute character for field 5.

FIELD 5: "LOCATION:"

The attribute byte for this field is the same as that specified for field 3; protected and high intensity should be specified. This attribute prevents the operator from keying a name longer than the maximum length desired. If the name is shorter than the maximum field size, the operator presses the TAB key when the name is complete. The TAB automatically skips the cursor past protected fields, such as this one, and stops at the first character position in which data can be entered (the next unprotected field). In this example, the cursor would be positioned for entry of location. If the operator attempts to key too many characters (a name greater than 17 characters in the example) the cursor is positioned under this attribute for the 18th character. The next keystroke attempts to destroy this attribute but fails to do so because attribute bytes are protected. The keyboard is inhibited, the clicker shuts off, and the "input inhibited" indicator is turned on. The operator's attention is assured since this condition requires pressing the RESET key to continue.

If the attribute byte for this field were omitted, the word "LOCATION:" would become part of field 4 and would be normal intensity and unprotected. This is undesirable since the operator could continue entering name information beyond the desired maximum length and could modify the heading information by entering data in the screen locations occupied by "LOCATION:".

FIELD 6: The area following "LOCATION:"

This field is for operator input and therefore must be unprotected. The rest of the default attribute values apply and so a default attribute may be used. You need specify only that a field is to begin following "LOCATION:". This field ends with the attribute byte at the beginning of field 7, which determines the length of the field.

FIELD 7: "SERIAL NUMBER:"

This field, like "NAME:" and "LOCATION:", should be specified as protected and high intensity. This also limits the location field length to 5 characters. Note that if field 6, the input field for location, were defined as always being a five-character code, field 7, "SERIAL NUMBER:", could be defined as auto-skip to save the operator from having to press TAB after filling in the location code.

FIELD 8: The area following "SERIAL NUMBER:"

The null area following "SERIAL NUMBER:" is an input area for the operator and must be unprotected. It should also be specified as numeric so that if the operator tries to enter alphabetic data in the field (and the keyboard has the Numeric Lock feature), the keyboard inhibits entry of the incorrect character, the keyboard clicker shuts

off, and the "input inhibited" indicator appears to notify the operator of the error. The improper character does not appear on the screen and the correct digit may be entered after the operator presses the RESET key.

The serial number in the example always contains a fixed number of digits and is the last field entered. The maximum length of the field is determined by the location of the attribute for the next field. But the next field in the example is too far away ("WHEN ALL ... KEY").

By placing an additional attribute byte following input field 8, the operator cannot enter a serial number that is too long. If the positions allocated to the serial number are filled, the next keystroke locks the keyboard, as in the name and location fields.

This additional length check is used here because this is the last field to be entered. If you had another field to enter after "SERIAL NUMBER:", it might be more advantageous to omit this length check, as explained in field 9.

FIELD 9: The area between the additional attribute described in Field 8 and "WHEN ALL ... KEY"

By definition, the additional attribute byte you used to delimit the serial number field begins a new field. The protected attribute alone is sufficient for this field, and this attribute limits length for the serial number field. Normally, however, protected (output) fields that follow fixed-length input fields should be defined as protected and numeric. The protected and numeric attribute defines a field as auto-skip. Auto-skip automatically positions the cursor at the location following the attribute byte for the next unprotected field, which is the next place you want to key data. This technique saves keystrokes for the operator. When the operator keys the last character of the preceding fixed-length field, the cursor normally enters the next field, which may be protected. But since the next field is auto-skip, the cursor skips this intervening protected field, and automatically positions itself for entry of the next field, without an extra keystroke.

FIELD 10: "WHEN ALL ... KEY"

This field is a heading which the operator should not be allowed to change. It need not be high intensity and thus it may be defined as protected only. Field 10 does not automatically terminate when the last screen position is reached. The field definition continues from the bottom right screen position to the upper left screen position until the next attribute character is reached. This is called "wraparound." Keep this in mind, particularly if you define the last field on a screen as unprotected!

Since fields 9, 10, and 1 are adjacent to each other (by wraparound) and all have the same attributes, they may be combined into a single field by the omission of attributes before "WHEN" and "SIGN-CN." The result is a single protected field beginning after the input area for serial number, wrapping around the screen, and terminating either at "PLEASE" or at "NAME" if fields 1 and 2 have been previously combined.

Combining fields in the above manner may be convenient but may cause confusion and error if you change the screen layout later. It is a better practice to specify separate fields in all cases.

The panel is completely formatted when the fields are positioned, the attribute bytes are all defined, and the cursor is placed. You must now begin the transition from the visual image, or human-oriented panel, to the detailed data necessary for the 3270 to implement your panel design.

PANEL DESIGN

You can think of a panel as a single 3270 display screen image created by your program. (The terms "screen" or "screen image" or "display image" could also have been used.)

If the terminal operator filled in the information requested in the panel in Figure 2-4, he might receive another panel such as the one shown in Figure 2-5.

An Example of a Sequence of 3270 Panels

Assume you are given the assignment of designing the panels for an accounts receivable application. You are to create the panels that will allow a terminal operator to post a customer payment against his unpaid invoices. The terminal operator will be sitting at a 3270 work station, removing checks and invoice copies from envelopes. If the invoice copies are returned with the check, the terminal operator will for each invoice enter the customer number, payment, and invoice number. If the invoice copies are not returned, the terminal operator will have to find the customer number based on the customer name and then decide which open invoices to apply the payment against. It will be helpful if the operator has some way to add various open invoices to find a combination that totals the payment.

The 1920-character panels that follow show one possible solution.

The first panel in the application is shown in Figure 2-6. If the invoice copies come with the check, the terminal operator can enter the customer number, amount, and invoice number, and press the ENTER key.

This posts the payment against the specified invoice. The terminal operator can then post the next payment and so forth; so long as the customer number and invoice number are known, only panel 1 is displayed.

```
SIGN-ON PROCEDURE  
PLEASE ENTER YOUR SIGN-ON INFORMATION  
  
NAME: _           LOCATION:  
  
SERIAL NUMBER:  
  
WHEN ALL INFORMATION IS COMPLETE  
YOU MAY PRESS THE ENTER KEY
```

Figure 2-4. An Example of a Panel

ITEM	CUST #	NAME/ADDRESS	ITEM	CUST #	NAME/ADDRESS
1	0010341	CAPITAL AVIATION 711 HILLSBOROUGH ST. RALEIGH, N.C. 27611	5	0052693	CAPITOL ELECTRIC 56 STATE ST. MONTPELIER, VT. 05602
2	0028472	CAPITOL BAKERIES 1800 MAIN ST. COLUMBIA, S.C. 29201	6	0084362	CAPITOL FEATHER CO. 899 LOGAN ST. DENVER, COLO. 80217
3	0034020	CAPITOL COLA CORP 1439 PEACHTREE ST. NE ATLANTA, GA. 30309	7	0048729	CAPITAL GLASS CO. 121 STATE ST. ALBANY, N.Y. 12201
4	0041938	CAPITAL DRUG CO. 201 NORTH 9TH ST. RICHMOND, VA. 23219	8	0038492	CAPITOL HOLDING CO. 1609 SHOAL CREEK B AUSTIN, TEXAS 78701

PANEL 2

Figure 2-7. Panel 2, Showing the Results of a Search on a Customer Name

As a result of terminal operator response to Panel 2, Panel 3 (shown in Figure 2-8) displays all open invoices for the identified customer. The terminal operator can now use the selector pen or cursor select to specify the open invoices to which the payment applies. He does this by touching the selector pen to the question mark adjacent to each desired invoice number or positioning the cursor in the invoice number field and processing the cursor select keys; selection is verified immediately by the question mark changing to a > character. To post the payment against the selected invoice numbers, the operator can select APPLY. If, however, the operator cannot easily tell the invoices to which the payment is applied, he can select CALC instead of APPLY.

ACCOUNTS RECEIVABLE						
CUST #	NAME	INVOICE #	DATE	(D)	GROSS	NET
0028472	CAPITOL BAKERIES	? A984632	11/01/71		\$182.50	\$182.50
		? B000312	12/05/71		\$778.00	\$778.00
CHK AMT	\$4,000.00	? B000418	12/07/71		\$98.50	\$98.50
TOT DUE	\$5,358.40	? B000964	12/11/71		\$1,250.00	\$1,250.00
		? B001200	12/21/71		\$682.40	\$682.40
		? B001439	12/25/71		\$395.00	\$395.00
		? B001800	01/11/72	*	\$1,029.75	\$1,009.15
		? B002015	01/15/72	*	\$982.50	\$962.85

MANUAL APPLY
CALC NEXT

PANEL 3

Figure 2-8. Panel 3, Showing the Customer's Open Invoices

Selecting CALC displays Panel 4 (Figure 2-9); this is the same as Panel 3 except that ACCOUNTS RECEIVABLE, which was high intensity in Panel 3, is now normal intensity in Panel 4. A new line with CALCULATOR in high intensity indicates the screen mode and explains the PF keys' functions. The terminal operator can now use the lower right-hand quadrant of the screen as a "scratch pad" to figure out a combination of open invoices that will total the payment check. This use of one part of the screen for a separate function is sometimes called a "split-screen capability."

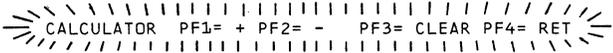
The calculator could be programmed a number of different ways. It could, as our example illustrates, show all invoice numbers selected (shown with > in Figure 2-9) prior to selecting CALC in one column in the CALCULATOR quadrant and in another column show any balance remaining from the check amount after subtracting the selected invoice numbers. In Figure 2-9, Panel 4 is shown as it would appear if the terminal operator had first selected four invoice numbers and then selected CALC. In this example, the selected invoices equal the check amount so .00 is shown as the balance after subtracting the selected invoices.

Panel 4 shows that the CALCULATOR could also allow the operator to key in amounts and add or subtract them from the check amount (pressing PF1 in our example adds keyed-in amounts; PF2 subtracts one keyed-in amount from another). To start over at any point, the operator can press PF3 to clear the calculator quadrant. In our example, the selected invoices equal the check amount, so they can now be posted. But first the terminal operator must leave the CALCULATOR routine by pressing PF4 (RETURN). This displays Panel 5, shown in Figure 2-10.

Panel 5 is the same as Panel 4 except that, with the operator having signaled completion of the CALCULATOR, that word now appears in normal intensity and ACCOUNTS RECEIVABLE once again appears in high intensity. The terminal operator can now, using the selector pen or cursor select, select the invoices against which to apply the payment and then select APPLY to post the payment.

Panel 6, in Figure 2-11, shows the ACCOUNTS RECEIVABLE file for the customer after posting the payment, with the new balance and the total amount applied. To continue to the next customer, the operator selects NEXT and returns to Panel 1.

ACCOUNTS RECEIVABLE						
CUST #	NAME	INVOICE #	DATE	(D)	GROSS	NET
0028472	CAPITOL BAKERIES	? A984632	11/01/71		\$182.50	\$182.50
		> B000312	12/05/71		\$778.00	\$778.00
CHK AMT	\$4,000.00	? B000418	12/07/71		\$98.50	\$98.50
TOT DUE	\$5,358.40	> B000964	12/11/71		\$1,250.00	\$1,250.00
		? B001200	12/21/71		\$682.40	\$682.40
		? B001439	12/25/71		\$395.00	\$395.00
		> B001800	01/11/72	*	\$1,029.75	\$1,009.15
		> B002015	01/15/72	*	\$982.50	\$962.85

MANUAL	APPLY				
CALC	NEXT	CALCULATOR PF1= + PF2= - PF3= CLEAR PF4= RET			

\$778.00	.00
\$1,250.00	
\$1,009.15	
\$962.85	

PANEL 4

Figure 2-9. Panel 4, Showing Use of the Calculator

ACCOUNTS RECEIVABLE						
CUST #	NAME		INVOICE #	DATE (D)	GROSS	NET
0028472	CAPITOL BAKERIES	?	A984632	11/01/71	\$182.50	\$182.50
		>	B000312	12/05/71	\$778.00	\$778.00
CHK AMT	\$4,000.00	?	B000418	12/07/71	\$98.50	\$98.50
TOT DUE	\$5,358.40	>	B000964	12/11/71	\$1,250.00	\$1,250.00
		?	B001200	12/21/71	\$682.40	\$682.40
		?	B001439	12/25/71	\$395.00	\$395.00
		>	B001800	01/11/72 *	\$1,029.75	\$1,009.15
		>	B002015	01/15/72 *	\$982.50	\$962.85
MANUAL	APPLY		CALCULATOR	PF1= + PF2= -	PF3= CLEAR	PF4= RET
CALC	NEXT					
					\$778.00	.00
					\$1,250.00	
					\$1,009.15	
					\$062.85	
PANEL 5						

Figure 2-10. Panel 5, Showing Selection of Invoices after Using the Calculator

ACCOUNTS RECEIVABLE						
CUST #	NAME		INVOICE #	DATE (D)	GROSS	NET
0028472	CAPITOL BAKERIES	?	A984632	11/01/71	\$182.50	\$182.50
CHK AMT	\$4,000.00	?	B000418	12/07/71	\$98.50	\$98.50
TOT DUE	\$5,358.40	?	B001200	12/21/71	\$682.40	\$682.40
NEW BAL	\$1,358.40	?	B001439	12/25/71	\$395.00	\$395.00
SEL INV	\$4,000.00					
MANUAL	APPLY					
CALC	NEXT					
PANEL 6						

Figure 2-11. Panel 6, Showing New Balance after Posting

Not all of the 3270's possibilities are shown in these six panels and not all users will have the selector pen or cursor select; this example was designed only to show what panels are and how the 3270 can be used.

Note that, in the above example, the terminal operator does not see as many panels as the programmer must create; not all panels necessarily appear to the operator in any given application. What the programmer regards as separate panels may appear to the terminal operator as one changing panel.

In the above example, a number of additional panels or variations to the panels shown would be required. For example, if the terminal operator presses an invalid PF key, a variation of the panel would be required to send a message to the operator over the panel presently at his display. In programming panels that are variations of one main panel, it may be useful to assign panel designations (for example, Panel 4A, 4B, and so forth) for variations of Panel 4.

Planning a Sequence of Panels

After an application program has been defined, the information that will be passed between the program and the terminal operator must be defined. This information can be thought of as output panels and input response to panels. Usually, you will be able to approximate the sequence of panels. The exact sequence of output panels often depends on the input response to panels. The following discussion shows one way to define a sequence of panels.

Defining the Purpose of Each Panel

Assuming you have a good understanding of the type of application program (such as data entry, order entry, or inquiry) and the kind of information that must be exchanged and processed (such as customer name, invoices, and check amounts), you can consider which panels come first. Suppose the first panel required is a sign-on panel, as shown in Figure 2-12.

After sign-on, the next panel might allow the terminal operator to choose one of several different applications or procedures that he would use. But what if the name or word entered was not an authorized sign-on? Another panel might tell the terminal operator about this and ask him to reenter a sign-on name. Figure 2-13 illustrates a technique, sometimes called "block diagramming," that may help in laying out a sequence of panels.

Using the Panel Layout Sheet

After block-diagramming the panels in the application or procedure, you are ready to decide on the exact contents of each panel: the fields that will be in the panel, what attributes each field will have, and what words will be displayed in the panel. This can be done on graph paper. The IBM 3270 Information Display System Layout Sheet, GX27-2951 or GX27-0014, is useful for layout.

One of these sheets can be used for each panel. After laying out a sequence of panels, you have a collection of panel layout sheets. Using the information on these sheets and the block diagram showing the relationship between panels, the program can be written to send the panels to a terminal and handle an operator's response to them.

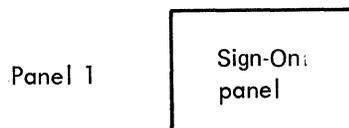


Figure 2-12. Sign-On Panel Block Diagram

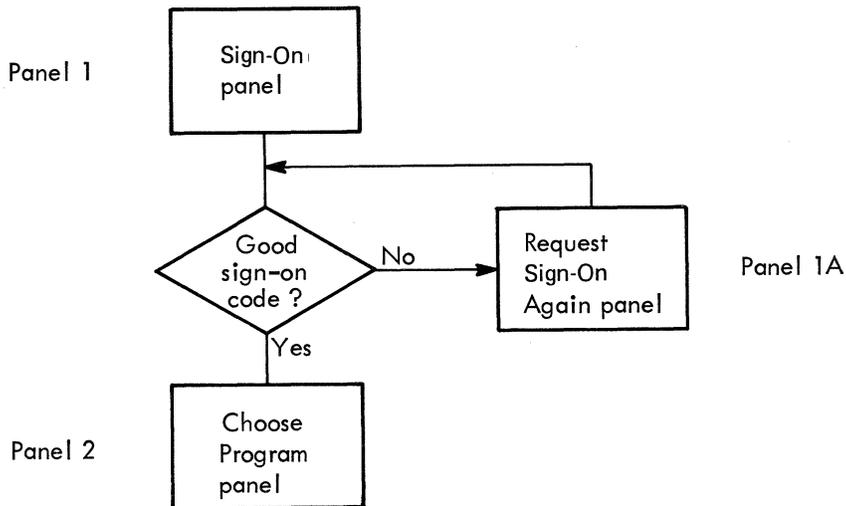


Figure 2-13. Block-Diagramming a Sequence of Panels

An Example of Laying Out a Panel

To lay out a panel, consider the sign-on panel shown in Figure 2-12. You might jot down on a piece of paper the information required for the panel, or you might write it directly on the panel layout sheet. Figure 2-14 shows what the panel part of the layout sheet might look like after you put the text you wanted for your sign-on panel on the layout sheet. A 480-character display is shown here.

Now that you have written out what you want the terminal operator to see, you can define as fields the separate items of displayed text and spaces you are allowing for operator input. Remember that a field is always preceded by an attribute byte. The attribute byte occupies a space on the panel even though it appears as a blank space to the operator. Before deciding the attributes of a field, insert some character such as A on the layout sheet to indicate the space for the attribute byte. As you get used to creating panels, you may want to enter the A at the same time you are laying out the text. You should also show the cursor location on the panel layout sheet to indicate to the operator where to start his response. The cursor position can be indicated by an underscore (_) under the space where you want it to appear, or you might enclose the space or characters in a rectangle. After adding the indications for attribute bytes and the cursor position, the sign-on panel appears as shown in Figure 2-15.

You could have designed the panel as one long field (or even no field at all), but if you did, you would not be taking advantage of the 3270's capabilities. If you designate various items on the panel as fields, each field can have different attributes, as discussed in "What Attributes May Be Assigned to a Field."

For example, you might want the fields NAME:, LOCATION:, and SERIAL NUMBER: to have high-intensity attribute to focus the operator's attention on them, because these fields indicate where the operator enters information. You might want to protect the fields other than the operator input fields so the operator could not erase them; the operator input fields following NAME:, LOCATION:, and SERIAL NUMBER: should be unprotected so the operator can type in information. The operator input field following SERIAL NUMBER: can be numeric to allow some work station editing; the operator would not be allowed to accidentally enter an alphabetic character. Field length can be defined by beginning a new field where you want the previous field to end (in some cases, this new field serves only to give a length attribute to a previous field).

		COLUMN																																																						
		1 - 10										11 - 20										21 - 30										31 - 40																								
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2													
01																																																								
02																																																								
03																																																								
04																																																								
05																																																								
06																																																								
07																																																								
08																																																								
09																																																								
10																																																								
11																																																								
12																																																								
13																																																								

Figure 2-14. Sign-On Panel As Written Out on Layout Sheet

		COLUMN																																																							
		1 - 10										11 - 20										21 - 30										31 - 40																									
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2														
01																																																									
02																																																									
03																																																									
04																																																									
05																																																									
06																																																									
07																																																									
08																																																									
09																																																									
10																																																									
11																																																									
12																																																									
13																																																									

Figure 2-15. Panel Layout, Including Attribute and Cursor Positions

Having decided on these attributes, you can use the columns on the right side of the layout sheet to record the locations and attributes of the fields you have created. Your recording in these columns might appear as in Figure 2-16.

The use of these columns depends on whether the panel designer also codes the panels or only designs them. The information now on the layout sheet can be used to write a line of code that, when sent to the display, displays your panel with its specified field characteristics. The next section, "Data Stream Coding," shows how the panel in this example is coded.

Item 5. LOCATION: The control unit must have two orders for this item which (1) give the starting buffer address (SEA) of the field as R6, C25, and (2) indicate that it is the start of a new field (SF), that it is protected, and that it has high intensity.

Item 6. Input field for operator's location code. This item immediately follows the text of the last item so there is no need to set the buffer address. Write only the SF order to indicate the start of a new unprotected field, and use default attributes.

Item 7. SERIAL NUMBER: This field requires an SBA to location R7, C1, and an SF to begin a new field. The attribute is specified the same as that for Item 5.

Item 8. Input field for serial number. The attribute byte for this input field immediately follows the last character of the previous field so an SBA is not required. The attribute is numeric only.

Item 9. An extra field created to limit the size of the serial number input field. This follows the input field and is protected only. An SEA is required for location R7, C23, for proper placement of the attribute.

Item 10. "WHEN ALL ... COMPLETE." The control unit must have two orders for this item: an SBA order that gives the starting address of R10, C3, and an SF order to indicate that it is the start of a new field. The attribute byte defines a protected field, and the rest of the field attributes take the default values.

Item 11. "YOU MAY ... KEY." All the words from "WHEN ALL" through "KEY" could have been treated as a single item, but 8 blank spaces would have to be sent between "COMPLETE" and "YOU" to position "YOU" properly at R11, C5. Use only the 3 characters required for an SBA order and its associated address, breaking the field into 2 items, to position "YOU" at R11, C5.

Coding the Panel

To write a panel in assembler language so that it can be part of the application program, you must transfer the panel's text and orders to an assembler coding sheet or to any other form you find suitable.

On the coding sheet (and in your program), a panel is represented by a series of assembler IC statements, each with a name to which your program can refer. In the example given below, SIGNPANL is the name of the sign-on panel. When the application program wants to send the sign-on panel to a display unit, it issues an Erase/Write or Erase/Write Alternate command and designates SIGNPANL as the panel for display.

The display orders must be written in the IC statements in the hexadecimal codes listed in Figure 2-21. Thus, SF is represented by 1D, SEA by 11, IC by 13, SFE by 29, SA by 28, and MF by 2C.

Each part of each order must be written in hexadecimal, including the attribute byte that follows the SF order and the buffer address that follows the SEA order. The IBM 3270 Reference Summary, GX20-1878, contains the hexadecimal codes for all the attribute byte combinations and the hexadecimal code for every buffer location in both EBCDIC and ASCII.

Begin coding with the first item on the panel layout sheet: the title, SIGN-CN PROCEDURE. Start with the orders for the panel text, which must always precede the text itself so that the control unit knows what to do with the text. The extended data stream equivalent is written to the right of the 3270 data stream coded statement.

The first order for the title is the SEA order. Figure 2-21 shows that the SEA hexadecimal code is 11, so you write this code in a DC statement as:

DC X'11'

Now look up the R2, C11 address that must follow the SEA order. The EBCDIC address is 40F2 and it follows the SEA code in the DC statement:

DC X'1140F2'

You should also record this statement in the buffer address HEX column to the left of the SEA on the layout form for possible future reference. You may, if you prefer, lock up all the addresses and record them in a similar manner before you begin to write your DC statements. See Figure 2-22 for an example.

The next order for the title is the SF order, which is followed by the attribute byte. Attributes are shown in Figure 2-23. The SF code, 1D, and the attribute code, 60, are read from the table and added to the DC statement, which is then closed with a single quotation mark:

(SF) DC X'1140F21D60' or (SFE) DC X'1140F22901C060'

Order Sequence Order	Byte 1 (Order Code)		Byte 2	Byte 3	Byte 4	Byte n	Byte n+1
	EBCDIC Hex	ASCII Hex					
Start Field (SF)	10	10	Attribute				
Set Buffer Address (SBA)	11	11	Address	Address			
Insert Cursor (IC)	13	13					
Program Tab (PT)	05	09					
Repeat To Address (RA)	3C	14	Address	Address	Char.		
Erase Unprotected to Address (EUA)	12	12	Address	Address			
Set Attribute (SA)	28	NA	Attr type	Attr value			
Start Field Extended (SFE)	29	NA	Number of pairs	Attr type 1	Attr value 1	Attr type n	Attr value n
Modify Field (MF)	2C	NA	Number of pairs	Attr type 1	Attr value 1	Attr type n	Attr value n

Figure 2-21. Buffer Control Orders and Order Codes

Item	Display Printer		Buffer Address		Orders	Prot	No.	High Int	Sel Det	Non-Disp Prt	MDT On
	Row	Col	Dec	Hex							
1	02	11		40F2	SBA SF	AH	✓				
2	04	02		C1F9	SBA SF	AH	✓				
3	06	01		C3C8	SBA SF	AH	✓	✓			
4	06	07			SF IC	AH					
5	06	25		C360	SBA SF	AH	✓	✓			
6	06	35			SF	AH					
7	07	01		C3F0	SBA SF	AH	✓	✓			
8	07	16			SF	AH	✓				
9	07	23		C4C6	SBA SF	AH	✓				
10	10	03		C56A	SBA SF	AH	✓				
11	11	05		C6D4	SBA						

Figure 2-22. Sign-On Procedure Panel Orders and Attributes

Following the DC statement containing the orders for the title is the DC statement containing the text for the title:

```
(SF) DC X'1140F21D60'          or (SFE) DC X'1140F22901C060'
      DC C'SIGN-ON PROCEDURE'      DC C'SIGN-CN PROCEDURE'
```

To code an input field that contains no text, such as the input field for NAME:, write just one DC statement that contains the orders for that field:

```
(SF) DC X'1D4C13'              or (SFE) DC X'2901C04013'
```

1D is the hexadecimal code for the SF order, 40 is the hexadecimal code for an attribute byte that defines an unprotected field (and all other default attributes), and 13 is the hexadecimal code for the IC order.

ATTRIBUTE BYTE BIT DEFINITIONS

ATTRIBUTE						Hex
Prot	A/N	MDT ON	High Intens	Sel Det	Non Disp PRT	
U		Y				40
U				Y		C1
U		Y		Y		C4
U				Y		C5
U		Y	H	Y		C8
U			H	Y		C9
U			-	-	Y	4C
U		Y	-	-	Y	4D
U	N	Y				50
U	N			Y		D1
U	N			Y		D4
U	N	Y		Y		D5
U	N	Y	H	Y		D8
U	N		H	Y		D9
U	N		-	-	Y	5C
U	N	Y	-	-	Y	5D
P		Y				60
P				Y		61
P		Y		Y		E4
P				Y		E5
P		Y	H	Y		E8
P			H	Y		E9
P			-	-	Y	6C
P		Y	-	-	Y	6D
P	S	Y				F0
P	S			Y		F1
P	S			Y		F4
P	S	Y		Y		F5
P	S	Y	H	Y		F8
P	S		H	Y		F9
P	S		-	-	Y	7C
P	S	Y	-	-	Y	7D

S = Skip Y = Yes
 U = Unprotected H = High
 P = Protected N = Numeric

Figure 2-23. Attribute Byte Combinations in Hexadecimal

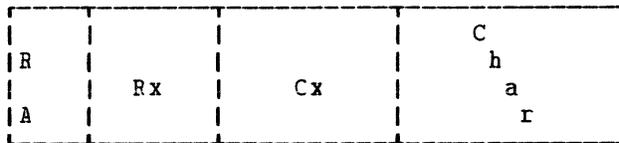
A DC statement can be written as two or more statements. The DC statement above, for example, could be written as:

```
(SF) DC X'1D40'           or (SFE) DC X'2901CC4C'
      DC X'13'             DC X'13'
```

Each item from the panel layout sheet is coded in this fashion. Figure 2-24 shows the complete code required to display the sign-on panel. Except for one control character, it consists entirely of the panel text, preceded by the display orders for that text. (The control character is described under the heading "Write Control Character (WCC).")

Repeat to Address Order

The Repeat to Address (RA) order stores a specified alphanumeric or null character in buffer locations, starting at the current buffer address and ending at (but not including) the specified stop address. The specified stop address then becomes the current buffer address. You specify the stop address immediately following the RA order, just as you specify an address after an SEA order. After the stop address you specify the character that you want repeated. When attribute values are defined by the SA order, those attributes are applied as each repeated character is written into the character buffer. Symbolically this appears as:



RA is 3C in hexadecimal. RA can repeat null characters and can erase selected parts of the screen. You may also use it to repeat any other character. To put a row of asterisks under the last title in the sign-on panel, after the DC statement for "YOU MAY PRESS THE ENTER KEY" you specify an SBA for R12, C1. The RA order should repeat the asterisk character to location R1, C1 (the address after the last *). This is noted on the layout form as shown in Figure 2-25.

PROGRAM		DATE		FUNCTIONS	GRAPHIC	PAGE	OF
PROGRAMMER		STATEMENT		FUNCTION	FUNCTION	CARD	ELECTRO NUMBER
Line	Op	Op	Op	Op	Op	Op	Op
SIGNPANEL	DC	X'CT'	WCC				
	DC	X'1140F21D60'	SBA R2C11	ATT	P		
	DC	C'SIGN-ON PROCEDURE'					
	DC	X'11C1F91D60'	SBA R4C2	ATT	P		
	DC	C'PLEASE ENTER YOUR	SIGN-ON INFORMATION'				
	DC	X'11C3C81DE8'	SBA R6C1	ATT	PH		
	DC	C'NAME:'					
	DC	X'1D4013'		ATT	U		CURSOR
	DC	X'11C3601DE8'	SBA R6C25	ATT	PH		
	DC	C'LOCATION:'					
	DC	X'1D40'		ATT	U		
	DC	X'11C3E01DE8'	SBA R7C1	ATT	PH		
	DC	C'SERIAL NUMBER:'					
	DC	X'1D50'		ATT	ON		
	DC	X'11C4C61D60'	SBA R7C23	ATT	P		
	DC	X'11C56A1D60'	SBA R10C3	ATT	P		
	DC	C'WHEN ALL INFORMATION IS COMPLETE'					
	DC	X'11C6D4'	SBA R11C5				
	DC	C'YOU MAY PRESS THE ENTER KEY'					

Figure 2-24. Assembler Language Statements for Sign-On Panel

Item	Display Printer		Buffer Address		Orders	Prot	No.	High Int	Sel Det	Non-Disp Prt	MDT On
	Row	Col	Dec	Hex							
1	02	11		40F2	SBA						
					SF	ATT	✓				
2	04	02		C1F9	SBA						
					SF	ATT	✓				
3	06	01		63C8	SBA						
					SF	ATT	✓		✓		
4	06	07			SF	ATT					
					IC						
5	06	25		C360	SBA						
					SF	ATT	✓		✓		
6	06	35			SF	ATT					
7	07	01		C3F0	SBA						
					SF	ATT	✓		✓		
8	07	16			SF	ATT		✓			
9	07	23		C4C6	SBA						
					SF	ATT	✓				
10	10	03		C56A	SBA						
					SF	ATT	✓				
11	11	05		C6D4	SBA						
12	01	01			RA	*					

Figure 2-25. Example of RA Order

The order in the example is coded as:

```
DC X'3C4040'
DC C'*
```

If you want to delete a field already on the screen, you can repeat the "null" character to delete it.

Write Control Character (WCC)

The control unit to which the display unit is attached uses the orders to format the panel. One control character for the control unit must be included as the first character of every panel you write: the Write Control Character (WCC). The WCC is a hexadecimal code that provides control information for the control unit and defines printer information for printing panels. The other information in the WCC specifies:

- Whether to sound the audible alarm. The audible alarm is an optional display unit and printer feature that sounds a tone at the display unit upon program request. You can request this function by selecting the appropriate WCC hexadecimal code. If this feature is not installed on a display unit, the request is ignored.
- Whether to restore the keyboard at the end of your panel operation. If this option is requested, the keyboard, which locks when the operator completes a panel operation, is automatically unlocked when the program has finished processing the operator's input. Keyboard restoration means the operator does not have to press the RESET key.

You might not want to unlock the keyboard after each panel is displayed. For example, if you plan to write out another panel before you want to accept input, locking the keyboard prevents the operator from entering data before it is needed. Also, after writing an incorrect panel, you may want to force the operator to press the RESET key to make sure you have gained his attention.

- Whether to reset the modified data tag (MDT). If this option is specified, the attribute bytes of all modified fields are reset. This function resets all input fields to their original (unmodified) status when an operation is completed so they are ready for the next operation.

Each panel written to a display unit or printer must begin with the WCC to identify whether these functions are requested.

The hexadecimal code for each possible WCC combination is shown in Figure 2-26.

Note: Bit number 1 of the WCC is the reset bit and can be used to reset characteristics to their system defined values. Refer to the IBM 3270 Information Display System Component Description manual for more information on the WCC reset.

The sign-on panel data is now complete and can be sent to the display unit.

WCCs for the Display

Start Printer	Sound Audible Alarm	Restore Keyboard	Reset MDTs	Code This Hex Value
No	Yes	Yes	Yes	C7
No	Yes	Yes	No	C6
No	Yes	No	Yes	C5
No	Yes	No	No	C4
No	No	Yes	Yes	C3
No	No	Yes	No	C2
No	No	No	Yes	C1
No	No	No	No	40

WCCs for the Printer

Start Printer	Sound Audible Alarm	Restore Keyboard	Reset MDTs	Code This Hex Value If You Want			
				NL and EM Codes Honored	40-Char. Line	64-Char. Line	80-Char. Line
Yes	Yes	Yes	Yes	4F	5F	6F	7F
Yes	Yes	Yes	No	4E	5E	6E	7E
Yes	Yes	No	Yes	4D	5D	6D	7D
Yes	Yes	No	No	4C	5C	6C	7C
Yes	No	Yes	Yes	4B	5B	6B	7B
Yes	No	Yes	No	4A	5A	6A	7A
Yes	No	No	Yes	C9	D9	E9	F9
Yes	No	No	No	C8	D8	E8	F8

Figure 2-26. WCC Hexadecimal Codes

PANEL DESIGN WITH THE EXTENDED 3270 DATA STREAM ATTRIBUTES

Consider now the example of the Sign-On Procedure panel from Figure 2-4. This panel will be designed for a color display with extended-color capability.

Field 1: "SIGN-ON PROCEDURE"

Again, this field is a heading that should not be altered and not be returned to the computer when the ENTER key is pressed. This field will be protected, alphameric, displayed at normal intensity, not detectable by the selector pen or cursor select, not modified, and blue. The SEA order would still be the same, but now a Start Field Extended order is used. The hexadecimal data stream would be

```
X'11C15A2902CC6042F1'
```

The X'11' (SBA) sets the buffer address to row 2, column 11 (X'C15A'). The Start Field Extended order X'29' indicates two (02) attribute "type and value" pairs. The C0 indicates a 3270 field attribute of 60, which is protected with the defaults mentioned above. The 42 applies a color type with the attribute value F1, indicating this field will be blue.

Field 2: "PLEASE ENTER ... INFORMATION"

This field would have the same characteristics as field 1 and would not have to be defined separately, but, if desired, this field could be defined with different attributes.

Field 3: "NAME"

This field should be protected, alphameric, not modified, and not detectable by the selector pen. Since this field will request information, it should be distinguished from the panel heading; therefore, it will be displayed with high intensity, which will be white on a color display.

The hexadecimal data stream would be X'11C6502901C0E8'. This stream decodes as set buffer address (11) to row 6, column 1 (C650). A Start Field Extended order (29) with one attribute "type and value" pair (01), a 3270 field attribute (C0) of protect and intensify (E8).

Field 4: The Area Following "NAME"

The null area following NAME is an operator input area and must be unprotected. As before, the 3270 marks this field as modified if anything is entered into it. To call the operator's attention to this field, it will also have the Extended Highlighting attribute of underscore, and the cursor will be set at this location for entering the information. The hexadecimal representation would be X'2902C04041F413'. This again would be a Start Field Extended order (29), indicating two "type and value" pairs (02) with the 3270 field attribute (C0) unprotected (40) and the Extended Highlighting attribute type (41) of underscore (F4) with the cursor set (13) at the buffer address for the first space in the unprotected field, and with the defaults mentioned above.

ANALYZING INPUT DATA

The Operator's Response

When the sign-on panel is displayed, the operator responds by entering name, location, and serial number as shown in Figure 2-28. As the operator keys this information, the entered data characters are stored in the display unit's buffer and are displayed as part of the panel. Data that is entered in a nondisplayable field is stored in the buffer, but does not appear on the panel.

When the operator finishes entering the requested sign-on data, he indicates the end of this operation by pressing the ENTER key, which causes an automatic Read Modified command execution and sends the following information to your program:

- An attention code to identify that the ENTER key was pressed
- The address of the cursor's location
- The start buffer address code to identify the next two characters as addresses
- The starting addresses of every modified field, followed by the data in the modified fields

Figure 2-29 shows this sequence of input data, which is explained below.

Attention Identifier (AID)

The Attention Identifier (AID) is a hexadecimal code. By identifying this code, your program can determine in which of several possible ways the operator contacted the program and determine what request is being made. For example, pressing the ENTER key requests "Please enter this data."

The AID code is always the first code received from the display unit by your program. The hexadecimal codes for all AID codes are shown in Figure 2-30.

For a Read Modified, the AID code is followed by the cursor address, which is the hexadecimal code for the row and column location of the cursor when the operator contacted your program.

```
SIGN-ON PROCEDURE
PLEASE ENTER YOUR SIGN-ON INFORMATION
NAME: JOHN SMITH      LOCATION: BOSTN
SERIAL NUMBER: 963981
WHEN ALL INFORMATION IS COMPLETE
YOU MAY PRESS THE ENTER KEY
```

Figure 2-28. Sign-On Panel with Operator's Input

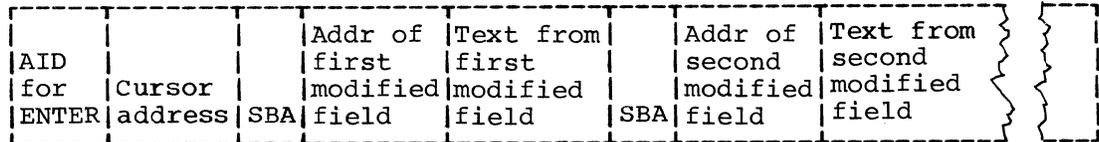


Figure 2-29. Input Data Sequence

ATTENTION IDENTIFICATION (AID) CONFIGURATION

AID VALUES FOR TEXT READ

Graphic Character	EBCDIC Hex	Operator Action
-	60	No action by display operator
	88	No Action by display operator (Inbound Structured Field Identifier)
Y	E8	No action (printer)
'	7D	ENTER key depressed
1	F1	PF key 1 depressed
2	F2	PF key 2 depressed
3	F3	PF key 3 depressed
4	F4	PF key 4 depressed
5	F5	PF key 5 depressed
6	F6	PF key 6 depressed
7	F7	PF key 7 depressed
8	F8	PF key 8 depressed
9	F9	PF key 9 depressed
:	7A	PF key 10 depressed
#	7B	PF key 11 depressed
@	7C	PF key 12 depressed
A	C1	PF key 13 depressed
B	C2	PF key 14 depressed
C	C3	PF key 15 depressed
D	C4	PF key 16 depressed
E	C5	PF key 17 depressed
F	C6	PF key 18 depressed
G	C7	PF key 19 depressed
H	C8	PF key 20 depressed
I	C9	PF key 21 depressed
c	4A	PF key 22 depressed
-	4B	PF key 23 depressed
<	4C	PF key 24 depressed
=	7E	Immediately detectable field selected
0	F0	TEST REQUEST key depressed
W	E6	Data transferred from card reader

AID VALUES FOR SHORT READ

—	6D	CLEAR key depressed (screen cleared)
%	6C	PA1 key depressed
>	6E	PA2 (cancel) key depressed
,	6B	PA3 key depressed

Figure 2-30. Attention Identifiers (AIDs) in Hexadecimal Codes

Input Data

All the modified fields from the panel follow the AID code and the cursor address. A modified field is any field whose attribute byte has the MDT on. A modified field can be one that was modified by the operator or one that was defined by you in your program with the MDT on in its attribute byte.

When any character location in an input field is modified by the operator, the MDT in the attribute byte for that field is automatically turned on. An input field is not necessarily a modified field. If the operator made no entry in the SERIAL field, for example, only his name, location, and the date would be sent as modified fields to your program.

The display unit sends all the data in a modified field except nulls. When an operator finishes an operation, the display unit reads through the buffer for every attribute whose code indicates its MDT is on. Each time one is found, the display unit provides an SBA code and the starting address (the attribute byte's address plus one) of the modified field. The SEA code identifies to your program that an address follows. It is the same X'11' code that you coded in your panel to identify the starting locations of the panel's text.

SEA Codes

SEA codes identify the incoming data by cross-referencing it to the correct input field.

For the sign-on panel, your program knows that row 6, column 8 (X'C34F') is the start of the name input field. When it receives the first SBA code (X'11'), it checks the address that follows to see if it is (X'C34F'). If it is, your program knows the text that follows it (until the next SBA code) is the operator's name and can process the input accordingly.

The first part of the input from the sign-on panel is as follows:

7D	C4	C6	11	C3	4F	J	O	H	N	S	M	I	T	H	...
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	-----

The hexadecimal codes are:

- 7D: The AID code for the ENTER key (see Figure 2-30).
- C4C6: The cursor address R7, C23. The cursor is at the next character location after the entered serial number.
- 11: The SEA (Set Buffer Address) order code which tells the program that the next 2 characters are addresses. (See Figure 2-21.)
- C34F: The location (R6, C8) where the following text is located on the panel.
- JOHN SMITH ...: The first modified field containing the operator's name.

PROGRAM ATTENTION KEYS

Program Access (PA) Keys

Each 3270 keyboard has at least one program access (PA) key that the operator can use to request program attention without sending any input data.

The AID codes for the PA keys are shown under a separate heading in Figure 2-30, because they are not followed by input data even though there may be modified fields on the panel when a PA key is pressed. All four short read codes consist of just the AID code.

Your program should use these keys for operator requests for immediate action such as trouble alerts or requests for termination. For example, the assignment of several PA keys might be:

- PA1: Terminate current application
- PA2: Return to starting (master) panel
- PA3: Explain system message

Program Function (PF) Keys

Program function (PF) keys are a keyboard feature. Your program defines the function that each key requests when it is pressed by the operator.

There is a separate AID code for each PF key so that your program can quickly identify which key was pressed and consequently which function was requested. When a PF key is pressed, all modified fields on the panel and their addresses are sent with the AID code and cursor address, the same as the ENTER key. For this reason, a PF key can be a valuable time-saving device for the operator. For example, the assignment of several PF keys might be:

- PF1: Return to previous panel
- PF2: Clear (without using data) and repeat current panel
- PF3: Set up next panel
- PF4: Page forward
- PF5: Page backward
- PF6: Return to page #1

SELECTOR PEN AND CURSOR SELECT INPUT AND OUTPUT

Positioning data for selector pen (optional feature) or cursor select (basic feature on the 3278 and 3276) use and setting the attribute characters is the same as for any other type of data, but the select function has additional data-stream requirements.

Selector Field Format

A field for selector pen operations must be defined as shown in Figure 2-31. The cursor select does not require the three part character that must precede the selector pen field, although they can be present. Also, the cursor selection can be on any character in the field.

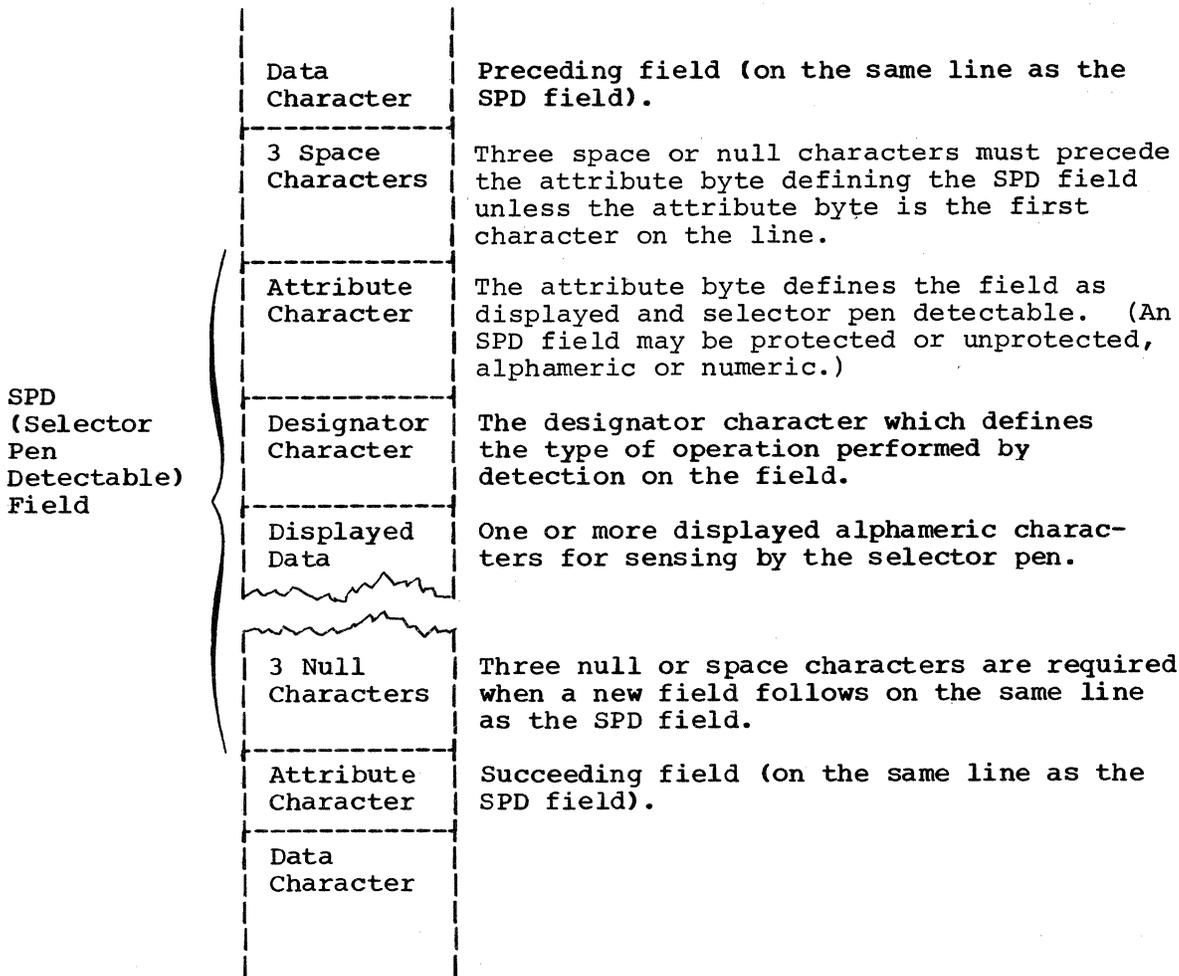


Figure 2-31. Definition of Field for Selector Pen Operation

The attribute byte, the designator character (described in the next section), and displayed alphameric characters must be on the same line. If the field is longer than one line, only those characters on the same line as the attribute byte can be detected by the selector.

Designator Characters

Designator characters define three types of selector fields: selection and two types of attention. Each type of field performs a different operation.

The selection field is defined by a question mark (?) designator character. When the selector pen detects a selection field, the MDI bit in the attribute byte for that field is set in the display buffer. Also, the designator character is automatically changed on the screen to a greater-than (>) sign to provide a visible indication to the operator that the detection was successful. If a mistake was made and the operator again detects on that same field, the > reverts to a ? and the MDI is reset. The first type of attention field is defined by a space or null designator character. Probing an attention field or selecting it with the cursor is similar to using an ENTER key. The input

our example, transmits only the addresses of the fields in which the MDT was set unless you are using a 3274 or 3276 control unit in which case the address and data are transmitted.

In the example, if you pick RED and 4 DCOR the symbolic input would appear as follows:

Pen								
A	Cursor	S			S			
I	ADDR	E	R3	C10	B	R4	C26	
D		A			A			

Shortening transmissions by eliminating unnecessary data requires some caution. If you design a panel requiring both pen selection and keyboard entry, do not put an attention designator (space or null) on the panel. An attention designator after keyboard entry transmits only the address of the keyboard input field and causes the loss of its contents. Not having an attention designator on the panel assures you that an ENTER or PF key will be used and the modified field contents will be transmitted (and the words "RED" and "4 DOOR" in the example).

THE RELATIONSHIP OF ONE DATA STREAM TO ANOTHER

The examples used so far have assumed that you started with a blank screen and that you built the entire panel into your data stream with ERASE or WRITE commands. This approach may lead to tedious work and lengthy data streams, which you can avoid if the panel you wish to display differs only slightly from the one that is presently displayed.

MODIFYING EXISTING PANELS

Suppose the displayed panel is the sign-on panel in the previous sections. If the operator keys an invalid serial number, you may wish to notify him of his error and request reentry of the serial number field only. You could create a new error message panel, write it to the display, require that the operator acknowledge its receipt, create a special serial number entry panel, write it, and finally read the corrected serial number. A better way might be to use the existing sign-on panel.

After the operator has keyed the data and it has been read into the computer, the screen appears as shown in Figure 2-33.

You would like the screen to look like Figure 2-34.

Most of the information you want displayed is already there. An Erase/Write or Erase/Write Alternate command would clear the screen and require writing a data stream containing all the information for the new panel. You could use a Write command which modifies existing data in the 3270's buffer.

To change the panel in Figure 2-33 to look like Figure 2-34, you would:

1. Position the cursor at R7, C17;
2. Replace the message beginning at R10, C5 with the error message;
3. Change the attribute at R10, C4 to high intensity for the error message.

		COLUMN																																																			
		1 - 10										11 - 20										21 - 30										31 - 40																					
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2										
01																																																					
02																																																					
03																																																					
04																																																					
05																																																					
06																																																					
07																																																					
08																																																					
09																																																					
10																																																					
11																																																					
12																																																					
13																																																					

Figure 2-33. Modifying an Existing Panel -- Basic Panel

		COLUMN																																																						
		1 - 10										11 - 20										21 - 30										31 - 40																								
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2													
01																																																								
02																																																								
03																																																								
04																																																								
05																																																								
06																																																								
07																																																								
08																																																								
09																																																								
10																																																								
11																																																								
12																																																								
13																																																								

Figure 2-34. Existing Panel with Error Message

To do this the right side of your panel layout for the error panel might (in abbreviated form) look like Figure 2-35.

- ITEM 1. Repositions the cursor to R7, C17.
- ITEM 2. Changes the attribute at R10, C4 to protected and high intensity. If the designer of the sign-on panel had combined the original field at this location with the previous field, the field "SIGN-CN PROCEDURE," and the following field by omitting the attributes at R10, C4, R2, C11, and R4, C2, (as you saw under the discussion of attributes) the result would be undesirable. The attribute placed at R10, C4 would begin a new field. This would not affect the preceding field but, by wraparound, would cause "SIGN-ON PROCEDURE" and "PLEASE ... INFORMATION" to be high intensity even though they were neither intended to be so, nor were they rewritten. For this reason you should adhere closely to the "Field Concept" and not combine fields unless necessary for efficiency; if you must combine fields, be very careful to avoid undesired results.
- ITEM 3. Repositions the data flow to correctly place the second line of the error message. 3 characters are used instead of 6 null characters.
- ITEM 4. Repositions the data flow for the third line of the error message.

Since there are two different types of Write commands for the 3274 and 3276, you must tell the I/C portion of your program which type to use for the data stream. You may want to indicate the type you want in a comment in the data stream. It is suggested that you establish some convention for indicating command selection by discussing it at your installation with the people responsible for the I/O portion of the program.

Item	Display Printer		Buffer Address		Orders	Prot	No.	High Int	Sel Det	Non-Disp Prt	MDT On
	Row	Col	Dec	Hex							
1	07	17			SBA IC						
2	10	04			SBA SF						
					"LINE 1 OF ERROR MESSAGE"						
3	11	05			SBA						
					"LINE 2 OF ERROR MESSAGE"						
4	12	05			SBA						
					"LINE 3 OF ERROR MESSAGE"						

Figure 2-35. Panel Layout Changes for Error Message (Keyed to Text)

Write Control Character (WCC)

When the operator presses the ENTER key after filling in the sign-on panel, the keyboard automatically locks, as it always does after an operator-initiated input operation. One of the functions of the Write Control Character, which was also discussed under "Coding the Panel," is to enable the keyboard. You should now decide if you want the WCC at the beginning of the error panel data stream to enable the keyboard for the operator. While it is normal to enable the keyboard at this point, you may not want to do it here. It might be better for the operator to press the RESET key, calling further attention to the error panel. (See the discussion of the Copy Function for additional information for the 3274 and 3276 control units.)

In Figure 2-33, assume that the operator now keys "9" and presses the ENTER key. The "9" corrects the original entry error and the serial number field now reads "963981". What goes into the computer? The prior discussion of input data streams shows the basic format, but which fields can you expect? You know that the serial number input field will be received in its entirety, since keying the "9" caused the 3270 to turn on the MDT for this field, and any field which has been modified is transmitted in its entirety (except nulls).

The input field MDTs for NAME, LOCATION, and SERIAL NUMBER were all turned on by the data entered into those fields in the sign-on panel. While an Erase/Write or Erase/Write Alternate resets all MDTs, a Write does not; therefore, if you do not reset them, all 3 input fields are returned to the computer. Because not all of them have changed, all 3 should not return to the computer. You may specify in the WCC that all MDTs in the device are reset "off" or "not modified" (you should do so here).

You may also want to sound the audible alarm, if you have one, with the error panel. A WCC to reset the keyboard, reset all MDTs, and sound the alarm is defined as IC X'C7' (see Figure 2-26). You can now use the Write command to change the sign-on panel into the error message panel.

CAUTION: As you have seen, the Write command allows you to modify an existing screen image while retaining all or a portion of the information already displayed. With the Write command, you can treat the 3270 as a typewriter-type terminal and write your panel line by line or field by field. Using multiple Write commands to create a panel, while technically possible, may create problems.

The operator might start keying data into the panel before you have finished writing it all to the screen. You can prevent this problem by not enabling the keyboard (see WCC above) until the last Write in the series.

Using successive Write commands to accomplish what one Write command can do is an inefficient use of the communication line on remote 3270s, and unnecessary I/O overhead on local 3270s. In addition, in both local and remote use, successive Write commands without an intervening READ may result in a "blinking" effect while you build up the panel. "Blinking" may be annoying to the operator. (The 3274 and 3276 control units update screens without the blinking screen.)

Wherever possible, use a single Write command to avoid the inconveniences noted above.

Erase Unprotected to Address

The error panel shown in Figure 2-34 displayed the erroneous serial number. All the operator had to do was key over the incorrect digits. This may sometimes be confusing. You might instead want to erase only the serial number input field as shown in Figure 2-36.

Begin again with the desired WCC. Place the cursor at R7, C17 with an SEA to R7, C17, followed by an IC order. To erase what was entered in the serial number input field, use the Erase Unprotected to Address order, or EUA (watch the sequence of these letters so you do not confuse them with EAU, which is discussed next). EUA inserts nulls (erases all unprotected positions) from the current buffer address up to, but not including, the specified stop address. It will also set any character attributes of the nulled characters to X'00'.

The specified stop address then becomes the current buffer address. The format of the order is similar to an SBA; the code for the order itself (X'12' for EUA) is immediately followed by a row and column address.

At the first position to be erased (a result of prior operation) you should include an EUA order. For a terminating address, you may use R7, C23 (the first position after the last to be erased). There is a better stop address, however. Since EUA only erases unprotected fields, and since the field beginning at R7, C23 is protected, it can be included in the range covered by the EUA. If R10, C4 is used as the stop address, nothing additional is erased, but you can then write the next attribute without using an SEA, saving three characters of transmission (see Figure 2-37). The current buffer address is the stop address. Any data or SF order that follow go into the buffer at this address.

EUA erases all unprotected fields within its range and can erase multiple fields. Suppose you wanted all three input fields erased on the error panel, as shown in Figure 2-38.

First place the cursor at R7, C17, then "back up" with an SEA to R6, C8 (the name input field) before issuing the EUA to R10, C4 (see Figure 2-39).

		COLUMN																																												
		1 - 10								11 - 20								21 - 30								31 - 40																				
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2			
01																																														
02																																														
03																																														
04																																														
05																																														
06																																														
07																																														
08																																														
09																																														
10																																														
11																																														
12																																														
13																																														

Figure 2-36. Error Message Panel with Serial Number Field Erased

Item	Display Printer		Buffer Address		Orders	Prot	No.	High Int	Sel Det	Non-Disp Prt	MDT On
	Row	Col	Dec	Hex							
1	07	17			SBA						
					IC						
	10	04			EUA						
2					SF	AH	✓		✓		
	"LINE 1 OF ERROR MESSAGE"										
					.						
					.						
					.						

Figure 2-37. Example of EUA Use

ROW	COLUMN																																							
	1-10										11-20										21-30										31-40									
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
01																																								
02																																								
03																																								
04																																								
05																																								
06																																								
07																																								
08																																								
09																																								
10																																								
11																																								
12																																								
13																																								

Figure 2-38. Sign-On Panel with Three Erased Fields

Item	Display Printer		Buffer Address		Orders	Prot	No.	High Int	Sel Det	Non-Disp Prt	MDT On
	Row	Col	Dec	Hex							
01	07	17			SBA						
					IC						
	06	08			SBA						
	10	04			EUA						
					SF	AH	✓		✓		
					.						
					.						
					.						

Figure 2-39. Erasing Multiple Fields with EUA

You could have started at R6, C8 with an SEA to R6, C8, followed by the EUA to R10, C4. However, sometime later in the data stream you would have had to "back up," probably with an SEA to insert the cursor.

Erase All Unprotected Command

In the preceding example, you wanted to erase all unprotected data, reposition the cursor, and add some new titles to the sign-on panel to make it an error panel. The Erase All Unprotected command:

- Clears all unprotected character locations and associated character attributes to nulls
- Resets MDTs in all unprotected fields
- Unlocks the keyboard
- Resets the AID (see "Program Attention Keys")
- Repositions the cursor to the first character of the first unprotected field

This command appears to do what you want (it even does what the WCC would have done), but it does not write any data to the screen. You could issue an Erase All Unprotected command before the Write command. Then you would just write the new titles in their proper positions. You have then issued two commands to create one panel. What, then, is EAU for? It logically resets the panel for repetitive input using the same panel. Do not use EAU to change panels.

Data Entry Example: You can use the EAU command to change a sign-on panel slightly and make it a data entry panel. Then the operator just keys in NAME, LOCATION, and SERIAL NUMBER for the first employee. If an error is made, an error panel is shown. If there is no error, you may want to clear the input, reset the MDTs, unlock the keyboard, and reposition the cursor.

The data entry panel might appear as shown in Figure 2-40.

		COLUMN																																								
		1 - 10										11 - 20										21 - 30										31 - 40										
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
01																																										
02																																										
03																																										
04																																										
05																																										
06																																										
07																																										
08																																										
09																																										
10																																										
11																																										
12																																										
13																																										

Figure 2-40. Example of Data Entry Panel

The operator keys JOHN SMITH, presses TAB, keys BOSTN, presses TAB, keys 963981, and presses ENTER (Figure 2-41).

You simply send the 3270 an FAU command to unlock the keyboard. The operator then sees the same panel as in Figure 2-40. The operator may now key data for the next employee. You have used your knowledge of what is displayed already to arrive at the next panel or re-create the present panel.

Repetitive Output

In the data entry example you used one panel repetitively for input of employee information. You can reverse the requirement and design an employee data screen. For this example, assume the application is inquiry with "browsing" capability. Assume also that the operator has previously used another panel to request the information for employee number 963981. The display might appear as shown in Figure 2-42.

At the bottom of the panel the operator is instructed to use the PA1 key to see the next employee page, probably number 963982. The PA2 key is assigned to page backwards. Remember, PA keys are assigned by the program. Program access keys cause a short transmission; they do not even transmit the contents of changed fields. For an inquiry and browsing application, there should be no input. The PA key assures there is no input even if the operator changes one of the unprotected fields, so its use is preferred to the ENTER or PF keys.

		COLUMN																																																	
		1 - 10										11 - 20										21 - 30										31 - 40																			
		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2								
01																																																			
02																																																			
03																																																			
04																																																			
05																																																			
06																																																			
07																																																			
08																																																			
09																																																			
10																																																			
11																																																			
12																																																			
13																																																			

Figure 2-41. Data Entry Panel with Entered Data

ROW	COLUMN																																									
	1 - 10										11 - 20										21 - 30										31 - 40											
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
01																																										
02																																										
03																																										
04																																										
05																																										
06																																										
07																																										
08																																										
09																																										
10																																										
11																																										
12																																										
13																																										

Figure 2-42. Employee Data Panel

Program Tab

The input fields in the previous examples are output fields in this example. You could designate them as protected, but if you did, you could not use another 3270 function called "Program Tab." The Program Tab (PT) order advances the current buffer address to the address of the first character location of the next unprotected field. When the PT order immediately follows an alphanumeric or null character (not another order) in the WRITE data stream (other than the character specified by the Repeat to Address order, which is discussed earlier), it also inserts nulls in all the character positions from the current buffer address to the end of the current field. The PT order can be used to page through the employee data file.

When the operator is ready to view the information for the next employee, he presses the PA1 key. Since you want to modify only the present panel, not erase it or blank the unprotected fields, you request a WRITE command with a WCC to unlock the keyboard. Because you are not sure of the present buffer address, you might begin with an SBA order to R6, C8 followed by the next employee name from the disk file - JOE AMES. Because this name contains fewer characters than JOHN SMITH, the screen would look like this if you did not clear the remainder of the field:

03																																									
04																																									
05																																									
06																																									
07																																									
08																																									

CHAPTER 3. SCREEN MANAGEMENT

A screen management program module is a set of subroutines physically separate from application programs and from the telecommunications management program module of an online 3270 system. Figure 3-1 illustrates this relationship.

Support functions in a screen management program may reduce the amount of detail work required by the application programs, and effectively use the features of the 3270. The separation of screen management from the other programs also allows screen management to be modified with little or no impact on application programs or the telecommunications management programs.

Screen management might include:

- Decoding input data streams.
- Dynamic building of output data streams.
- Generating multiple I/O requests to the Line Control Module based upon a single request from an application program (that is, WRITE then READ).
- Automatic paging; the application program passes multiple pages to screen management, which asks the line control module to write a particular page to a display, depending on the display operator's request.
- Automatic copying (providing a hard copy of a display image).

The COPY function supports data movement between any types of device attached to the same control unit: display to display, display to printer, printer to display, and printer to printer. To prevent copying information from an unauthorized device, the control unit provides a program-controlled copy-lock for devices attached to it. If the first position of a device buffer contains an attribute character with the protected option, the control unit rejects any attempt to copy from that device. (The differences in copy functions for the various control units are described at the end of this chapter.)

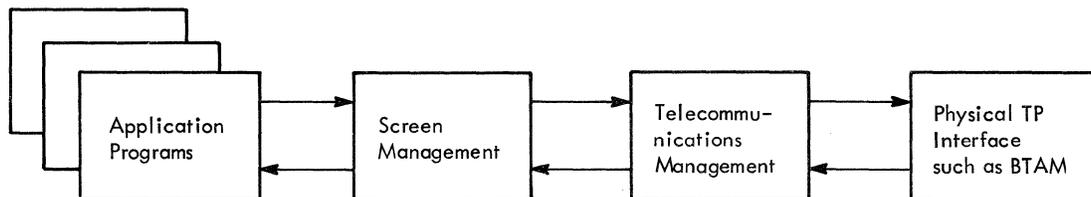


Figure 3-1. Relationship of Screen Management to Telecommunications Management and Application Program

DECODING AND GENERATING DATA STREAMS

The data streams sent between application programs and the 3270 contain unique orders that request particular operations by the 3270 displays and printers. Generalized subroutines can be written to assist the application programmer's interface with the 3270 system, and an interface can be built to simplify online programs.

This chapter discusses several approaches to the development of a screen management module whose functions can be used by the application programmer to prepare output data streams and to decode input data streams. The approaches demonstrate how some 3270 device-dependent considerations can be removed from the application programmer's responsibility. The different techniques for 3270 input or output data stream manipulation can be used in various combinations to suit the needs of the installation.

This discussion assumes that the device management routines (line control) discussed in Chapter 4 make the local and remote 3270 transparent to the application program. Therefore, discussion of data streams in this chapter ignores all header data in the input stream up to and including the AID character, and all header data in the output stream up to but not including the Write Control Character (WCC).

DECODING READ MODIFIED INPUT DATA STREAM

A Read Modified command for a display station with a formatted screen (a screen with at least one attribute character defined) produces a data stream consisting of the data from each field whose modified data tag has been turned on (either by program control or by data entered in the field). Each transmitted data field is preceded by the 3270 buffer address where that data is located on the display. The order of the fields transmitted from the screen is from left to right for each line, starting at the top of the screen and ending at the bottom of the screen. All null characters in a transmitted field are stripped out by the control unit during transmission.

The data stream, ignoring the header information up to and including the AID character, appears as:

S	A	A	DATA	S	A	A	...
B	1	2		B	1	2	
A				A			

If the data entered in a field is variable-length or if a field can be skipped by the terminal operator, the data from a particular field on a given panel can appear in a different location within the data stream for each set of operator input. A Read Modified command produces a variable-length data stream of fixed-length fields and variable-length fields concatenated together.

Each two-character screen address in the data stream is immediately preceded by a Set Buffer Address (SBA) order. The detection of each SBA order in the data stream identifies the next two characters in the stream as a 3270 screen address and also indicates the end of the preceding data field. The System/360 and System/370 translate and test instruction (TRT) can be used to scan the data stream and to stop at each main storage address containing an SBA order. If the detected main

storage address of the current SBA order is known, the following calculations can be performed for a given data stream:

```
SBA (1), ADD (1A), ADD (1E), DATA FIELD (1),  
SBA (2), ADD (2A), ADD (2E), DATA FIELD (2),  
SBA (3),
```

The numbers in parentheses are used as subscripts to provide unique identification:

- The length of data field(1) = (Address of SEA(2) - Address of SBA (1)) -3.
- The two-character 3270 screen address of data field(1) can be found at the address of SEA(1) +1.
- The length of data field(2) = (Address of SEA(3) - Address of SBA (2)) -3.
- The two-character screen address of data field(2) can be found at the address of SEA(2) +1.

The two-character 3270 screen address as it appears in the input stream does not provide a direct decimal or binary numeric value that can be used to calculate the relative position in the 3270 buffer from which the data was read. However, you can use the following routine to convert the 3270 address as it appears in the input data stream to a binary value which directly indicates the position (relative to zero) of the data in the 3270 buffer.

Assume that R3 contains the address of SEA(1) and that R4 and R5 are work registers. R5 will contain the result at the end of the routine.

```
ALDCNVRT EQU *  
SR R4, R4 CLEAR WORK REG  
SR R5, R5 CLEAR WORK REG  
IC R4, 0(R3) GET FIRST ADDRESS CHAR (ADD (1A))  
N R4, = F'63' TURN OFF ALL EITS EXCEPT LAST SIX  
IC R5, 1(R3) GET SECOND ADDRESS CHAR (ADD (1B))  
N R5, = F'63' TURN OFF ALL EITS EXCEPT LAST SIX  
SLL R4, 6 SHIFT FIRST ADDRESS SIX BITS TO THE LEFT  
AR R5, R4 ADD THE RESULTS TOGETHER
```

By using the above technique, several approaches may be developed to a general purpose subroutine that decodes the variable field length data stream for the application program, and returns the data in a more easily processed format.

Nonselector Pen or Non-Cursor Select Data Streams

DISPLAY BUFFER IMAGE TECHNIQUE: By using the READ BUFFER command you can use the display buffer image technique to return to the application program a main storage buffer area the same size as the display buffer (480, 960, 1920, 2560, 3440, 3564). The data read from the display is placed in the same relative position in the main storage buffer as it occupied in the display buffer, with all other positions in the returned buffer cleared to spaces.

For this technique, use the TRT instruction and the 3270 address conversion routine. You must know the relative locations in the display buffer where data can be entered by the operator, so that the decoded buffer can be processed when returned by the mapping subroutine. The completed layout sheet for the panel in which the operator enters data will give you the required addresses relative to the respective buffers.

Using the image technique, all data received from the 3270 is left-justified in its respective fields. This has no effect on fixed-length fields, variable-length alphanumeric fields (which are normally left-justified), or omitted input fields. However, you must be aware of variable-length numeric fields where the operator can omit leading zeros.

Although the image technique requires little main storage for the mapping subroutine, main storage can be wasted if the routine returns a complete buffer with little data. To help overcome this problem, the decoding routine can pass back to the application program, a field at the beginning of the buffer. The field indicates the total length of the buffer, which allows the decoding routine to use a buffer area just large enough to accommodate the relative address of the last data field read.

MAPPING FROM A TABLE OF REQUIREMENTS: This mapping technique requires a table assembly for each unique input panel that the mapping subroutine decodes for the application program. The table provides information to the subroutine so that the input data stream in one main storage buffer can be decoded a field at a time and moved to a specified relative offset in another main storage buffer (the target buffer) according to the directions assembled in the table. The preassembled table could be used to specify the following information to the mapping subroutines:

1. The 3270 buffer address preceding each field, which could be read from a particular panel. This is the buffer address as it appears in the data stream which corresponds to the first data position in a field, not to the buffer location of the attribute character that defines the field. Any data fields in the 3270 input stream that do not have a matching buffer address in the table would be ignored by the typical mapping routine using the table approach.
2. An offset relative to zero that provides the starting position of each field in the target buffer. This information allows the application programmer to order the fields in the target buffer in a sequence that may or may not agree with the field sequence in the transmitted data stream.
3. A value that indicates the maximum length of each field in the target buffer. This information allows the mapping routine to truncate data stream fields that are too long for the target fields. The maximum field length value is also required if the mapping routine supports right-justification of fields during mapping.
4. A flag byte consisting of bit switches that could indicate:
 - Whether left justification with low-order blank padding is requested
 - Whether right justification with high-order zero fill is requested

- Whether the field should be translated to ensure uppercase characters only
- Any additional functions the installation wishes to implement in the mapping routine

Figure 3-2 shows some typical logical contents of the table. The order of the elements within each table entry is optional.

Assume that you map the following input data stream in hexadecimal using the sample table in Figure 3-2:

```
1140D4F1F2F31140E8818283848511C1C6E385A7A3
```

The following target buffer, also in hexadecimal, would be returned to the application program:

```
C1C2C3C4C54C4C404040F0F0F1F2F3E385A7A34C40
```

This approach to mapping makes the application program's input processing routine device-independent.

Instead of the mapping table, you could write a macro instruction to prepare the table; the macro would convert written requests into the proper machine language constants.

A typical format for a macro instruction to build the sample table shown in Figure 3-2 might be:

```
MAP NAME=TABLE,MCDEL=2
MAP ADD=(1,21),OFFSET=11,MAXL=5,JUST=RIGHT
MAP ADD=(1,41),OFFSET=1,MAXL=10,JUST=LEFT,TRAN=YES
MAP ADD=(1,71),OFFSET=16,MAXL=6,JUST=LEFT
```

Note: The ADD parameter specifies the 3270 buffer in row and column notation relative to one. For example, buffer position zero equals row 1, column 1. The offset values are expressed relative to one. The macro instruction can have default options; for example, if JUST=RIGHT is not specified, JUST=LEFT can be assumed.

TABLE	DS 0H	
ENTRY1	DC X'40D4'	ACTUAL 3270 ADDRESS FOR POS 20
	DC H'10'	RELATIVE OFFSET IN TARGET BUFFER
	DC HL1'5'	MAX FIELD LENGTH OF TARGET FIELD
	DC X'80'	RIGHT JUSTIFY, NO TRANSLATE FLAG
ENTRY2	DC X'40E8'	ACTUAL 3270 ADDRESS FOR POS 40
	DC H'0'	RELATIVE OFFSET IN TARGET BUFFER
	DC HL1'10'	MAX FIELD LENGTH OF TARGET FIELD
	DC X'40'	LEFT JUSTIFY, TRANSLATE FLAG
ENTRY3	DC X'C1C6'	ACTUAL 3270 ADDRESS FOR POS 70
	DC H'15'	RELATIVE OFFSET IN TARGET BUFFER
	DC HL1'6'	MAX FIELD LENGTH OF TARGET FIELD
	DC X'00'	LEFT JUSTIFY, NO TRANSLATE FLAG
ENDOLIST	DC X'FF'	END OF LIST INDICATOR

Note: 3270 buffer addresses in the table are shown relative to buffer location zero; relative offsets in the target buffer are shown relative to zero.

Figure 3-2. Table of Requirements

The following example shows the logic flow for a table-driven input mapping technique:

1. Find the 3270 buffer address of a data field to be processed in the input data stream using the TRT instruction.
2. Determine the length of the data field in the data stream using the techniques discussed in this section.
3. Search the table of requirements, using the 3270 buffer address found in step 1 as a search argument to find a matching entry.
4. Add the offset value from the entry found in the table to the starting address of the main storage map buffer, to produce the main storage address of the start of the receiving field.
5. If the length of the data field determined in step 2 is greater than the maximum field length value in the entry found in the table, go to step 10.
6. Check the flag byte in the entry found in the table. If left justification is requested, go to step 10. Otherwise proceed to step 7 for right justification.
7. Move zoned decimal zeros to the receiving field, using the field starting address determined in step 4. Use the maximum field length value in the entry found in the table as the length for the move.
8. Develop a new main storage address for the start of the receiving field to accommodate the request for right justification. The right-justified starting address for the receiving field = (field starting address determined in step 4 + maximum field length value in the entry found in the table) - length of the data field in the data stream found in step 2.
9. Move the data field from the data stream to the main storage address developed in step 8, using the length of the data in the data stream determined in step 2. Return to the start of this routine to find the next data field in the data stream.
10. Move blanks to the receiving field using the starting address of the field as determined in step 4. Use the maximum field length value in the entry found in the table as the length for the move.
11. Move the data field from the data stream to the receiving field using the field address determined in step 4. Use the length of the data in the data stream (determined in step 2) as the length for the move.
12. Check the flag byte in the entry found in the table to determine if uppercase translation is requested. If it is not requested, return to the start of this routine to find the next data field in the data stream.
13. Translate the data in the receiving field to uppercase, then return to the start of this routine to find the next data field in the data stream. The translation can be done in two ways:
 - Use the TRANSLATE instruction with the translation table built to convert lowercase alphabetic characters to uppercase.

- Use the OR instruction to place blanks in the field. This will change the DUP and FM characters. The FM appears as a ; on the screen, but appears in the data stream as X'1E'. It will be converted to a true ; that is, X'5E'. The DUP appears as an * on the screen, but appears in the data stream as X'1C'. It will be converted to a true * (X'5C').

Immediate Selector Pen or Cursor Select Data Stream

When a Read Modified command is executed for a display station as a result of an immediate detection by the selector pen or cursor select, the resulting data stream consists of address strings that identify which fields on the screen have the modified data tag set; the 3274 and 3276 control units also transmit the modified data if the proper designator character is used.

The data stream, ignoring the header information up to and including the AID character, appears as:

S	A	A	S	A	A	
E	1	2	E	1	2	. . .
A			A			

If the operator keys into a field and an immediate selector field is selected, the keyed data is not transmitted. However, if keyed data is entered by the operator, delayed selector fields are selected, and the ENTER key or a PF key is pressed; then the address and data for all fields, whether selected or keyed, are included in the data stream.

You can use a subroutine to free the application program from determining which fields were selected on a panel. A table can be built that consists of the 3270 buffer addresses, giving the location of each selectable field on a panel. The mapping routine can then compare the addresses in the table, and return to the application program a list of indicators that identifies the selected fields.

The list of indicators can be returned to the application program. A string of one-position fields can be used, and each position can indicate with a unique character that a field was selected. The first position in the returned list can be marked if a field in the data stream has the same address as the first element in the address table; the second position in the returned list can be marked if a field in the data stream has the same address as the second element in the address table. The application program can then determine which relative positions in the list have been marked to determine which fields have been selected by the operator.

Because the input from a display using selector pen or cursor select detection is a series of fixed-length addresses, the mapping routine can analyze the input stream and decode it.

For example, using the selector panel illustration in Figure 3-3, assume that the operator has selected the delayed-detectable fields located at row 5, column 10 and row 3, column 26 and the immediate-detectable field located at row 7, column 18. The input data stream transmitted in hexadecimal from the display would be:

11C1E911C2E911C4C1

ROW	COLUMN																																							
	1 - 10										11 - 20										21 - 30										31 - 40									
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
01	PICK ONE FROM EACH COLUMN																																							
02																																								
03	0?RED																				0?2, D.O.R.																			
04	0?BLUE																				0?4, D.O.R.																			
05	0?YELLOW																				0?6, D.O.R.																			
06																																								
07																			0 ENTER																					
08																																								
09																																								
10																																								
11																																								
12																																							480	
13																																								

Figure 3-3. Example of Selector Pen Panel

Using the sample table in Figure 3-4, the mapping routine returns a list in hexadecimal to the application program:

406F40406F406F

This list indicates that the second, fifth, and seventh fields were selected. Note that the addresses of the selected fields appear in the data stream in the same sequence as the fields appear in the display buffer. When a selector pen panel is designed by columns, the address of the field selected from the first column may not occur before the address of the field selected from the second column in the input data stream.

You can write a macro instruction similar to the one used to build the table in Figure 3-2 to build the selector pen table:

```
MAP NAME=SELTABLE,MODEL=1
MAP ADD=(3,10)
MAP ADD=(3,26)
MAP ADD=(4,10)
. . .
```

SELTABLE	EQU *	FOR MODEL 1 DISPLAY
DC	X'C1D9'	ROW 3 COL 10
DC	X'C1E9'	ROW 3 COL 26
DC	X'C2C1'	ROW 4 COL 10
DC	X'C2D1'	ROW 4 COL 26
DC	X'C2E9'	ROW 5 COL 10
DC	X'C2F9'	ROW 5 COL 26
DC	X'C4C1'	ROW 7 COL 18
DC	X'FF'	TABLE STOP INDICATOR

Note: The 3270 addresses used in the above table correspond to the buffer position of the Selector Pen designator character in a field, not to the location of the attribute character which defines the field.

Figure 3-4. Sample Mapping Table

Mixed Read Modified Input Data Streams

When some keyed input and some delayed selector pen or cursor select detection occur in a panel during the same input operation from a display, you can use the table-driven mapping technique for non-selector-pen or cursor select panels. Specify the table elements so that all delayed selector fields have a maximum length of one character. The mapping routine places the first character from the appropriate data stream field into the target field. The first character in a delayed selector field that has been selected is always a (>); that is, X'6E'. The application program can examine the target buffer for that character in the proper target field to determine if the field has been selected.

BUILDING OUTPUT DATA STREAMS

The 3270 requires specific bit patterns for order sequences, control characters, and buffer addressing. The data streams can be prepared in several different ways. A data stream to build a static panel (a panel which will always be displayed in exactly the same manner) can be assembled in an application program as a set of data constants. A semi-dynamic panel, which may occasionally be modified or added to, can have the static portion assembled in the application program and have the program dynamically modify or add to the data stream. A data stream for a dynamic panel (a panel with a high degree of change) must be created or assembled as a unit at execution. This section discusses how to reduce the considerations of device-dependency required to support static, semi-dynamic, and dynamic output data streams.

Static Data Streams

You can write macro instructions to simplify the preparation of static data streams for the 3270. One approach is to write a set of macro instructions in which each macro instruction prepares a single order sequence. Another approach is to write one macro instruction that can prepare all types of order sequences, but prepares only one sequence for each execution of the macro instruction in a program.

A sample macro instruction of the first type might be:

```
$MOD MODEL = 1, 2, 3, 4, 5
```

This macro instruction sets a global value so that the specified model number is used until another \$MOD macro instruction is encountered. The model number is required to correctly calculate 3270 buffer addresses. The buffer address X'C2D5' represents column 4, row 30 for a Model 1 display, and column 2, row 70 for a Model 2 display.

The following are also examples of the first type of macro instruction:

```
$SBA (1,10) generates the SEA order sequence X'1140C9'
```

```
$SF (PROT,NUM,SKIP,MDT,HI,DET,NONDISP)
```

generates an SF order (X'1D') followed by the appropriate attribute character defined by the options selected in parentheses. Notice that if PROT is not specified, unprotected is assumed; if numeric is not specified, alphameric is assumed.

\$RA (1,10,'*') generates the RA order sequence X'3C40C95C'.

\$EUA (1,10) generates an EUA order sequence X'1240C9'.

\$WCC (RESET,RESTORE,ALARM,PRINT,40CHAR,64CHAR,80CHAR,NLEM)

generates the proper WCC, depending on the options selected in parentheses.

\$CCC (PRINT,40CHAR,64CHAR,80CHAR,ALARM,ATT,UNPROT,PROT,ALL)

generates the proper copy control character (CCC), depending on the options selected in parentheses. (The CCC identifies the type of data to be copied.)

\$IC generates X'13'

\$KBD KEYBOARD = APL or Text

Used with the Data Analysis feature to identify the keyboard providing 3277-2 display input.

\$SI generates the Suppress Index character, valid for the 3288-2 or 3289 printer. Other printers receive | (the or bar) in place of the Suppress Index character.

After you have defined the macro instruction, the data stream required to build the sign-on panel shown in Figure 2-12 could be created as follows:

```
SIGNON  $MOD      MODEL=1
        $WCC      (RESET,RESTORE)
        $SEA      (2,11)
        $SF        (PRCT)
        DC        C'SIGN-ON PROCEDURE'
        $SEA      (4,2)
        $SF        (PROT)
        DC        C'PLEASE ENTER YOUR SIGN-ON INFORMATION'
        $SEA      (6,1)
        $SF        (PRCT,HI)
        DC        C'NAME:'
        $SF
        $IC
        $SEA      (6,25)
        $SF        (PRCT,HI)
        DC        C'LOCATION:'
        $SF
        $SEA      (7,1)
        $SF        (PROT,HI)
        DC        C'SERIAL NUMBER:'
        $SF        (NUM)
        $SEA      (7,23)
        $SF        (PRCT)
        $SEA      (10,4)
        $SF        (PRCT)
        DC        C'WHEN ALL ... ENTER KEY'
```

You could also write the second type of instruction, a single 3270 data stream macro instruction, which might have the format:

[symbol]	\$MAC	op-type	,(attributes)					1
			,(row,column)	[,character]	,MODEL=			2
								3
								4

symbol
specifies a symbol that refers to the data stream

op-type
specifies the type of screen control operation to generate. Valid values are: SF, SEA, IC, RA, EUA, WCC, and CCC.

(row,column)
specifies the row (1 to 43) and column (1 to 132) where the operation starts or ends (depending on the op-type). This parameter is required for op-types SBA, RA, and EUA.

(attributes)
indicates attributes or control bits for SF, WCC, and CCC:
Some valid values for SF are: PRCT, SKIP, NUM, MDT, HI, DET, NONDISP.

Some valid values for WCC are: RESET, RESTORE, ALARM, PRINT, 40CHAR, 64CHAR, 80CHAR, NLEM.

Some valid values for CCC are: PRINT, 40CHAR, 64CHAR, 80CHAR, ALARM, ATT, UNPROT, PROT, ALL.

character
specifies the character used in the RA function.

MODEL=
indicates the model of 3270. This model number is used to calculate the buffer address. This parameter is specified only once in the first macro instruction of a data stream series or whenever the data stream to be generated is for a different model than the preceding series. Model numbers 3 and 4 can be specified only for the 3278 Display Station.

After you have defined the macro instruction, the data stream required to create the sign-on panel shown in Figure 2-14 could be as follows:

```
SIGNON      $MAC      WCC, (RESET,RESTORE),MODEL=1
            $MAC      SEA, (2,11)
            $MAC      SF, (PROT)
            DC        C'SIGN-ON PROCEDURE'
            $MAC      SEA, (4,2)
            $MAC      SF, (PRCT)
            DC        C'PLEASE ENTER YOUR SIGN-ON INFORMATION'
            $MAC      SEA, (6,1)
            $MAC      SF, (PROT,HI)
            DC        C'NAME:'
            $MAC      SF
            $MAC      IC
            $MAC      SEA, (6,25)
            $MAC      SF, (PROT,HI)
            DC        C'LOCATION:'
            $MAC      SF
```

```

$MAC SEA, (7, 1)
$MAC SF, (FRCT, HI)
DC C'SERIAL NUMBER:'
$MAC SF, (NUM)
$MAC SEA, (7, 23)
$MAC SF (FRCT)
$MAC SEA, (10, 4)
$MAC SF, (FRCT)
DC C'WHEN ALL ... ENTER KEY'

```

These two types of macro instructions can generate either a total static data stream or static sections of data streams that can be dynamically assembled at execution by the application program.

Semi-Dynamic Output Streams

A semi-dynamic panel requires some dynamic modification. Perhaps an error message must be written to a particular part of the panel and the cursor must be moved to the input field in which an error was detected during editing. The application program can concatenate preassembled static data stream segments into the program, such as field error messages. The same macro instructions that build static data streams can build partial static streams. As the input from a panel is edited, the standard error message for each field can be assembled in the output buffer, thus allowing multiple brief messages to be sent to the display in one operation.

You may have to change one or two attribute characters from high intensity to low intensity and erase the unprotected fields on a display. For example, an error message segment may have changed a field to high intensity to call the operator's attention to the field; the operator has recognized the error and re-entered the correct information. The display must now be made ready for the next input on the panel. Concatenate the order stream segments to change the attribute characters and use the Erase Unprotected to Address (EUA) order to restore the panel; do not transmit all the data and orders to completely refresh the panel.

Dynamic Output Streams

It may become physically impossible to hold in main storage all possible output data and order stream combinations that could occur during the execution of an application. You can incorporate a subroutine into screen management to accept parameters from an application program to decode the parameters and to create the data stream. You can also write for the application program a macro instruction that builds a parameter list inline from entries you specify in the macro instruction, and then branches to the screen management routine to build the required orders and data in the buffer area.

The macro instruction could appear as follows:

```
$BUILD ADD=ADDFIELD,ATT=(R3),DATA=(R4),LEN=(R5)
```

The ADDFIELD contains the 3270 buffer address in either row-column format, binary offset, or 3270 address form. R3 contains the address of the attribute byte, R4 contains the address of the data to be entered in the field, and R5 contains the length of the data. The attribute character parameter is optional.

The subroutine could convert row and column buffer addresses relative to one to decimal offsets relative to zero with the following formula:

Model 1 Buffer: $((R-1) \times 40) + (C-1)$
 Model 2, 3, 4 Buffer: $((R-1) \times 80) + (C-1)$

If the row and column buffer addresses relative to one are in two single-byte areas in binary, the conversion to binary offsets relative to zero can be coded as follows:

```
SR      R3,R3
IC      R3,COLUMN
BCTR    R3,C
SR      R4,R4
IC      R4,RCW
BCTR    R4,C
MH      R4,=H'40' USE VALUE OF 80 FOR MODEL 2
AR      R4,R3  RESULT IN R4
```

The following subroutine converts a binary halfword that represents the offset relative to zero of a position in a 3270 buffer to an equivalent two-character 3270 address. R3 is a work register, and R4 points to the binary halfword to be converted. The converted result is found at ANSWER.

```
LH      R3,C(R4)
STC     R3,ANSWER+1
SRL     R3,6
STC     R3,ANSWER
NI      ANSWER+1,X'3F'
TR      ANSWER(2),TAE
.
.
.
ANSWER DC      X'0C00'
TAE     DC      X'40C1C2C3C4C5C6C7C8C9A4B'
        DC      X'4C4D4E4F50E1D2D3D4D5D6D7'
        DC      X'D8D95A5B5C5E5F6061E2E3'
        DC      X'E4E5E6E7E8E96A6B6C6D6E6F'
        DC      X'FCF1F2F3F4F5F6F7F8F97A'
        DC      X'7E7C7D7E7F'
```

Large Screen Size

Application programs written for systems that use 480- or 1920-character screen size will run on large screen displays with the same width but with a greater number of lines. Terminals with large screen capacity (960, 2560, 3440, and 3564 characters) will automatically default to smaller screen size unless the large screen size has been specified explicitly by the application program. The Erase Write Alternate command is used to switch a display into large screen mode.

Since buffer address wrapping is screen size dependent, application programs should not depend on buffer wrap during write operations. Also field attributes must be appropriately placed to delimit the end of the screen image.

Copy Function for the 3271, 3272

Many applications require complete and unaltered hard copy (printout) of the terminal's current screen contents for the display station operator. The printer on which the display contents are printed may support one or more display stations, depending on the 3270 configuration.

When using the copy function to obtain a printout on a 3288-2 or 3289 printer, remember that various print belts can be installed on these printers.

You should define a Program Attention key so that a terminal operator can request hard copy on an assigned terminal printer. The screen management program can be notified of the operator's request and perform the appropriate action.

When a data transfer to the computer occurs from pressing a Program Attention key, a remote BSC 3277, 3275, 3276, 3278, or 3279 transmits AID and cursor address, a local 3277 only transfers the AID (Attention Identifier) character. The AID character identifies which key transferred the data. No screen data is transmitted, so the program is notified of a specific request.

Once the request is identified by inspecting the AID character, the program must identify the type of unit that made the copy request. This can be done by examining the characteristics of the specific device in a terminal characteristics table that you can create. For example, depending on the type of device, the following procedures can be used to produce hard copy:

- To copy from a remote 3275 to the printer attached to the 3275, the program should send WCC to the 3275. The WCC (Write Control Character) restores the keyboard, starts the printer, and prints 40 or 80 characters per line. Because the printer attached to the 3275 uses the same buffer as the display, all that is necessary to print the buffer (which contains the screen data) is the start print bit in a WCC sent in a valid WRITE command sequence.
- To copy from a 3277 attached to a remote 3271 to a printer attached to the same 3271, the program should send the following data stream to the printer: STX, ESC, COPY command, CCC, from-device address, ETX. The CCC (Copy Control Character) specifies start printer, the option to copy all data, and either 40 or 80 characters per line. A model 2 display cannot be copied to a model 1 printer, but all other copy combinations are valid. The device address following the CCC is a single-character address which identifies the device to be copied from, and which is identical with the device address used to specifically poll the display requesting the copy function.

The COPY command allows the buffer contents of a device attached to a 3271 to be copied to the buffer of another device attached to the same 3271, without moving the data to be copied to and from the computer. Once the prior data stream has been sent to the printer, the program should send the following data stream to the display station that requested the copy: STX, ESC, WRITE command, WCC, ETX. The WCC restores the keyboard. The operator has a positive response that the request has been honored, and the keyboard allows the operator to continue without manual intervention.

- To copy from a local 3277 to a local terminal printer, the program should execute a Read Buffer command to the display that made the copy request. The Read Buffer command is executed, and the display station transmits AID, a two-byte cursor address, and the screen data to the computer. The program should then remove the AID character and the cursor address from the received data and, immediately preceding the remaining data, insert a WCC that specifies start printer and 40 or 80 characters per line. The altered data stream beginning with WCC should then be sent to the printer to copy the data. The program should then send a WCC with the restore keyboard option to the display that requested the copy function.

If the program determines that the receiving printer is busy, and the requested copy function cannot be immediately completed, one of the following actions should be taken:

- 3271: Notify the terminal operator of the situation and ask the operator to wait or cancel the request.
- 3271 or 3272: Perform a Read Buffer to bring the screen data into the computer where it can be queued until the printer is available, without delaying the operator.

3274 AND 3276 COPY FUNCTION

The 3274 and 3276 control units operating in BSC mode can process the COPY command identically with the 3271 control unit. However, these control units can also handle the local copy function as follows:

1. With the 3274 a local copy can be initiated by using the Print key and the print authorization matrix. A local copy involves the transfer of data directly from the display buffer to the printer buffer and its subsequent printing.
2. The host can initiate a copy via the print authorization matrix by setting the start-print bit in the WCC of a write command.

THE PRINT AUTHORIZATION MATRIX FOR THE 3274 AND 3276

With the exception of processing the BSC copy command, the print authorization matrix is always used by the 3274 and 3276 control units for copy operations, that is, to direct data from a display to a printer attached to the same control unit. For the 3274 the print authorization matrix allows each installation to define destination, printer mode, and classes of print devices and to authorize their use by displays attached to the same control unit. The 3276 uses a default matrix to control this.

The definition of a class of printers can be based on physical characteristics, location, or security of the printer. For example, in a particular installation class, "72" may have been defined as referring to all printers with a text character set and yellow paper. Thus an operator may select an authorized printer on the basis of these characteristics rather than by address.

The print authorization matrix allows a maximum of 16 classes to be defined in each subsystem. In any configuration a single printer can be in one class, or several classes, or none. A destination device may be in one of three modes specified in the print authorization matrix: local, system, or shared.

A printer in local mode is used solely for operator-initiated local copy functions. This means that displays within the cluster can contend for use of printers but the host cannot. The printer is not available for direct print operations from the host. A printer in local mode using BSC also is unavailable to the host via the COPY command. In particular a printer in local mode cannot validly be specified as a "from" device in a COPY command.

A local copy can be initiated by an operator, using the Print key on the 3274, or by the host when the printer is operating in shared mode. In ESC mode the start-print bit in the WCC or a WRITE command to the source display initiates the copy operation.

A printer in system mode is entirely under host (system) control. The printer cannot be used for operator-initiated local copy requests. The printer is likewise not available for host-initiated copy operations. However, when operating with BSC the printer can honor a COPY command when in system mode. The COPY command directed to the "to" device specifies "from" device as a command parameter. The print authorization matrix is not used to direct the copy operation.

THE MATRIX STRUCTURE

The print authorization matrix consists of a number of destination device descriptors with the following format:

Destination Address	Mode	Class	Source Device List
---------------------	------	-------	--------------------

Destination Address is the first field of the descriptor. Addresses from 1 to 7 for the 3276 and 1 to 31 for the 3274 allow printers to be attached to any port on the control unit (port 0 cannot be used for a printer). Addresses are sequential, by adapter.

Mode defines the printer to be in local, system, or shared mode.

Class is the third field of the descriptor and provides the ability to group printers into classes. This field is bit-coded, one bit for every 16 classes so that a single printer can be in more than one class. Valid classes are designated 70 through 85.

Source Device List is a bit-coded field that specifies which displays are authorized to use the printer associated with this device descriptor. Each bit position is associated with a port number on the cluster.

A printer can belong to one or more classes, and several printers can belong to a single class. It is important to note that source devices are associated with destination devices, not with classes. Thus several printers may be defined to be in class 75, but a particular display can only be authorized for some subset of all printers in that class. When class identification is displayed in the indicator row of the display, copying is performed only to authorized printers in that class.

The print authorization matrix is required to perform local copy operations. If the matrix is not loaded in the 3274, all printers are in system mode, and local copy operations are not possible. (The 3276 has a default matrix.) The exception occurs during BSC line discipline, where the host can initiate a local copy by sending a COPY command to the printer.

DEFINING THE MATRIX

The matrix must be defined by the application program at the host system and loaded into the subsystem as follows:

1. The operator at the display attached to port zero initiates a transaction with the host program responsible for defining, managing, and loading the print authorization matrix. This transaction may, through appropriate interaction with the operator, define a new print authorization matrix, redefine an existing matrix, or retrieve a previously defined matrix from storage.
2. The application program must then transmit the matrix data to the display as normal application data, in a data stream that resides in the regeneration buffer as normal character data.
3. The operator then holds down the ALT key and presses the Erase End of Field (EOF) key on the keyboard. This will cause the buffer to be scanned one character at a time and the configuration data to be stored in internal form in the control unit.

During the loading process, the Wait indicator is displayed and the keyboard is locked. If the load is successful, the Wait indicator is turned off and the keyboard unlocked. The operator can then return to normal activity. Local printing can take place according to the authorization established in the matrix. If the loading process fails, the Program Check indicator is displayed and the keyboard remains locked. The operator can reset the keyboard and resume operation. Only those device descriptors that have been loaded take effect.

The application program must ensure that correct matrix data is loaded. If invalid data is loaded, unexpected results may occur when the matrix is used by the subsystem. Loading of the matrix will terminate abnormally only when there is a format violation.

When the operator initiates the load operation from the keyboard, the print authorization matrix must appear in the buffer as shown in the following text.

The first two lines of the display are reserved for the use of the host program to display descriptive information to the display operator. These positions are not scanned during the load process.

There must be a sequential string of attribute characters, beginning at the first character position on the third row of the display (buffer address X'A0') as follows:

Bits 23 4567	Prct Unprot	Alpha Numer	MDT	High Inten	Sel Pen	Ncn Disp
10 C000	F					
00 0001	u		yes			
01 C100	u	n			yes	
10 0000	F					

This 4-byte sequence uniquely identifies the buffer data that follows as print authorization data. If the sequence does not appear exactly as shown, the load process will not occur. The remainder of the third row is not scanned.

The remaining rows of the display contain the destination device descriptors. One descriptor is contained in each row. The format of each descriptor is as follows:

Col 1	Col 2, 3	Col 4	Ccls 5-20	Cols 21-52
Protected Attribute- one byte	Address of Printer- two bytes	Printer Mode -- one byte	Print Class -- 16 bytes	Source Device List -- 32 bytes

The protected attribute, '110 0000', defines the next 51 bytes as a destination device descriptor. If it does not appear in the first column of the row, a format violation occurs and the loading process will terminate at this point.

The two bytes immediately following the attribute provide the character-coded decimal address of the printer being described. For example, the printer at port 03 is identified by the character data '03', X'F0F3'. Addresses are not validated at the time the matrix is loaded. For example, the device at the specified address must not be a printer.

Printer mode is expressed as follows, as a 1-character field:

<u>Mode</u>	<u>Character</u>	<u>Hex (EBCDIC)</u>
Local	L	C3
System	S	E2
Shared	J	D1

Any other coding of this byte results in the printer's being defined as unaccessible for either local-copy or direct-print operations. There is no validation of this byte during loading of the matrix. If there is a conflict between the mode definition and the coding of the source device list, the mode byte takes precedence.

The next 16 characters define the printer classes that are applicable to the device. By appropriate coding of this field, a device can be defined for multiple classes. Each character in this field is defined to be a character-coded digit representing one entry in the class field of the device descriptor.

<u>Display Column</u>	<u>Class</u>
5	7C
6	71
7	72
8	73
9	74
10	75
11	76
12	77
13	78
14	79
15	8C
16	81
17	82
18	83
19	84
20	85

The character '1', EBCDIC X'F1', in one of these character positions defines the device being described as a member of the class associated with the corresponding position in the class field of the device descriptor. Any other character in this position means that the device is not in the associated class.

The source device list is a 32-byte field. Source devices authorized for printers are character-coded. The character '1', EBCDIC X'F1', in any character location specifies the associated device as an authorized source device for the destination device defined. Other values in this location indicate that the associated device is not a valid source device.

<u>Display Column</u>	<u>Device Address</u>
21	00
22	01
23	02
24	03
25	04
26	05
27	06
28	07
29	08
30	09
31	10
32	11
33	12
.	.
.	.
.	.

Each descriptor takes 52 bytes, including the attribute byte; thus, each row contains 52 bytes of significant information. Other data on the row is not scanned during the load process. The first descriptor begins at buffer address X'0CF0', the second at X'0140', etc.

The end of the matrix is signaled by the following sequence of attribute bytes beginning in the first column of the row following the last valid destination device descriptor:

Bit	Prot	Alpha	MDT	High	Sel Pen	Non
23 4567	Unprot	Numer		Inten	Detect	Disp
10 CC00	F					
00 0101	u		Y		Y	
01 0101	u	n	Y		Y	
00 C100	u				Y	

Local Operation

After the print authorization matrix is loaded the operator initiates a local copy, using the Print key on the 3278 keyboard. Print data from a terminal is always directed to the authorized printer in the associated printer class. By using the Print IDENT key, however, the display operator can alter this defined association from the display keyboard. A new print class can be selected by pressing the Print IDENT key and keying in a two-digit class identification number between 70 and 85.

Host-Initiated Copy in Shared Mode for the 3274 and 3276

The host application program can initiate a local copy function in shared mode by sending to the display a WRITE command with the start print bit turned on the WCC. The control unit performs the local copy function as required, using the print class assigned to the display. When a write-type command is sent to the display with the start-print bit on, the display first interprets the orders and data in the write data stream, and then updates the display buffer.

Note: On copy operations, if the required display and printer features do not match, the desired printed output may not result.

Using Katakana Character Set Codes

Programs written for the 3271/72 using the Katakana character set might encounter problems in certain instances if run on the 3274/76. The 3274/76 control units handle four characters differently from the 3271/72. These characters are #, .', ", -. The table below summarizes the handling of these characters.

3274/76 Characters	I/O Code	3271/72 Display	Hex Code Back to Host
# (pound sign)	4A	x LC	55
!	5A	lc	56
"	7F	space	nothing
-	A1	space	00

The 3271/72 handle code 7F as the escape control character used with NL (new line) and EM (end of message). Although the character " will display as blank, the printer will treat it as NL or EM. Thus, systems that have a mix of 3271/72 and 3274/76 control units should avoid using the four characters.

CHAPTER 4. BTAM SUPPORT

This chapter, a supplement to the 3270 information in the ETAM and ETAM-ES manuals, describes some aspects of the support BTAM and BTAM-ES provide. (Chapters 5 and 6 describe TCAM and VTAM support.) It discusses how you might manage devices and keep track of device status, and how BTAM and BTAM-ES analyze 3270 operations and perform error recovery. Before writing any application programs that use BTAM and BTAM-ES macro instructions, however, refer to the BTAM and BTAM-ES publications and the other 3270 publications listed in the preface to this book for a more complete description of BTAM and BTAM-ES support for the 3270 display system. Use of the term BTAM throughout this chapter is meant to include ETAM-ES.

TELECOMMUNICATIONS MANAGEMENT WITH BTAM

BTAM provides support for the 3270 under both DOS and DCS/VS and OS, CS/VS1, and OS/VS2. ETAM-ES provides support for the 3270 under DOS/VSE. BTAM and BTAM-ES support local 3272, 3274-1B, 3274-1D, and remote BSC 3271, 3274-1C, 3274-51C, 3275, and 3276. This support includes generating channel programs for the 3270, starting and supplementing I/O operations, handling attentions and line interruptions, and performing error recovery.

Using BTAM as part of the telecommunications management of a 3270 display system involves several factors, including the following:

- Different devices on the lines have different characteristics.
- Application programs require information contained in the data stream.
- Screen management should receive the same data from a telecommunications management program, regardless of device type, to maintain a standard interface.

BTAM and BTAM-ES include support for the extended 3270 data, which includes the Write Structured Field command, Prepare to Read, and transparency macros.

WRITE STRUCTURED FIELD (WSF)

This command causes a 3274 (with the capability) to enter Write Structured Field mode. This operation transfers a data stream containing structured fields to a specified device. The structured field contains a type field that determines device action.

Local 3270 operation devices unable to support structured fields, that is those attached to a 3272 or 3274-1B Control Unit, will reject the Write Structured Field with a command reject error. On the 3274-1D, except for the optype, operands for this macro are the same as those for the WRITE ERASE macro instruction. The optype for the WSF macro is TSF.

For remote 3270 operations, Write Structured Field requires data transparency because full 8-bit bytes describe many of the device actions. Data transparency means that data, including normally restricted data-link line-control characters, are treated as specific

bit patterns. Thus, any bit pattern can be transmitted as information data and not as line-control data. To enter transparent mode, use the WRITE transparent macros.

If contact has not been previously established with the selected device, a WRITE TIE or WRITE TIX is issued, with the entry operand specifying the address of the device's entry in an addressing terminal list. If contact is already established, a WRITE TTE or WRITE TTX is issued, and the entry operand is not applicable and should be omitted. In both cases, the area operand specifies an output area containing:

DLE	STX	ESC	3 ¹	Structured Field(s)	DLE	ETX or ETB
-----	-----	-----	----------------	---------------------	-----	------------

¹This is X'F3' in EBCDIC

Notes:

1. The DLE ETX or DLE ETB characters are supplied by BTAM when the application program uses BTAM transparency macro instructions.
2. Any application program using Write Structured Field should return the device to its 3270 default mode before terminating. This will avoid starting a second application program, which may not use WSF, with the device in WSF mode. To reset WSF mode, issue an ERASE WRITE or ERASE/WRITE ALTERNATE command with a WCC of .1.. BTAM will automatically reset WSF mode during a Request for Test (RFT) processing. ETAM CPEN and LOPEN routines will also reset WSF mode.

PREPARE TO READ OPERATIONS (PTR)

The Prepare to Read operation applies to local 3274 Model 1D devices. Prepare to Read does not apply to the 3272, 3274-1B, or remote 3270 control units.

Display systems with PTR capability communicate with a host program so that, at selection time of a particular display or printer, the next program action (for example, Read Modify, Read Buffer) is known by the display system. The display system performs initialization sequences while the host prepares and issues the next action. This overlap of host and control unit activity frees the channel and control unit resources.

Prepare to Read does not have to be specified, because the 3274 Model 1D will operate in default mode, using the previously defined (local 3270) select command.

To use the Prepare to Read commands, the devices attached to a 3274-1D must be specified as supporting the Prepare to Read select commands. In OS/VS1 or OS/VS2 MVS operating systems, specifying FEATURE=PTREAD in the ICDEVICE macro sets an indicator in the Unit Control Block (UCB). In DCS/VSE, a value specified on the MODE operand sets an indicator in the PUB. If these indicators are on, BTAM constructs channel programs containing the PTR select commands. If the indicator is off, they use the local 3270 select command in the channel programs. The result is that BTAM application programs do not require modification to use the PTR select commands. For more information on the Prepare to Read select commands, refer to OS/VS BTAM, GC27-6980, or Basic Telecommunications Access Method Extended Support (BTAM-ES) Programming, SC38-0293.

Note: PTR is supported only on OS/VS1 and OS/VS2 (MVS) BTAM and ETAM-ES.

The following examples show how an application program using BTAM macro instructions can be used to concatenate and standardize a data stream.

Example 1: The normal Read Modified message from a remote ESC 3270 (3271, 3275, 3274-1C, 3274-51C, 3276) on a nonswitched or switched network backup line appears as follows:

INDEX	STX	CUA	DVC	AID	CA1	CA2	SEA	A1	A2	TEXT	ETB
-------	-----	-----	-----	-----	-----	-----	-----	----	----	------	-----

Header for First Block Only

For Structured Fields:

INDEX	DLE	CUA	DVC	STX	AID	Structured Field(s)	ETB
-------	-----	-----	-----	-----	-----	---------------------	-----

The application program can concatenate the blocks (which are generally 256 bytes or less) in a particular data stream, and strip the index, STX, ETX and ETB characters. Control unit address and device address can be converted to a specific terminal name or ID with a table. The attention identification may be used to take a standard action (such as printing the buffer contents) defined for the terminal key that caused the interruption. A subroutine may be used to convert the cursor address into screen position (by number, such as 440, or row-column, such as 10, 15). The program can then pass the combined text (preceded with the SBA and address characters) to a screen management routine.

For a 3274-1C or a 3274-51C control unit with the capability of interpreting the structured fields, the inbound write structured field would be as below.

Index	DLE	STX	CUA	DVC	AID	Structured Field	Structured Field	ETB or ETX
					X'88'			

Example 2: A message from a local 3272, 3274-1E, and 3274-1D appears as follows:

AID	CA1	CA2	SEA	A1	A2	TEXT
-----	-----	-----	-----	----	----	------

The application program should know where the data came from so it can send data to the source. You can check the relative line number in the DECE for the device address. The attention identification and cursor address information may be used as described in Example 1, and the text then sent to the screen management routine.

For an inbound data stream from a 3274-1D control unit with the structured field capability, the data stream would be:

AID	Structured Field
X'88'	

Example 3: A 3275 with the Dial feature sends a message that appears as follows:

STX	AID	CA1	CA2	SEA	A1	A2	TEXT	ETB	...
-----	-----	-----	-----	-----	----	----	------	-----	-----

As in Example 1, the application program can concatenate the blocks, strip the STX and ETB or ETX characters, use the AID and cursor address data to provide meaningful information, and supply a complete data stream to the screen management routine.

A program using BTAM macro instructions should include error recovery procedures to prevent unnecessary system or program failure. The program should be able to recognize, record, analyze, and correct error conditions and isolate a defective terminal, line, or control unit after a specific number of retries. Human intervention should be avoided by including error recovery procedures in the creation of the program.

TECHNIQUES FOR MANAGING DEVICES

THE ADVANTAGES OF A TERMINAL CONTROL PROGRAM

A terminal control program may be part of your BTAM application program or it may serve a number of applications. The terminal control program issues the BTAM macro instructions that initiate input and output. Usually it handles the error recovery you have specified. By separating this program from the processing application or applications, you allow future expansion of individual modules in your teleprocessing programming system without having to change all of them. A terminal control program can:

- Free the application program from the details of I/O, including error recovery. The terminal control program can be invoked by an instruction such as GET or PUT.
- Provide some buffering for the calling program. The terminal control program might collect all 256-byte blocks to be read from a 3270 terminal in its input area, then return with the address of the entire message.
- Simplify input for the processing program. For example, the AID byte in the data stream from entry of data might not matter to the processing program; the terminal control program can strip the AID byte or bypass it.
- Insert certain data stream characters. The terminal control program might contain some or all of the mapping functions suggested in the section "Screen Management" or only the I/O macro instructions and error recovery, and interface directly with a mapping module.

THE ADVANTAGES OF A MASTER TERMINAL PROGRAM

A master terminal program allows changes (in configuration, for example) in a teleprocessing application while the system is in operation. It provides a central control that allows the teleprocessing application or system to react flexibly to variables such as time of day, user or system priority, and system operator or remote supervisor messages. A master terminal program can usually be invoked (perhaps by the terminal control module) from a message by the console operator or from a local

or remote 3277, 3278, 3279, or a remote 3276 designated as a master terminal.

The master terminal can communicate exclusively with the master terminal program or serve as a work terminal and be used as a master terminal when required. Access to the master terminal program may be available to any operator at a terminal designated as a master terminal, to a supervisor using his identification card at any of a number of terminals equipped with a card reader, to any terminal operator who entered a password authorizing use of the master terminal program, or to an operator at the system console.

Here are some uses for a master terminal program:

- A common use would be to change the configuration of a teleprocessing network; add or remove one or more terminals to a line. A supervisor at a master terminal in a Denver office could send a message to the central office in Kansas City to remove a temporarily inactive terminal from a line; the master terminal program would then (perhaps using the BTAM CHGNTY macro instruction) set the skip bit for that device in the appropriate terminal list. Time would not be wasted polling that terminal.
- One or more application programs that depend on input and output from a system console or master terminal could use the master terminal program as a common interface to the master terminal or system console operator.
- On receipt of a master terminal message that the teleprocessing system will be switched from one operating system to another, the master terminal program could arrange an orderly collection of outstanding messages prior to system shutdown, then start up the teleprocessing system again after the new operating system is running.
- If it is desirable to switch disk files at a particular time for a given data entry application, a supervisor at a master terminal could request the switch and the master terminal program could send a request message to the system operator.
- A master terminal program could broadcast messages to all or designated terminals in a system. For example, operators could be notified of temporary system shutdown. A bank might use such a broadcast message to send branches of affiliates the serial numbers of stolen \$100 bills.
- A master terminal program could maintain the time of day and assure that no terminals were polled in time zones that were not yet at work.

TECHNIQUES FOR KEEPING TRACK OF DEVICE STATUS

There are several reasons why you may want to maintain tables in your program with entries for each control unit or terminal. These tables can be used to store logical or symbolic names for use in messages (you may want to refer to a particular terminal in a message as MIAMI rather than by its device address number), to record the activity of each device, or to store other information such as dial digits used when calling a nonswitched device through a switched network backup facility (for example, using the IBM 3872 Modem).

The tables in Figure 4-1 may be used to:

- Associate each control unit or terminal with a name based on its geographic location or work station number.
- Keep track of the number of transactions (inquiries or entries, for example) from a terminal either for billing purposes or to see how much the terminal is being used.
- Keep track of various kinds of errors.
- Keep a priority assignment number when the network is heavily used.
- Keep the phone number to be dialed when using switched network backup to a control unit.

One way to associate a table with line activity is to build your own extension to the DECB. Figure 4-2 suggests some of the uses for such a DECB extension.

TRANPTR is an area that stores a pointer to a translation table used after an input operation. Another byte, PREVOP, stores information on what I/O operation took place prior to the one the DECB is presently associated with; this could be useful in determining error recovery actions. Another use would be to store the return address associated with an I/O operation when one Read or Write macro instruction and DECB is branched to from more than one place in a program. Here is one DOS, DCS/VS coding technique for saving a return address with a DECB extension:

```

BAL      R11,WRITTT
.
.
.
WRITTT  WRITE  (R6),TT,DTFBT1,AREA1,200,,3,MF=E
B        TWAIT
.
.
.
TWAIT  EQU     *
ST      R11,RETURN
TWAIT  (R6),TERMTST,FCBLIST=LIST
L       R11,RETURN
BR      R11

```

RETURN, shown in Figure 4-2, is a three-byte area in the DECB extension.

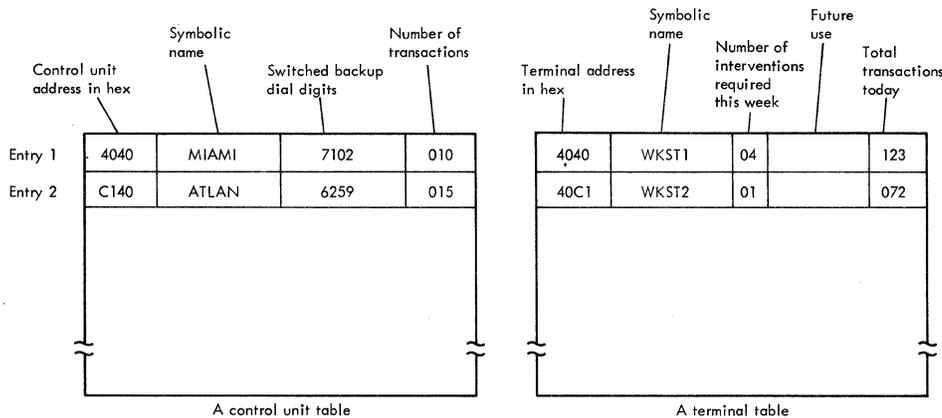


Figure 4-1. Table of Control Unit and Terminal Information

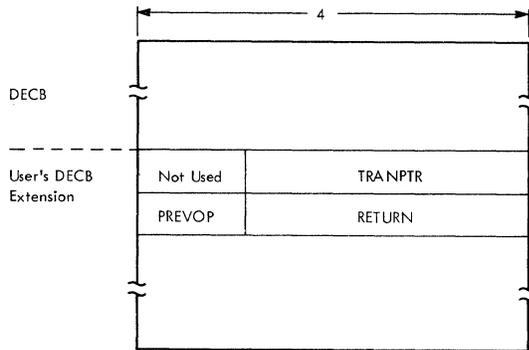


Figure 4-2. Example of a User-Built DECB Extension

RELIABILITY AND ERROR RECOVERY

REMOTE LEASED LINE EVENT COMPLETION ANALYSIS

On completion of a 3270 I/C operation, the terminal control program should analyze the circumstances of the completion and decide what action to take. This section applies to any terminal control program that uses ETAM.

The BSC 3270 remote leased line completion analysis is organized in six parts. Four of the parts are the flowcharts in Figures 4-3 through 4-6, which are a logical sequence for analyzing completion information after a read or write operation. The flowcharts refer to the Read action descriptions or Write action descriptions which follow the Read or Write flowcharts.

The action descriptions are in the following format:

- The ETAM operations to which the action applies
- An explanation of the causes of the completion condition
- The advised actions and an explanation, where appropriate

Certain completion conditions indicate that a control unit sense/status message has been received. These messages are generated by the remote BSC 3270 in a variety of circumstances to inform the computer of changes in the status of 3270 devices. Examples of such changes are the completion of a mechanical print operation or the receipt by the control unit of an invalid command. For further information on the sense/status message, refer to the description of remote operations in the IEM 3270 Information Display System, Component Description. Where completion conditions exist, the action description contains the advised procedure for processing the receipt of the message as input. However, the description of sense/status analysis should be consulted to interpret the information in the message and the actions that follow. The 3270 sense/status message must be processed to maintain the availability of the remote BSC 3270 devices.

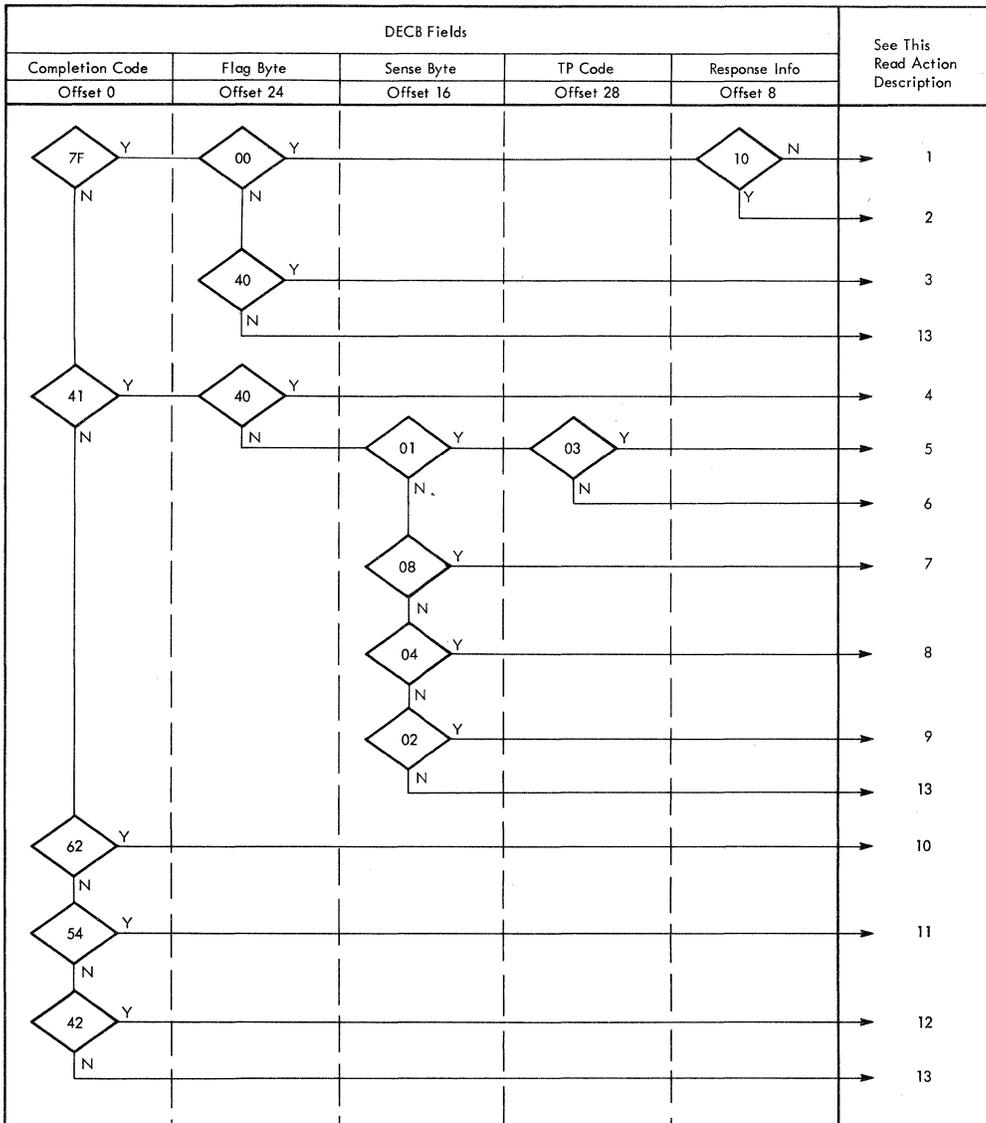


Figure 4-3. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Nonswitched Line Read Completion Analysis

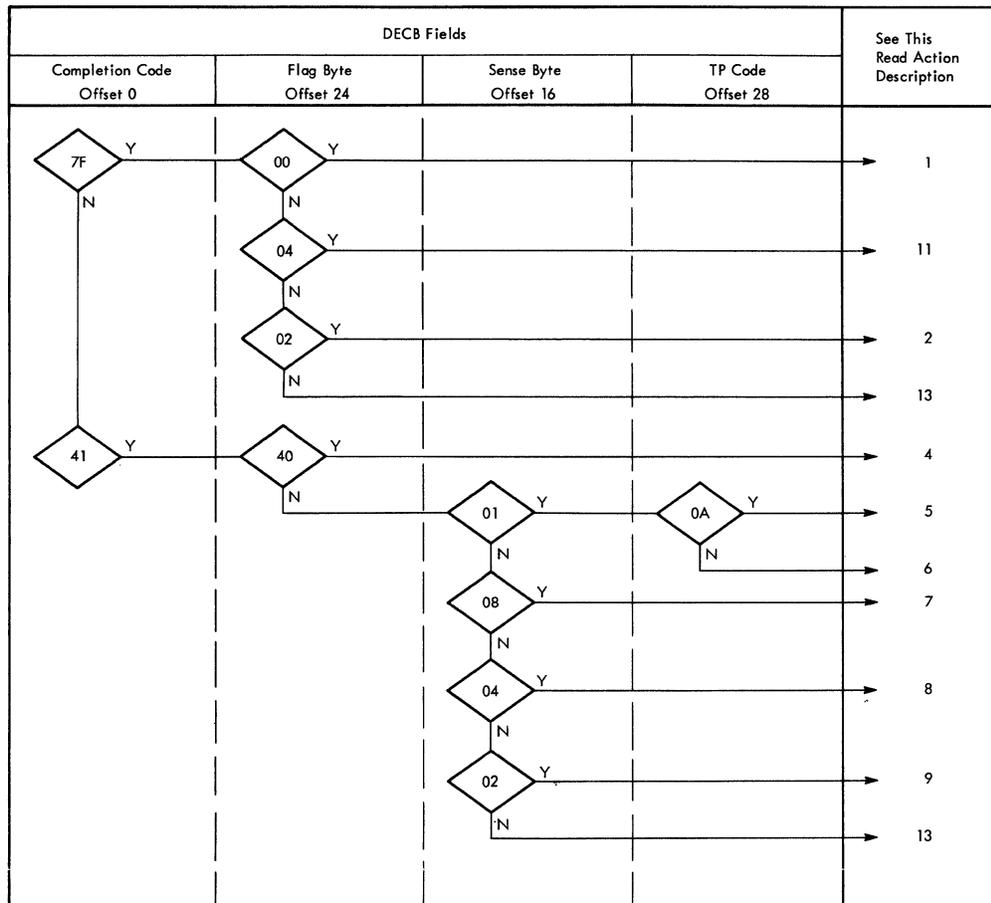


Figure 4-4. OS BTAM and OS/VS BTAM, Remote Nonswitched Line Read Completion Analysis

Read Action Description One

EIAM Operations: Follows completion of a Read Initial (TI) or Read Continue (TT).

Explanation: A text block has been received without hardware or line error. The input message may take one of several formats. The format generally appears as follows:

AUTCPOLL	STX	CU	DVC	AID	CURSOR	FIELD DATA	ETB
INDEX		ALDR	ADDR		ADDRESS		ETX
BYTE							

If an operator has initiated the message other than with the Test Request key, the first block from a device has the above format. The maximum block length is 256 characters from Auto Poll index byte through ETX or ETB. The block could be less than 256 even if there are subsequent blocks.

The following variations to the above format are also possible:

- If Auto Poll is not used or if the message is not from the first device to respond to a general poll, STX or DLE STX is the first character in the input area.
- If the CLEAR key or a Program Attention key was pressed, the ETX is the only character following the AID byte.
- If there are no modified fields, the ETX is the only character following the cursor address.
- If the input block is not the last from the device, an ETB terminates the block. ETX terminates the last or only block.

STX	FIELD DATA	ETB	or	DLE STX	FIELD DATA	ETB
		ETX				ETX

If the input block is not the first from the device, it has the above format. STX or DLE STX is the first character in the input area and there is no 3270 header information.

AUTOPOLL						
INDEX	SOH	%	/	STX	TEXT	ETX
EYTE						

The above format could be received if the operator pressed the Test Request key on the 3277 or System Request key on the 3278 or 3279 and the Binary Synchronous test facility is not included in the ETAM support.

STX	ETX	or	DLE STX	ETX
-----	-----	----	---------	-----

The above null message format may be received as the last block under unusual circumstances.

EOT

BTAM passes the EOT character to the user in the input area as a normal completion. Bits are set in the DECB to indicate that an EOT has been received (DOS, DOS/VS, and DOS/VSE BTAM-ES).

Action:

- Issue READ Continue (TT) until EOT is received if multiple messages from a control unit on general poll are acceptable.

- Issue READ Interrupt (TRV) if multiple screens from any additional devices pending on the cluster are not desired.

General polling is being performed and all blocks associated with the first message have been read. All blocks of a screen should be read and concatenated before processing. To do this, move the STX plus one location of succeeding blocks to the ETB location of preceding blocks. Receipt of a null block can be processed in the same way. If a Test Request message is received in this manner, the Binary Synchronous test facility should be included. If this is not possible, the message may be processed as a CLEAR key depression.

If CS or OS/VS BTAM is used and an EOT is received, the response depends on the application. See Read Action Description Three.

Read Action Description Two

BTAM Operation: Follows completion of a Read Initial (TI) or Read Continue (TT).

Explanation: A text block containing a sense/status message has been received from the remote control unit.

AUTCPOLL								
INDEX	SOH % R	SIX	CU	DVC	S/S 1	S/S 2	ETX	
BYTE			ADDR	ACDR				

The Auto Poll index is not present if the Auto Poll feature is not used or if the message is not the first received in response to a general poll.

Action: Same as Read Action Description One.

In order to maintain the availability of the remote 3270 devices, the sense/status message must be analyzed and acted upon. For guidance in processing this message, see the section "Sense/Status Analysis."

Read Action Description Three

BTAM Operation: Follows completion of a Read Continue (TT).

Explanation: An EOT was received in response to the previous Read Continue (TT) for a text block ending in ETX.

Action: The action taken depends on the line control program.

- A held line system holds the communications line open after receipt of a message in anticipation of a response to the device. For this type of system the line is left inactive until a response is created. At this time a Write Initial (TI) is issued to select the device and send the response.
- A non-held line system does not hold the line open after receipt of a message. In this type of system the line control program might check and issue a Write Initial if output is available. If there is no output for the line, polling might be initiated with the Read Initial (TI) macro instruction.

In most systems where message throughput is a primary objective, lines should not be held.

Read Action Description Four

ETAM Operation: Follows completion of a Read Initial (TI) or Read Continue (TC).

Explanation: A text block has been received that terminates with an ENQ character.



This message format indicates that the 3270 control unit has detected an internal parity check or a cursor check during transmission. A character with invalid parity is transmitted as a SUB character (EBCDIC '3F' or ASCII '1A') and the ending ETX or ETB is replaced with the ENQ character. In addition, a data check sense/status condition is recorded at the 3270 control unit.

Action: Issue a Read Initial (TI) using the specific polling character for the sending device to retrieve the sense/status message and reset the status condition at the control unit. Use Action Description Two after receipt of the sense/status message.

Read Action Description Five

ETAM Operation: Follows completion of a Read Initial (TI).

Explanation: A timeout has occurred. No response to the previous polling sequence has been received in the time allowed by the transmission control unit. Possible causes are:

- The 3270 control unit is unable to respond, perhaps due to lack of power, a malfunction, or the keylock has not been unlocked. The 3274 or 3276 control unit might be busy executing a local diagnostic.
- Conditions on the communications line prevent transmission.
- A modem is not functioning.
- The transmitted polling sequence is not valid for any control unit on the communications line.

Action: To retry the polling operations, issue a Read Initial (TI). ETAM will have retried the operation; reissuing the macro instruction begins a new sequence.

After retrying the polling operation, if the condition persists: Take the control unit or terminal out of service and off the polling list. This can be done either under program control or in response to operator intervention through a master terminal (see "Advantages of a Master Terminal Program" in this section). Whether the action is automatic or in response to a command entered by an operator, you should issue the ETAM CHGNTRY macro instruction.

The following is an example of using the CHGNTRY macro instruction to remove a control unit from a polling list:

```
CHGNTRY      (R2),AUTOWLST,(R3),5,SKIP
```

(R2) is a register with the address of the polling list, (R3) is a register with the relative position of the entry to be changed, and 5 is the number of characters in a 3270 polling list entry. The example specifies the list as an Auto Poll wrap list but should agree with the type specified in the DFTRMLST macro instruction used to create the list. A CHGNTRY macro instruction with the ACTIVATE parameter can reinstate the control unit when the difficulty has been corrected.

The fact that the control unit is not available should be recorded for use of the terminal control program, and the operator should be notified to take manual recovery action if the system has not previously informed him.

Read Initial (TI) should be reissued after forcing the DECB polling entry address (OFFSET 21) to another control unit. If there is no other control unit on the line, or all on the line are out of service, the line should be recorded as out of service and no further operations initiated until it is placed back in service, perhaps by a master terminal.

Read Action Description Six

ETAM Operation: Follows completion of a Read Initial (TI) or Read Continue (TT).

Explanation: A time-out has occurred. No further transmission has been received after a text block acknowledgement (ACK-0 or ACK-1), or text flow has stopped without a proper ending sequence (ETB, ETX, ENQ). The possible causes include those in Read Action Description Five, except an invalid polling sequence does not apply.

Action:

- Issue a Read Repeat (TP) to acknowledge no transmission received and to receive the response, if you want more retries than ETAM error recovery provides.
- If the problem is not corrected issue a Write Reset (TR) to reset the line with an ECT. Remove the control unit from the polling list as in Read Action Description Five.

Read Action Description Seven

ETAM Operation: Follows completion of a Read Initial (TI) or Read Continue (TT).

Explanation: The transmission control unit has detected an erroneous parity or BCC check on the received data.

Action: See Read Action Description Six.

Read Action Description Eight

ETAM Operation: Following completion of a Read Initial (TI) or Read Continue (TT).

Explanation: An overrun condition has occurred. The I/C channel has not maintained the speed of the incoming data.

Action: See Read Action Description Six.

Read Action Description Nine

BTAM Operation: Follows completion of Read Initial (TI) or Read Continue (TT).

Explanation: A lost data condition has occurred. This is usually due to receipt of a data stream that exceeds the length specified for the Read operation.

Action: See Read Action Description Six.

Read Action Description Ten

BTAM Operation: Follows a Read Continue (TT).

Explanation: The positive acknowledgement of the preceding text block was not properly received by the remote control unit, which responded with an ENQ character.

Action: Issue a Read Continue (TT) to retry the acknowledgement. If the condition persists, see Read Action Description Five.

Read Action Description Eleven

BTAM Operation: Follows a Read Initial (TI).

Explanation: A negative response was received from the last active terminal in an open polling list (DFTRMLST AUTOLST), or a RESETPL macro instruction terminated polling.

Action: The appropriate action depends on the line control program:

- If output is available for the line, issue a Write Initial (TI) macro instruction to send the message.
- Resume polling at the beginning of the list.
- Suspend polling long enough to reduce the impact of processing negative polling responses.

Read Action Description Twelve

BTAM Operation: Follows a Read Initial (TI) or Read Continue (TT).

Explanation: A Test Request message has been received but the TWAIT macro instruction has not been issued.

Action: Issue a TWAIT macro instruction of the form:

```
TWAIT (R1),TERMTST,ECBLIST=(R2)
```

where (R1) specifies a register which will contain the address of the DECE posted complete when the TWAIT is satisfied, and (R2) is loaded with the address of the DECE with the X'42' completion.

Do not alter the completion code prior to issuing the TWAIT macro instruction.

Read Action Description Thirteen

BTAM Operation: Follows Read Initial (TI) or Read Continue (TT).

Explanation: This is an unrecognized completion and should not occur; it is probably a software problem.

Action:

- Take a SNAP dump or EDUMP of the system and analyze it.
- Notify the operator of the condition.
- Issue a Read Initial (TI) to reset the line and resume polling.

TRANSPARENCY

For remote 3270 devices, the write transparent macros are used for sending Write Structured Field messages. These macros are the following:

Macro	OS/VS BTAM	DCS, DCS/VS BTAM and DOS/ VSE BTAM-ES
WRITE Initial Transparent	TIX	----
WRITE Initial Transparent Text	----	TIX
WRITE Initial Transparent Block	TIE	TIE
WRITE Initial Transparent and Reset	TIXR	TIXR
WRITE Continue Transparent	TTX	----
WRITE Continue Transparent Block	TTE	----
WRITE Transparent Text	----	TX
WRITE Transparent Block	----	TE
WRITE Initial Conversational Transparent	TIVX	TIVX
WRITE Continue Conversational Transparent	TTVX	TXV

In the following descriptions of write actions, the above macros would replace their counterparts if the text being sent were a structured field.

Write Action Description One

BTAM Operation: Follows Write Initial (TI), Write Continue (TT), Write Conversational (TV), or Write Initial Conversational (TIV).

Explanation: Text transfer has completed normally.

Action:

- If the previous operation was Write Initial (TI), issue Write Continue (TT) if blocked output is being sent and more blocks remain. Note that the 3270 does not accept conventionally blocked output.

Issue Write Reset (TR) to send an EOT which resets the line. If it is desirable to resume polling on the line, issue Read Initial (TI), which resets the line and begins polling.

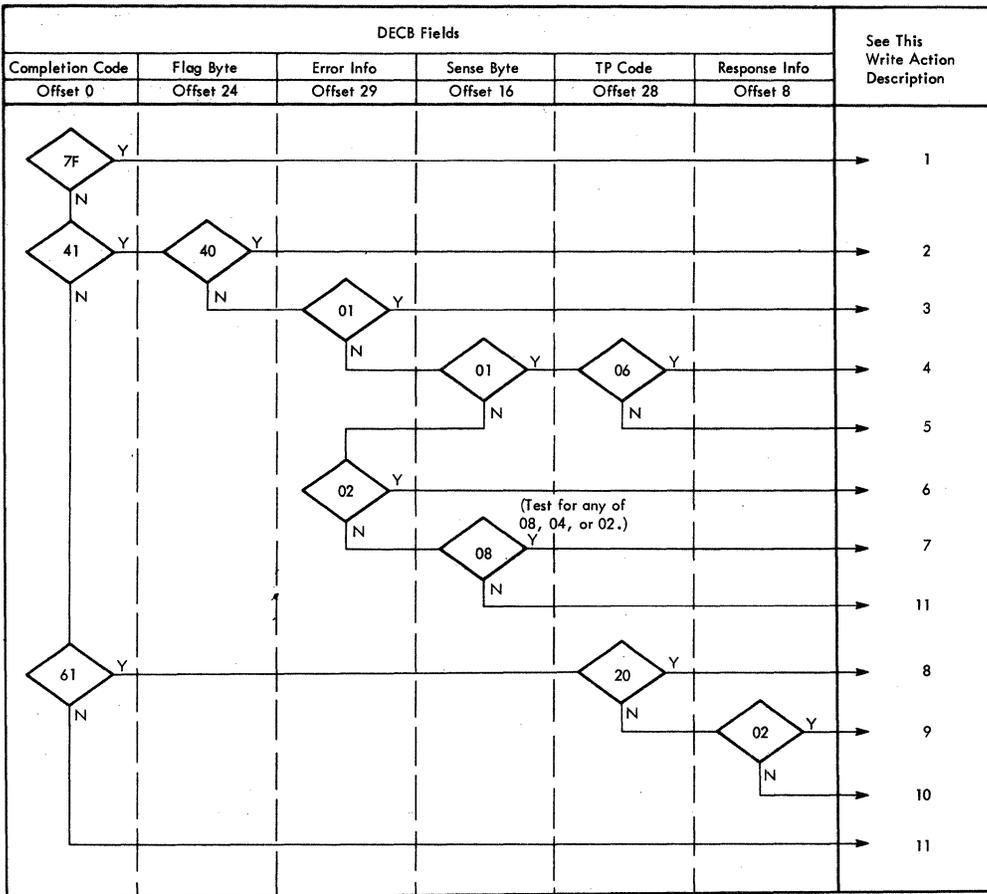


Figure 4-5. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Nonswitched Line Write Completion Analysis

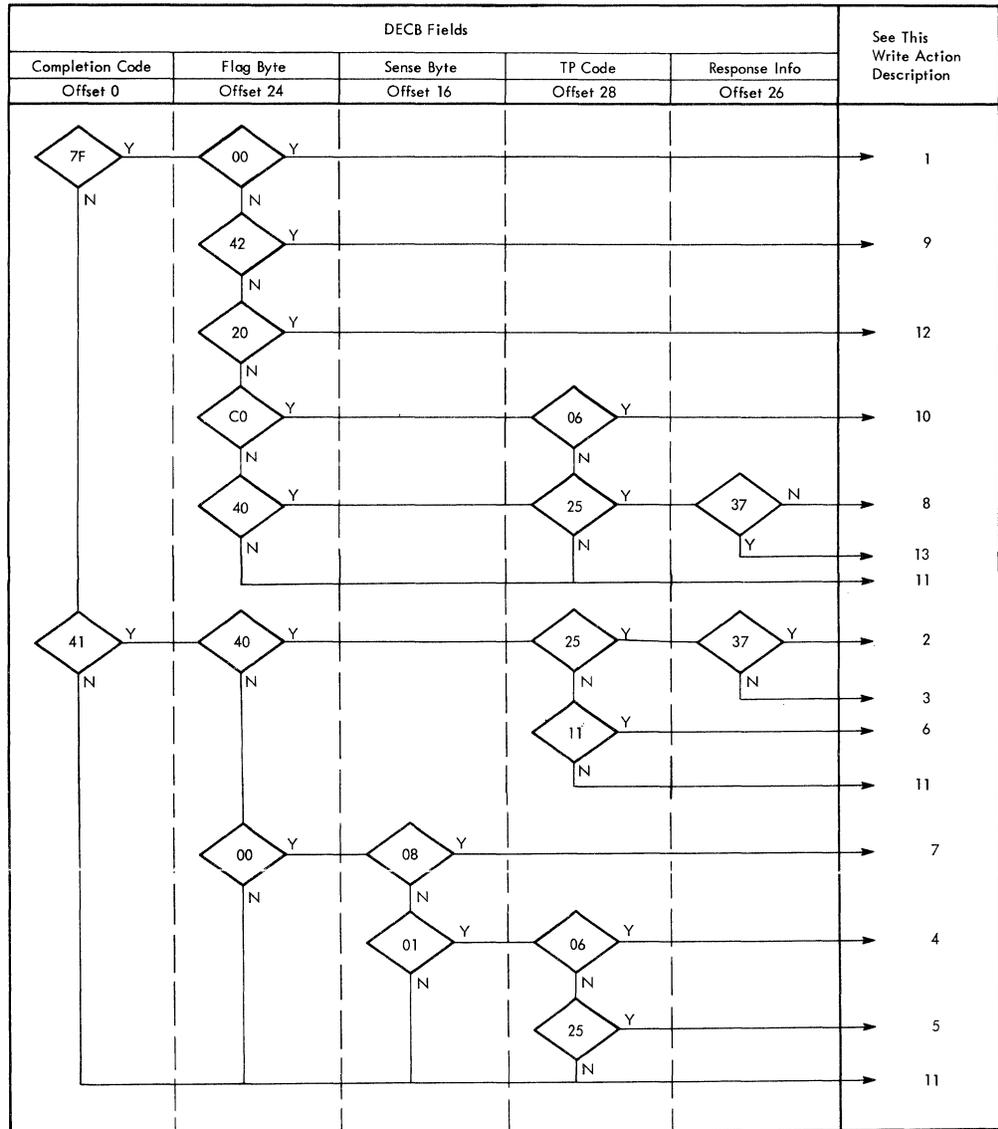


Figure 4-6. OS BTAM and OS/VS BTAM, Remote Nonswitched Line Write Completion Analysis

- If the previous operation was Write Initial Conversational (TIV), Write Continue Conversational (TTV) or Write Conversational (TV), issue Read Continue (TT) to read all blocks and the final EOT.

Write Action Description Two

BTAM Operation: Follows a Write Initial Conversational (TIV), Write Continue Conversational (TTV), Write Conversational (TV), Write Initial (TI), or Write Continue (TT).

Explanation: An EOT has been received in response to a text transmission. This response indicates that the device could not perform the operation specified by the command code in the text. Examples are a busy or unavailable device, or a check condition. The exact cause has been recorded at the control unit.

Action: Issue a Read Initial (TI) command specifying as the polling entry a list containing the specific polling characters of the control unit and device that returned the EOT. You could code a specific polling list with the DFTRMLST macro instruction for each terminal in the system or code one DFTRMLST macro instruction and modify the entry prior to issuing the Read Initial (TI) macro instruction. See the description of the DFTRMLST format in the appropriate BTAM SRL.

Write Action Description Three

BTAM Operation: Follows Write Initial (TI), Write Continue (TT), Write Initial Conversational (TIV), Write Continue Conversational (TTV), or Write Conversational (TV).

Explanation: A NAK has been received in response to a text transmission. This indicates that the 3270 control unit has detected an ENQ character in the transmission or that the 3271, 3276 or 3274 (not 3275) has detected an invalid ECC. A 3275 sends an ECT and indicates an invalid ECC in the sense/status message. The cause could be a transmission error or invalid text. BTAM will have retried the operation.

Action:

- If the preceding operation was Write Initial (TI) or Write Continue (TT), issue a Write Initial (TI) to retry the operation.
- If the preceding operation was Write Initial Conversational (TIV) or Write Conversational (TV), issue a Write Initial Conversational (TIV) to retry the operation.
- If the condition permits, take a SNAP dump or PDUMP of the text block and notify the operator of the condition so that the fault can be isolated.

Write Action Description Four

BTAM Operation: Follows a Write Initial (TI) or Write Initial Conversational (TIV).

Explanation: No response to the previous selection sequence has been received. Possible causes include:

- There is no device on this line for the selection sequence that was sent.

- A hardware transmission error prevented recognition of the selection sequence or the acknowledgement.
- The 3270 is unavailable due to lack of power or a malfunction.

BTAM will have retried the operation.

Action:

- If the previous operation was Write Initial (TI), issue a Write Initial (TI) to retry the operation.
- If the previous operation was Write Initial Conversational (TIV), issue a Write Initial Conversational (TIV) to retry the operation. If the condition persists, take the device out of service. Record the fact that the terminal is out of service for use by the terminal control program and notify the operator. See "Advantages of a Master Terminal Program" in this chapter.

Write Action Description Five

BTAM Operation: Follows Write Initial (TI), Write Continue (TT), Write Initial Conversational (TIV), Write Continue Conversational (TTV), or Write Conversational (TV).

Explanation: No response to the preceding text transmission has been received. Possible causes include:

- The preceding text was received by the 3270 without valid framing characters (STX/ETX) or (DLE STX/ETX).
- The 3270 has been unavailable.
- A transmission error has prevented receipt of the response.

Action: Reissue the preceding macro instruction to retry the operation. If the condition persists, take the terminal out of service and proceed as in Write Action Description Four.

Write Action Description Six

BTAM Operation: Follows Write Initial Conversational (TIV), Write Continue Conversational (TTV), or Write Conversational (TV).

Explanation: Text ending in ENQ has been received. This indicates that the 3270 has detected an internal parity check or a cursor check. A SUB character (EBCDIC '3F' or ASCII '1A') has been substituted for the error character and the ENQ character is transmitted in place of the ETX/ETB and BCC. A status condition has been stored at the 3270's control unit.

Action: Issue a Read Initial (TI) with a polling list containing the specific polling sequence for the device transmitting the ENQ character. This retrieves the 3270 sense/status message (see "Sense/Status Analysis" in this chapter) and resets the status condition.

Write Action Description Seven

BTAM Operation: Follows Write Initial Conversational (TIV).

Explanation: Text was received in error:

X'08' - data check
X'04' - overrun
X'02' - lost data

See Read Action Descriptions Seven through Nine.

Action: Issue a Read Repeat (TP) to send a NAK, which transmits the lost text again.

Write Action Description Eight

BTAM Operation: Follows Write Initial (TI) or Write Continue (TT).

Explanation: A WACK was received in response to a text write. This is a normal response to a text data stream that contains a copy control character or write control character specifying 'start printer.' It implies that the print operation has begun and that the printer is now busy.

Action: Issue a Write Initial (TI) to send to another device on the line, or issue a Read Initial (TI) to begin polling on the line.

Write Action Description Nine

BTAM Operation: Follows Write Initial (TI) or Write Initial Conversational (TIV).

Explanation: An RVI has been received in response to addressing. This response indicates that the 3270 has pending status, other than device end or device busy, which must be retrieved prior to writing to the 3270.

Action:

- Issue a Write Reset (TR) to reset the line.
- Then issue a Read Initial (TI) using a polling list with the specific polling sequence for the device.

Write Action Description Ten

BTAM Operation: Follows a Write Initial (TI).

Explanation: A WACK has been received in response to addressing. This indicates that the addressed device is busy.

Action:

- Issue a Write Reset (TR) to terminate the operation.
- Then check for output to another device on the line and issue a Write Initial (TI) or initiate polling on the line with a Read Initial (TI).

Write Action Description Eleven

BTAM Operation: Follows Write Initial (TI), Write Continue (TT), Write Initial Conversational (TIV), Write Continue Conversational (TTV), or Write Conversational (TV).

Explanation: This is unrecognized completion and should not occur. The probable cause is a software problem.

Action:

- Take a SNAP dump or PDUMP of the system for analysis.
- Notify the operator of the condition.
- Issue a Write Reset (TR) to reset the line, then issue a Read Initial (TI) to resume polling.

Write Action Description Twelve

BTAM Operation: Follows a Write Initial (TI).

Explanation: An incorrect alternating acknowledgement was received in response to the text transmission. BTAM has verified that the incorrect acknowledgement is incorrect by sending an ENQ to request retransmission of the ACK.

Action:

- Issue a Write Reset (TR) to reset the line. Then retry the Write Initial (TI).
- If the problem persists, notify the operator of the condition. Record the control unit out of service and proceed as in Read Action Description Five.

Write Action Description Thirteen

BTAM Operation: Follows a Write Initial (TI).

Explanation: EOT has been received in response to a Write Initial trying to start a 3284-3 printer attached to a 3275.

Action: Read the sense/status message.

REMOTE DIAL EVENT COMPLETION ANALYSIS

This section should help you design or code the portion of a terminal control program which, upon completion of a 3270 I/O operation, analyzes the completion and decides the proper action.

The description of the 3270 remote dial event completion analysis is organized in six parts. Four of the parts are flowcharts contained in Figures 4-7 through 4-10. These flowcharts are a logical sequence in which completion information can be analyzed after a Read or Write operation. The flowcharts refer to the action descriptions that immediately follow the flowcharts.

The action descriptions are in the following formats:

- An explanation of the causes of the completion condition.
- The advised actions, and comments, where appropriate.

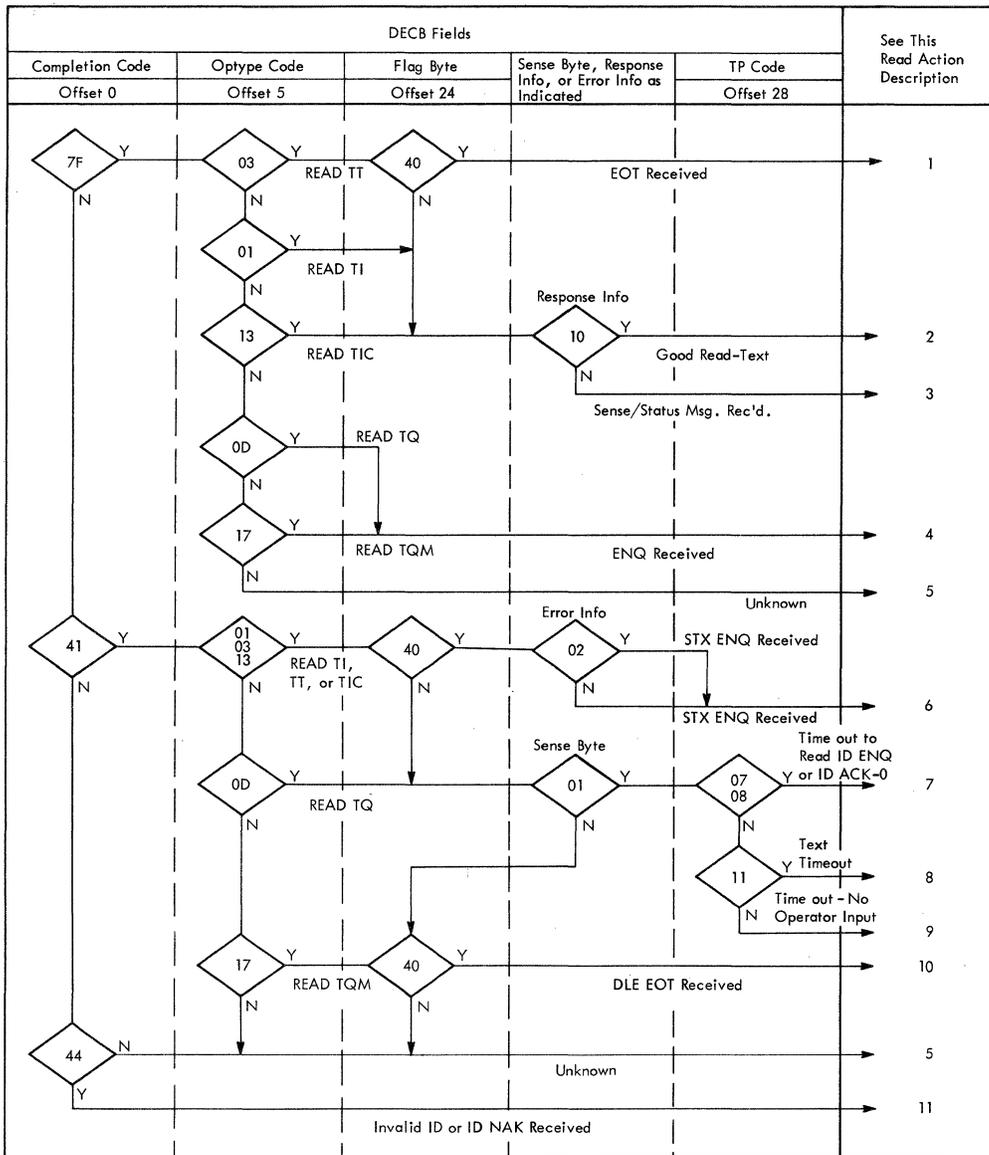


Figure 4-7. DOS BTAM, DOS/VSE BTAM, and DOS/VSE BTAM-ES, Remote Dial Read Completion Analysis

Read Action Description One

Explanation: The Read command has completed properly without error. If an EOT has been received, the 3275 has responded EOT to a Read continue, indicating it has no more text to send and is releasing control of the line.

Action: Normal completion of a read sequence from the 3275 allows you to:

1. Issue a Write Disconnect if no more input or output is desired from or to the 3275.
2. Issue a Read Inquiry or Read Inquiry Monitor to await another bid from the 3275. The Read Inquiry Monitor does not allow a 3275 time-out. It maintains the connection with the 3275 for further

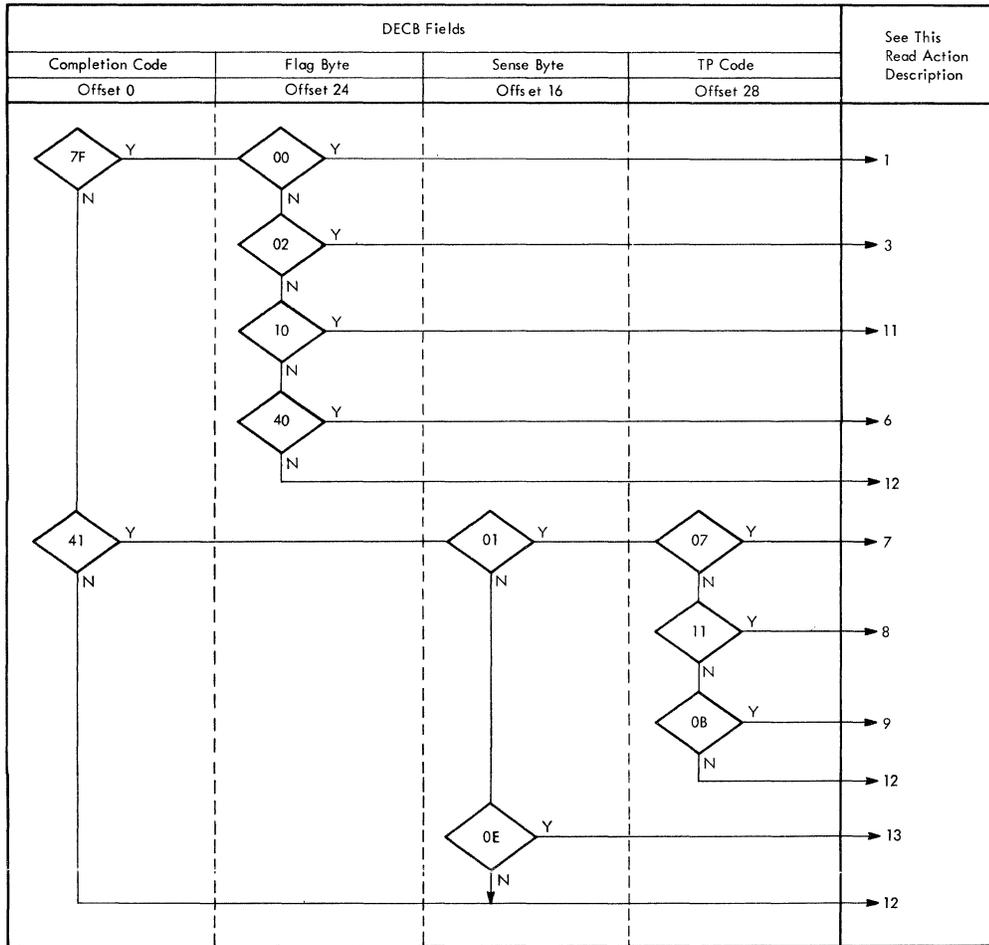


Figure 4-8. OS BTAM and OS/VS BTAM, Remote Dial Read Completion Analysis

interaction. You might set a timer interval with the Monitor version and end the read with RESETPL after the interval, followed by a Write Disconnect, or a Write Inquiry followed by a Write Continue to prompt the operator.

3. Issue a Write Inquiry to bid for the line, if you want to send a message to the terminal.

If an EOT was not received, then a text block has been successfully received from the 3275 in response to a Read Initial, Read Continue, or Read Connect instruction.

The input may have the following format:

STX	AID	Cursor Address	FIELD DATA	ETB ETX
-----	-----	-------------------	------------	------------

If the operator initiated the message other than with the Test Request key, the first block has the above general format. The maximum block length is 256 characters even if there are subsequent blocks because the 3275 does not break an SEA sequence (3 bytes represent a buffer address).

The following variations in the above format are possible:

- If the Clear key or a Program Attention key was pressed, the ETX is the only character following the AID byte.
- If there are no modified fields, the ETX is the only character following the cursor address.
- If the block is not the last from the device, an ETB terminates the block. ETX or ETB terminates the last block.

STX	FIELD DATA	ETB ETX
-----	------------	------------

If the input block is not the first block of the transmission series, it has the above format (no AID or cursor address). ETX or ETB terminates the last block.

SOH	%	/	STX	TEXT	ETX
-----	---	---	-----	------	-----

The above format could be received if the operator pressed the Test Request key and the Binary Synchronous Test Facility is not included in ETAM.

STX	ETX
-----	-----

The above format could be received as the last block under certain unusual circumstances, and can be ignored.

Action: Issue Read Continue macro instructions until EOT is received from the 3275, ending its control of the line. The data blocks can then be concatenated and passed to the application modules, as discussed in "Remote Leased Line Event Completion Analysis" in this chapter.

Read Action Description Two

Explanation: A text block has been successfully received from the 3275 from a Read Initial, Read Continue, or Read Connect macro instruction. The input may have the following format:

STX	AID	Cursor Address	FIELD DATA	ETB ETX
-----	-----	-------------------	------------	------------

If the operator initiated the message other than with the Test Request key, the first block has the above general format. The maximum block length is 256 characters from STX to ETX or ETB. The block could be fewer than 256 characters even if there are subsequent blocks because the 3275 does not break an SEA sequence (3 bytes represent a buffer address).

The following variations in the above format are possible:

- If the Clear key or a Program Attention key was pressed, the ETX is the only character following the AID byte.
- If there are no modified fields, the ETX is the only character following the cursor address.
- If the block is not the last from the device, an ETB terminates the block. ETX or ETE terminates the last block.

STX	FIELD DATA	ETB
		ETX

If the input block is not the first block of the transmission series, it has the above format (no AID or cursor address). ETX or ETE terminates the last block.

SCH	%	/	STX	TEXT	ETX
-----	---	---	-----	------	-----

The above format could be received if the operator pressed the TEST REQUEST key and the Binary Synchronous Test Facility is not included in FIAM.

STX	ETX
-----	-----

The above format could be received as the last block under certain unusual circumstances, and can be ignored.

Action: Issue Read Continue macro instructions until EOT is received from the 3275, ending its control of the line. The data blocks can then be concatenated and passed to the application modules, as discussed in "Remote Leased Line Event Completion Analysis" in this chapter.

Read Action Description Three

Explanation: A 3275 sense/status message has been received (see "Sense/Status Analysis" in this chapter), indicating either a "busy" condition, a return to "ready" from "not-ready" status, or a hardware error condition at the terminal.

Action: Issue a Read Continue macro instruction to receive EOT from the 3275 and allow it to relinquish control of the line. Analyze the two sense/status bytes as outlined in "Sense/Status Analysis" in this chapter.

The message has the format:

SOH	%	R	STX	SENSE/STATUS BYTES	ETX
-----	---	---	-----	-----------------------	-----

Read Action Description Four

Explanation: The 3275 has responded ENQ to a Read Inquiry or Read Inquiry Monitor macro instruction, indicating the operator wishes to send a message.

Action: Issue a Read Continue to receive the first text block.

Read Action Description Five

Explanation: A completion condition unknown to the 3275 Dial support has occurred. This is probably caused by a programming error such as issuing an invalid macro instruction sequence.

Action: See Write Action Description Ten.

Read Action Description Six

Explanation: The 3275 has detected an internal buffer parity error (STX-ENQ received) or a malfunction other than a parity error, and has aborted its current transmission. This indicates that the 3275 has an error sense/status message pending, which you must retrieve and analyze before continuing.

Following Write Text, the 3275 responds with an EOT instead of an ACK-0, ACK-1, or NAK to indicate that a sense/status message is pending and must be cleared before any further operation can continue.

During the connect sequence, the 3275 sent the ID followed by a NAK instead of ACK-0 to indicate that the sense/status is pending and must be cleared.

Action: To reset the line, issue a Write Reset for OS. Then issue a Read Continue macro instruction to read the sense/status message from the 3275. See Read Action Description Three for the format of a sense/status message.

Read Action Description Seven

Explanation: While establishing a connection with the Read Initial or Read Connect macro instruction, the 3275 failed to transmit its ID-ENQ sequence (CPU answering operation) or ID-ACK-0 sequence (CPU called the terminal) within the TCU's timeout interval. This is probably a terminal hardware error.

Action: Issue a Write Disconnect to disconnect and disable the line, followed by another Read Initial or Read Connect, as appropriate.

Read Action Description Eight

Explanation: The 3275 has failed to transmit text within the TCU's time-out period in response to a Read Initial, Read Connect, or Read Continue macro instruction. This can occur due to incorrect operator procedures on Read Initial or Read Connect. You should retry the operation. On Read Continue, it is probably a hardware error and is not worth retrying.

Action: Issue a Write Disconnect to disable and disconnect the line, followed by another Read Initial or Read Connect, as appropriate. Alternately, for Read Initial or Read Connect, you might issue a Read Inquiry or Read Inquiry Monitor to await input, or issue a Write Inquiry to bid for the line and send a prompting message.

Read Action Description Nine

Explanation: The 3275 operator has failed to bid for the line within the TCU's timeout interval, following the Read Inquiry. This is a normal occurrence, depending on your usage of Read Inquiry and Write Reset instead of Read Inquiry Monitor and Write Reset Monitor, which cannot timeout.

Action: You may issue Write Reset or Read Inquiry to initiate another timeout interval, or use the Monitor form of either macro instruction to wait indefinitely for operator action. The Monitor form of the macro instruction may cause extended toll costs if the operator is not at the display station.

You may also issue a Write Disconnect (TD) to disconnect the line. This action may be appropriate after some specified interval has elapsed following a Read Inquiry Monitor Operation.

Read Action Description Ten

Explanation: The 3275 operator has requested immediate termination of the connection by pressing the Disconnect key on the 3275.

Action: Issue a Write Disconnect to disconnect and disable the line, followed by another Initial or connect-type macro instruction to establish another connection with the same or another operator.

Read Action Description Eleven

Explanation: The 3275 has responded with an invalid ID sequence or with ID-NAK to your Read Initial or Read Connect, indicating either a wrong terminal or a hardware error. In either case, further communication is not desired.

Action: If expanded ID verification is in use, BTAM automatically disconnects the line. Issue the Read Connect (TC) macro. If expanded ID verification is not in use, issue a Write Disconnect (TD) macro.

Read Action Description Twelve

Explanation: A completion condition unknown to the 3275 dial support has occurred; it is probably caused by a programming error such as issuing an invalid macro instruction sequence.

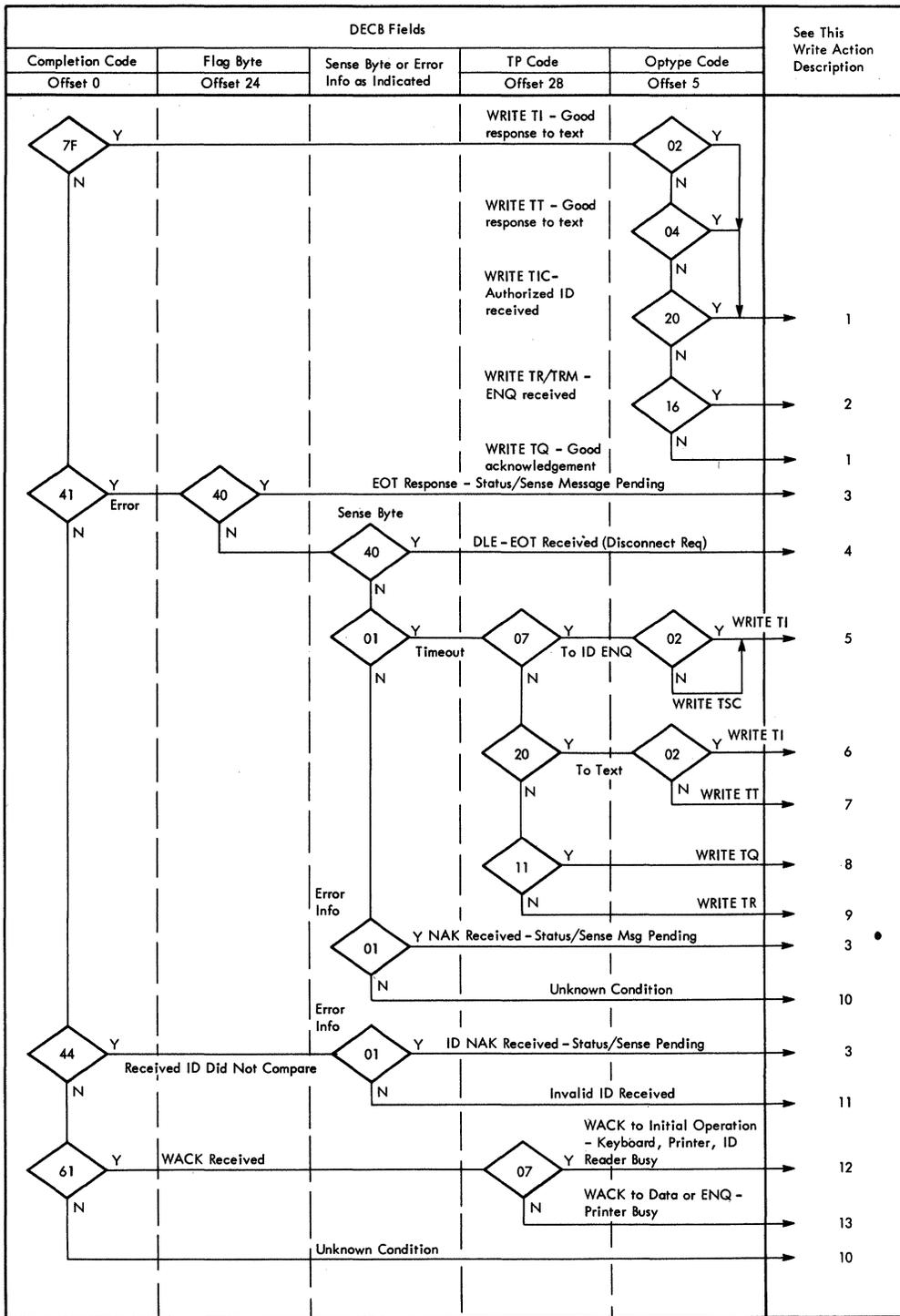


Figure 4-9. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Remote Dial Write Completion Analysis

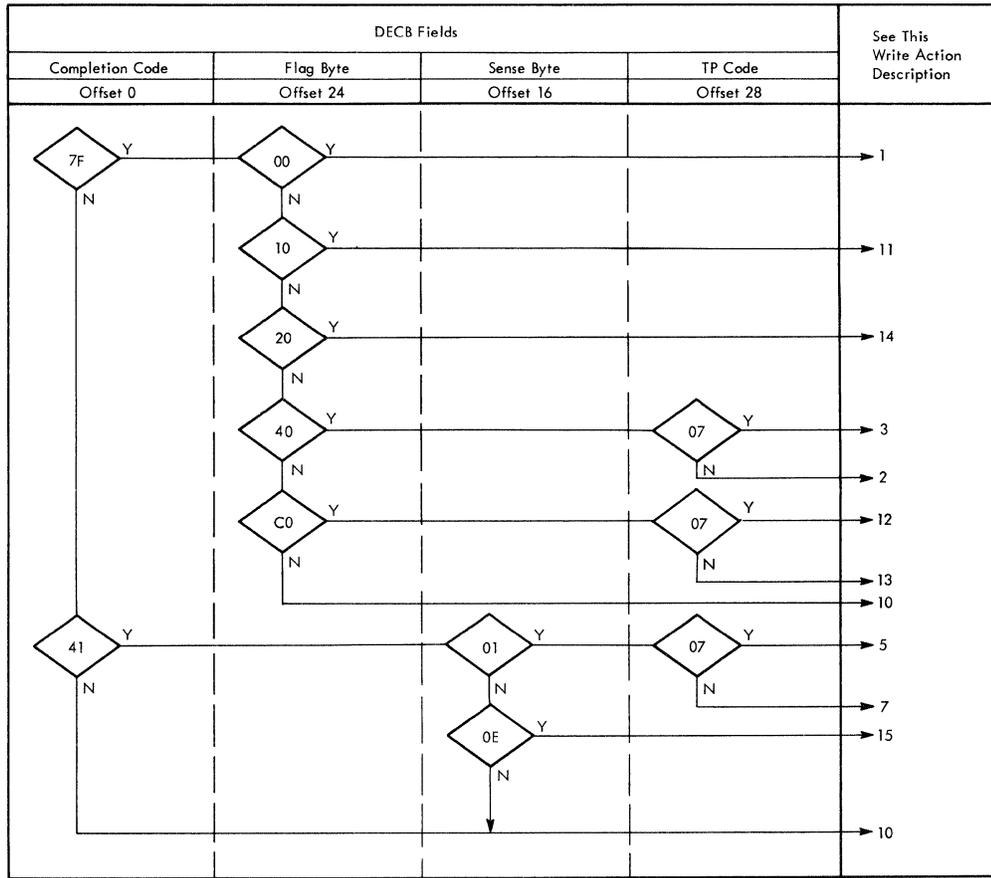


Figure 4-10. OS BTAM and OS/VS BTAM, Remote Dial Write Completion Analysis

Action: Obtain as complete a picture of current system status as possible, including, in order of importance: the DECB (with any user extensions you may have appended); DTFBT for the line group; any terminal-status tables you may have in your program; I/O buffers; and application program areas. These should be dumped to an external device for printing and analysis. You should then issue a Write Disconnect to disable the line, followed by an Initial or Connect macro instruction to prepare for further operation.

Read Action Description Thirteen

Explanation: An I/O error has occurred on a read operation. The error can be a result of lost data, data check, or data overrun. The text received is in error.

Action: Issue a Write Disconnect (TD) to disconnect the line. This is probably the most likely action because BTAM has already tried unsuccessfully the specified number of times.

As an alternative, you can issue a Read Repeat (TP) to reattempt to read the text from the 3275.

Write Action Description One

Explanation: The 3275 has responded positively to the last write operation on the line. The response means:

Write Initial: Text received properly
Write Continue: Text received properly
Write Connect: Proper terminal ID has been sent by the 3275;
terminal is now ready for data.
Write Inquiry: Positive response to CPU bid for the line;
3275 is ready to receive data.

Action: Issue a Write Continue to transmit text, or a Write EOT Monitor, or Write Reset/Write Reset Monitor to terminate the transmission.

Write Action Description Two

Explanation: The 3275 has responded ENQ to a Write EOT or Write EOT Monitor, indicating that it desires to transmit a message.

Action: Issue a Read Continue macro instruction to read the message.

Write Action Description Three

Explanation: The 3275 has responded with an ID NAK during connection and ID verification sequence. The 3275 is sending the ID for verification but indicates that it has a sense/status message pending which you must read before continuing.

If the connection process has been completed and a Write Inquiry solicits a NAK from the 3275, then the 3275 is again indicating that it has a sense/status message pending.

The third instance that could lead to this condition is a Write Initial or Write Continue issued during a Write Text sequence. A response of ECT from the 3275 instead of ACK-0 or ACK-1 indicates an outstanding sense/status message at the 3275.

Action: Issue a Write Reset macro instruction to reset the line; then issue a Read Continue macro instruction to read the sense/status message from the 3275. You should issue another Read Continue after receipt of the sense/status to allow the 3275 to transmit EOT and clear its status connection. The message has the form:

S			S	S	S	E
O	%	R	T	S	S	T
H			X	1	2	X

You should now analyze the two SS bytes, as described under the heading "Sense/Status Analysis" and take the appropriate corrective action.

Write Action Description Four

Explanation: The remote operator has requested immediate disconnection through use of the DISCONNECT switch on the display, and has probably already hung up the phone.

Action: Depending on data set options at the transmission control unit:

1. If the data set has Auto Disconnect and the device was dialed by the computer, issue a CONTROL Disable or a Write Break macro instruction on the same DECB.

Example: [symbol] CONTROL DECBNAME,TD,MF=E

2. If the data set does not have the Auto Disconnect Feature, or if the data set has Auto Disconnect and the device was not dialed by the computer issue a Write Disconnect macro instruction against the same DECB. Then issue a Write Initial or Write Connect to call another terminal, or a Read Initial or Read Connect to await another call.

Write Action Description Five

Explanation: While establishing a connection with the Write Connect or Write Initial macro instructions, the 3275 failed to respond to your program's transmitted ID sequence within the TCU 3-second timeout interval. This is probably a terminal hardware error; this particular terminal should probably not be called again until checked.

If the Write Initial has successfully established a connection with the 3275 (including receipt of the ID sequence), then this completion indicates that the 3275 has failed to respond to the text block transmitted as part of the Write Initial.

Action:

1. Issue a Write Disconnect macro instruction on the same DECB to disconnect and disable the line.
2. Issue another Write Initial or Write Connect macro instruction to the same terminal (if immediate retry is desired), or to another terminal.

Write Action Description Six

Explanation: After your Write Initial has successfully established a connection with the 3275 (including receipt of the ID sequence), the 3275 has failed to respond to the text block transmitted as part of the Write Initial.

Action: Same as Write Action Description Five.

Write Action Description Seven

Explanation: A text block was sent to the 3275 using a Write Continue, but the 3275 failed to acknowledge the block within the TCU's 3-second timeout interval.

Action:

A. Issue a Write Inquiry macro instruction to ask the 3275 to retransmit its last response. The response may have been garbled or a momentary line loss may have occurred. OS and OS/VS provide an area for the response to be read into. There are two ways to provide BTAM with pointers to this area:

1. The ENTRY operand technique. You provide a parameter list pointer with the ENTRY operand of the Write TQ. The parameter list contains the fullword address of the response input area, and a fullword constant containing the length of the input area (2 is sufficient). For example:

```
(symbol) WRITE DECBNAME,TQ,,,,PARMLIST,MF=E
PARMLIST DC A (RESPAREA) ADDR OF RESPONSE INPUT AREA
          DC F'2'          LENGTH OF RESPONSE INPUT AREA
RESPAREA DS CL2          RESPONSE INPUT AREA
```

2. The DECB extension technique. You provide the address and length of the response input area in the Write TQ macro instruction itself, which then stores the information in the DECB extension. For example:

```
(symbol) WRITE DECBNAME,TQ,,(,RESPAREA),(,2)MF=E
```

The method you use must agree with the specification of the ETMOD macro instruction DECBEXT parameter. If BTMCD specifies DECBEXT=YES, you must use the DECB extension technique; similarly, if you use the DECB extension technique, ETMOD must specify DECBEXT=YES. The BTMCD parameter choice may be dictated by other terminal types in the system. Conversational reads and writes to a 2770 require the DECB extension, and you would then have to use the same technique.

B. If the Write TQ completes successfully, you may send another block of data with Write Continue, or end with Write EOT. If Write TQ is not successful, issue a Write Disconnect followed by Write Initial or Write Connect, as desired.

Write Action Description Eight

Explanation: The 3275 has failed to respond to the Write Inquiry macro instruction; BTAM has retried the number of times specified in the DIFBI. Further attempts are probably useless.

Action: Same as Write Action Description Five.

Write Action Description Nine

Explanation: You issued a Write EOT or Write Reset macro instruction to relinquish control of the line, and the 3275 has not bid for control within 7 seconds for OS, or the number of seconds represented by 3 times the number of retries specified in DTFBT in DOS and DOS/VS.

Action: You should keep count of the number of consecutive occurrences of this type of time-out, and if a reasonable number is exceeded, disconnect the line and reinitialize for another call.

You could also issue a Read Inquiry Monitor, which will not time out, following this first completion of the Write EOT or Write Reset.

Alternately, you could issue a Write EOT Monitor or Write Reset Monitor, neither of which will time out, instead of the Write EOT.

If the Monitor macro instructions are used, be aware of the possibility an operator may leave the terminal without requesting disconnection, which could result in substantial line toll costs. You should perhaps set a timer interval when issuing a Monitor operation, and stop the operation with a RESETPL macro instruction if the operation is not complete within the time interval.

Write Action Description Ten

Explanation: A completion condition unknown to 3275 Dial support has occurred, which is probably caused by a programming error such as issuing an invalid macro instruction sequence.

Action: Obtain as complete a picture of current system status as possible, including, in order of importance: the DECB (with any user extensions you may have appended); DTFBT for the line group; any terminal-status tables you may have in your program; I/O buffers; and application program areas. These should be dumped to an external device for printing and analysis. You should then issue a Write Disconnect to disable the line, followed by an Initial or Connect macro instruction to prepare for further operation.

Write Action Description Eleven

Explanation: An ID sequence other than any specified in the DFTRMLST macro instruction has been received from the 3275 during a Write Initial or Write Connect operation.

Action: Issue a Write Disconnect to hang up and disable the line, followed by another Write Initial or Write Connect to enable the line, as appropriate.

Note: Because terminal ID lists are not reentrant and reusable, it may be appropriate to do an ID check against a common list at this point.

Write Action Description Twelve

Explanation: The 3275 has responded with WACK to your attempt to establish communications with a Write Initial or Write Connect. This indicates a "busy" condition involving the keyboard, operator ID card reader, or printer.

Action: Issue a Write Disconnect to disconnect and disable the line, followed by another Write Initial or Write Connect to establish connection with another terminal.

Write Action Description Thirteen

Explanation: The 3275 has responded WACK to your transmission of text. This condition occurs only if you had the Start-Print bit on in the Write Control character. WACK indicates that the text has been successfully received and that printing has started. No further text transfers are possible until the printing is finished.

Action: Issue a Write EOT Monitor or Write Reset Monitor to relinquish control of the line and monitor for completion. When this WRITE TRM completes, issue a Read Continue to receive the sense/status message from the 3275 indicating device end (sense/status bytes = X'C240'). On receipt of this message and its following EOT, you may initiate any new transmissions you desire, or monitor the line for terminal activity.

Write Action Description Fourteen

Explanation: The acknowledgment received from the 3275 in response to a write command is not the expected alternating acknowledgement (ACK-0 or ACK-1). A message may be lost.

Action: Issue a Write Inquiry (TQ) macro asking the 3275 to retransmit its last response (for example, ACK-0, ACK-1, or NAK).

If you are unable to reconstruct the message or resolve the sequence of acknowledgements, issue a Write Disconnect (TD) to disconnect the line.

Write Action Description Fifteen

Explanation: Data received from the 3275 in response to a conversational write is in error. The indicated cause of the error is lost data, data check or overrun.

Action: Issue a Read Repeat (TP) to reattempt to read the text. BTAM has already tried a specified number of times so if the condition continues, issue a Write Disconnect to disconnect the line.

LOCAL EVENT COMPLETION ANALYSIS

The information in this section should help you design or code the portion of a terminal control program which, upon completion of 3270 I/O operation, analyzes the circumstances of the completion and decides the proper action. A local 3270 includes a 3272, a 3274-1B, or a 3274-1D control unit.

The description of the 3270 local event completion analysis is organized in six parts. Four of the parts are flowcharts contained in Figures 4-11 through 4-14. These flowcharts are a logical sequence in which completion information can be analyzed after a Read or Write operation. The Erase/Write Alternate command performs the same function as the ERASE/WRITE command and causes a device with alternate buffer capability to enter alternate buffer mode. On local 3274-1D devices, the Prepare to Read command enables the display system to communicate with the host program so that, at selection time of a particular display or printer, the display system knows the next program action. The Write Structured Field (TSF) command causes a 3274 to enter Write Structured Field mode.

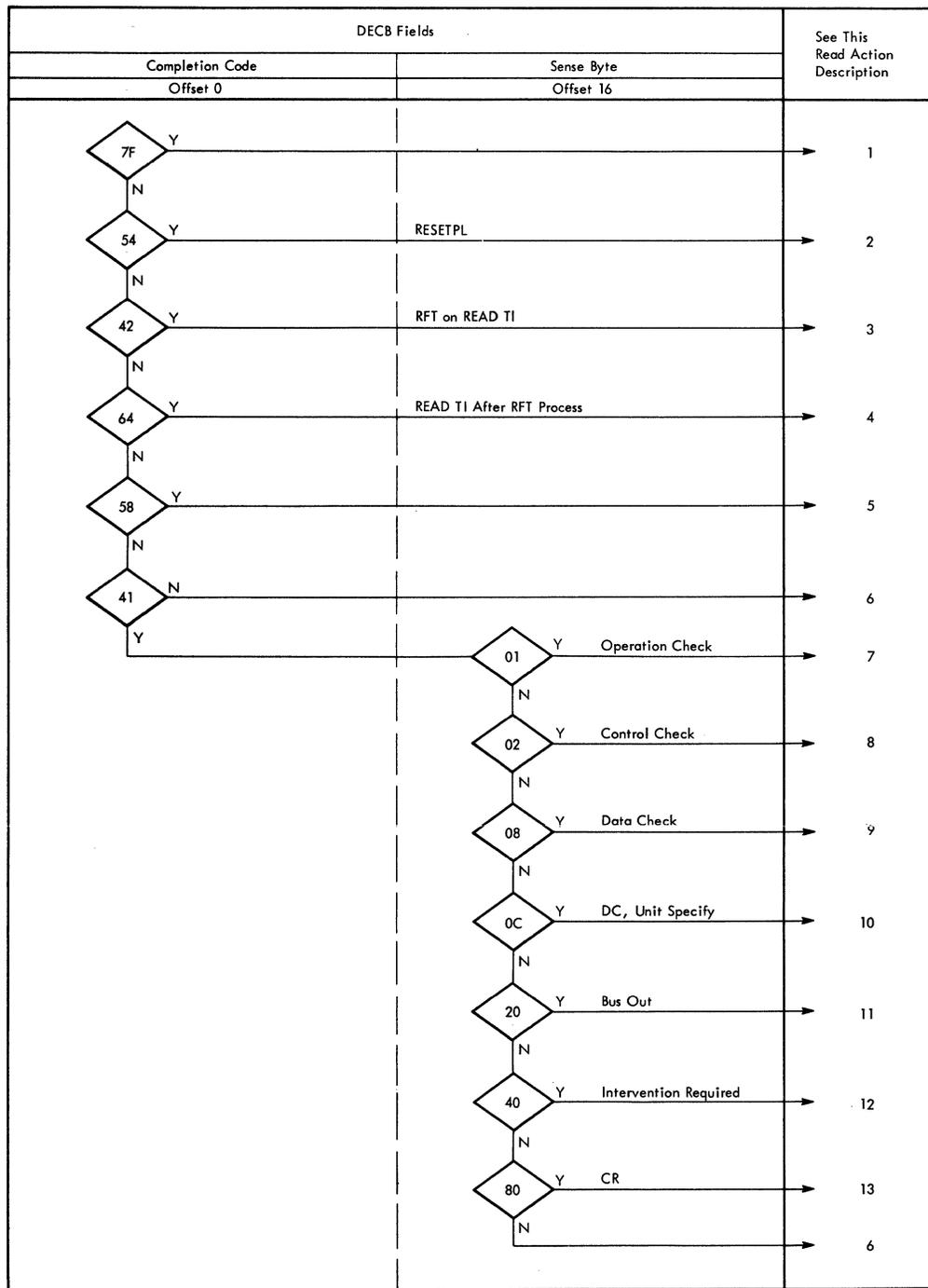


Figure 4-11. DOS BTAM, DOS/VS BTAM; and DOS/VSE BTAM-ES, Local Read Completion Analysis

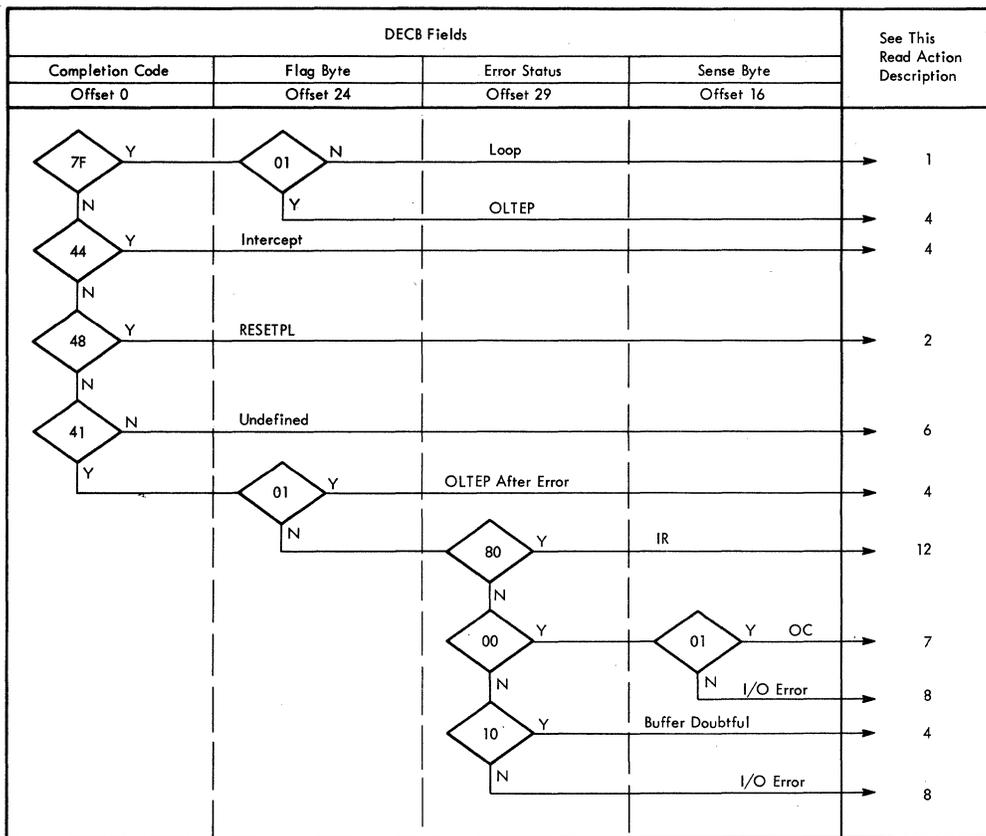


Figure 4-12. OS BTAM, OS/VSE BTAM, and DOS/VSE BTAM-ES, Local Read Completion Analysis

Figures 4-11 through 4-14 refer to the action descriptions that immediately follow. The action descriptions are in the following format:

- An explanation of the causes of the completion condition
- The advised actions and comments, where appropriate

Read Action Description One

Explanation: The Read operation has been completed successfully. The input message may take one of several formats.

AID	CURSOR ADDRESS	FIELD DATA	or	AID	Structured Field(s)
				X'88'	

If the operator has initiated the transaction other than with the TEST REQUEST key, the message has the above general format. The following variations are also possible:

- If the CLEAR key or Program Attention key was pressed, the AID byte is the only character received.
- If there are no modified fields and the operation was a Read Modified, only the AID byte and cursor address are received.

Action: The action taken depends on the line control program. You may:

- Check for the availability of output and, if present, write to a terminal with a Write Initial (TI), Write Erase (TS), Erase/Write Alternate (TSA), or Write Structured Field (TSF) or issue a Write Unprotected Erase (TUS) to the same format again. Write Unprotected Erase does not transmit text.
- If no output is available, issue a Read Initial (TI) so that the next operator action is recognized.

Read Action Description Two

Explanation: The previously initiated Read Initial (TI) was terminated without message receipt, by a RESETPL macro instruction.

Action: The appropriate action depends on the line control program. You may:

- If there is output for the control unit, issue a Write Initial (TI), Write Erase (TS) or Erase/Write Alternate (TSA) to send the message.
- Resume the Read Initial (TI).
- Issue a CLOSE macro instruction to terminate operations on the control unit.

Read Action Description Three

Explanation: A REQUEST-FOR-TEST message has been received in response to a Read Initial (TI), and a TWAIT macro instruction has not been issued.

Action: Issue a TWAIT macro in the form:

```
TWAIT (R1),TERMTST,(R2)
```

(R1) is a register that contains the address of the DECE, which is posted as complete on the satisfaction of the TWAIT; (R2) is loaded with the address of the DECE, which is posted with the X'42' completion code. The completion code should not be altered prior to the issuance of the TWAIT.

If the online terminal test facility is not available, the REQUEST-FOR-TEST message might be processed the same as a CLEAR key depression.

Read Action Description Four

Explanation: The contents of the 3270 buffer are unreliable because of previous processing of a REQUEST-FOR-TEST message.

Action: The entire buffer must be reinitialized with a Write Erase (TS) or Erase/Write Alternate (TSA). This could require maintaining an image of the device buffer during processing because the buffer may be the cumulative result of multiple I/O transactions.

Read Action Description Five

Explanation: ETAM has detected a cancel condition.

Action: Take a PDUMP or SNAP dump of the system. Issue an operator awareness message and terminate the system after recording checkpoint/restart information, if required.

Read Action Description Six

Explanation: This is an unrecognized error condition that should not occur. The probable cause is a program error.

Action:

- Take a PDUMP or SNAP dump of the system for analysis.
- Notify the operator of the condition.
- Issue a Read Initial (TI) to resume input.

Read Action Description Seven

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected an operation check. Possible causes are:

- The data stream transmitted as the result of a Read Modified from Position (TMP) or Read Buffer from Position (TBP) contains an illegal buffer address.
- The data streams contain a Write Control Character that specified start print.

Action:

- Take a SNAP dump or PDUMP of the offending data streams for analysis.
- Issue a transaction-cancelled message to the terminal.
- Inform the system operator of the occurrence.

Read Action Description Eight

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a control check condition. The addressed device failed to perform an operation or respond to the control unit within a period of time determined by the control unit.

Action:

- Retry the failing operation the specified number of times.
- Notify the operator of the occurrence.
- Mark the terminal as out of service.

- Remove the terminal from the line group by issuing a CHGNTRY macro instruction of the form:

(for OS and OS/VS
CHGNTRY (R1),ATTLST,(R2),,SKIP

(for DOS and DOS/VS)
CHGNTRY (R1),ATTLST,(R2),SKIP

R1 is loaded with the address of the DTF/DCB and R2 is loaded with the relative line number of the terminal.

- After repairs have been made, the terminal can be placed back in service in response to an operator command through the use of a CHGNTRY macro instruction with the activate parameter.

Read Action Description Nine

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a data check.

Action: Same as Read Action Description Eight.

Read Action Description Ten

Explanation: A printer or display has detected a data check condition.

Action:

- The entire device buffer must be reconstructed with an Erase Write (TS) or Erase Write Alternate (TSA) command.
- If desired, the failing operation may then be retried. This may require maintaining an image of the current buffer content which may consist of several I/C operations.
- It may be preferable to issue a Erase Write (TS) or Erase/Write Alternate (TSA) command indicating that the transaction has aborted.
- Proceed as in Read Action Description Eight.

Read Action Description Eleven

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a bus-out check (incorrect parity on a command or data received from the channel).

Action: Same as Read Action Description Eight.

Read Action Description Twelve

Explanation: The addressed device is unavailable (powered off or not operational).

Action:

- Notify the system operator of the condition.
- Take the terminal out of service. See Read Action Description Eight for an example of the CHGNTY macro instruction.
- Reissue the failing macro instruction if it was Read Initial (TI).

Read Action Description Thirteen

Explanation: The 3272, 3274 Model 1B, or Model 1D has detected an invalid command.

Action: Same as Read Action Description Seven.

Write Action Description One

Explanation: The Write operation has been posted complete without error. For a DOS, DOS/VS, or DOS/VSE printer, this only signals that the print operation has begun. Print errors which occur after channel end are posted through a return code '30' on the subsequent Read or Write operation.

Action: Check for the availability of additional output for other terminals on this control unit and send a Write Erase or Erase/Write Alternate (TSA) or Write Initial (TI). Issue a Read Initial (TI).

Note: Channel end posting for printers releases the DECB after the print cycle. Separate DECBS are recommended for printer operations to allow the application program to perform I/O operations to other devices while print operations are in progress. The separate DECBS avoid extended waiting for completion of a print operation.

Write Action Description Two

Explanation: BTAM has detected a cancel condition, which is probably a user program problem.

Action: Same as Read Action Description Five.

Write Action Description Three

Explanation: This is an undefined error that should not occur, probably caused by a program error.

Action: Same as Read Action Description Six.

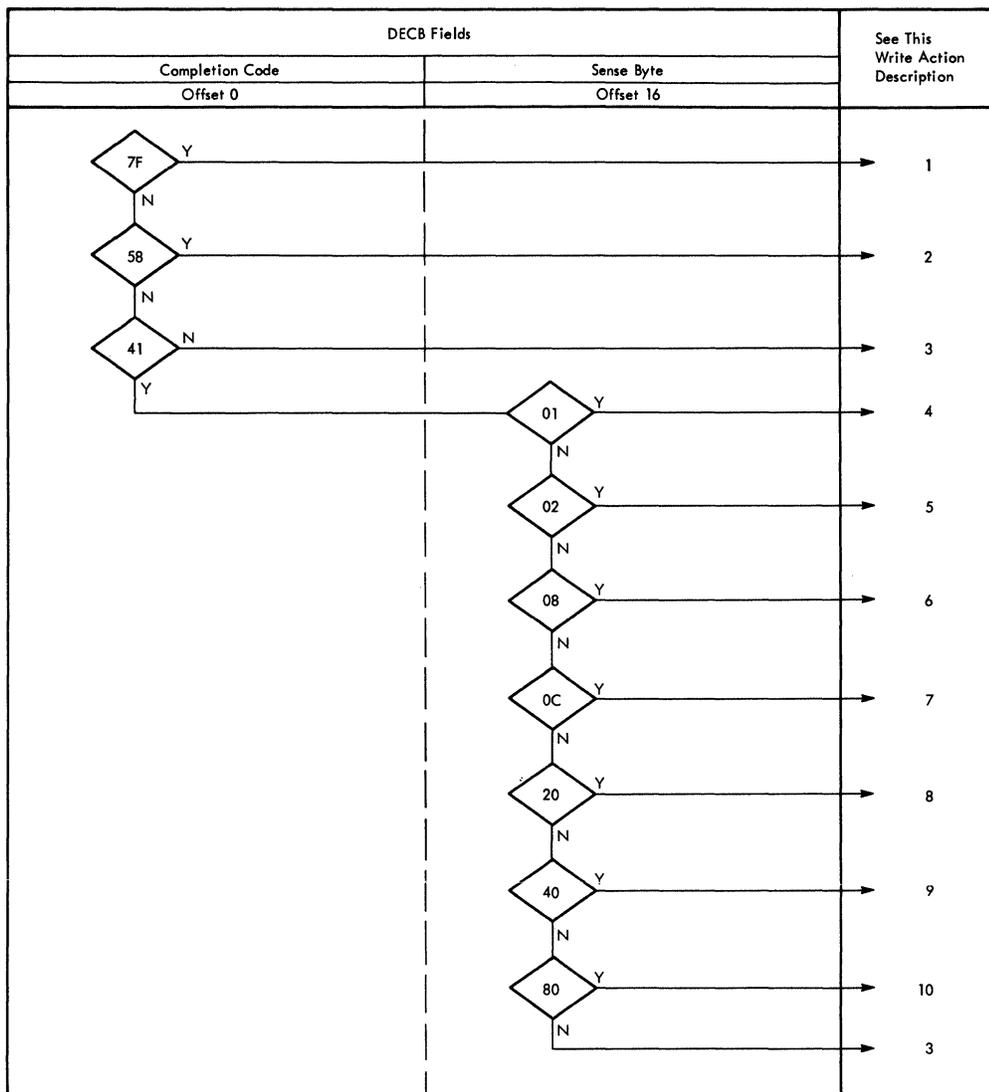


Figure 4-13. DOS BTAM, DOS/VS BTAM, and DOS/VSE BTAM-ES, Local Write Completion Analysis

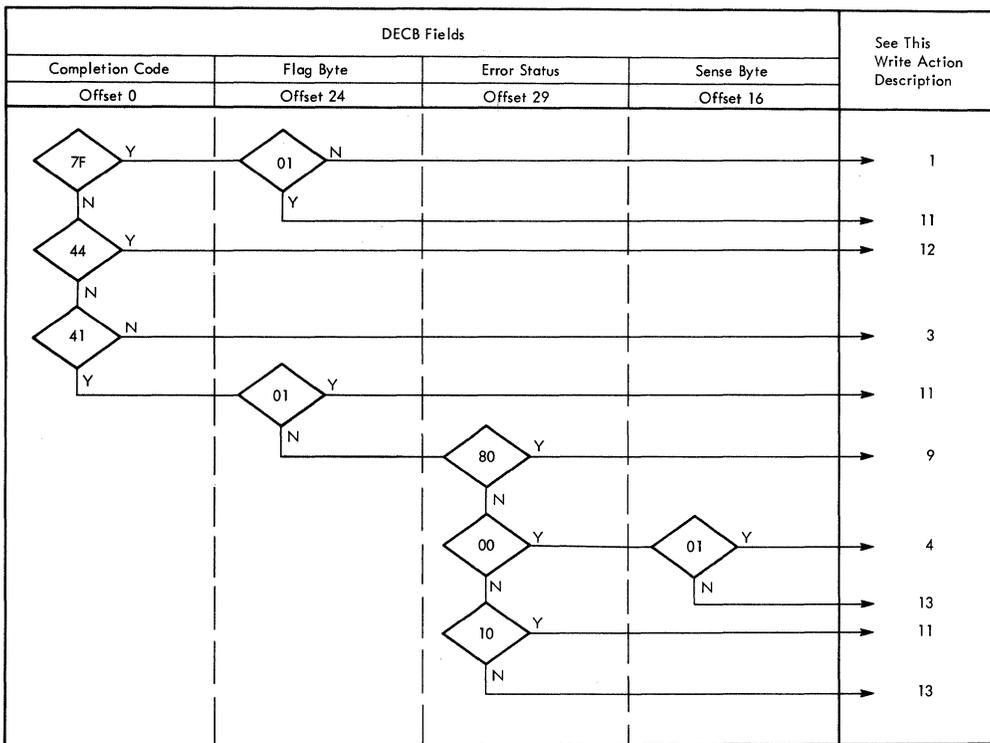


Figure 4-14. OS BTAM and OS/VS BTAM, Local Write Completion Analysis

Write Action Description Four

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected an operation check. Possible causes are:

- The data stream contains an invalid buffer address.
- The data stream ends prior to the completion of an order sequence.

Action: Same as Read Action Description Seven.

Write Action Description Five

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a control check. The device failed to perform an operation or respond to the 3272 or 3274 in the time allowed by the control unit.

Action: Same as Read Action Description Eight.

Write Action Description Six

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a data check. This is a hardware error.

Action: Same as Read Action Description Eight.

Write Action Description Seven

Explanation: A printer or display has detected a data check. This is a hardware error.

Action: Same as Read Action Description Ten.

Write Action Description Eight

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected a bus-out check (incorrect parity on a command or data received from the channel).

Action: Same as Read Action Description Eight.

Write Action Description Nine

Explanation: The addressed device is unavailable (powered off or not operational).

Action: Same as Read Action Description Twelve.

Write Action Description Ten

Explanation: The 3272, 3274 Model 1B, or 3274 Model 1D has detected an invalid command. This could be due to an ERASE/WRITE ALTERNATE command or a WRITE STRUCTURED FIELD command's being sent to a 3272 or 3274-1B, or to a PREPARE TO READ command's being sent to a 3272 or 3274-1B.

Action: Same as Read Action Description Seven.

Write Action Description Eleven

Explanation: The device buffer is unreliable because of diagnostic testing after completion of the previous output operation.

Action: Same as Read Action Description Four.

Write Action Description Twelve

Explanation: The input/output request was rejected because an error occurred following the previous operation or request-for-test processing.

Action: See Read Action Description Four.

Write Action Description Thirteen

Explanation: A permanent I/C error has occurred.

Action: Same as Read Action Description Eight.

SENSE/STATUS ANALYSIS

Unlike previous terminal systems which told you only that an error had occurred, the 3270 Information Display System has a self-diagnosis system to inform the central site of error or completion conditions. In remote configurations this sense/status information is communicated in a special message format as illustrated in the Read Completion Analysis sections (Read Action Description Three for remote dial, Read Action Description One for remote leased multipoint).

Proper analysis and use of the sense/status bytes may improve system availability. In many cases, a specific retry operation can correct an error condition and allow normal system operation to proceed. Also, conditions requiring manual intervention, such as a powered-down display or printer, or a printer out of paper, can be quickly identified. Proper personnel can be notified to correct the situation. Serious hardware malfunctions may be identified, logged out, and communicated to proper maintenance personnel as an aid in diagnosing and correcting the problem. For a detailed description of the Sense/Status Analysis table, see the IBM 3270 Information Display System: Component Description, GA27-2749.

CHAPTER 5. TCAM SUPPORT

Another access method that can be used in the telecommunications management of the 3270 display system is the Telecommunications Access Method (TCAM). TCAM controls data transfer between main storage and local or remote display stations. It operates under OS/VS1 and OS/VS2 and supports the 3270 display system in areas such as device scheduling, diagnostic testing, and error handling and recording. (See Figures 5-1 through 5-4.)

This chapter presents only an overview of TCAM support for the 3270; it does not give a complete description of TCAM support. The following TCAM publications provide detailed information about TCAM:

OS/VS TCAM System Programmer's Guide TCAM Level 10, GC30-2051

OS/VS TCAM Macro Reference Guide TCAM Level 10, GC30-2052

Operator's Library: OS/VS TCAM (Level 10), GC30-3037

Advanced Communications Function for TCAM, System Programmer's Guide, SC30-3117

Advanced Communications Function for TCAM Macro Reference Guide, SC30-3118

Advanced Communications Function for TCAM, Prog Prod, Operator's Guide, SC30-3123

Advanced Communications Function for TCAM, Version 2 Installation; Guide, SC30-3132

Advanced Communications Function for TCAM, Version 2 Installation; Reference SC30-3133

Advanced Communications Function for TCAM, Version 2 Operation, SC30-3136

TCAM addresses, polls, and manages binary synchronous line control (BSC) and synchronous data link control (SDLC) according to the user's definition. Read modified is the only TCAM read operation for the remote 3270.

DEFINING THE 3270 NETWORK

The following sections discuss some of the TCAM macro instructions and operands that affect the 3270; for complete information about their use, see the TCAM publications cited above. Figure 5-5 indicates which TCAM macros are used to define 3270 stations according to their attachment mode. The switched 3275 is not supported by TCAM.

	Non-SNA Local	BSC		SNA		
		Non- NCP	NCP	Local	LU.T0	LU.T2
TCAM10 Direct ¹	X	X	X	—	X	—
ACF/TCAM Version 0	X	X	X	—	X	X
ACF/TCAM Version 2 ²	X	X	X	X	X	X
TCAM10 Direct with TSO ^{1,3}	X	X	X	—	X	—
ACF/TCAM/TSO Version 0 ³	X	X	X	—	X	X
ACF/TCAM/TSO Version 2 ³	X ²	X	X ²	X ²	X ²	X ²

Note: The switched 3275 is not supported by any release of TCAM.

¹ The 3278 Model 5 is not supported.

² The extended 3270 data stream is supported.

³ The 3270 printers are not supported by TCAM/TSO.

Figure 5-1. TCAM Release Support of 3270 Attachment Modes

	OS/VS1	OS/VS2	
		SVS	MVS
TCAM10 Direct	X	X	X
ACF/TCAM Version 0	X	X	X
ACF/TCAM Version 2	X	—	X
TCAM10 Direct with TSO	—	X	X
ACF/TCAM/TSO Version 0	—	X	X
ACF/TCAM/TSO Version 2	—	—	X

Figure 5-2. TCAM Release Support under OS/VS1 and OS/VS2.

	DCB API	TSO (TIOC) API	Record API
TCAM10 Direct	X	X	—
ACF/TCAM Version 0	X	X	—
ACF/TCAM Version 2 Release 1	X	X	—
ACF/TCAM Version 2 Release 2	X	X	X

Figure 5-3. TCAM Release Support of Application Program Interface Types

	Non-SNA Local	BSC		SNA		
		Non-NCP	NCP	Local	LU.T0	LU.T2
TCAM10 Direct	-		-	-	-	-
ACF/TCAM Version 0	-		X	-	X	X
ACF/TCAM Version 2 Release 1	-		X	X	X	X
ACF/TCAM Version 2 Release 2	X		X	X	X	X
TCAM10 Direct with TSO	-		-	-	-	-
ACF/TCAM/TSO Version 0	-		X	-	X	X
ACF/TCAM/TSO Version 2 Release 1	-		X	X	X	X
ACF/TCAM/TSO Version 2 Release 2	X		X	X	X	X

Figure 5-4. TCAM Release Support of Cross-Domain Networking for 3270 Attachment Modes

	Non-SNA Local	BSC		SNA		
		Non-NCP	NCP	Local	LU.T0	LU.T2
DCB	X	X	-	X	-	-
INVLIST	X	X	-	-	-	X ¹
TERMINAL	X	X	X	X	X	X
GROUP	-	-	X	X	X	X
OPTION	X	X	X	X	X	X

¹Switched lines only.

Figure 5-5. TCAM Macros Used to Define the 3270 Device According to Attachment Mode

TERMINAL MACRO INSTRUCTION

The **TERMINAL** macro instruction defines the cluster control unit and the devices attached to it (for ESC and SNA). It is only required for 3270 control units if general polling is specified.

Figure 5-6 gives the **TERMINAL** macro operands used to define 3270 devices according to attachment mode.

Note: The **USS** operand is the only distinguishing operand for LU.T0 or LU.T2

To specify LU.T0 code USS=3270
 To specify LU.T2 code USS=SCS

	Non-SNA	BSC		SNA		
	Locals	Non-NCP	NCP	Local	LU.T0	LU.T2
ACTIVE	-	-	O	O	O	O
ADDR	-	R	-	-	-	-
ALTDEST	O	O	O	O	O	O
BFRPAD	-	-	-	O	-	-
BUFSIZE	O	O	O	O	O	O
CALL ¹	-	-	-	-	-	O
CDABLE	-	-	O	-	-	-
CINTVL ¹	-	-	-	-	-	O
CLOCK ¹	-	-	-	-	-	O
DCB	R	R	-	R	-	-
DIALNO ¹	-	-	-	-	-	O
ENCRPT	-	-	-	O	-	O
FEATURE	O	O	O	O	O	O
GROUP	-	-	R	R	R	R
LEVEL	O	O	O	O	O	O
LLAT	-	-	-	R	-	-
LMD	O	O	O	O	O	O
LOCADDR	-	-	-	R	-	-
MAXBFRU	-	-	-	O	-	-
MAXDATA ¹	-	-	-	-	-	O
MAXOUT ¹	-	-	-	-	-	O
OPDATA	O	O	O	O	O	O
PASSLIM ¹	-	-	-	-	-	O
QBY	R	R	R	R	R	R
QUEUES	R	R	R	R	R	R
RETRY ¹	-	-	-	-	-	O
RLN	R	R	R	R	R	R
SCRSIZE	O	O	O	O	O	O
SECTERM	O	O	O	O	O	O
SPECOUT	-	-	O	-	-	-
TBLKSZ	-	O	-	-	-	-
TCMSESN	-	-	-	O	O	O
TERM	R	R	R	R	R	R
USS	-	-	-	R	R	R
UTERM ¹	-	-	-	-	-	O

O = optional R = required - = not used

Note: Not all these operands are valid for every release of TCAM. Consult the macro reference guide for your release of TCAM to determine which operands can be used.

¹For switched lines only.

Figure 5-6. TERMINAL Macro Operands Used to Define 3270 Devices According to Attachment Mode

INVLIST MACRO INSTRUCTION

The INVLIST macro instruction generates the polling list for BSC non-NCP 3270s, the type of read operation for 3270 non-SNA locals, and the IDs for SNA switched 3270s. Each INVLIST macro defines the polling characters for all devices on a BSC non-NCP line. TCAM requires a specific poll entry for every 3270 printer and display, whether polled or not. If general polling is desired, an ORER= suboperand specifying the general poll characters for each BSC cluster control unit must also be included in the INVLIST macro specification.

IEDRH MACRO INSTRUCTION

The IEDRH macro instruction is used for LU.2 SNA 3270s to set Begin Bracket, End Bracket, and Change Direction flag bits in the SNA Request Header in outgoing messages.

GROUP AND DCB MACRO INSTRUCTIONS

TCAM has the concept of "line groups," sets of lines that have common characteristics and that permit operator control of a collection of lines with single commands.

For non-NCP BSC and non-SNA local stations, line groups must be defined for lines to which 3270 devices are attached with the line group DCB macro instruction. Figure 5-7 indicates which DCB macro operands are used to define such line groups.

For NCP BSC and all SNA 3270 devices, line groups are defined with the GROUP macro instruction. Figure 5-8 indicates which GROUP macro operands are supported for various modes of 3270 attachment.

The DCB macro is also used to define 3705 controllers operating in NCP mode and SNA local clusters. The principal information supplied by the DCB macro for these types of resources is the DDNAME of a JCL DD card, which defines the subchannel to which the physical unit is attached. See the TCAM publications for further information on this use of the DCB macro.

DEFINING LOCAL NON-SNA 3270 CLUSTERS

Each non-SNA local 3270 control unit is considered to be a line group and therefore requires a DCB macro. Each device attached to the control unit is treated as a "line," since it has its own subchannel address, and therefore requires a unique RLN= operand value on the TERMINAL macro representing the device. A separate INVLIST macro must be coded for each station on the control unit.

Dynamic buffering is not supported for the local non-SNA 3270. FCI=(N,N) must be used.

If you are defining a local 3270 that uses the 3278, 3279, 3287, or 3289, you should note that they must be specified as some other 3270 device of the same type (for example, the 3278 display must be specified as a 3277 display).

	Non-SNA	BSC		SNA		
	Locals	Non-NCP	NCP	Local	LU.T0	LU.T2
BUFIN	O	O				
BUFMAX	O	O				
BUFOUT	O	O				
BUFSIZE	O	O				
CPRI	O	O				
DDNAME	R	R				
DSORG	R	R				
EXLST	O	O				
INTVL	-	O				
INVLIST	R	R				
MACRF	R	R				
MH	R	R				
PCI	R	O				
RESERVE	O	O				
SCT	O	O				
TRANS	R	R				

O = optional R = required - = not used

Note: Not all these operands are valid for every release of TCAM. Consult the macro reference guide for your release of TCAM to determine which operands can be used.

Figure 5-7. LCB Macro Operands Used to Define 3270 Devices According to Attachment Mode

	Non-SNA Locals	BSC		SNA		
		Non-NCP	NCP	Local	LU.T0	LU.T2
BUFMAX			O	O	O	O
BUFOUT			O	O	O	O
BUFSIZE			O	O	O	O
INVLIST			-	-	-	O
MH			R	R	R	R
OPACING			-	O	-	O
PACK			O	O	O	O
PCI			O	O	O	O
TRANS			O	O	O	O
RAPIMH			O	O	O	O

O = optional

R = required

- = not used

Note: Not all these operands are valid for every release of TCAM. Consult your TCAM macro reference guide for your release of TCAM to determine which operands can be used.

Figure 5-8. GROUP Macro Operands Used to Define 3270 Devices According to Attachment Mode

SAMPLE RESOURCE DEFINITIICNS

Following are sample definitions for each type of 3270 display attachment supported by TCAM. In each sample are defined a line group, line, cluster controller or physical unit, one device or logical unit, and an invitation list (if necessary). In real-life configurations, a line group includes several lines, each line has several cluster controllers or physical units attached to it, and each cluster controller or physical unit has many devices or logical units attached to it.

Remote LU Type 2

```
GRP3274      GROUP      BUFOUI=2,BUFMAX=2,MH=SNAEDSMH,PCI=(,N),
                RESERVE=(30,27),OPACING=5,
                TRANS=EBCD,BUFSIZE=2000

HDX3274      TERMINAL   AL RLN=1,
                GROUP=GRP3274,
                TERM=LINE,ACTIVE=NO

HP13274      TERMINAL   AL TERM=PUNT,
                ACTIVE=NO
```

HL13274 TERMINAL AL RLN=1,ACTIVE=(YES,NO),USS=SCS,
TERM=LUNT,TCMSESN=NORMAL,FEATURE=(NOBREAK,NOATTN),
GROUP=GRP3274,SCRSIZE=(24,80,43,80),
QBY=T,LEVEL=(0,255),
QUEUES=DRT,
SECTERM=YES,
LMD=YES

Remote LU Type 0

GRP3270 GROUP EUFOUT=2,BUFMAX=2,BUFSIZE=2000,MH=SNAEDSMH,
PCI=(,N),TRANS=EBCD
HDX3270 TERMINAL RLN=1,GROUP=GRP3270,TERM=LINE,ACTIVE=NO
HP13270 TERMINAL TERM=PUNT,ACTIVE=NO,FMPROF=0,TSPROF=1
HL03270 TERMINAL RLN=1,ACTIVE=(YES,NO),USS=3270,TERM=LUNT,
TCMSESN=NORMAL,FEATURE=(NOBREAK,NOATTN),
GROUP=GRP3270,SCRSIZE=(24,80),QBY=T,
QUEUES=DRT,SECTERM=YES,LMD=yes

Non-NCP Remote ESC

DCB3274R DCB LSCRG=TX,MACRF=(G,P),DDNAME=DD3274E,EUFSIZE=2000,
INVLIST=INV3274R,MH=BSCEDSMH,TRANS=EBCD,
RESERVE=(30),INTVL=7,BUFOUT=2,PCI=(,N),SCT=EBCD,
BUFIN=2,BUFMAX=2,CPRI=S
INV3274R INVLIST CRDER=(EP3274CL-40407F7F2D,EP3274ST+404040402D)
EOT=37,AUTO=NO
EP3274CL TERMINAL QBY=1,TERM=327C,DCB=DCB3274R,RLN=1
EP3274ST TERMINAL QUEUES=DNT,QBY=T,SECTERM=YES,TERM=327R,
LEVEL=(0,255),DCB=DCB3274R,NTELKSZ=3500,
TELKSZ=2000,ADDR=606040402D,RLN=1,
SCRSIZE=(24,80,43,80)

Non-SNA Local

DCE3274L DCB DSORG=TX,MACRF=(G,P),DDNAME=DD3274L,
EUFOUT=1000,MH=BSCEDSMH,TRANS=EBCD,EUFMAX=10,
INVLIST=(INVYAL),RESERVE=(30),BUFIN=10,BUFOUT=10,
PCI=(N,N),SCT=EECL,CPRI=S
INVYAL INVLIST ORDER=(YAL=+06)
YAL TERMINAL QUEUES=MRT,QBY=T,SECTERM=YES,TERM=327L,
LEVEL=(0,255),DCB=DCB3274L,RLN=1,
SCRSIZE=(24,80,43,80)

MESSAGE HANDLING

TCAM message handler macro instructions can be used for editing and manipulating of data when the data arrives and before transmission. The user, however, must design and control the panels and handle the data stream.

For all but non-SNA local 3270s, TCAM always reads data with a Read Modified command. For non-SNA local 3270s, TCAM uses either a Read Modified or a Read Buffer command, depending on how the INVLIST macro is coded.

Output to a 3270 may originate either in an application program or in a message handler.

Normally, a TCAM application program generates all 3270 data streams as if they were going to be sent to an NCP-attached remote BSC 3270 (Escape character, command character, WCC character, data, and orders). In the message handler, the macros described below and other message handler macros are used to convert the data stream to the format expected by the destination station. For each type of attachment, a brief description of the conversions necessary follows:

- Non-SNA Locals: Set up the channel command with the RETRV function of the SCREEN macro instruction; then delete the Escape and remote command characters with the MSGEDIT macro instruction.
- SNA Clusters: Delete the Escape character with the MSGEDIT macro instruction.
- NCP Remote BSC: No conversion necessary.
- Non-NCP Remote ESC: Use the MSGFORM macro instruction to frame the data stream with STX and ETX characters.

In the input side of the message handler, control unit and device ID characters are deleted from the data streams sent in by ESC 3270s. Once this is done, input data streams have the same format regardless of station attachment mode.

Certain message handler macros perform special or critical functions for 3270 data streams. These macros are described below.

MSGFORM Macro Instruction

The MSGFORM macro instruction is used in output message handlers for the following types of attachment, as indicated:

- For non-NCP remote ESC 3270s
Supplies the framing characters (STX, ETX) for a data stream if it does not already include them, and breaks messages into blocks (ending in ETB).
- For remote BSC extended 3270 data streams
Causes the data stream to be sent in transparent mode.

The MSGFORM macro is not used for local or SNA 3270 devices.

SCREEN Macro Instruction

The SCREEN macro instruction is used in the output message handler for the following types of attachment, as indicated:

- For local non-SNA 3270s
To set up the channel command used to write the data stream that corresponds to the remote command found in the data stream.
- For all 3270s
To set the command used to write the data stream to the command specified on the SCREEN macro instruction.

MSGLIMIT Macro Instruction

For non-NCP attached ESC 3270 devices, code the MSGLIMIT macro in the input message handler. On input, MSGLIMIT 1 forces TCAM to the next entry on the polling list. If MSGLIMIT 1 is not coded and device status is generated, a loop could result with specific polling because TCAM repolls the device.

HANDLING REMOTE PRINTERS

In a normal print operation, an application program sends a message for a non-NCP BSC 3270 printer to the TCAM message control program, which places the message on the printer queue. When the message has arrived at the printer buffer, the remote 3271, 3274, 3276 returns a WACK. TCAM marks the message as complete, and the buffer's contents are dumped to the printer. When the printing is completed, the printer sends device end status to indicate that it can now accept the next print command.

If a printing error occurs, some portion of the message may be lost, even though TCAM has marked the message as complete. You must provide code in the input message handler or application program that scans input SOH%R sense/status messages, recognizes printer error conditions, and resends messages that may not have printed successfully.

If another message arrives at the printer queue while a printout is occurring, TCAM tries to send the message. The response is a WACK, and TCAM merely reschedules the message without informing your message-handling program.

This is, in effect, a loop until the message currently being printed is completed. You should ensure that only one message is sent to a printer at a time, either by centrally routing all printer messages through one application program or by holding the printer's queue after a message is sent until Device End status is received.

HANDLING OTHER TYPES OF PRINTERS

There are similar considerations for sending messages to 3270 printers attached to TCAM in the other attachment modes supported. See the TCAM manuals for more information.

HANDLING THE AID BYTE

Pressing any of the PA or PF keys, the ENTER key, or the CLEAR key transmits a message to TCAM. Pressing a PA key or the CLEAR key sends only the AID byte, followed by ETX. Pressing the PF key or ENTER key sends the AID byte and all data that has the modified data tag set.

HANDLING SENSE/STATUS CONDITIONS

The 3270 sends a sense/status message to the incoming message handler program when any of the following occur:

- TCAM addresses the device during the 2 milliseconds that the device is busy after the operator presses a key.
- TCAM is sending data to a device and an error occurs.
- TCAM is polling and a device is turned off and then turned on again.
- A printer ends a print operation, with or without an error.

TCAM retries addressing errors six times. A control unit's line control responses tell TCAM when status is pending.

If TCAM addresses a device and receives a Reverse Interruption (RVI), or sends data and receives an EOT, sense/status is pending for the device. TCAM sets the appropriate Message Error Record (MER) bits for a permanent error and sends the message through the output message handler (for an RVI) or a zero-length error buffer through the output message handler (for an EOT). The reason for the error message is not yet known, but you should issue a HOLD so that the current message is not lost and additional I/C is not attempted until the sense/status message is interrogated.

TCAM stops all activity on the line and does a specific poll of the device in question, referring to the invitation list. The sense/status message is posted to the message handler with the SOH%EC or SOH%R bit set in the message error record. You must analyze the sense/status message and take corrective action, as discussed below.

If TCAM cannot poll successfully, a zero-length buffer is sent through the message handler. You should take action forcing TCAM to stop communicating with the device until the problem is fixed. All sense/status messages appear at the incoming message handler.

If the device is busy and TCAM receives a WACK in response to addressing, the operation is rescheduled. However, to optimize line use, HOLD should be used specifying a time interval to minimize WACK responses.

It is advisable to have a separate application program to handle status, rather than to try to include such programming in the message handler programs. If necessary, the application can work with operator control, issue QRESET, MRELEASE, and ICHNG macro instructions, and retrieve messages.

If a permanent error occurs on the outgoing side of the message handler, record the pending status in an option field for the terminal until the sense/status message arrives at the input message handler. If an unexpected sense/status message arrives, you may need to issue a QRESET macro instruction to resend a message that had been marked as complete.

In the OUTMSG subgroup, specify a short duration (INVTL) for the HOLD macro instruction if a high percentage of cases requires retrying. After three retries, issue a permanent HCLD. An ERRORMSG exit to an application program is probably required to release the device. The exit must be to a valid destination or the exit is not taken.

TCAM FACILITIES FOR ERROR HANDLING

TCAM has various facilities for error handling, which can be divided into three classes: basic operator control commands, message handler macros, and TCAM application program macros.

- Basic operator control commands

VOFFTP/VONTP - Commands used to halt and restart communication with a device or devices on a line or all lines in a group.

- Message handler macro instructions

HCLD - Holds a TCAM message queue so that TCAM will not send messages on a held queue to the associated device until the queue is released.

CANCEL - Cancels the current message being processed by the message handler.

ERRORMSG - Generates a message from within a message handler to a specified destination (such as an error handling application program).

IEDPSTOP - Stops polling of a specific device.

IEDVOFF - Stops communication with a device or with all devices on a line.

IEDRELS - Releases a held message queue; any messages on the queue will subsequently be sent.

SLOWPOLL - Suspends polling on a line for a specified interval.

- TCAM application program macros

MRELEASE - Releases a held message queue; any messages on the queue will subsequently be sent.

ICOPY - Copies a line's invitation list into application program storage for examination.

ICHNG - Changes a line's invitation list; for example, makes a device's entry inactive or changes a device's polling characters.

POINT - Used to set up message retrieval for data stream error analysis.

QCOPY - Copies a device's queue control blocks into application program storage for examination.

QRESET - Marks messages previously marked complete as not sent so that they will be resent.

TCOPY - Copies a device's terminal table entry control block into application program storage for examination.

TCHNG - Changes a device's terminal table entry control block.

There are similar error conditions and tasks to be performed for the other forms of attaching 3270s to TCAM. For NCP BSC 3270s, SOH%R sense/status messages flow through the input message handler as for ncn-NCP BSC 3270s. For LU.T0 3270s, device sense/status is returned to TCAM in the user sense field of SNA exception responses and requests. For LU.T2 3270s, errors and status are reported to TCAM via SNA exception responses and LUSTAT DFC requests. For all these forms of attachment, refer to the TCAM publications for more information on how to handle status and errors, and to the 3270 Component Description for information on errors and status sent by 3270 stations and the form in which the information is encoded.

CHAPTER 6. VTAM SUPPORT

The Virtual Telecommunications Access Method (VTAM), Advanced Communication Function (ACF/VTAM), and Advanced Communications Function VTAM Entry (ACF/VTAME) are access methods that can be used in the telecommunications management of the 3270 display system. They support both local and remote 3270s under DOS/VS, OS/VS1, and OS/VS2. ACF/VTAM and ACF/VTAME provide large screen support for the 3278 and 3276 displays plus the LOGMODE capability in record mode. ACF/VTAME supports Color, Extended Highlighting, and Programmed Symbols for SNA devices only.

The VTAM, ACF/VTAM, and ACF/VTAME publications listed in the preface to this manual are required for:

- A detailed introduction to VTAM and a definition of the appropriate terminology (VTAM Concepts and Planning, ACF/VTAM Concepts and Planning), and Advanced Communications Function for VTAM Entry (ACF/VTAME) General Information: Concepts, GC27-0451.
- An explanation of how to define a VTAM network that includes 3270s (the DOS/VS, OS/VS1, or OS/VS2 ACF/VTAM, or VTAM System Programmer's Guide)
- An explanation of how to write a VTAM application program that communicates with 3270 terminals using VTAM (VTAM Macro Language Reference, ACF/VTAM Macro Language Reference, VTAM Macro Language Guide, ACF/VTAM Macro Language Guide, and Advanced Communications Function for VTAM Entry (ACF/VTAME) Programming, SC24-0442)

For explicit information on how to generate each 3270 device, see IBM 704 and 3705 Control Program Generation and Utilities Guide and Reference Manual for VTAM and ACF/VTAM.

This chapter is only a summary of the VTAM information for the 3270, and should be used in conjunction with the VTAM publications. The chapter discusses the considerations for a 3270 in a network using VTAM, and describes:

- VTAM Support for the 3270
- Appearance of the 3270 in VTAM Record Mode
- Defining the 3270
- Managing the 3270
- Migrating from Non-SNA to SNA 3270s

For detailed reference material, see the IBM 3270 Information Display System Component Description.

VTAM WITH BTAM AND TCAM

VTAM can coexist with ETAM under DOS/VS and with BTAM and TCAM under OS/VS. ETAM programs that do not use the communications controller in network control mode can be executed concurrently with VTAM as long as they have telecommunication networks separate from VTAM's. With this concurrent execution, a single application program can use both BTAM and

VTAM or TCAM and VTAM to communicate with 3270 devices, provided that all the requirements of both access methods are met.

ETAM and VTAM can operate concurrently only in separate networks. For a detailed description of TCAM programs under VTAM, refer to VTAM Concepts and Planning. Note, however, that ACF/VTAM does not provide support for TCAM.

APPEARANCE OF THE 3270 IN VTAM RECORD MODE

Three major classes of 3270 control units are supported by VTAM. (See Figure 6-1.) They differ in their means of physical attachment to the host and in their SNA capabilities. There are two classes of SNA 3270 control units. The first is called a physical unit type 2 (PU.T2); it is an SNA control unit, which can attach by a 370 channel or by a switched or nonswitched SDLC link; it supports logical unit types 1, 2, and 3 for its terminals. The other SNA control unit is called a physical unit type 1 (PU.T1); it can attach only by a nonswitched SDLC link and supports only type 0 logical units for its terminals. The final class of 3270s consists of non-SNA control units; they can attach by a 370 channel or a nonswitched BSC link. Although these control units do not directly support SNA, VTAM (with some exceptions) and ACF/VTAM programming permit the non-SNA 3270 terminals to appear to the VTAM application program as if they were type 0 logical units (that is, to appear the same as terminals attached to a PU.T1 3270 control unit).

With minor exceptions the data streams for all three 3270s appear the same to a VTAM application program. Also the SNA protocols used by the type 0 LUs are the same for the PU.T1 and non-SNA 3270 terminals. However, different protocols are used for the PU.T2 logical units (LU types 1, 2, and 3). The data streams and protocols are discussed in detail in the Component Description manual. Some of the key differences between type 0 and type 2 logical unit protocols are discussed in following sections.

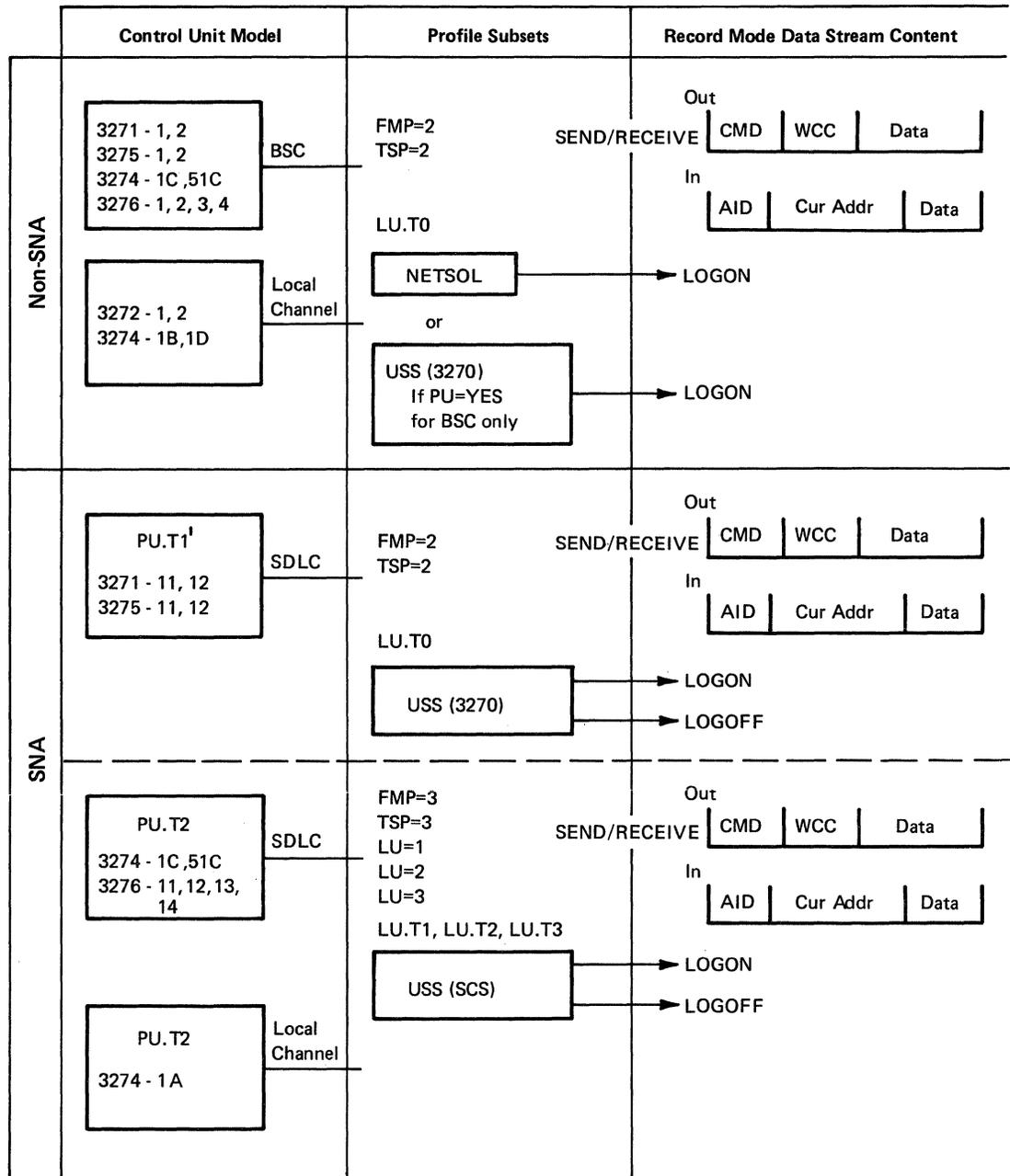
DEFINING THE 3270

DEFINING THE LOCAL NON-SNA 3270

A set of local non-SNA 3270s is one of the types of major nodes you can define to VTAM.

VTAM'S LOCAL definition statement provides definitions of local non-SNA major nodes. A LOCAL statement defines either one printer or one display unit; each locally attached non-SNA 3270 terminal must be defined by at least one LOCAL statement. The LOCAL statement provides the following information:

- The name of the terminal
- The channel and unit address of the terminal
- The features available on the terminal
- The name of the interpret table to be used in analyzing logon requests for the terminal



¹ACF/VTAME does not support PU.T1.

Figure 6-1. VTAM Support Summary

- The name of an application program to which VTAM is to automatically transmit a logon for the terminal whenever the terminal is available for connection. (VTAM can automatically submit a logon if this option is used. The application program can be the IBM network solicitor (NETSOL), a user-written network solicitor, or the application program name.)
- The buffer limit for the terminal
- Whether the terminal is to be considered active or inactive when the logical set of which it is a part is activated
- The terminal type: 3277, 3284, or 3286
- For ACF/VTAM the name of the logon mode table to be used in determining session parameters for the terminal in a record mode session and a default logon mode name

If you are defining a local non-SNA 3278, 3287, or 3289, you must use the LOCAL macro and define the 3278 or 3279 as a 3277, the 3287 as a 3284, and the 3289 as a 3286, since these are the only devices recognized as valid operands.

An LBUILD statement identifies a set of local 3270s as a major node. The LBUILD statement and the LOCAL statements that define the 3270s in the set are filed together as a member (member in OS/VS, book in DOS/VS) of the VTAM definition library. Figure 6-2 illustrates the defining of localnon-SNA 3270s.

DEFINING THE ESC 3270

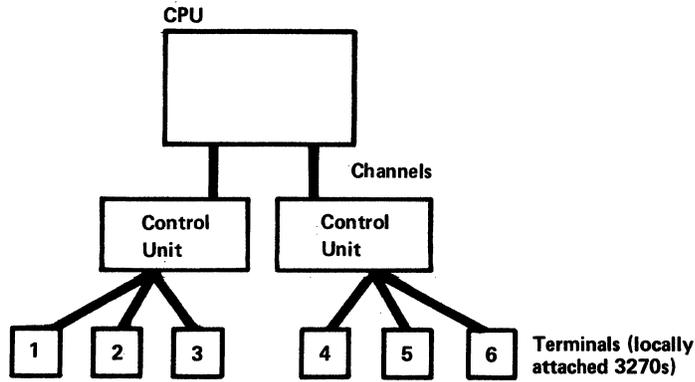
A CLUSTER and a TERMINAL definition statement define the ESC 3270 terminal. These statements can specify to VTAM:

- The name of the terminal
- The features of the terminal
- Automatic logon and interpret table requirements
- The initial status of the terminal or cluster control unit when VTAM activates the NCP
- Logon mode table and default logon mode name requirements (ACF/VTAM and ACF/VTAME only)
- USS definition table requirements (ACF/VTAM and ACF/VTAME only)

The TERMINAL definition statement when used to define printers and displays must specify any of these as a 3277, 3284, or 3286, since these are the only terminals that are valid operands. (For example, if your cluster includes a 3287, you should specify it as a 3286.)

The CUTYPE operand on the CLUSTER macro accepts only the 3271 as valid; therefore, if you wish to specify a 3274 Model 1C, 3274 Model 51C, or 3276, substitute a 3271 as the operand.

PHYSICAL CONFIGURATION



NETWORK DEFINITION

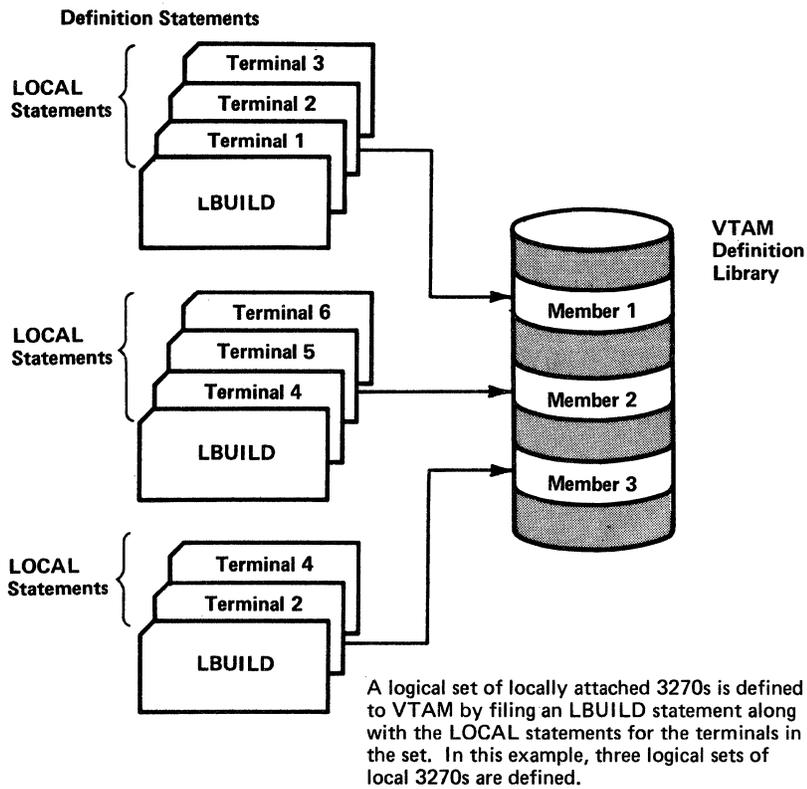


Figure 6-2. Grouping Locally Attached 3270s into Logical Sets.

LOGCN REQUESTS (ESC and Local Attachment)

The network solicitor acts as VTAM's logon monitor facility for locally attached and BSC 3270 terminals assigned to it (if PU=YES is not specified). To assign a 3270 to the network solicitor, specify the name of the network solicitor in the APPL operand of the LOCAL statement for the local 3270 or in the LCGAPPL operand of the TERMINAL statement for the ESC 3270. (NETSOL is the name of the IBM-supplied network solicitor.)

When the terminal operator enters a message at the 3270 terminal, the network solicitor determines whether the message is a valid logcn request by checking an interpret table. If the message is valid, the network solicitor passes the request to the logon exit routine of the application program specified in the interpret table for that message.

Note: NETSCL does not exist in the ACF/VTAM Release 2 or ACF/VTAME.

Note: If no interpret table is specified and the operating system is OS/VS, the network solicitor checks the message for usual format. If the message is the usual logcn, the network solicitor passes the terminal to the application program specified in the message itself.

If the logon message is invalid, if the application program associated with the message is not active, or if that program is not accepting logons, the network solicitor notifies the terminal operator that the logcn has been rejected and invites the operator to enter another logcn message.

Figure 6-3 diagrams the network solicitor's processing of a 3270's logon message. For more information about the network solicitor and interpret tables, refer to VTAM Concepts and Planning.

Terminal operator enters logon from active Non-SNA terminal.

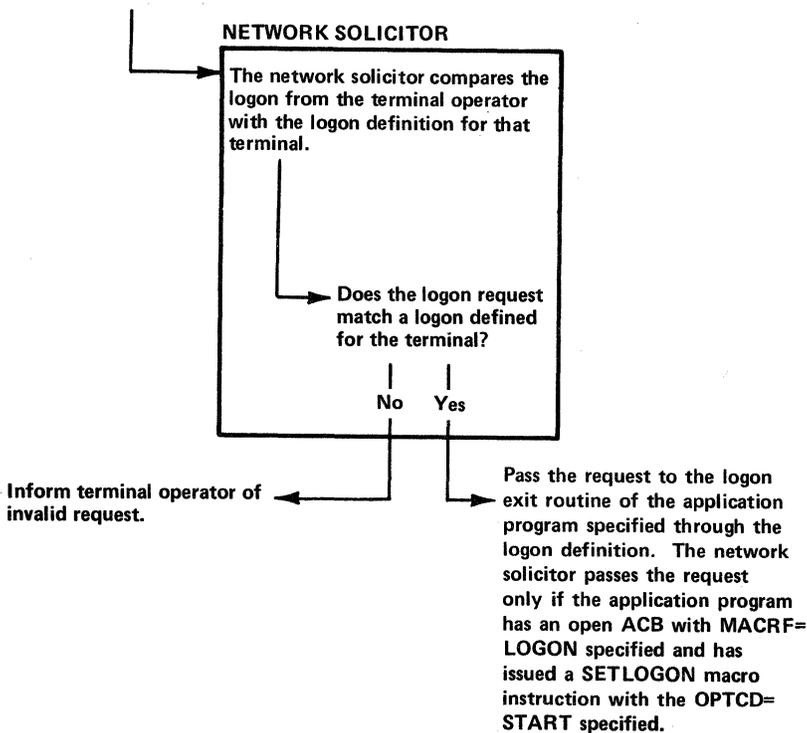


Figure 6-3. Processing a Terminal-Initiated Logon with the Network Solicitor

DEFINING AN SNA 3270

To define the SNA 3270 control unit and its terminals, use the same statements (PU and LU) and the same tables (USS definition table and logcn mode table) that you use to define other SNA devices.

When defining the SNA 3270 devices you should refer to the appropriate System Programmer's Guide for details; you can use the following guidelines for the various control units.

PU Options

For the 3276 Models 11, 12, 13, 14 use PU parameters associated with PU.T2 support.

For the 3276 Models 1A, 1C in SDLC use PU parameters associated with PU.T2 support.

For the 3271 Models 11, 12 use PU parameters associated with PU.T1 support (not supported by ACF/VTAME).

For the 3275 Models 11, 12 use PU parameters associated with PU.T1 support (not supported by ACF/VTAME).

LU Options

The LU types are

- Type 1 - the device attached to the 3274 or 3276 is a printer and the data stream is the SNA character string (SCS).
- Type 2 - the device attached to the 3274 or 3276 is a keycard/display and the data stream is in 3270 data stream compatibility format.
- Type 3 - the device attached to the 3274 or 3276 is a printer and the data stream is in the 3270 data stream compatibility format.

For the 3276 Models 11, 12, 13, 14 the PU and LU parameters for switched line support apply.

For the 3274 Models 1A and 1E LU local statements apply.

For the 3276 and 3274 SDLC attachments are defined like any other LU types.

Local SNA 3274s (Model 1A) must be specified as a 3791 on the IODEVICE system macro in OS/VS systems and on the DVGEN macro in DOS/VS systems.

You can specify unformatted system services (USS) definition tables for the 3270 terminals associated with an SNA 3270 controller; with the 3270 controller, USS monitors all traffic on the SSCP-LU session.

MANAGING THE 3270

DATA TRANSFER MODES

To communicate with a ncn-SNA 3270 through VTAM, an application program can use either of two modes of data transfer: basic mode or record mode. (If PU=YES was specified, only record mode can be used.) Basic mode supports only the ncn-SNA 3270s, but a program can use record mode to communicate with both ncn-SNA and SNA 3270s. Using record mode with ncn-SNA 3270s permits the VTAM (with some exceptions) and ACF/VTAM application program to communicate with ncn-SNA 3270s using the same VTAM instructions as those used to communicate with the SNA 3270s.

Note: Basic mode is not supported in ACF/VTAM Release 2 and ACF/VTAME.

DATA TRANSFER USING RECORD MODE

The 3270 terminal is treated as a logical unit if the NIB's MODE field is set to RECORD (indicating record mode) when the terminal is connected by CPNDST. Using record mode, the VTAM application program exchanges messages and responses with SEND and RECEIVE macro instructions. The application program cannot use any basic mode macro instructions with terminals connected in record mode. An application program using ncn-SNA 3270s can communicate with different devices on the same control unit in different modes at the same time if PU=YES was not specified. Ncn-SNA 3270 devices can also be disconnected in one mode and reconnected in another if PU=YES was not specified.

With record mode, the application program can be independent of whether the 3270 is locally or remotely attached, because VTAM deletes all line control characters sent from remote devices.

Some restrictions apply to all SEND, RECEIVE, and SESSICNC communication with the 3271, 3275, and 3277. These restrictions are described in the "Considerations" sections below.

Since in ACF/VTAM logon mode tables are used for all record mode sessions, even those with ncn-SNA 3270s, care must be taken that any logon mode name used for a session leads to a valid logon mode table entry for the session. If no logon mode name is specified for the terminal (either directly or indirectly) a default set of session parameters is used. See the ACF/VTAM System Programmer's Guide for further information.

Note: It might be necessary to update the LCGMCDE entry to specify the primary and alternate screen sizes for new devices if the application program requires that information.

Using SEND/RECEIVE

The following is an example of 3270 input and output data using the SEND and RECEIVE macro instructions.

On completion of the RECEIVE macro instruction:

```
RECEIVE RPL=(2),RIYPE=LFSYN,AREA=AREA1,AREALEN=100
```

the program could look in the RECLen field of the RPL pointed to by register 2 to find the exact length of the data received in AREA1. AREA1 might contain data in this format:

AID	Cursor Address	SEA	Buffer Address	Trust Account Number
-----	----------------	-----	----------------	----------------------

The AID might indicate a particular processing routine that was required, such as a trust account information program. The reply to this input, prepared by the trust account information program, might consist of data in the following format:

Erase/Write Char	WCC	Orders and Text
------------------	-----	-----------------

and be sent with this macro instruction:

```
SEND RPL=(2),SType=REQ,AREA=AREA1,RECLen=50,OPTCD=ASY,
      POST=RESP,ECE=ECB1,RESPOND=(NEX,FME)
```

When VTAM determines that the message has arrived, it posts ECB1. (FCST=SCHED could also be specified and completion determined by receiving a response with a RECEIVE(RTYPE=RESP) or as a result of VTAM scheduling the program's RESF exit routine.) Note that VTAM inserts a binary synchronous ESC character in front of the Erase and Write (or other) command character for remote BSC 3270s.

Note: Refer to Sample Program 2 in the VTAM Macro Language Guide for another example of communicating with the 3270 terminals using record mode macro instructions.

Input Considerations for Non-SNA and PU.T1 3270s (LU.T0)

For information on SNA considerations see the 3270 Component Description.

The VTAM application program can receive no SESSIONC commands from the 3270 terminal. The application program sends only the Clear command to the 3270. The Clear command resets both incoming and outgoing sequence numbers to 0, terminates any current bracket, and allows data traffic to continue. While a Clear is in progress, data traffic is not allowed; the application program gets a return code to this effect (RTNCD=20, FDBK2=65) if it attempts to send data while CLEAR is in progress.

If the application program has no use for brackets, the entire interval between the first I/C request following connection or CLEAR, which must begin a bracket, and disconnection can be considered to be one bracket. The BIND parameters can also indicate that brackets are not to be used; for non-SNA 3270s this is available only with ACF/VTAM. Both the application program and the 3270 can begin a bracket, but only the application program can end one. The application program can begin and end a bracket with the same message; that is, it can specify BRACKET=(EB,EB) for the SEND RPL.

The first input from a local non-SNA 3270 or a remote 3270 while not in bracket is marked as the beginning of a bracket. This input includes any power-on device end condition that is passed to the application program as an exception request message after OPNDST. All subsequent messages received from the 3270 indicate that the bracket is being continued; only the application program can end the bracket.

If both the application program and the 3270 attempt to begin a bracket at the same time, or if the application program attempts to begin a new bracket without ending the current one, the response to the application program's SEND indicates Request Reject (SSENSEI=RR). If the application program sends an NBB bracket indicator while not in a bracket, a STATE error (SSENSEI=STATE) is returned.

When the SEND data stream contains a 3270 Read command, the resulting input is received as a separate message, not as a response to the SEND operation. The application program should not begin or end a bracket when the data being sent contains a Read command. The application program must request a definite response to each message sent to the 3270 that begins or ends a bracket. A definite response is requested by setting RESPOND=(NEX,FME,NRRN) for the SEND RPL. A definite response should also be requested when a message is sent to a printer. All other output can indicate that either exception responses only -- RESPOND=(EX,FME,NRRN) -- or definite responses -- RESPOND=(NEX,FME,NRRN) -- are expected. Definite Response 2's (formerly called RRN responses) are not used. Only single-element chains can be used.

Output Considerations for Non-SNA and PU.T1 3270s (LU.T0)

Note: The restrictions noted in this section do not apply to the SNA 3275/3276. For information on SNA considerations see the ACF/VTAM and VTAM System Programmer's Guide and the 3270 Component Description.

The application program must place all 3270 commands and orders in the output data in the format expected by a BSC 3270 except for ESC; that is, everything, beginning with the command code, is used as though the 3270 were a BSC 3270. BSC communication control characters are provided by NCP or ACF/VTAME.

A SEND macro instruction may point to a data area containing a Read Modified command to be sent to the device. The data retrieved from the device is placed in the application program's storage area of a subsequent RECEIVE macro instruction in the format:

AID	CUR ¹	CUR ²	device control characters, orders, text
-----	------------------	------------------	---

AID is the attention identification byte and CUR¹ and CUR² form the 2-byte cursor address. If the terminal operator causes a "short read" to occur at the terminal (by pressing the CLEAR key or a Program Access key, for example), the input data consists of the AID byte only.

No responses may be sent to the non-SNA terminal unless your system includes a 3274 or 3276 Control unit. All incoming messages indicate that no response of any type is expected (RESPOND=(NEX,NFME,NRRN)).

Messages sent to the 3270 should contain only data and control characters for the 3270. No SNA data flow control commands, such as Quiesce, Signal, Bit, Chase, or Cancel commands should be sent; that is, only CONTROL=DATA is allowed. If the application program attempts to send one of these commands, the SEND is completed with RTNCD=20 and FDBK2=71. The terminal may return a negative response. Bracket

indicators are used as described above in "Input Considerations." Chaining indicators must always mark the message as the sole element of a chain; that is, CHAIN=ONLY. The format indicator (OPTCD=FMHDR) and direction indicator (CHNDIR) are not used with LU.T0 and should be set to 0.

Copy Considerations for Non-SNA and PU.T1 327Cs

When a Copy Command is placed in the data stream for a BSC or SDLC 3271, remote BSC 3274 or 3276, the application program must include the physical device address of the "from" device. It can obtain this address by issuing INQUIRE (CPTCD=DEVCHAR). Note that a copy operation is not valid for a locally attached 3270 terminal and should not be used in an application program that is intended to be attachment-independent. Also this device address information is not available by INQUIRE if the 3270 is owned by a TCAE system in a multi-system networking configuration.

When using the Copy command to obtain a printout on a 3288-2 or 3289 printer, remember that varicus print belts can be installed on this printer.

Network Solicitor Considerations

Logon requests for the non-SNA 3270 originate from these 3270 terminals (unlike a logical unit) by means of the network solicitor. When PU=YES has been specified, the network solicitor cannot be used, and USS logon is provided by means of the ENTER key or magnetic card input. (For SDLC 3271 and 3275 USS logon and logoff are provided by means of the system request key.)

If a non-SNA 3270 device is polled and the 3271, 3274, or 3276 control unit is not powered-on, the terminal is disconnected from the network solicitor. When the 3270 device and control unit are subsequently powered-on, the user must call up the network operator and request NETSOL connection with a VARY LOGON command, if the network solicitor is to be used for the device.

CAUTION: If a 3270 terminal is in session with a VTAM application program, the terminal operator must terminate the session before switching off the terminal. If the operator turns off the device before successfully terminating the session and before a general poll reaches the device, no status is returned until the device is turned back on; thus, another terminal operator could turn on the 3270 device and have access to the first terminal operator's session. If, however, the application issues a SEND to the device, an error indication is received.

Sense and Status Information

Status and sense information will be available in the RPL's USENSEI field when an exception request or response is received. The format of the information is shown below. The 2 high-order bits of each byte are

set to 0; this makes it easier to design the application program to be independent of the terminal's mode of attachment.

<u>First byte</u>	<u>Secnd byte</u>	<u>Meaning</u>
08	00	Device busy
04	00	Unit specify
02	00	Device end
01	00	Transmission check
C0	20	Command reject
00	10	Intervention required
C0	08	Equipment check
00	04	Data check
C0	02	Control check
00	01	Operation check

Only the low-order 6 bits of each byte are significant. These bits may be set in combination and should be tested individually

The SSENSEI (system sense) field is set following the receipt of exception responses (see VTAM Macro Language Reference for complete SSENSMI settings). The SSENSEI field can be set to indicate a PATH error (SSENSEI=PATH), a STATE error (SSENSEI=STATE), a Request Reject error (SSENSEI=RR), a Function Interpret error (SSENSEI=FI), or no system error (SSENSEI=0). In the case of SSENSEI=0, the USENSEI field should be used to determine the cause of the exception condition.

DATA TRANSFER USING EASIC MCLE

In basic mode, the MODE field of the NIB is set to BASIC when the 3270 terminal is connected. In basic mode, the VTAM application program exchanges data using READ and WRITE macro instructions; the program cannot use any record mode macro instructions with terminals connected in basic mode. It can, however, communicate with different devices on the same control unit in different modes at the same. A 3270 device can also be disconnected in one mode and reconnected in the other. Basic mode cannot be used if PU=YES is specified.

Because VTAM deletes all line control characters sent from remotely attached devices, input processing does not have to consider whether the device is locally or remotely attached.

Input Considerations

To avoid losing incoming data when the input area is too small, specify the KEEP processing option in the NIB used to connect the device. Then if the data is too long, VTAM fills the input area, sets the second bit (DATAFLG=EOB) of the FLEK field off, and holds the remaining data for the next read request.

To send a Read Buffer command to a terminal, use a DO macro instruction that specifies an LDO with CMD=READBUF. The data in the application program's input area upon completion of the DC macro instruction is in the format:

AID	CUR ¹	CUR ²	SF	ATTR	text
-----	------------------	------------------	----	------	------

AID is the attention identification byte and CUR¹ and CUR² form the 2-byte cursor address. The SFs (Start Fields) and ATTRs (attribute bytes) are present only if the device buffer is formatted.

Output Considerations

The four output operations available are selected by setting the ERASE-EAU-NERASE-CONV option code in the RPL of a WRITE macro instruction:

- WRITE (OPTCD=ERASE) clears the device's entire buffer, and then fills it with the output data whose address you provide in the RPL's AREA field. At the beginning of that data, you must provide the Write Control Character (WCC) followed by the appropriate device control characters, orders, and text. Because you set the BLK-LBM-LBT option code to LBT, do not select the embedded line control character option.
- WRITE (OPTCD=EAU) sends an Erase All Unprotected command to the device. Because this form of WRITE involves no output data, set the RPL's RECLN field to 0.
- WRITE (OPTCD=NERASE) sends a Write command to the device. You must prepare the output data in exactly the same manner as is specified above for OPTCD=ERASE: begin the output data with WCC, followed by the appropriate device control characters and orders.
- WRITE (OPTCD=CCNV) writes a block of data to a terminal and then reads a block from the same terminal. This form of WRITE (the conversational WRITE) uses the AREA field of the RPL to contain the address of the output data and the AAREA field to contain the address of the input data.

For a Read/Write sequence, the write operation is not suspended pending the completion of the solicitation of the device. Instead, the write operation is completed and the solicitation of the device continues.

Other Basic Mode Considerations

If a 3270 device is addressed and the control unit is not powered-on, a X'0C01' return code is received. The terminal is no longer usable and should be disconnected. When the 3270 device and 3271, 3274, 3275, or 3276 control units are subsequently powered-on, the user must call up the network operator and request that the control unit be reactivated using the VARY commands.

If a locally attached 3270 device is polled while it is powered-off and the control unit is powered-on, an error indicator is received. A device end status is returned when the device is powered-on. When the device end status is received, reissue the request.

See VTAM Macro Language Reference for options on the NIB macro instruction that are invalid for 3270 devices.

Set the BLK-LBM-LBT option code (applicable for output) to LBT; BLK is invalid, and LBM requires that you be aware of whether the device is locally or remotely attached (because no line control characters are sent regardless of the attachment mode).

CAUTION: If a 3270 terminal is in session with a VTAM application program, the terminal operator must terminate the session before switching off the terminal. If the operator turns off the device before successfully terminating the session and before a general poll reaches the device, no status is returned until the device is turned back on; thus, another terminal operator could turn on the 3270 device and have access to the first terminal operator's session.

Sense and Status Information

When the terminal sends sense and status information in response to a READ, WRITE, or DO macro instruction, VTAM places the 2 bytes of information in the RPL's SENSE field. VTAM also sets the RPL's RTNCD field to 4 and sets the FDEK2 field to 2 to signal that the SENSE field has been set. The failed operation may be retried after execution of a RESET macro instruction with OPTCD set to UNCCND or LOCK.

If the SENSE field is extracted with SHOWCB, the 2 bytes of information are right-adjusted in the fullword work area. The SENSE field codes are described below; the possible hexadecimal values of the 2 bytes are:

<u>First byte</u>	<u>Second byte</u>	<u>Meaning</u>
C8	40	Device busy
C4	40	Unit specify
C2	40	Device end
C1	40	Transmission check
40	60	Command reject
40	50	Intervention required
40	C8	Equipment check
40	C4	Data check
40	C2	Control check
40	C1	Operation check

Only the low-order 6 bits of each byte are significant. These bits may be set in combination and should be tested individually.

USING VTAM WITH SNA 3270S

VTAM supports SNA 3270s on nonswitched lines only with the exception of the 3276 on which VTAM supports the use of switched lines. The application program communicates with the SNA 3270 in record mode. Although this 3270 is defined to the application program as a logical unit, it has no programming capabilities. All other aspects of communication with the SNA 3270 are the same as those described in the VTAM publications for other SNA terminals (refer to the publications listed in the preface to this book).

LOGON REQUESTS

To log on from a 3270 (FU.T2, PU.T1 or non-SNA 3270 for which PU=YES is specified), the terminal operator follows device procedures for getting to the SSCP session. The operator can indicate a logon mode previously defined by the installation as valid and specify a logon message to be passed to the VTAM application program.

The request the operator enters is a character-coded request; that is, a request that uses the SNA unformatted system services of VTAM. To specify in a VTAM network definition that a terminal uses character coded requests, specify SSCPFM=USS3270 in the LU statement defining the 3270 for PU.T1 and specify SSCPFM=USSSCS in the LU statement defining the 3270 for PU.T2.

When VTAM receives this request, it converts it into a field-formatted connection request; that is, a request that uses the SNA formatted system services (FSS) of VTAM. VTAM uses a logon request format and related definition table for the conversion of the request; the format and table may be IBM-supplied or defined by the user. Refer to the appropriate VTAM or ACF/VTAM System Programmer's Guide for more information about request formats and definition tables for USS and logon modes.

SNA LOGOFF REQUESTS

To log off an SNA 3270, an installation can write the VTAM application program so that it examines each message from the 3270 for a predesignated request for disconnection or logoff.

Instead of using the predesignated request, the installation can also have the terminal operator send a logoff request as a character-coded system service request by using the system request key. This request arrives as a LOGOFF request, which VTAM converts into a formatted Terminate-Self command. As with the logon requests, VTAM can use either an IBM-supplied or user-defined logoff request format and associated definition table.

NCN-SNA LOGOFF REQUESTS

An installation can write the VTAM application program to examine each message from the local 3270 terminal for a predesignated request for disconnection or logoff request. This logoff request and the LOSTERM exit (with return code) are the type of logoff requests supported for ESC and locally attached 3270s.

MIGRATION FROM NON-SNA TO SNA 3270

Following are some of the areas that you should consider when migrating in a VTAM record mode environment from ncn-SNA 3270 to PU.T1 or PU.T2 3270s or from PU.T1 3270s to PU.T2 3270s.

VTAM network definition:

CLUSTER, TERMINAL, and LOCAL statements define non-SNA 3270s; PU and LU statements define SNA 3270s.

For PU.T2 3270s, specify SSCPFM=USSSCS; for PU.T1 and for 3270 terminals for which PU=YES, specify SSCPFM=USS3270.

LOGCN data passed to the application program:

If logon is done through the network solicitor (non-SNA 3270s with PU=NO), the whole logon message is passed to the application program. If USS is used (SNA 3270s or terminals with PU=YES), only the user data part of the logon is passed to the application program. See the VTAM or ACF/VTAM System Programmer's Guide for a limited means (using the interpret function) by which the whole logon message is passed to the application program. A user-written network solicitor can also pass the data part of a logon to the application program.

LOGON format:

Any logon message defined for non-SNA terminals should have a format acceptable to USS. This will allow migration without changing the logon format used by the terminal operator. See the VTAM or ACF/VTAM System Programmer's Guide for rules defining acceptable formats and also for a limited means of bypassing this requirement by using the interpret function.

LOSTERM exit:

SNA 3270s can invoke an application program LOSTERM exit for a conditional logoff or a segmenting error. These conditions do not occur for non-SNA 3270s.

Figure 6-4 shows SNA protocol differences between LU types 0, 1, 2, and 3 for the 3270.

<u>LU Type 0</u>	<u>LU Types 1, 3</u>	<u>LU Type 2</u>
Uses full duplex mode	Uses half-duplex flip-flop mode	Uses half-duplex flip-flop mode
Uses single-element chains	Uses multi-element chains	Uses multi-element chains
Never requests responses	Can request responses	Can request responses
Uses no data flow control commands	Sends LUSTAT, SHUTC, LU.T1 can send SIGNAL	Sends CANCEL, LUSTAT, SHUTC, SIGNAL
	Receives CANCEL, CHASE, BID, SHUTD, SIGNAL	Receives CANCEL, CHASE, BID, SHUTD, SIGNAL
Supports COPY command	Uses WRITE with start print bit set ²	Uses WRITE with start print bit set
Uses limited sense codes	Uses extensive sense codes	Uses extensive sense codes
Does not support Start Data Traffic	Supports Start Data Traffic	Supports Start Data Traffic
Ignores pacing and RU sizes in BIND	Uses pacing and RU sizes in BIND	Uses pacing and RU sizes in BIND
Does not use unconditional bracket termination ¹	Uses conditional bracket termination	Uses conditional bracket termination
Supports READ command to printers	Does not support READ command	Supports READ command

¹PU.T1 3270s support only unconditional bracket termination; so does ACF/VTAM for non-SNA 3270s. For BSC 3270s, VTAM (VS, ACF/VTAM) uses conditional bracket termination; this requires that the VTAM application send another end bracket if the request carrying the first end bracket indicator fails.

²For LU.T1 only

Figure 6-4. LU Type Differences

A sequence number is returned in the 2-byte RPL SEQNO field when a SEND is completed. For PU.T1 and PU.T2 the whole SEQNO field is used; this is also true for OS/VS VTAM support of non-SNA 3270s. However, for DOS/VS VTAM and OS/VS and DOS/VS ACF/VTAM, only the low-order byte of the SEQNO field is used; the high-order byte is always set to 0.

Switched Line Support for SNA (Switched PU Support)

The IDBLK 12-bit binary number required on the switched PU macro is defined for the 3276 as 018. The IDNUM 20-bit binary identification number for the 3276 is factory-assigned and is the same as the device serial number. (Although the 3274 does not support switched PUs, the comparable value used as part of the XID for the 3274 is always 0000.)

GLOSSARY

This glossary defines 3270 Information Display System terms and other data processing and communication terms used in this publication. For definitions of terms not included in this glossary, see IBM Data Processing Glossary, GC20-1699. IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard Vocabulary for Information Processing (copyright 1970 by American National Standards Institute, Inc.), which was prepared by Subcommittee X3K5 on Terminology and Glossary of American National Standards Committee X3. A complete commentary taken from ANSI is identified by an asterisk (*) that appears before the term and the beginning of the commentary; a single definition taken from ANSI is identified by an asterisk after the item number for that definition.

A

access method: A technique for moving data between main storage and input/output devices.

AID: See attention identifier.

alphanumeric field: A field that may contain any alphabetic, numeric, or special character that is available on any of the 3270 keyboards.

attention: An I/O interruption generated asynchronously by a display station, usually as the result of an action taken by the operator of the device.

attention identifier (AID): A code that is set in the display station when the operator takes an action that produces an I/O interruption. The character identifies the action or key that generates the condition. The AID is set when the display station operator presses a program access key, when a selector pen attention occurs, or when a successful operator identification card read-in occurs. The AID also identifies device addresses assigned to printers.

attribute: A characteristic of a 3270 display field. The attributes of a display field include: protected or unprotected (against manual input and copy operations); numeric-only or alphanumeric input control; displayed, nondisplayed, display-intensified; selector-pen-detectable or -nondetectable; and modified or not modified.

attribute byte: A code that defines the attributes of the display field that follows. An attribute character is the first character in a display field, but it is not a displayable character.

audible alarm: A special feature that sounds a short, audible tone automatically when a character is entered from the keyboard into the next-to-last character position on the screen. The tone can also be sounded under program control.

automatic polling: (1) A hardware feature of a telecommunications unit that processes a polling list, polling the terminals in order and handling negative responses to polling without interrupting the central processing unit. At the end of the list, polling is automatically begun again at the beginning of the list. Synonymous with autopoll. (2) See also polling.

automatic skip: After entry of a character into the last character position of an unprotected display field, automatic repositioning of the cursor from a protected and numeric field to the first character position of the next unprotected display field.

autopoll: Same as automatic polling.

auto-skip: Same as automatic skip.

B

Base Color: On color displays, one of four colors in which fields can be displayed: red, blue, green, or white.

Base Color mode: Provides four different colors, which can be produced by existing 3270 application programs with little or no reprogramming.

basic mode (VTAM): A set of facilities (including the macro instructions needed to use them) that enable the application program to communicate with BSC and start-stop terminals, including the locally attached 3270 Information Display System. READ, WRITE, SOLICIT, RESET, DO, and LDO macro instructions are basic-mode macro instructions.

Basic Telecommunications Access Method (BTAM): An access method that permits read/write communications with remote devices.

Binary Synchronous Communications (BSC): Data transmission in which character synchronism is controlled by timing signals generated at the sending and receiving stations.

Block matrix: The total array of dots that can be used to describe a graphic character for a 3270 display or printer.

Bracket: In VTAM, an exchange of data between an application program and a logical unit which accomplishes some task.

ESC: See Binary Synchronous Communications.

ETAM: See Basic Telecommunications Access Method.

buffer address: The address of a location in the buffer at which one character can be stored.

C

CCC: See copy control character.

character attribute: An attribute associated with individual characters to define character color, character set, and/or character highlighting.

character buffer: An area of storage that contains codes for graphic characters or field attribute characters.

character position: A location on the screen at which one character can be displayed; also, an addressed location in the buffer at which one character can be stored.

clear indicator: In VTAM, a SESSIONC indicator sent by one node to another that prevents the exchange of messages and responses.

cluster control unit: (1) A device that can control the control the input/output operations of more than one device. A remote cluster control unit can be attached to a host CPU only via a communications controller. A cluster control unit may be controlled by a program stored and executed in the unit, or it may be controlled entirely by hardware. (2) See also communications controller.

command: An instruction that directs a control unit or device to perform an operation or a set of operations.

communications controller: (1) A type of communication control unit whose operations are controlled by a program stored and executed in the unit. Examples are the IBM 3704 and 3705 Communications Controllers. (2) See also cluster control unit.

connection: In VTAM, in response to a request from an application program, the linking of VTAM control blocks in such a way that the program can communicate with a particular terminal. The connection process includes establishing and preparing the network path between the program and the terminal.

control character: A character used in conjunction with a Write command to specify that a control unit is to perform a particular operation.

conventional 3270: A locally-attached 3270 terminal or a remotely-attached 3270 terminal that uses the BSC line discipline.

copy control character (CCC): A character used in conjunction with the Copy command to specify the type of data to be copied.

copy operation: An operation that copies the contents of the buffer from one display station or printer to another display station or printer attached to the same control unit.

cursor: A unique symbol (an underscore) that identifies a character position in a screen display, usually the character position at which the next character to be entered from the keyboard will be displayed.

D

data stream: All data transmitted through a channel in a single read or write operation to a display station or printer.

data transfer: In telecommunications, the sending of data from one node to another.

data transfer mode: (1) A set of facilities (including the macro instructions needed to use them) that enable the application program to communicate with terminals. (2) See also basic mode and record mode.

definite response 1: In VTAM, a response that indicates whether its associated message was successfully forwarded to its final destination (such as the display screen of an output device).

definite response 2: In VTAM, a response that indicates that the node sending the response has accepted recovery responsibility for the associated message.

definition statement: The means of describing an element of the telecommunication system to VTAM.

designator character: A character that immediately follows the attribute character in a selector-pen-detectable field. The designator character controls whether a detect on the field will or will not cause an attention. For a nonattention-producing field, the designator character also determines whether the modified data tag for the field is to be set or reset as the result of a selector-pen detect.

detectable: An attribute of a display field; determines whether the field can be sensed by the selector pen.

disconnection: In VTAM, the disassociation of VTAM control blocks in such a way as to end communication between the program and a connected terminal. The disconnection process includes suspending the use of the network path between the program and the terminal.

display field: A group of consecutive characters (in the buffer) that starts with an attribute character (defining the characteristics of the field) and contains one or more alphanumeric characters. The field continues to, but does not include, the next attribute character.

dot: One point in a printer or display block matrix.

F

erase all unprotected (EAO) command: A command that clears all unprotected fields to nulls, resets modified data tags in all unprotected fields, unlocks the keyboard, resets the attention identifier, and repositions the cursor to the first character of the first unprotected field.

erase unprotected to address (EUA) order: An order that erases all unprotected positions (inserts nulls) from the current buffer address up to, but not including, the specified stop address.

Extended Attribute Buffer (EAB): A buffer for storing extended field attributes and character attributes.

Extended Data Stream 3270: Enhancements to the 3270 Data Stream that allow color, highlighting, and/or character set definition.

Extended Field Attribute: An attribute that provides additional field attribute definition beyond that of the 3270 Field Attribute.

F

field: See display field.

FME response: See definite response 1.

formatted display: A screen display in which a display field, or fields, has been defined as a result of storing at least one attribute character in the display buffer.

G

general polling: (1) An input technique for remote 3270 devices in which special invitation characters are sent to a device control unit instructing that control unit to begin transmission from all devices ready to enter data. (2) See also polling and specific polling.

I

incoming group: (1) In systems with TCAM, that portion of a message handler designed to handle messages arriving for handling by the message control program. (2) See also outgoing group.

insert cursor (IC) order: An order that displays the cursor at the current buffer address.

intensified display: An attribute of a display field; causes data in that field to be displayed at a brighter level than other data displayed on the screen.

interpret table: In VTAM, an installation-defined correlation list that translates an argument into a string of eight characters. Interpret tables can be used to translate a logon message into the name of an application program for which the logon request is intended.

invitation list: In systems with the telecommunications access method (TCAM), a sequence of polling characters or identification sequences associated with the stations on line; the order in which the characters are specified determines the order in which the stations are invited to enter a message.

L

leased line: See nonswitched line.

line control characters: Characters that regulate the transmission of data over a line; for example, delimiting messages, checking for transmission errors, and indicating whether a station has data to send or is ready to receive data.

line group: In systems with the telecommunications access method (TCAM), a set of one or more communications lines of the same type, over which terminals with similar characteristics can communicate with the computer.

local: Pertaining to the attachment of directly by channels to a host CPU. Contrast with remote.

logical unit: The combination of programming and hardware of a teleprocessing subsystem that comprises a terminal for VTAM.

logoff: In VTAM, a request from a terminal to be disconnected from an application program.

logon: In VTAM, a request by or on behalf of a terminal to be connected to an application program.

logon message: In VTAM, the data that can accompany a logon request received by the application program to which the request is directed.

M

major node: A set of one or more minor nodes represented by a single symbolic name. A major node can be a set of local terminals, a set of application programs, or a network control program.

MCP: See message control program.

MDI: See modified data tag.

message control program (MCP): In TCAM, a program that is used to control the sending or reception of messages to or from remote terminals.

message handler: In systems with the telecommunications access method (TCAM), a sequence of user-specified macro instructions that examine and process control information in message headers, and perform functions necessary to prepare message segments for forwarding to their destinations. One message handler is required for each line group having unique message handling requirements.

modified data tag (MDT): A bit in the attribute character of a display field,

which, when set, causes that field to be transferred to the channel during a read-modified operation. The modified data tag may be set by a keyboard input to the field, a selector-pen detection in the field, a card read-in operation, or program control. The modified data tag may be reset by a selector-pen detection in the field, program control, or ERASE INPUT key.

Modify Field (MF): An order that allows specified field attributes to be modified.

N

NIB: See node initialization block.

node: A point in a telecommunication system defined to VTAM by a symbolic name. See also major node.

node initialization block (NIB): In VTAM, a control block, associated with a particular node, that contains information used by the application program to identify a node and indicate how communication requests directed at the node are to be implemented.

nonswitched line: A connection between a remote 3270 unit (3271 or 3275) and a computer that does not have to be established by dialing.

O

order code: A code that may be included in the write data stream transmitted for a display station or printer; provides additional formatting or definition of the write data.

order sequence: A sequence in the data stream that starts with an order code and includes a character address and/or data characters related to the order code.

outgoing group: (1) In systems with TCAM, that section of a message handler that manipulates outgoing messages after they have been removed from their destination queues. (2) See also incoming group.

P

PCI: See program controlled interruption.

polling: A technique by which each of the terminals sharing a communications line is periodically interrogated to determine whether it requires servicing.

Prepare to Read (PTR): A command for a local 3274-1D that allows the terminal to know the next program action.

program access (PA) key: A program attention key that may be defined to solicit program action that does not require data to be read from the buffer of the display station. If a Read Modified command is issued in response to the program attention key interruption, only the attention identification (AID) character is transferred to the program; no data from the buffer is transferred.

program attention key: Any key on the keyboard that solicits program action by generating an I/O interruption. The keys are the CLEAR key, ENTER key, TEST REQ key, CNCL key, program function keys, and program access keys. Each program attention key is associated with a unique attention identification (AID) character.

program-controlled interruption (PCI): An interruption that allows buffers to be deallocated continuously, replenishing the available unit pool.

program function (FF) key: A program attention key that may be defined to solicit program action that usually requires data to be read from the buffer of the display station. If a Read Modified command is issued in response to the program function key interruption, the attention identifier (AID) and all display fields in which the modified data tags are set are transferred to the program.

program tab (PT) order: An order that advances the current buffer address to the address of the first character location of the next unprotected field.

Programmable Symbols (PS): Customer-defined symbols, a maximum of 190 symbols to a programmed symbol set.

protected field: A display field for which the display operator cannot use the keyboard or operator identification card reader to enter, modify, or erase data.

R

record mode: A set of facilities (and the macro instructions needed to use them) that enable the application program to communicate with logical units or with the locally- or remotely-attached 3270 Information Display System. SEND and RECEIVE are record-mode macro instructions.

remote: Pertaining to the attachment of devices to a central computer through a communication control unit. Contrast with local.

repeat to address (RA) order: An order that stores a specified alphanumeric or null character in up to 480 buffer locations, starting at the current buffer address and ending at, but not including, the specified stop address.

request parameter list (RPL): In VTAM, a control block that contains the parameters necessary for processing a request for connection, communication, or a request for an operation related to connection or communication.

RPL: See request parameter list.

RRN response: See definition response 2.

S

SDLC: Synchronous data link control.

selector pen: A pen-like instrument that may be attached to the display station as a special feature. When pointed at a detectable portion of an image and then activated, the selector pen senses the presence of light at a display field and produces a selector-pen detect.

selector-pen detect: The sensing by the selector pen of the presence of light from data in a display field that has the detectable attribute. Depending on the designator character of that display field, the detection and location information is identified on the screen (and stored in the buffer) or may produce an interrupt that is transmitted to the CPU.

SESSIONC indicators: In VTAM, indicators that can be sent from one node to another without using SEND or RECEIVE macro instructions. SDT, clear, and STSN are SESSIONC indicators. All SESSIONC indicators are sent with a SESSIONC macro instruction.

Set Attribute (SA): An order that associates attributes in the EAB with individual characters.

set buffer address (SBA) order: An order that sets the buffer address to a specified location.

SNA 3270: A 3270 terminal that uses synchronous data link control (SDLC) and is treated as a logical unit by VTAM.

specific polling: (1) A polling technique that sends invitation characters to a device to find out whether the device is ready to enter data. (2) See also general polling and polling.

start field (SF) order: An order that indicates a specified location which contains an attribute byte and not a text character.

Start Field Extended (SFE): An order that generates an extended field attribute in the EAB and at the current buffer location.

Structured Field: A data stream format that permits variable-length data and controls to be parsed into its components without having to scan every byte.

suppress index (SI) order: An order that generates the suppress index character, valid only for the 3288-2 printer (other printers receive | an cr bar). This character inhibits a line index to allow overprinting.

switched line: A communication line in which the connection between the computer and a remote terminal is established by dialing.

T

TCAM: See Telecommunications Access Method.

Telecommunications Access Method (TCAM): A method used to transfer data between main storage and remote or local terminals. Application programs use either GET and PUT or READ and WRITE macro instructions to request the transfer of data, which is performed by a message control program.

telecommunications network: In a telecommunication system, the combination of all terminals and other telecommunication devices and the lines that connect them.

terminal: (1) *A point in a system or communication network at which data can either enter or leave. (2) Any device capable of sending and receiving information over a communication channel.

terminal-initiated logon: A logon request that originates from the terminal.

U

unformatted display: A screen display in which no attribute character (and, therefore, no display field) has been defined.

unprotected field: A display field for which the display station operator can manually enter, modify, or erase data.

V

Virtual Telecommunications Access Method (VTAM): A set of IBM programs that control communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2.

VTAM: See Virtual Telecommunications Access Method.

VTAM definition library: The DOS/VS files or OS/VS data sets that contain the VTAM definition statements filed during VTAM definition. These statements describe the telecommunication system to VTAM and can be used to tailor VTAM and the system to suit the needs of the installation.

W

WCC: See write control character.

wraparound: The continuation of an operation (for example, a read operation or a cursor movement operation) from the last character position in a buffer to the first character position in the buffer.

write control character (WCC): A character used in conjunction with a Write command to specify that a particular operation, or combination of operations, is to be performed at a display station or printer.

Write Structured Field (WSF) command: A command used for processing structured fields.

3270 data stream: A coded character data stream consisting of a command, and, for write type commands, a write control character followed by orders, attributes, and character data.

INDEX

- access methods 93,137,151
 - BTAM 93,136
 - BTAM-ES 93
 - definition of 169
 - TCAM 93,137-151
 - VTAM 93,151-167
 - VTAME 151
- accounts receivable application
 - example of 35
 - panels for 8,9,10,35,36,37,38
- address, converting to buffer position 2,75
- AID (see attention identifier)
- alphanumeric field, definition of 169
- attention, definition of 169
- attention identifier (AID)
 - codes for 57
 - generally 56
 - resetting 68
- attribute byte
 - auto-skip, combination for 33
 - codes for 50
 - default (assumed) values of 45
 - definition of 2,169
 - example of placement of 3,4,6,7-10
 - indicating on panel layout sheet 41
 - modified data tag (MDT) in 58
 - position occupied by 2,28
 - removal of 27,28
- attributes
 - assumed values of 44-45
 - blink 1,5,6,11
 - brightness 29
 - character 5,6,8,9,10
 - character content 29
 - color 5,9-12
 - combinations of 31
 - definition of 169
 - detectable for selector pen 29,30
 - displayable 29
 - extended field 6-12
 - field 2,5,6
 - high intensity, changing 82
 - modified data tag (MDT) in 30
 - nondisplayable 29
 - numeric 29
 - protected 29
 - reverse video 1,5-11
 - transmission 30
 - type 6-9,50,54,55
 - underscore 1,5-11
 - value 6-8,10-11,50,54,55
- audible alarm
 - definition of 52,169
 - sounding, generally 52
 - sounding, with WCC 52,65
- Auto Disconnect feature 123
- automatic copying, description of 73,84-85
- autopoll feature 101-126
- auto-skip 33
- auto-skip field, example of 33,40
- base color 4-5
- Basic Communications Access Method-Extended Support (BTAM-ES) 93-136
- basic mode (VTAM) 162-164,169
- Basic Telecommunication Access Method (BTAM), support for 3270 93-136
- binary synchronous communication (BSC) 152,155,170
- "blinking," how to avoid 65
- block diagramming 39,40
- brackets, in VTAM 159-160
- brightness, field 30
- "browsing" application 69
- BSC (see binary synchronous communication)
- BTAM (see Basic Telecommunications Access Method)
- BTAM-ES (see Basic Telecommunication Access Method)
- BTMOD macro instruction 124
- buffer addresses
 - converting 84-85
 - definition of 170
- buffer locations, hexadecimal codes for 46-47
- buffer positions, converting to screen address 75
- calculator, using screen as 37
- CCC (see copy control character)
- character attribute 6,9-12
- character definition 11
- character position, definition of 170
- character sets, loadable and nonloadable 12-19
- characters, repeating 81
- CHGNTY macro instruction, use of 104-105
- clear indicator, in VTAM
 - definition of 170
 - discussion of 159-160
- CLEAR key, with TCAM 148
- clicks, keyboard 32
- coded Graphic Character Set Local ID (CGLI) 11,13
- color 4,5,9-12,19-20
- commands
 - definition of 170
 - general discussion of 1
- connection, in VTAM 162,164
- control characters
 - definition of 170
 - general discussion of 1
- COPY command 73,86-92
- copy control character (CCC) 83,170
- copy-lock 73
- copying, automatic
 - data transfer by 84-85
 - device considerations 84-85
 - generally 84
 - PA key, use of 85
 - printer-busy considerations 86
- current buffer address 51,66
- cursor
 - definition of 170
 - indicating, on panel layout sheet 41
 - repositioning 64

data entry, example of 69-70
 data stream
 coding 42
 concatenation of 87
 decoding 73,75
 definition of 170
 dynamic 84
 elements of 42
 extended 3270 1,5-6
 generating 73
 macro instruction to build 80-82
 mixed read modified input 81
 output, generally 81
 read modified input 74
 relationship of 62
 scanning 74
 sections, generating 82,83
 segments, concatenating 83,84
 selector pen 79
 semi-dynamic 84
 static 81
 subroutine to create 83
 total, generating 82,83
 truncating 76
 variable-length 74
 data-transfer mode 158-164,170
 DECB extension, how to build 99,122
 definition statements, for 3270 152,170
 designator character, selector pen 60,171
 device address, checking DECB for 99
 device control 42
 device status
 errors, recording 98
 maintaining record of 98
 names, assigning 97
 priorities, assigning 98
 transactions, recording 98
 disconnection, in VTAM 162,164,170
 disk files, switching 97
 display buffer image technique
 format of data produced by 75
 1920-character screen check 75
 display field
 characteristics of 27-30
 definition of 171
 displaying, intensity of 30

 ENTER key, purpose of 56
 ENTRY operand technique 124
 Erase All Unprotected (EAU) command 68,171
 Erase Unprotected to Address (EUA) order
 definition of 171
 generally 66
 multiple fields 66
 positioning 66
 erasing screen data (see also Erase
 Unprotected to Address (EUA) order) 51
 error message, modifying panel to
 include 62
 error recovery
 purpose of 99
 under BTAM 99
 event completion analysis (see Read
 completion analysis and Write
 completion analysis)
 Extended Attribute Buffer
 (EAB) 9,10,15,171
 Extended Highlighting 1,5,11,22,28

 field concept 27
 field definition, example of 31
 field length
 attempt to key beyond length of 31,32
 defining 27,31,32
 fields
 characteristics of 27
 combining using w/password 33

 general polling
 definition of 171
 in TCAM 139

 hard copy (see copying, automatic)
 held-line system 103
 hexadecimal, coding orders in 46-49,54,55

 input, selector pen
 detectable 60
 generally 60,80
 keyboard input, combining with 62
 input area, screen
 defining 31,32
 example of 31,32
 input data
 analyzing 56
 eliminating unnecessary 62
 input field, defining 46
 inquiry application 68
 Insert Cursor (IC) order 46,171
 interpret table, in VTAM 154,171
 INTRO macro instruction 145

 Katakana character set codes 92
 keyboard
 data entry 29
 enabling with WCC 65
 resetting 53,65
 restoring 52
 typewriter 29
 unlocking 68

 large screen size 85
 length, field
 defining 2,11,30
 limiting 47
 line activity, table for 98,99
 line control characters 172
 local 3270
 with TCAM 137,141,145
 with VTAM 114
 logoff request, in VTAM 165
 logon request, in VTAM 154,156,164,166

 macro instruction
 default options for 77
 table, building 77
 managing devices, techniques for 92
 mapping
 generally 76
 logic flow for 78
 macro instructions for 77,80
 selector pen 81
 table-driven 76,77,80
 master terminal program
 generally 96,97
 uses of 96,97

message
 braodcasting 97
 collecting outstanding 97
 sending as part of a panel 84
 model number, using to calculate buffer
 address 82,83
 modified data tag (MDT)
 definition of 172
 generally 30
 resetting 53,65,68
 selector pen data stream 79
 Modify Field (MF) order 7,43,48
 Monitor macro instruction 125
 MSGFORM macro instruction 146
 multi-colored character 19,20

 non-held line system 103
 non-SNA 3270
 connection 162
 definition of 152,155
 disconnection 163
 with SNA 3270 164
 null 9
 numeric field
 attempt to enter alphanumeric data in 32
 defining 32
 Numeric Lock feature 29,30

 operating system, switching to another 97
 order sequence
 definition of 171
 using macro instructions to prepare 80
 orders
 codes for 47
 coding 47-48
 definition of 42, 172
 generally 42
 Insert Cursor (IC) 42
 Modify Field (MF) 7,43
 Program Tab (PT) 70
 Repeat to Address (RA) 49,51
 Set Attribute (SA) 8,43
 Set Buffer Address (SBA) 42
 Start Field (SF) 42
 Start Field Extended (SFE) 6,47

 paging (see Program Tab (PT) order)
 panel
 caution against creating with multiple
 Writes 65
 coding in Assembler language 47
 design 34-42
 dynamic 81
 example of sequence of 34-41
 modifying existing 62
 semi-dynamic 81
 static 81,82
 panel designations, assigning 34
 panel layout sheet
 contents of 43-44
 default (assumed) values of 45
 generally 39-40
 use of 39-40
 panel title, how to write 46
 panels, sequence of 39
 planes
 single 14,19
 triple 14,19-20

 polling
 general 172
 specific 173
 with TCAM 141
 Prepare to Read (PTR) 94
 definition 173
 print authorization matrix 87-88
 print key on 3278 91
 printer classes 87-88
 printer modes 87-88
 printers, with TCAM 147-150
 program access (PA) keys
 AID codes for 57
 assignment of, suggested 58-59
 definition of 173
 generally 59
 program attention keys (see CLEAR key,
 ENTER key, program access (PA) keys, and
 program function (PF) keys)
 program function (PF) keys
 AID codes for 57
 assignment of, suggested 59
 definition of 173
 generally 59
 Program Tab (PT) order
 definition of 173
 nulls, use to insert 70
 paging, use for 70
 panel defined with 71
 Programmed Symbols (PS) set 17
 protected field 27

 Query Reply 20-26

 Read completion analysis
 DOS local 127
 DOS remote Dial 120
 DOS remote leased line 67,100
 OS local 128
 OS remote Dial 121
 OS remote leased line 68,101
 Read Modified input data stream 74
 Read Partition Query 21
 record mode 152,158-161,173
 remote 3270
 with TCAM 137-147
 with VTAM 154
 Repeat to Address (RA) order
 definition of 51, 173
 example of 51
 Reply Mode 24

 SBA codes 58
 screen design 27-72
 SCREEN macro instruction 147
 screen management
 application program, relation to 73-75
 functions of 73,74
 generally 73,74
 line control, relation to 74
 screen size, large 85
 selector pen
 accounts receivable example, use
 in 36,37
 definition of 173
 fields detected by 29,30,80

- selector pen fields
 - attention 60
 - format for 59
 - macro instruction for 80
 - operator input, combined with 79,80
 - selection 59
 - table for 80
 - transmitting 60-62
 - which selected 80
- Set Attribute (SA) order 8,43,173
- Set Buffer Address (SBA) order 2,42,79,173
- Set Reply Mode 12
- SNA 3270
 - connection 170
 - definition of 157
 - disconnection 162
 - with non-SNA 3270 154
- source device list 88
- split-screen capability 37
- Start Field (SF) order 2,42,47-51,173
- Start Field Extended (SFE)
 - order 6,43,47-51,54-55,173
- structured fields 11
- switched network backup 65
- TCAM (see Telecommunications Access Method)
- Telecommunications Access Method, support for 3270
 - handling AID byte 148
 - handling remote printers 147
 - handling sense/status information 143
 - macro instructions 139-149
 - message handling 146
 - polling with 137
 - Read operation 137
- Telecommunications management
 - with BTAM 93
 - with TCAM 137
 - with VTAM 151
- terminal characteristics table 85
- terminal control program
 - advantages of 96
 - relationship to other modules 96-97
- terminal-initiated logon, in VTAM 156
- terminals
 - adding to a line 97
 - definition of 174
 - removing from a line 97
- Test Request key, result of pressing 102,116
- time of day, maintaining 97
- translate and test (TRT) instructions 74,75
- TSF (Write Structured Field Command) 93
- unprotected field, clearing (see Erase Unprotected to Address (EUA) order and Erase All Unprotected (EAU) command)
- Virtual Telecommunications Access Method
 - connecting with 162
 - defining 3270 to 114,157
 - disconnecting with 161
 - handling Test Request messages 163
 - in basic mode 162-165
 - in record mode 158-162
 - sense/status information 164
 - with non-SNA 3270 164
 - with other access methods 93,137
 - with SNA 3270 164
- VTAM (see Virtual Telecommunications Access Method)
- VTAME (see Virtual Telecommunications Access Method with SNA)
- WCC (see Write control character)
- wraparound
 - generally 33
 - undesired result of 64
- Write commands, multiple
 - "blinking" caused by 65
 - inefficiency of 65
- Write completion analysis
 - DOS local 133
 - DOS remote Dial 120
 - DOS remote leased line 108
 - OS local 134
 - OS remote Dial 121
 - OS remote leased line 109
- Write Continue Conversational Transparent (TTVX) 107
- Write Continue Transparent (TTX) 107
- Write Continue Transparent Block (TTE) 107
- Write control character (WCC)
 - enabling the keyboard 65
 - generally 52
 - hexadecimal codes for 53
 - in modifying panels 65
 - unlock keyboard, use to 70
- Write Structured Field (WSF) 94
- Write Transparent Block (TE) 107
- Write Transparent Text (TX) 107



**International Business Machines Corporation
Data Processing Division**

1133 Westchester Avenue, White Plains, N.Y. 10604

**IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591**

**IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601**

Introduction to Programming the
IBM 3270
Information Display System

READER'S
COMMENT
FORM

Order No. GC27-6999-3

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

How did you use this publication?

- | | | | |
|--------------------------|-------------------------------------|--------------------------|------------------------|
| <input type="checkbox"/> | As an introduction | <input type="checkbox"/> | As a text (student) |
| <input type="checkbox"/> | As a reference manual | <input type="checkbox"/> | As a text (instructor) |
| <input type="checkbox"/> | For another purpose (explain) _____ | | |

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address:

IBM branch office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

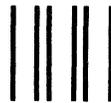
Reader's Comment Form

Cut or Fold Along Line
Intro to Programming the IBM 3270 Information Display System (File No. S360/S370/S3/4300/8100-30) Printed in U.S.A. GC27-6999-3

Fold and tape

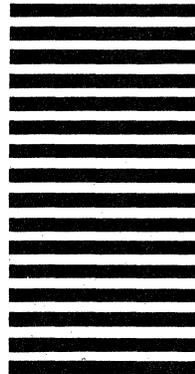
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 52Q
Neighborhood Road
Kingston, New York 12401

Fold and tape

Please Do Not Staple

Fold and tape



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601