GC27-6999-2
File No. S360/S370-30

Systems

Introduction to Programming the IBM 3270

IBM

Third Edition (August 1977)

This edition replaces GC27-6999-1. Changes to this third edition will be reported in subsequent revisions or technical newsletters.

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.

Copies of this and other IBM publications can be obtained through your IBM branch office.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to IBM Corporation, Department 63T, Neighborhood Road, Kingston, New York 12401.

BEFORE YOU USE THIS BOOK

Who This Book Is For

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

For those programmers who plan and code the messages seen on 3270 displays, this book may be the only book required.

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

How This Book Is Organized

This book is divided into these sections:

1: SCREEN DESIGN
   Introduces important 3270 concepts. 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.

2: SCREEN MANAGEMENT
   Suggests macro definitions and programming routines that might be written to encode and decode messages to and from the display.

3: BTAM SUPPORT
   Suggests including I/O operations (reading, writing, error recovery) in a module separate from message formatting. Contains descriptions and flowcharts to aid in writing error recovery routines for use with BTAM. Discusses sense/status analysis.

4: TCAM SUPPORT
   Suggests handling messages by means of two modules for the user's application program. Describes the TCAM macro instructions that affect the 3270. Suggests how to handle remote printers.

5: VTAM SUPPORT
   Summarizes the VTAM information for the 3270. Describes using VTAM with SNA and non-SNA 3270s. 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

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

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, GC30-2004

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 Programmers 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

CONTENTS

ILLUSTRATIONS

Summary of Amendments

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

3274 Control Unit
3276 Control Unit/Display Station
3278 Display station
3287 Printer
3289 Line Printer

Generally, these devices are compatible with other 3270 terminals; however, certain precautions are noted which must be observed in integrating them into your system. This edition also corrects several technical inaccuracies.

Changes to the text are indicated by a vertical bar in the left margin.

CHAPTER 1. 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 "O" 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 basis and specify the characteristics or "attributes" of each individual field. The 3270 then provides program and data control based on your individual field definitions.

How Fields are Defined

Each data field is established by writing a field attribute control code, or attribute character, as the first position of the field. A field is defined as the attribute character, plus all the data following it up to the next attribute character. The placement of attribute characters defines the field lengths, and the content of the attribute characters defines the other field characteristics. In the following examples, the symbol □ designates an attribute character.

All the characters in a field, except the attribute character itself, assume identical characteristics based on the specifications within the attribute character. In Figure 1 the characteristics of the field NAME: are controlled by the attribute 1 , and terminated by the attribute 2 . The placement of attributes controls the length of the fields.

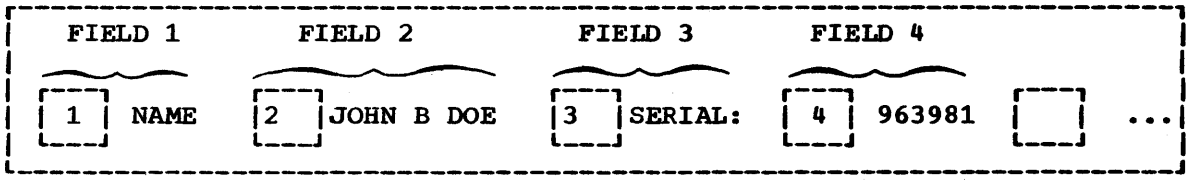


Figure 1. Example of 4 Fields and Attribute Characters

Field attributes can be modified or removed by a 3270 program. Removal of the attribute character [2] causes NAME: JOHN DOE to be considered by the 3270 as a single field. Changing the content of the attribute [3] alters the characteristics of SERIAL: even though SERIAL: itself has not been altered and it still remains associated with that attribute.

What Attributes May be Assigned to a Field

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

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 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.

Character Content: A field is either alphameric or numeric. An operator can enter alphameric, numeric, or special characters in an alphameric 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 shows what characters may be entered with various combinations of keyboards and field types.

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 will not be 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 3274's or 3276's 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 non-display attribute byte prior to sending the intended new non-displayable data to preclude its momentary appearance on the screen.
- do not overwrite a non-display attribute byte of the currently displayed image when partially changing field formats.

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. Results of Keyboard and Field Combinations

All characters within a displayable field can be displayed at regular brightness or at a high intensity so that they stand out among regular display fields. High intensity may be used to call attention to error conditions or to highlight protected or format fields. Normal intensity 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 space 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 or cursor-select 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

You will find that these attributes are the most commonly used.

The attribute character for each 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 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-ON 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.

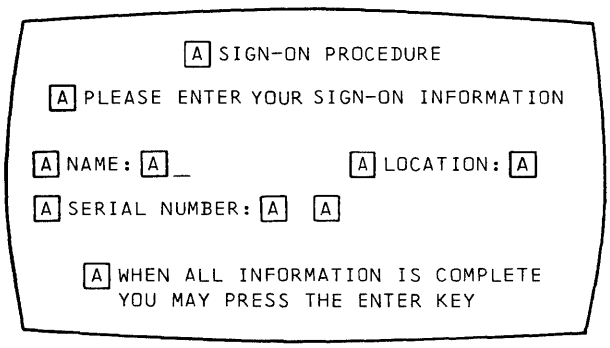


Figure 3. Example of Attribute Specification

FIELD 2: "PLEASE ENTER ... INFORMATION"

You should specify this field as protected. Remember that the characteristics of a field are determined by the attribute character 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 characters or you may choose to omit the attribute character 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).

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 (alphameric, 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 character 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 characters 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 character 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 character 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 character 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 character 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 character 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-ON." 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 characters 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 term "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 4, he might receive another panel such as the one shown in Figure 5.

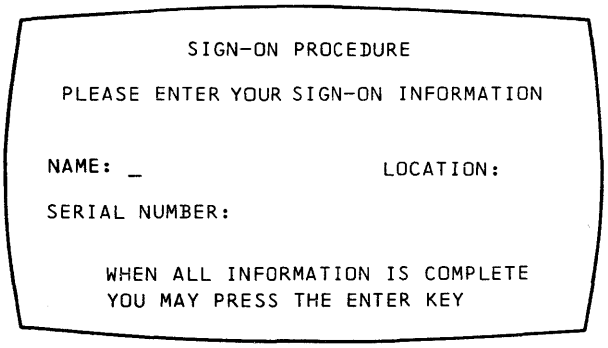


Figure 4. An Example of a Panel

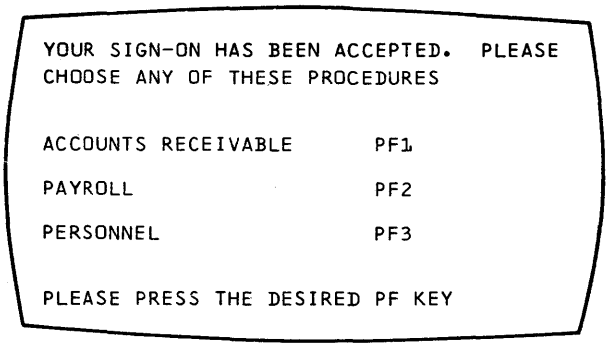


Figure 5. Another Example of a Panel

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 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.

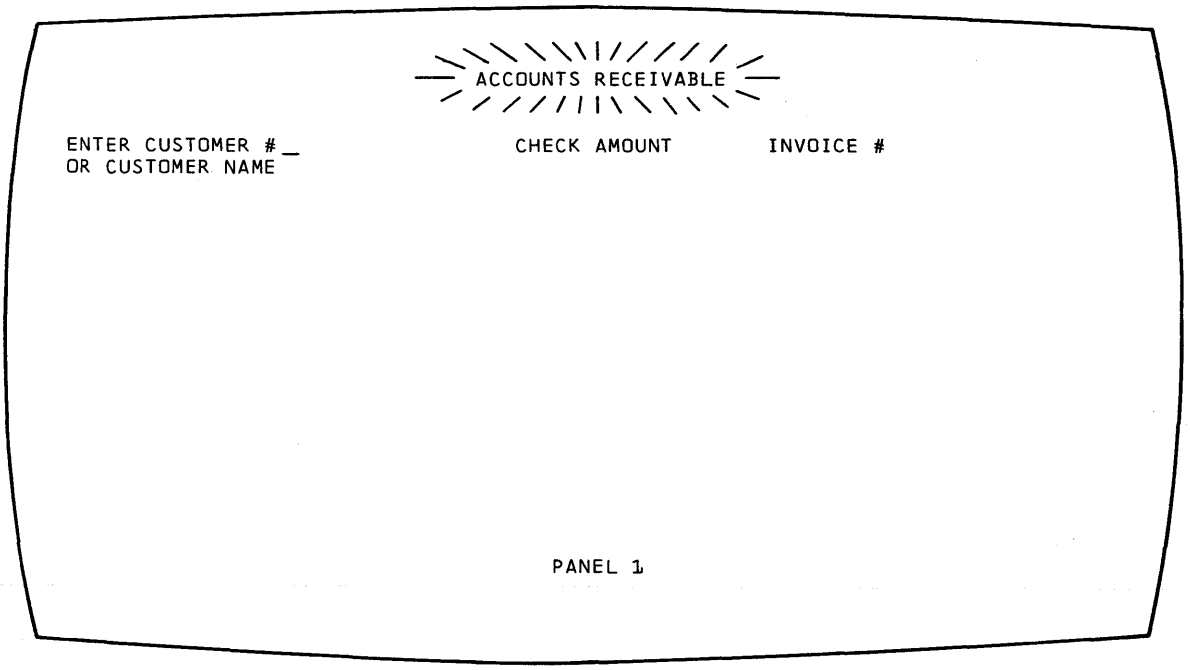


Figure 6. Panel 1 of an Accounts Receivable Application

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.

If, however, no invoice is returned and the customer number is not known, the customer name can be entered. The name need not be the complete name of the company; it can be the first name of the company. In our example, the check says only "CAPITOL" so that is what the operator enters. When the name has been entered, the terminal operator presses the ENTER key. The customer number is missing, so Panel 2 is displayed.

Panel 2, shown in Figure 7, shows all customers and customer numbers phonetically similar to the name entered in response to Panel 1. Item numbers in Panel 2 allow the terminal operator to select one by using a corresponding Program Function (PF) key (see "Program Attention Keys" in this section).

As a result of terminal operator response to Panel 2, Panel 3 (shown in Figure 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 can not easily tell the invoices to which the payment is applied, he can select CALC instead of APPLY.

```
ITEM    CUST #    NAME/ADDRESS          ITEM   CUST #    NAME/ADDRESS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 1      0010341   CAPITAL AVIATION       5     0052693   CAPITOL ELECTRIC
                  711 HILLSBOROUGH ST.                    56 STATE ST.
                  RALEIGH, N.C.                           MONTPELIER, VT.
                  27611                                   05602

 2      0028472   CAPITOL BAKERIES       6     0084362   CAPITOL FEATHER CO.
                  1800 MAIN ST.                           899 LOGAN ST.
                  COLUMBIA, S.C.                          DENVER, COLO.
                  29201                                   80217

 3      0034020   CAPITOL COLA CORP      7     0048729   CAPITAL GLASS CO.
                  1439 PEACHTREE ST. NE                   121 STATE ST.
                  ATLANTA, GA.                            ALBANY, N.Y.
                  30309                                   12201

 4      0041938   CAPITAL DRUG CO.       8     0038492   CAPITOL HOLDING CO.
                  201 NORTH 9TH ST.                       1609 SHOAL CREEK B
                  RICHMOND, VA.                           AUSTIN, TEXAS
                  23219                                   78701

                                PANEL 2                                   _
```

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

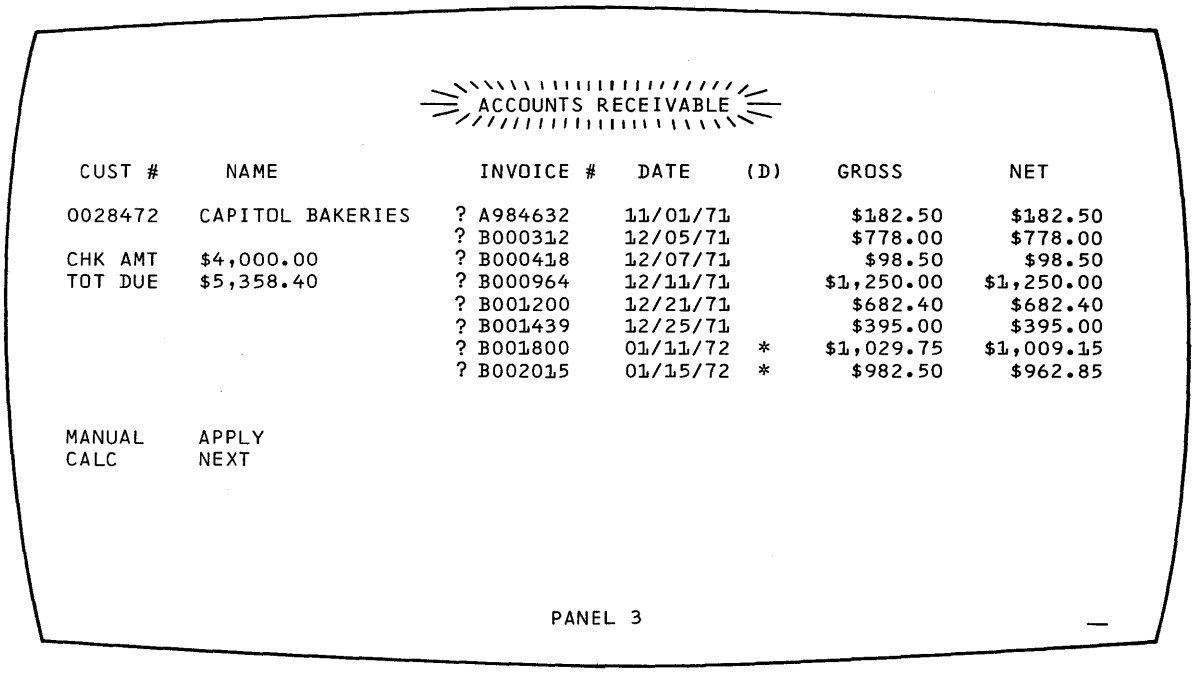


Figure 8. Panel 3, Showing the Customer's Open Invoices

Selecting CALC displays Panel 4 (Figure 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 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 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 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.

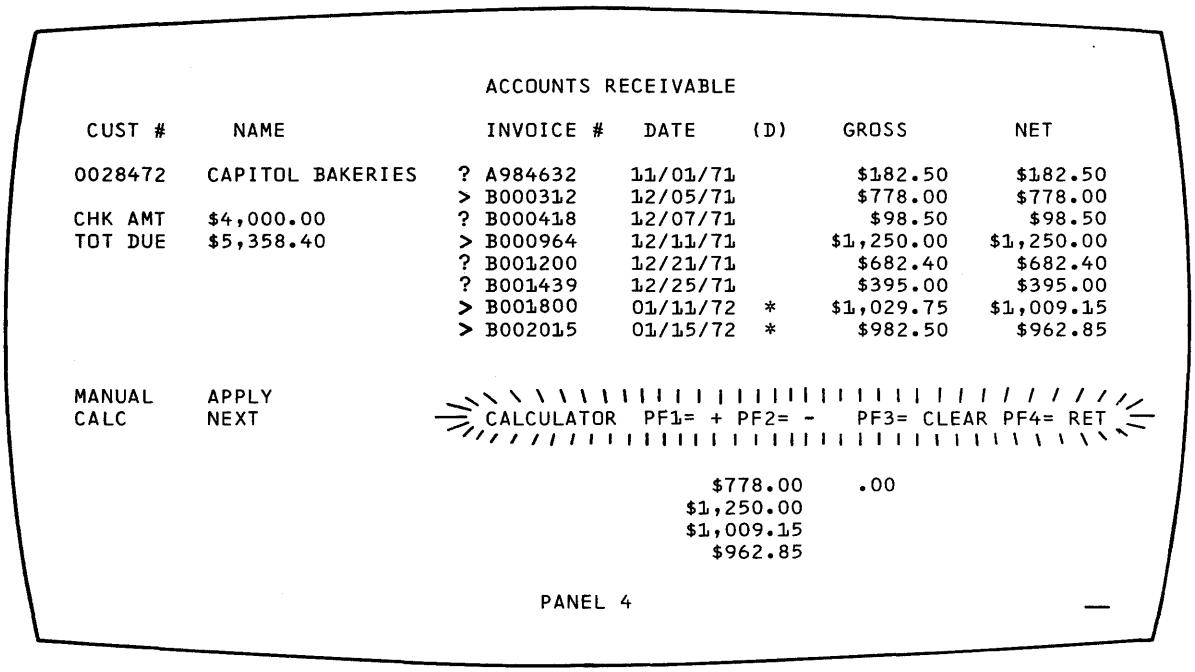


Figure 9. Panel 4, Showing Use of the Calculator

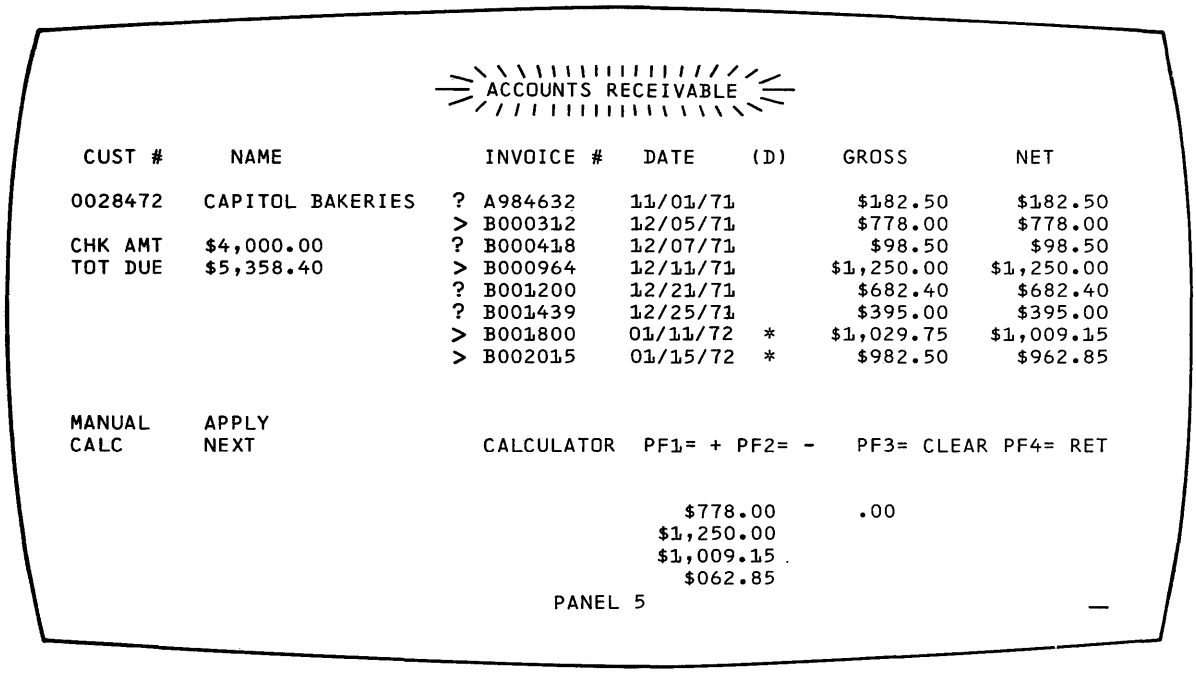


Figure 10. Panel 5, Showing Selection of Invoices after Using the Calculator

Panel 6, in Figure 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.

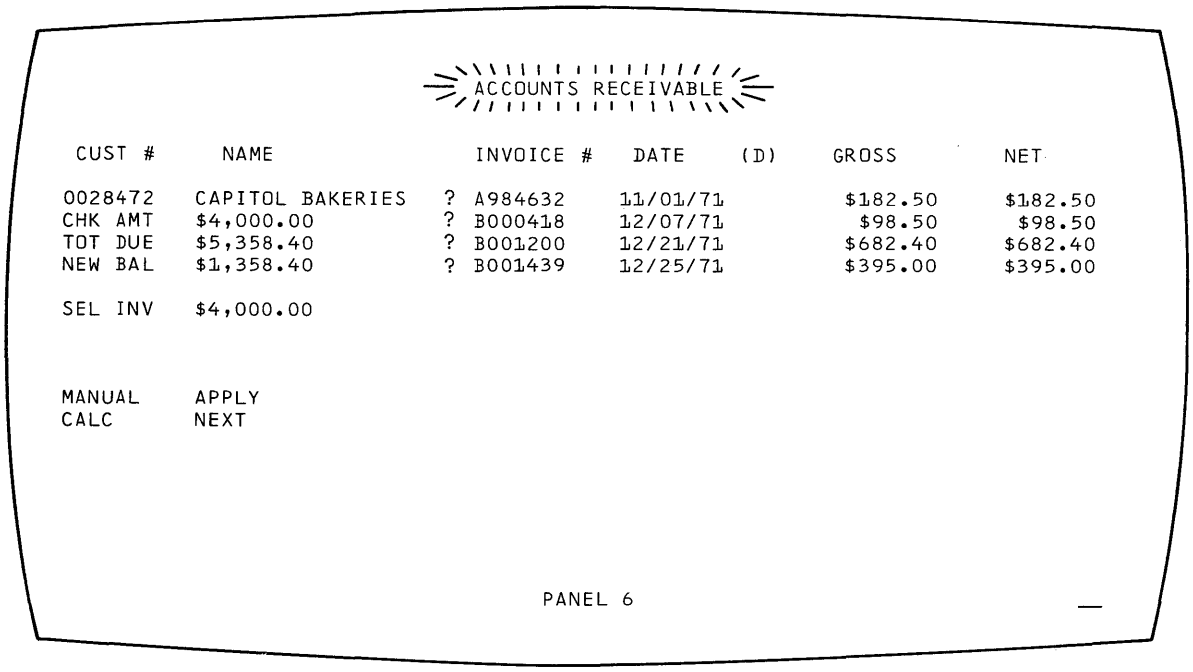


Figure 11. Panel 6, Showing New Balance after Posting

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 12.

Figure 12. Sign-on Panel Block Diagram

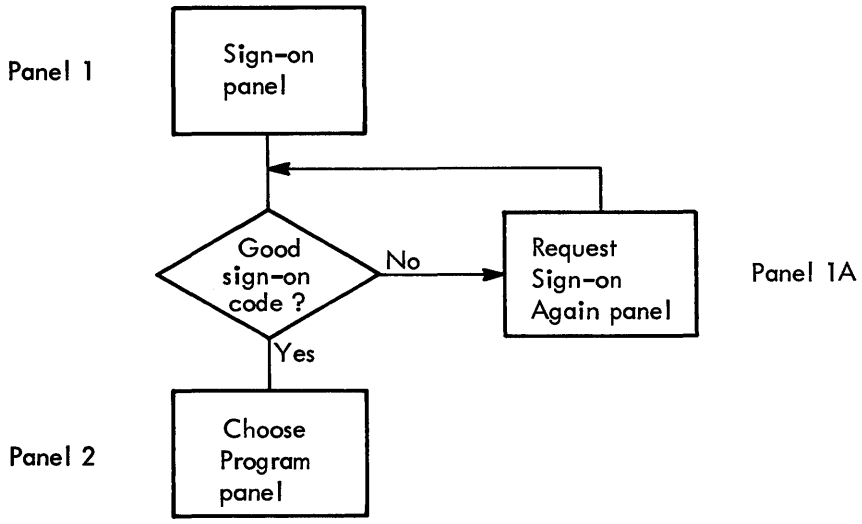


Figure 13. Block Diagrams

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 re-enter a sign-on name. Figure 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 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.

An Example of Laying Out a Panel

To lay out a panel, consider the sign-on panel shown in Figure 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 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. It is assumed you are using the 480-character display.

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 character. The attribute character cccupies 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 character. 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 characters and the cursor position, the sign-on panel appears as shown in Figure 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).

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 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.

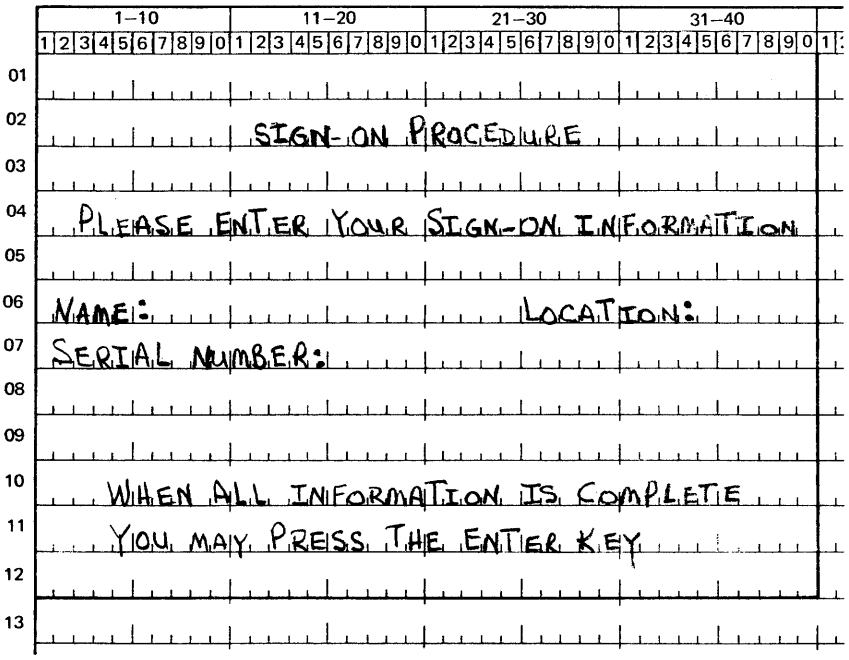


Figure 14. Sign-on Panel as Written Out on Layout Sheet

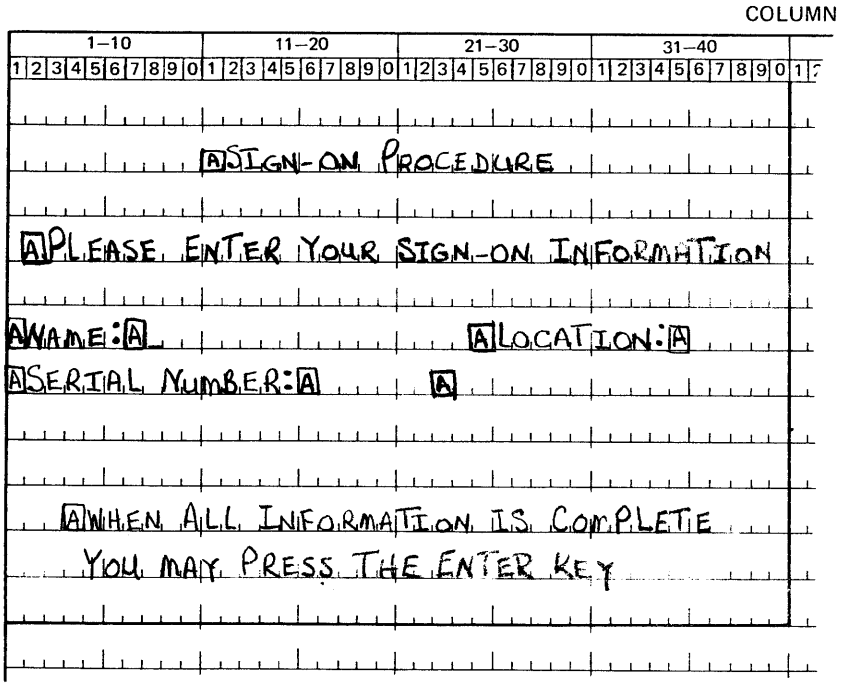


Figure 15. Panel Layout Including Attribute and Cursor Positions

ITEM	DISPLAY/PRINTER ROW	DISPLAY/PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS	ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP/PRT	ATTRIBUTE MDT ON
1	2	11				✓					
2	4	2				✓					
3	6	1				✓		✓			
4	6	7									
5	6	25				✓		✓			
6	6	35									
7	7	1				✓		✓			
8	7	16					✓				
9	7	23				✓					
10	10	4				✓					

Figure 16. Laying Out Field Attributes

DATA STREAM CODING

You must communicate certain information to a 3270 device or its control unit so that it can use the panels you have designed. This information includes commands, control characters, orders, and data.

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. For the examples given below, assume that you begin with a clear screen: all writes to the 3270 are Erase/Write or Erase/Write Alternate (for 960, 2560, 3440, or, optionally, 1920 character displays) commands and all positions are set to nulls. (Commands are discussed in more detail in this chapter under "The Relationship of One Data Stream to Another." Refer to the 3270 Component Description for the command codes. Note that the only command codes used for a 3270 with VTAM support are those listed under "Remote" in the command code table in the 3270 Component Description.) 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. (Control characters are discussed later in this section.) Orders are instructions written to the 3270 to tell the display unit how to format your panel. They control the creation and placement of fields and data. You may reduce the size of your data streams by careful order selection. (Orders are discussed below.)

Orders

Orders (1) position, define, and format data being written to the device; (2) erase selected unprotected data stored in the device; and (3) reposition the cursor.

Three orders provide enough instruction to format every panel:

- Start Field (SF) order: Specifies that the next character is an attribute character.
- Set Buffer Address (SBA) order: Specifies an address for data and successive orders.
- Insert Cursor (IC) order: Moves the cursor to the current buffer address.

These orders are included with the text, which is both the data you have in your computer for the terminal operator, such as field headings

or inquiry responses, and the data that the operator has that must be provided to the computer, such as serial number, part number, or quantity desired. The orders and text are sent to the display unit and are interpreted by a control unit to which the display unit is attached. The control unit formats the panel text before it is actually displayed at the display station.

Adding Orders to the Panel Layout Sheet

The back of the panel layout sheet is used for writing the panel orders. The headings at the top of the columns indicate what the columns should contain.

The first six columns as shown in Figure 17 identify items in the text, their addresses, and the orders required to format them. The column headings are explained below:

- ITEM: Refers to any part of the panel that requires one or more orders to the control unit to format it. There are 11 items in the sign-on panel:

  1. SIGN-ON PROCEDURE
  2. PLEASE ENTER YOUR SIGN-ON INFORMATION
  3. NAME:
  4. Input field
  5. LOCATION:
  6. Input field
  7. SERIAL NUMBER:
  8. Input field
  9. Field to limit size of serial number input
  10. WHEN ALL INFORMATION IS COMPLETE
  11. YOU MAY PRESS THE ENTER KEY

  It is only by coincidence that the number of items in this example equals the number of fields. Since each field requires an SF order, there are always at least as many items as fields. There are more items than fields when, for example, the SBA order is used to space over unused positions within a single large field, as in Item 11.

- ROW, COL: Contain the starting location (row, column) address of each item.

- DEC, HEX: Are for a different addressing format which you do not need if you use the row, column addressing format. Therefore, you may use these columns for any notes to yourself or leave them blank.

- ORDERS: Contains the orders you are writing such as SBA, SF, or IC.

ITEM	DISPLAY/PRINTER		BUFFER ADDRESS		ORDERS
	ROW	COL	DEC	HEX	

Figure 17. Text Items on Panel Layout Sheet

As shown in Figure 18, the next six columns under the word ATTRIBUTE provide the field attributes that can be defined with each attribute character. The programmer checks the appropriate columns of the attributes he is changing from the default values:

- PROT: Protected
- NO.: Numeric
- HI INT: High intensity
- SEL PEN DET: Selector-pen-detectable or cursor selectable
- NONDISP/PRT: Not displayed (nor printed at printer)
- MDT ON: Modified data tag cn

At the bottom of the six columns, the attribute values are shown (Figure 19) that are automatically provided unless you specify a change. You must, however, specify a hexadecimal order value for the default attributes, as discussed under "Coding the Panel" in this section. The default values are:

- UNPR: Unprotected
- A/N: Alphameric (alphabetic and numeric)
- NORM: Displayed at regular brightness
- NON: Not detectable by the selector pen or cursor select
- NORM: Displayed (at regular brightness)
- OFF: Not modified

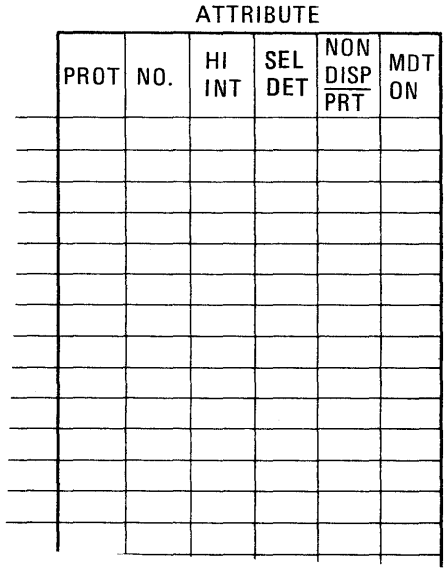


Figure 18. Field Attributes

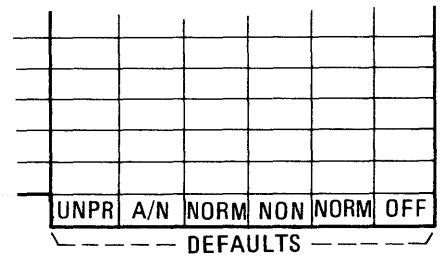


Figure 19. Attribute Default Values

You are now ready to add the required orders to the panel layout form. This may require that you rewrite the right half of the form if it was originally prepared without regard to orders or if insufficient space was allowed.

Figure 20 shows a completed layout sheet containing all the orders to be sent with the sign-on panel. The hexadecimal order values are discussed under "Coding the Panel" in this section and shown in Figure 22. Each item on the panel has been assigned an item number to help you correlate the text with its associated orders.

Item 1. SIGN-ON PROCEDURE. To write this title, you must tell the control unit:

- Where you want the title displayed on the panel. The SBA order sets the buffer address (SBA) to location R2, C11.
- That this location is the start of a field. The SF order tells the control unit that the location contains an attribute character and not a text character. You also indicate which attributes the attribute character is defining. In this case, the field is protected. The rest of the attributes for the field are default attributes and therefore do not have to be changed.

Item 2. PLEASE ENTER YOUR SIGN-ON INFORMATION: To write this information, the control unit must know only where the text is located. Therefore, you must write an SBA instruction followed by the address R4, C2. This is also the beginning of a protected field, so you should include an SF order and a protected attribute.

Item 3. NAME: As with Item 2, you must identify where this text is displayed. Therefore, you must write an SBA order followed by the buffer address R6, C1, where the text begins. R6, C1 is also the beginning of a protected, high-intensity field and you should include an SF and an attribute as shown.

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	OR-DERS		ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP/PRT	ATTRIBUTE MDT ON
1	02	11			SBA							
					SF	Att	✓					
2	04	02			SBA							
					SF	Att	✓					
3	06	01			SBA							
					SF	Att	✓		✓			
4	06	07			SF	Att						
					IC							
5	06	25			SBA							
					SF	Att	✓		✓			
6	06	35			SF	Att						
7	07	01			SBA							
					SF	Att	✓		✓			
8	07	16			SF	Att		✓				
9	07	23			SBA							
					SF	Att	✓					
10	10	03			SBA							
					SF	Att	✓					
11	11	05			SBA							

Figure 20. Completed Order and Attribute Information

Item 4. Input Field for operator's name. Since this item immediately follows Item 3, the control unit already knows the correct address. Therefore, there is no reason to issue an SBA order. Item 4 is the start of a new field, however, so you must issue an SF order to instruct the display to expect an attribute character next. The attribute character defines the input field as unprotected (U), alphameric (A), normal intensity, not detectable by selector pen, and no MDT on. Because these are the default attributes, you do not have to check anything in the attribute definition columns.

The cursor should follow the attribute character to indicate where the operator should begin to enter information. The Insert Cursor (IC) order displays the cursor at this current buffer address. After the display has stored the attribute character in location R6, C7, the new current address is R6, C8; this is the place where the cursor appears on the panel.

Item 5. LOCATION: The control unit must have two orders for this item which (1) give the starting buffer address (SBA) 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 character 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 SBA 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 character 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 DC 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 DC statements in the hexadecimal codes listed in Figure 21. Thus, SF is represented by 1D, SBA by 11, and IC by 13.

Each part of each order must be written in hexadecimal including the attribute character that follows the SF order and the buffer address that follows the SBA order. The IBM 3270 Reference Summary, GX20-1878 contains the hexadecimal codes for all the attribute character 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-ON 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 first order for the title is the SBA order. Figure 21 shows that the SBA 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 SBA order. The EBCDIC address is 40F2 and it follows the SBA 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 SBA on the layout form for possible future reference. You may, if you prefer, look up all the addresses and record them in a similar manner before you begin to write your DC statements. See Figure 22 for an example.

BUFFER CONTROL ORDERS AND ORDER CODES

Order Sequence / Order	Byte 1 (Order Code) EBCDIC Hex	Byte 1 (Order Code) ASCII Hex	Byte 2	Byte 3	Byte 4
Start Field (SF)	1D	1D	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	
Keyboard Only					
Duplicate (DUP)	1C	1C			
Field Mark (FM)	1E	1E			

Figure 21. Buffer Control Orders and Order Codes

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS		ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP PRT	ATTRIBUTE MDT ON
1	02	11		140F2	SBA							
					SF	AH	✓					
2	04	02		C1F9	SBA							
					SF	AH	✓					
3	06	01		C3C8	SBA							
					SF	AH	✓		✓			
4	06	07			SF	AH						
					IC							
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 22. Sign-on Procedure Panel Orders and Attributes

The next order for the title is the SF order, which is followed by the attribute character. Attribute characters are shown in Figure 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:

```
DC  X'1140F21D60'
```

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

```
DC  X'1140F21D60'
DC  C'SIGN-ON 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:

```
DC  X'1D4013'
```

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

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

```
DC  X'1D40'
DC  X'13'
```

Each item from the panel layout sheet is coded in this fashion. Figure 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 in the section "Write Control Character (WCC)".)

ATTRIBUTE CHARACTER BIT DEFINITIONS

ATTRIBUTE						Bits	Hex
Prot	A/N	MDT ON	High Intens	Sel Det	Non Disp PRT	23 4567	
U						00 0000	40
U		Y				00 0001	C1
U				Y		00 0100	C4
U		Y		Y		00 0101	C5
U			H	Y		00 1000	C8
U		Y	H	Y		00 1001	C9
U			–	–	Y	00 1100	4C
U		Y	–	–	Y	00 1101	4D
U	N					01 0000	50
U	N	Y				01 0001	D1
U	N			Y		01 0100	D4
U	N	Y		Y		01 0101	D5
U	N		H	Y		01 1000	D8
U	N	Y	H	Y		01 1001	D9
U	N		–	–	Y	01 1100	5C
U	N	Y	–	–	Y	01 1101	5D
P						10 0000	60
P		Y				10 0001	61
P				Y		10 0100	E4
P		Y		Y		10 0101	E5
P			H	Y		10 1000	E8
P		Y	H	Y		10 1001	E9
P			–	–	Y	10 1100	6C
P		Y	–	–	Y	10 1101	6D
P	S					11 0000	F0
P	S	Y				11 0001	F1
P	S			Y		11 0100	F4
P	S	Y		Y		11 0101	F5
P	S		H	Y		11 1000	F8
P	S	Y	H	Y		11 1001	F9
P	S		–	–	Y	11 1100	7C
P	S	Y	–	–	Y	11 1101	7D

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

Figure 23. Attribute Character Combinations in Hexadecimal

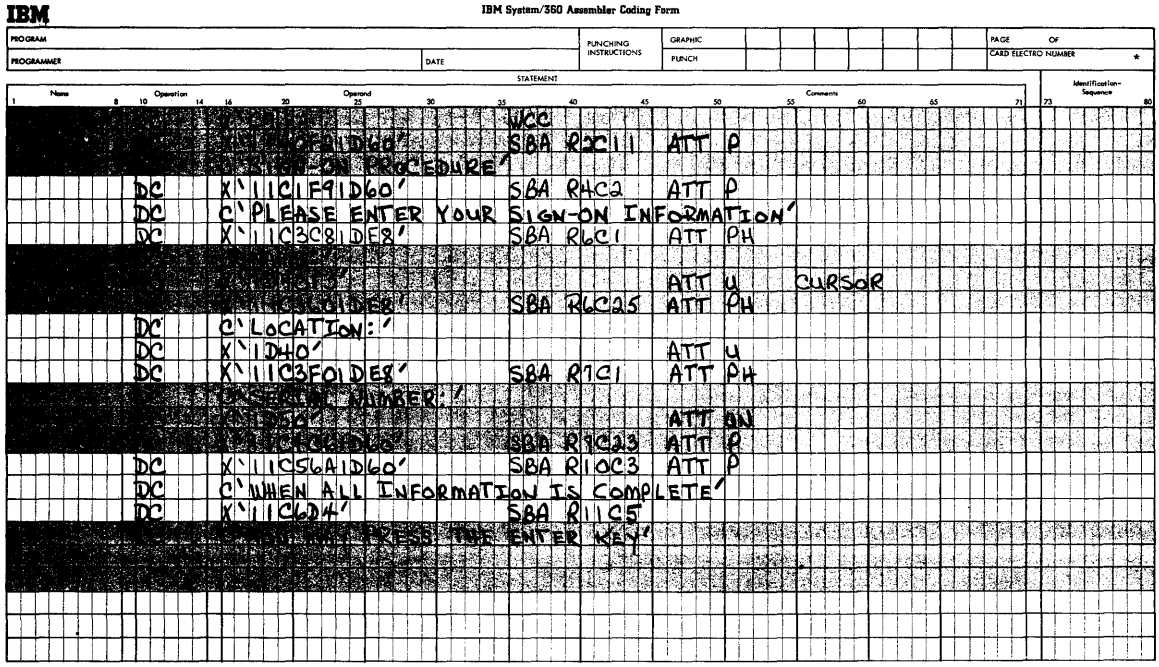
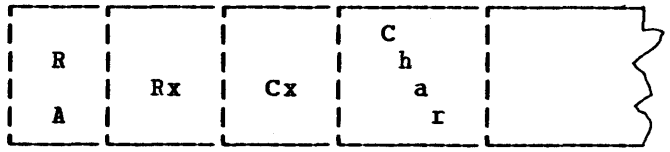


Figure 24. Assembler Language Statements for Sign-on Panel

Repeat to Address Order

The Repeat to Address (RA) order stores a specified alphameric 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 SBA order. After the stop address you specify the character that you want repeated. 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 25.

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.

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS		ATTRIBUTE PROT	NO.	HI INT	SEL DET	NON DISP PRT	MDT ON
1	02	11		40F2	SBA							
					SF	Att	✓					
2	04	02		C1F9	SBA							
					SF	Att	✓					
3	06	01		C3C8	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		C5CA	SBA							
					SF	Att	✓					
11	11	05		C6D4	SBA							
12	01	01			RA	*						

Figure 25. Example of RA Order

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 characters 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 26.

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 26. WCC Hexadecimal Codes

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 27. 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 28 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."

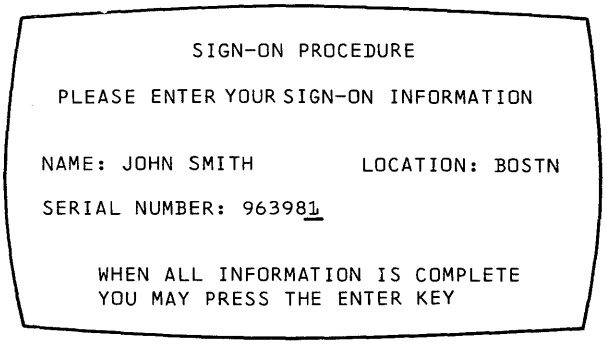


Figure 27. Sign-on Panel with Operator's Input

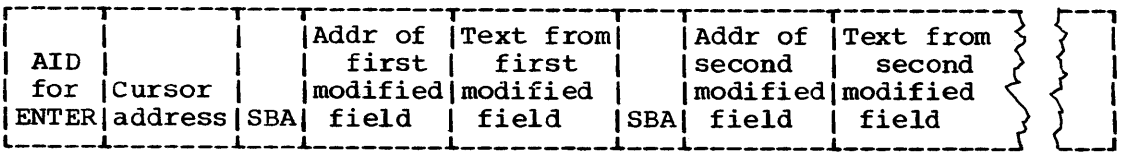


Figure 28. Input Data Sequence

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 29.

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.

ATTENTION IDENTIFICATION (AID) CONFIGURATION

AID VALUES FOR TEXT READ

Graphic Character	EBCDIC Hex	Operator Action
–	60	No action by display operator
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 29. Attention Identifiers (AID) 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 character 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 character.

When any character location in an input field is modified by the operator, the MDT in the attribute character 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 character 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 character's address plus one) of the modified field. The SBA 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.

SBA Codes

SBA 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:

The hexadecimal codes are:

7D: The AID code for the ENTER key (Refer to Figure 29)

C4C6: The cursor address R7, C23. The cursor is at the next character location after the entered serial number.

11: The SBA (Set Buffer Address) order code which tells the program that the next 2 characters are addresses. (Refer to Figure 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 29, 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 are 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 30. 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.

The attribute character, 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 character can be detected by the selector.

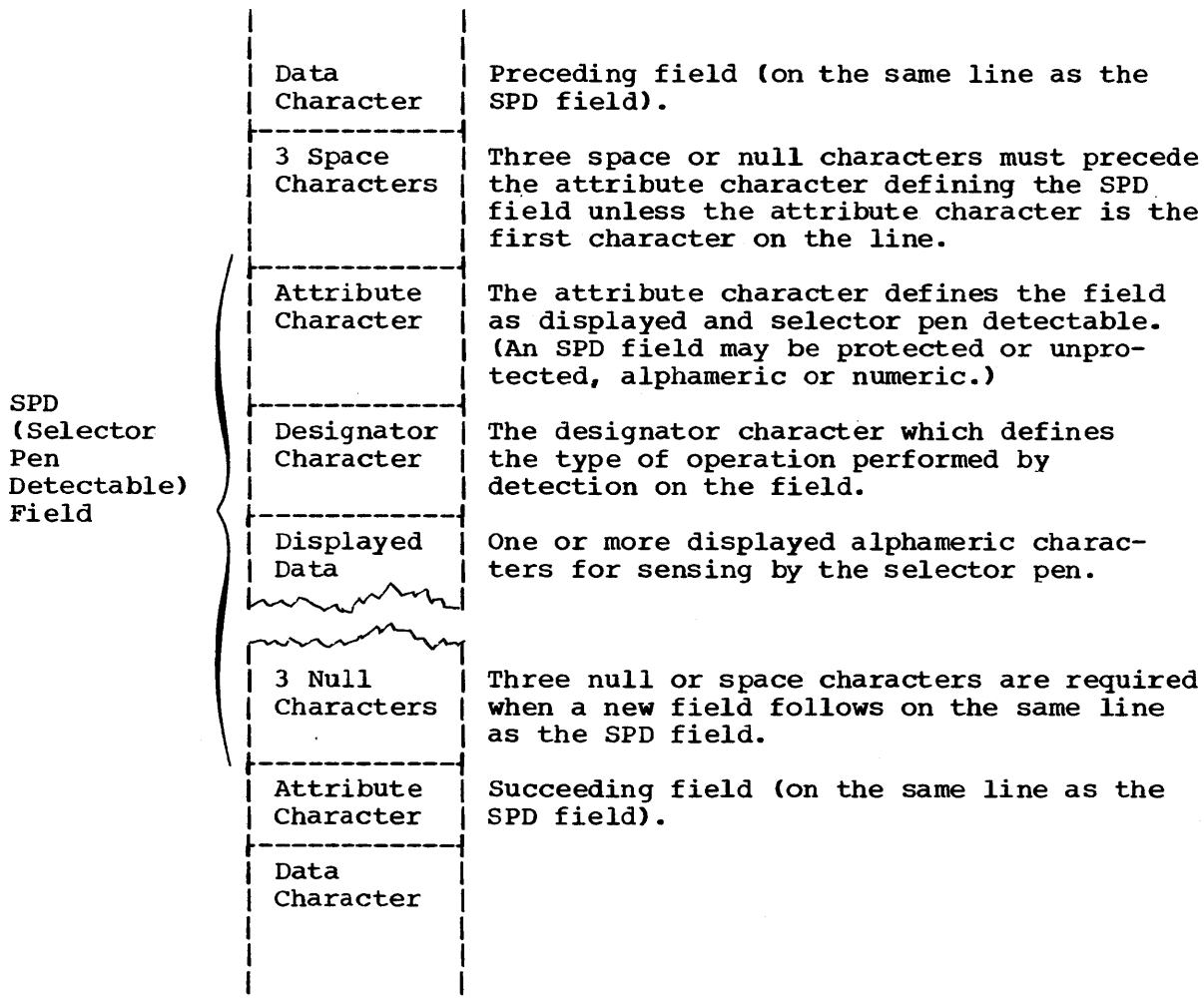


Figure 30. Definition of Field for Selector Pen Operation

Designator Characters

Designator characters define three types of selector fields: selection and two types of attentien. 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 MDT bit in the attribute character 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 MDT 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 information is released to be read by the computer when it is ready to do so. The second type of attention field is the ampersand (&) with the 3274 or 3276 control unit. Probing this field causes the program to issue a Read Modified command and obtain both the address and data of each field.

Figure 31 shows a sample selector pen panel that illustrates some of the special input and output data stream considerations.

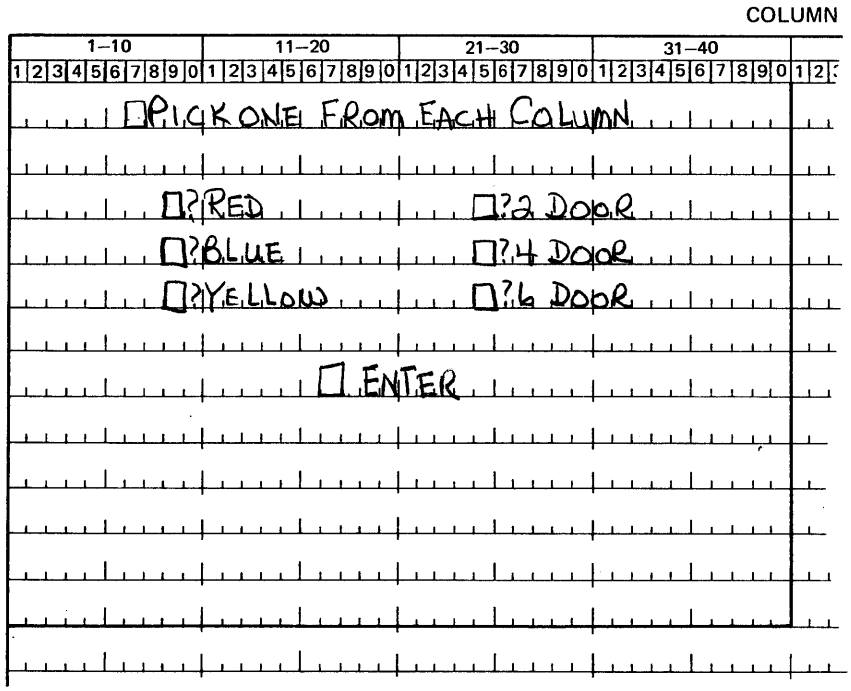


Figure 31. Sample Panel for Selector Pen or Cursor Select Detection

For output, an Erase/Write creates the panel. In the WCC, you enable input and optionally reset the MDTs. Next you specify an SBA sequence to get you to R1, C7, followed by an SF with a protected attribute.

This should be followed by the heading "PICK ... COLUMN" and another SBA to R3, C9. Then specify an SF order, followed by a protected (detectable fields may be protected) and detectable attribute. Next you need the designator "?" followed by "RED":

C	O	L	U	M	N	SBA	R3	C9	SF	P+D	?	R	E	D	

An SBA after "RED" to R3, C25, provides more than the 3 required null characters and positions the SF, attribute, and designator for "2 DOOR". This type of sequence is repeated for the remaining fields to location R7, C28. The designator here must be a null or a blank so that probing or selecting by cursor the "ENTER" field releases the selection to the computer.

As the operator uses the selector pen or cursor select, the program correlates the address of each selector-pen-detectable field with the data associated with it.

To combine selector-pen or cursor select detectable input with keyboard or cursor select input, use the keyboard to release the data to the computer by pressing the ENTER key or a PF key. Use of the selector pen or cursor select to release the data, such as by selecting "ENTER" in 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 DOOR the symbolic input would appear as follows:

Pen AID	Cursor ADDR	SBA	R3	C10	SBA	R4	C26	

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 ccntents. 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 tc 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 32.

You would like the screen to look like Figure 33.

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 32 to look like Figure 33, 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.

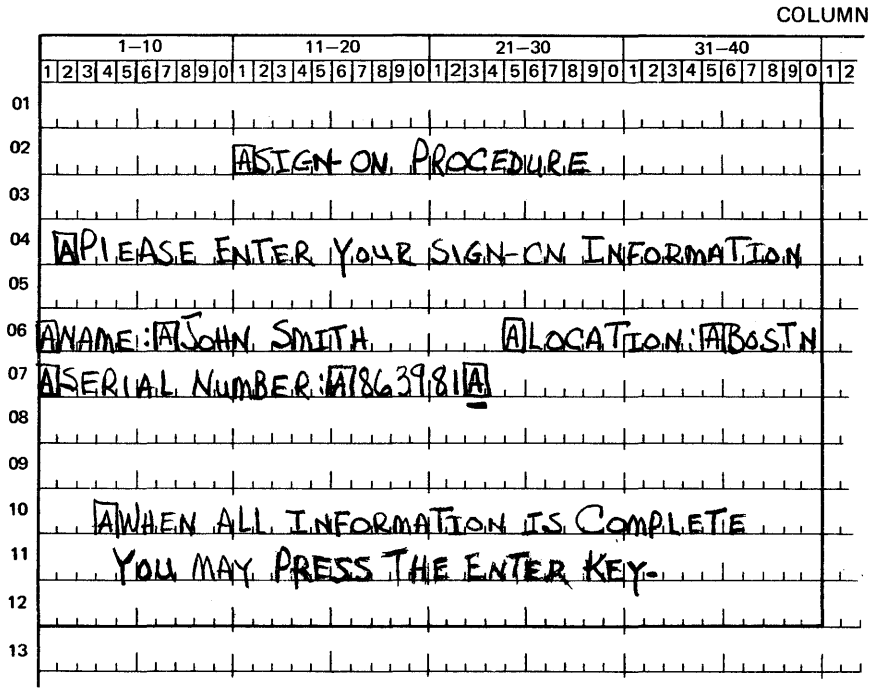


Figure 32. Modifying an Existing Panel -- Basic Panel

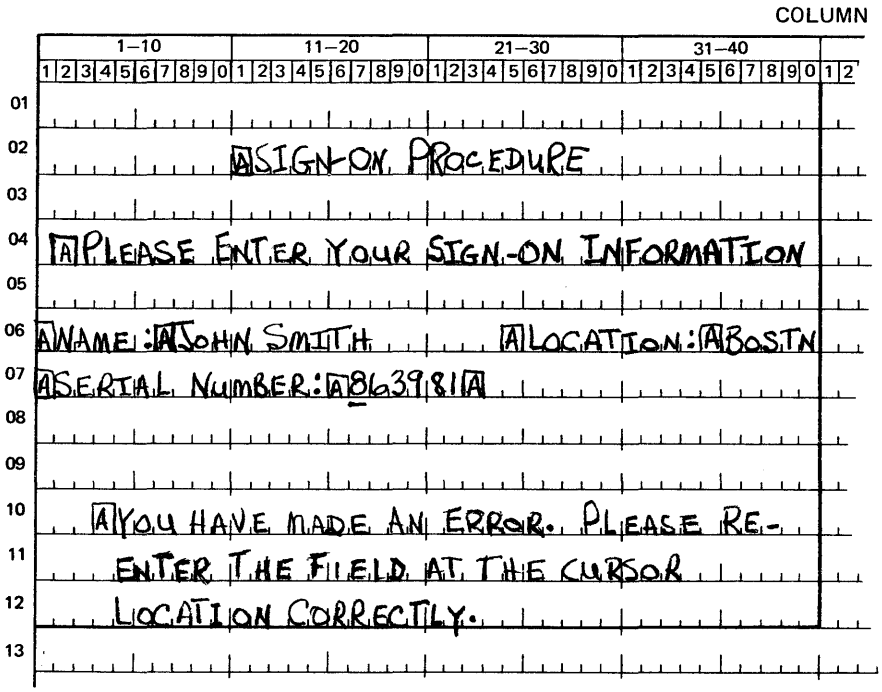


Figure 33. Existing Panel with Error Message

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS		ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP PRT	ATTRIBUTE MDT ON
1	07	17			SBA							
					IC							
2	10	04			SBA							
					SF	Att	✓		✓			
		"LINE	1		OF	ERROR	MESSAGE				"	
3	11	05			SBA							
		"LINE	2		OF	ERROR	MESSAGE"					
4	12	05			SBA							
		"LINE	3		OF	ERROR MESSAGE"						

Figure 34. Panel Layout Changes for Error Message (Keyed to Text)

To do this the right side of your panel layout for the error panel might (in abbreviated form) look like Figure 34.

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-ON 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-CN 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/O 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.

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 32, 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 DC X'C7' (see Figure 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 33 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 35.

Begin again with the desired WCC. Place the cursor at R7, C17 with an SBA 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, including attributes) from the current buffer address up to, but not including, the specified stop address.

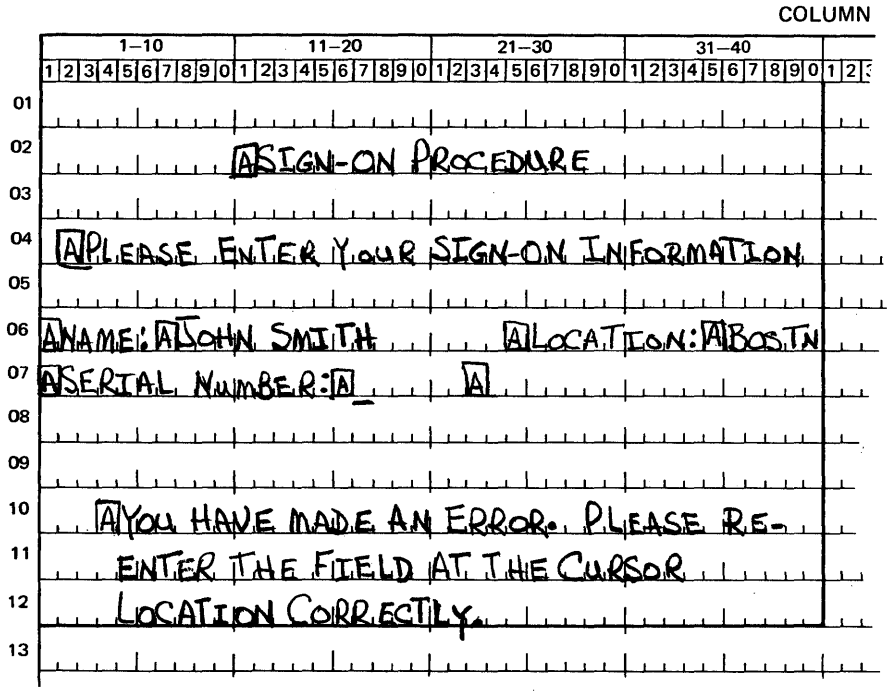


Figure 35. Error Message Panel with Serial Number Field Erased

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 SBA, saving three characters of transmission (see Figure 36). 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 37.

First place the cursor at R7, C17, then "back up" with an SBA to R6, C8 (the name input field) before issuing the EUA to R10, C4 (see Figure 38).

You could have started at R6, C8 with an SBA 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 SBA to insert the cursor.

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS		ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP/PRT	ATTRIBUTE MDT ON
1	07	17			SBA							
					IC							
	10	04			EUA							
2					SF	AH	✓		✓			
	"LINE 1 OF			ERROR	MESSAGE"							
					•							
					•							
					•							

Figure 36. Example of EUA Use

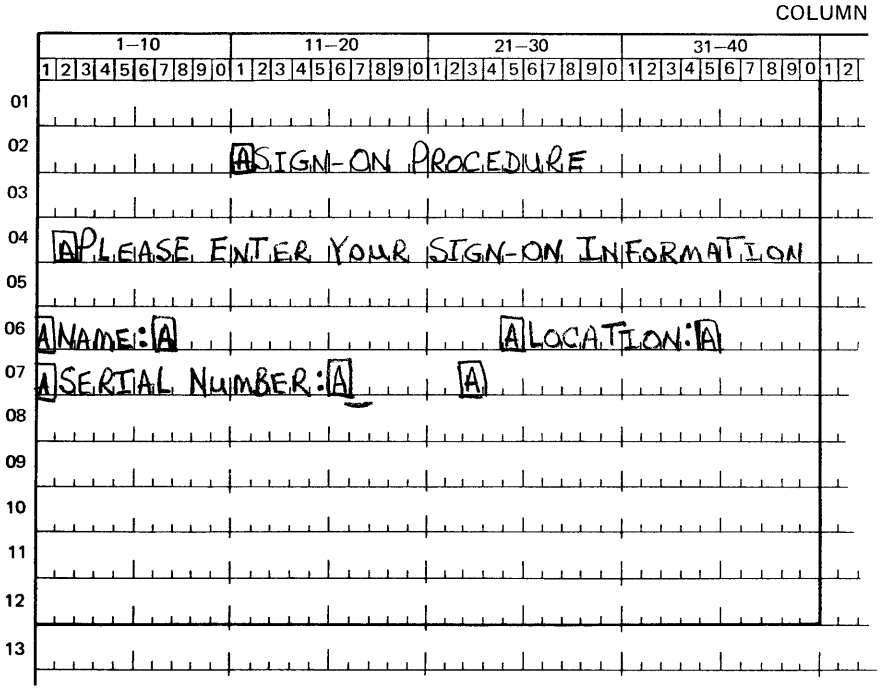


Figure 37. Sign-on Panel with Three Erased Fields

ITEM	DISPLAY PRINTER ROW	DISPLAY PRINTER COL	BUFFER ADDRESS DEC	BUFFER ADDRESS HEX	ORDERS		ATTRIBUTE PROT	ATTRIBUTE NO.	ATTRIBUTE HI INT	ATTRIBUTE SEL DET	ATTRIBUTE NON DISP/PRT	ATTRIBUTE MDT ON
01	07	17			SBA							
					IC							
	06	08			SBA							
	10	04			EUA							
					SF	AH	✓		✓			
					•							
					•							
					•							

Figure 38. Erasing Multiple Fields with EUA

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 fields (except 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 39.

The operator keys JOHN SMITH, presses TAB, keys BOSTN, presses TAB, keys 963981, and presses ENTER (Figure 40).

You simply send the 3270 an EAU command to unlock the keyboard. The operator then sees the same panel as in Figure 39. 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 recreate 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 41.

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.

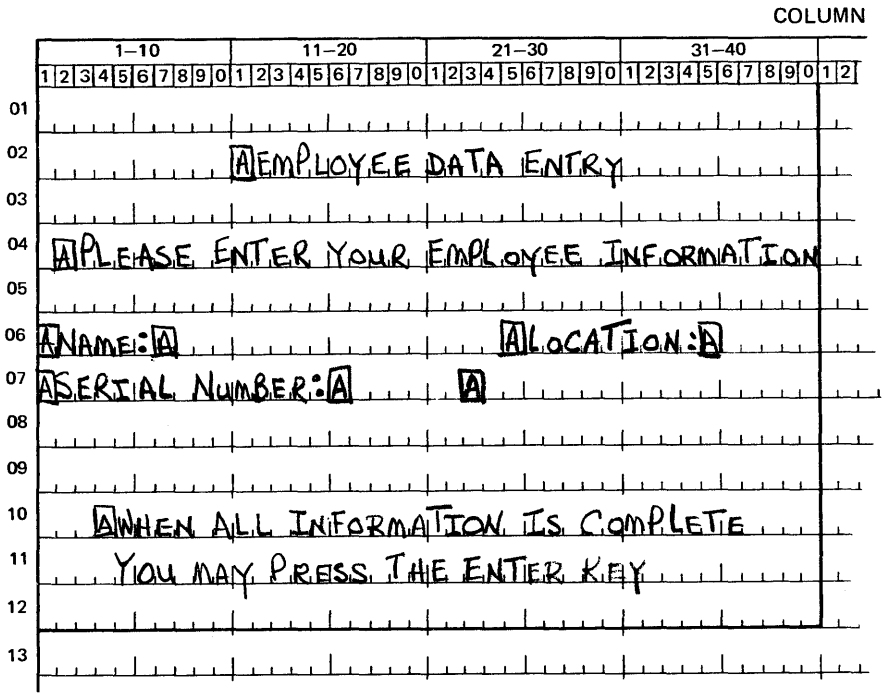


Figure 39. Example of Data Entry Panel

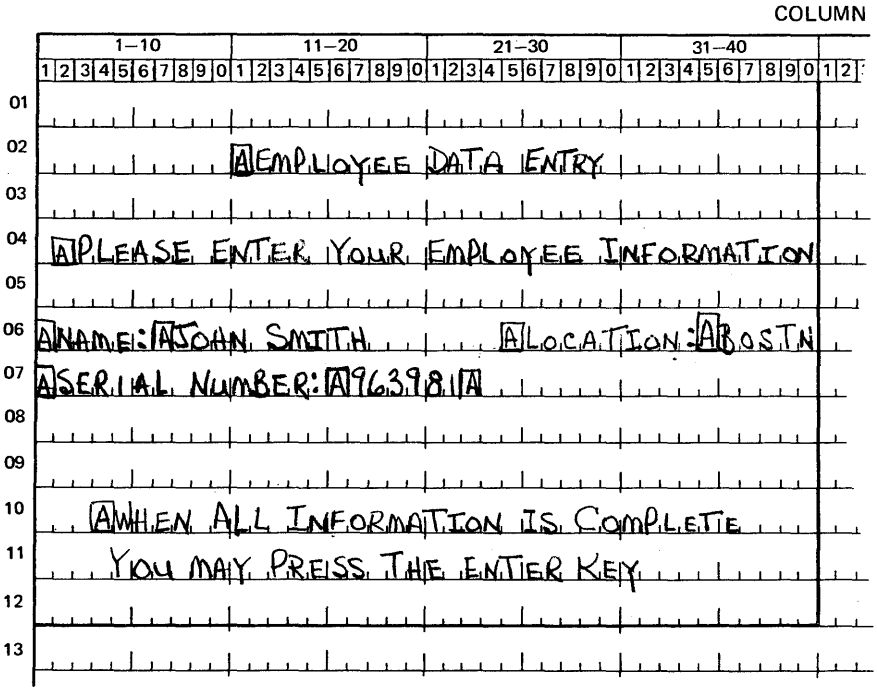


Figure 40. Data Entry Panel with Entered Data

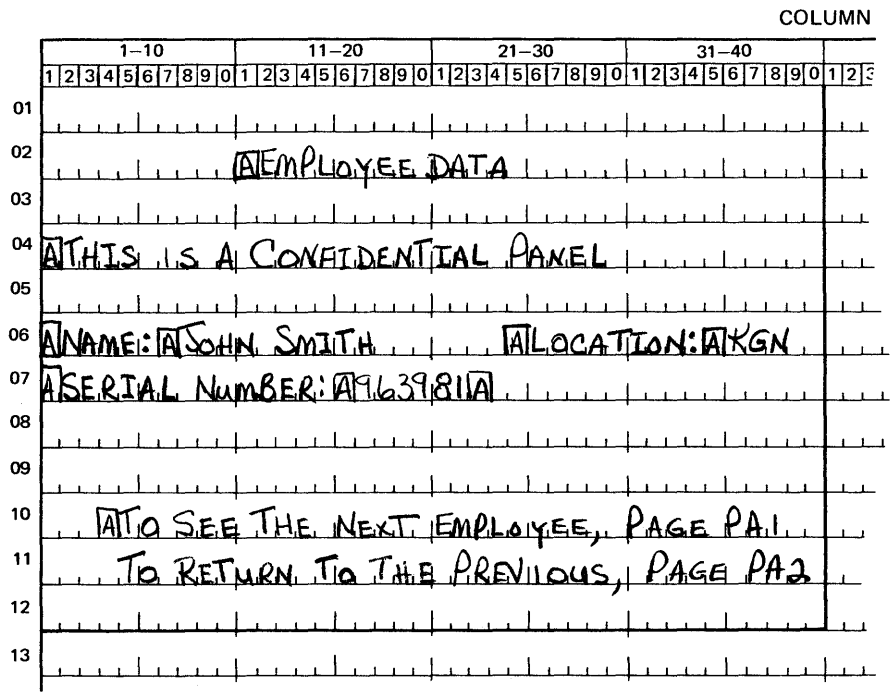
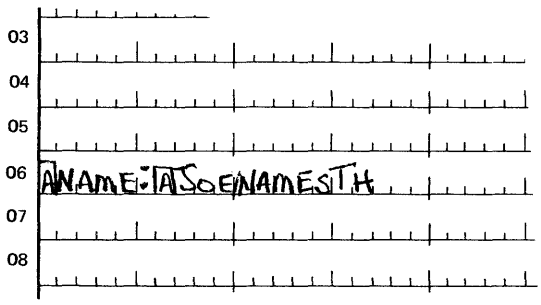


Figure 41. Employee Data Panel

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 alphameric 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 crder 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:



You must also place the location code at location R6, C36. You could use blanks after the name and an SBA sequence, or EUA with its associated address. Use PT instead. Insert a PT order after the "S" in "AMES". The single PT order clears the remainder of the unprotected name field to nulls and positions for the location code. PT should also follow the location code to position for the serial number. The data stream might look like this:

```
|W|S|   |   | | | | | | | |P| | | | | |P| | | | |
|C|B|R6|C36|J|O|E| |A|M|E|S|T|K|N|G|S|T|T|9|3|9|8|2|
|C|A|   |   | | | | | | | | | | | | | | | | | | |
```

The screen would appear as shown in Figure 42.

As you have seen, you can write each new panel out in its entirety with an Erase/Write or Erase/Write Alternate command. You should understand the relationship between the past data streams and the one you are building.

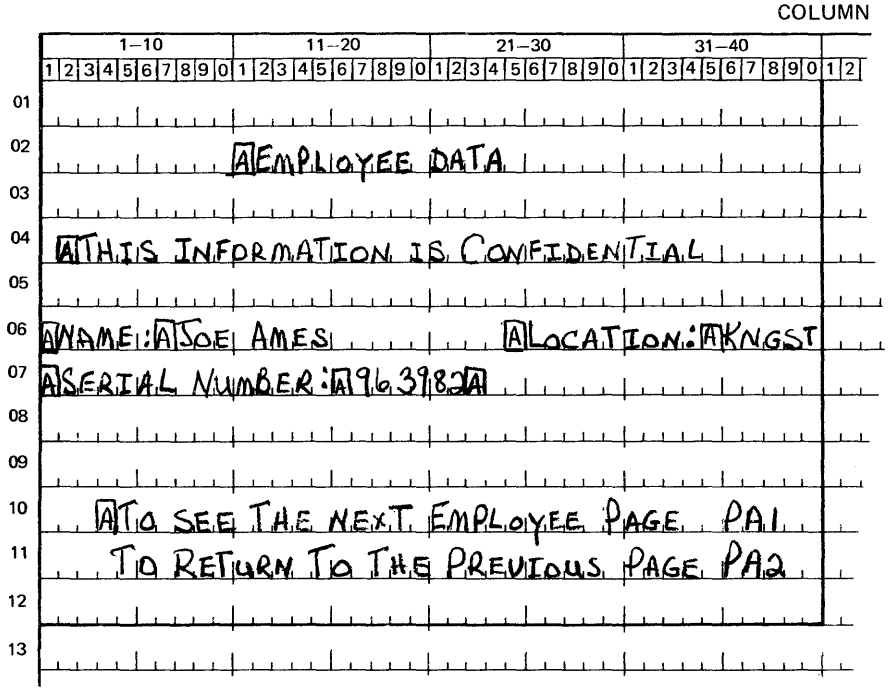


Figure 42. Panel Defined with Program Tab

CHAPTER 2: 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 43 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 and the second buffer position contains a null character, 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.)

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.

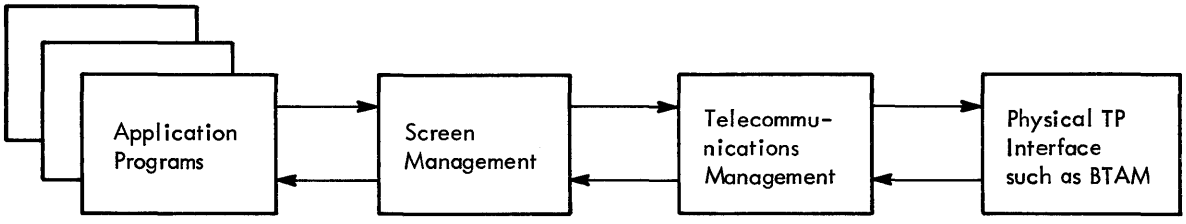


Figure 43. Relationship of Screen Management to Telecommunications Management and Application Programs

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 Section 3 make the local and remote 3270 transparent to the application program. Therefore, discussion of data streams in this section 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:

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(1B), DATA FIELD(1),
SBA(2), ADD(2A), ADD(2B), 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 SBA(2) - Address of SBA (1)) -3.

- The two-character 3270 screen address of data field(1) can be found at the address of SBA(1) +1.
- The length of data field(2) = (Address of SBA(3) - Address of SBA (2)) -3.
- The two-character screen address of data field(2) can be found at the address of SBA(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 SBA(1) and that R4 and R5 are work registers. R5 will contain the result at the end of the routine.

```
ADDCNVRT   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 BITS EXCEPT LAST SIX
           IC   R5, 1(R3)       GET SECOND ADDRESS CHAR (ADD (1B))
           N    R5, = F'63'     TURN OFF ALL BITS 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). 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-lengths fields, variable-length alphameric 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 acccmmodate 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 directicns 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-justificatien 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 44 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 44:

```
1140D4F1F2F31140E8818283848511C1C6E385A7A3
```

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

```
C1C2C3C4C54040404040F0F0F1F2F3E385A7A34040
```

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

```
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 44. Table of Requirements

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 shcwn in Figure 44 might be:

```
MAP     NAME=TABLE,MODEL=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 nctation 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.

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 cf 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 cf 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 spaces 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:

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 45, 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
```

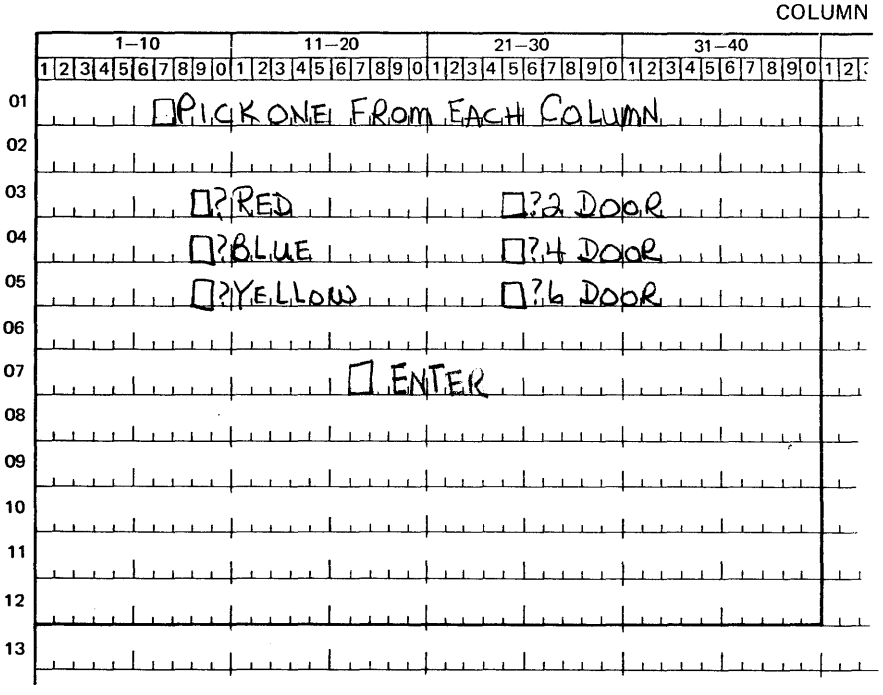


Figure 45. Example of Selector Pen Panel

```
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 46. Sample Mapping Table

Using the sample table in Figure 46, 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 44 to build the selector pen table:

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

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
```

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 '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 SBA 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,NL EM)

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 4 (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 12 could be created as follows:

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

```
┌──────────┬──────┬──────────────────────────────────────────────────┐
│          │      │          ,(attributes)                           1 │
│ [symbol] │ $MAC │ op-type  ,(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, SBA, IC, RA, EUA, WCC, and CCC.

(row,column)
   specifies the row (1 to 43) and column (1 to 80) 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: PROT, 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 14 could be as follows:

```
SIGNON      $MAC    WCC,(RESET,RESTORE),MODEL=1
            $MAC    SBA,(2,11)
            $MAC    SF,(PROT)
            DC      C'SIGN-ON PROCEDURE'
            $MAC    SBA,(4,2)
            $MAC    SF,(PROT)
            DC      C'PLEASE ENTER YOUR SIGN-ON INFORMATION'
            $MAC    SBA,(6,1)
            $MAC    SF,(PROT,HI)
            DC      C'NAME:'
            $MAC    SF
            $MAC    IC
            $MAC    SBA,(6,25)
            $MAC    SF,(PROT,HI)
            DC      C'LOCATION:'
            $MAC    SF
            $MAC    SBA,(7,1)
            $MAC    SF,(PROT,HI)
            DC      C'SERIAL NUMBER:'
            $MAC    SF,(NUM)
            $MAC    SBA,(7,23)
            $MAC    SF(PRCT)
            $MAC    SBA,(10,4)
            $MAC    SF,(PROT)
            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 character, 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)X40)+(C-1)
Model 2, 3,4 Buffer:    (((R-1)X80)+(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    R3,0
        SR      R4,R4
        IC      R4,ROW
        BCTR    R4,0
        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      R3,0(R4)
        STC     R3,ANSWER+1
        SRL     R3,6
        STC     R3,ANSWER
        NI      ANSWER+1,X'3F'
        TR      ANSWER(2),TAB
        .
        .
        .
        .
ANSWER  DC      X'0000'
TAB     DC      X'40C1C2C3C4C5C6C7C8C94A4B'
        DC      X'4C4D4E4F50D1D2D3D4D5D6D7'
        DC      X'D8D95A5B5C5D5E5F6061E2E3'
        DC      X'E4E5E6E7E8E96A6B6C6D6E6F'
        DC      X'F0F1F2F3F4F5F6F7F8F97A'
        DC      X'7B7C7D7E7F'
```

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, and 3440 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, or 3278 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 the 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 LOCAL 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.

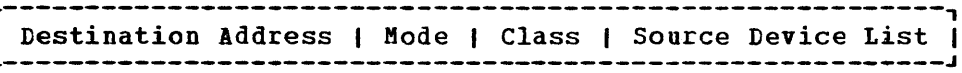
A printer in local mode is used solely for operator-initiated local copy functions. This means that displays within the cluster can contend for use of printers but the host cannot. The printer is not available for direct print operations from the host. A printer in local mode using BSC also is unavailable 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 DSC mode the start-print bit in the WCC of 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 the "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 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 opeator 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	Prot Unprot	Alpha Numer	MDT	High Inten	Sel Pen	Non Disp
10 0000	p					
00 0001	u		yes			
01 0100	u	n			yes	
10 0000	p					

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	Cols 2, 3	Col 4	Cols 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 fcllows, as a 1-character field:

Mode	Character	Hex (EBCDIC)
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.

Display Column	Class
5	70
6	71
7	72
8	73
9	74
10	75
11	76
12	77
13	78
14	79
15	80
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.

Display Column	Device Address
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'00F0', 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 23 4567	Prot Unprot	Alpha Numer	MDT	High Inten	Sel Pen Detect	Non Disp
10 0000	p					
00 0101	u		y		y	
01 0101	u	n	y		y	
00 0100	u				y	

Local Copy 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 Local 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 in 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.

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 3. BTAM SUPPORT

This chapter, a supplement to the 3270 information in the BTAM manuals, describes some aspects of the support BTAM provides. (Chapters 4 and 5 describe TCAM and VTAM support.) It discusses how you might manage devices and keep track of device status, and how BTAM analyzes 3270 operations and performs error recovery. Before writing any application programs that use BTAM macro instructions, however, refer to the BTAM publications and the other 3270 publications listed in the preface to this book for a more complete description of BTAM support for the 3270 display system.

TELECOMMUNICATIONS MANAGEMENT WITH BTAM

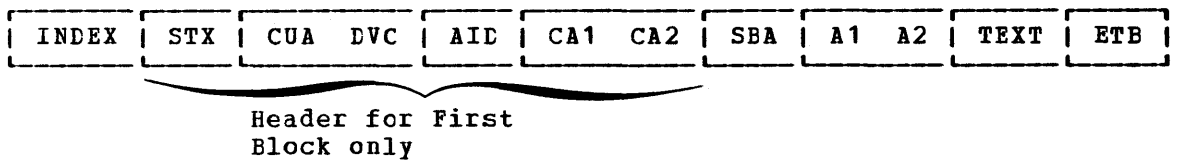
BTAM provides support for the 3270 under both DOS and DOS/VS and OS, OS/VS1, and OS/VS2. BTAM supports local 3272 and 3274-1B and remote BSC 3271, 3274-1C, 3275, 3276. This support includes generating channel programs for the 3270, starting and supplementing I/O operations, handling attentions and line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

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

- 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.

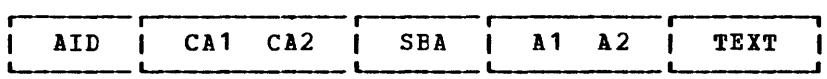
The following examples show how an application program using BTAM macro instructions can be used to concatenate and standardize a data stream.

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



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.

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



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 DECB 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.

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

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 or 3278 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 CHGNTRY 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 47 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.

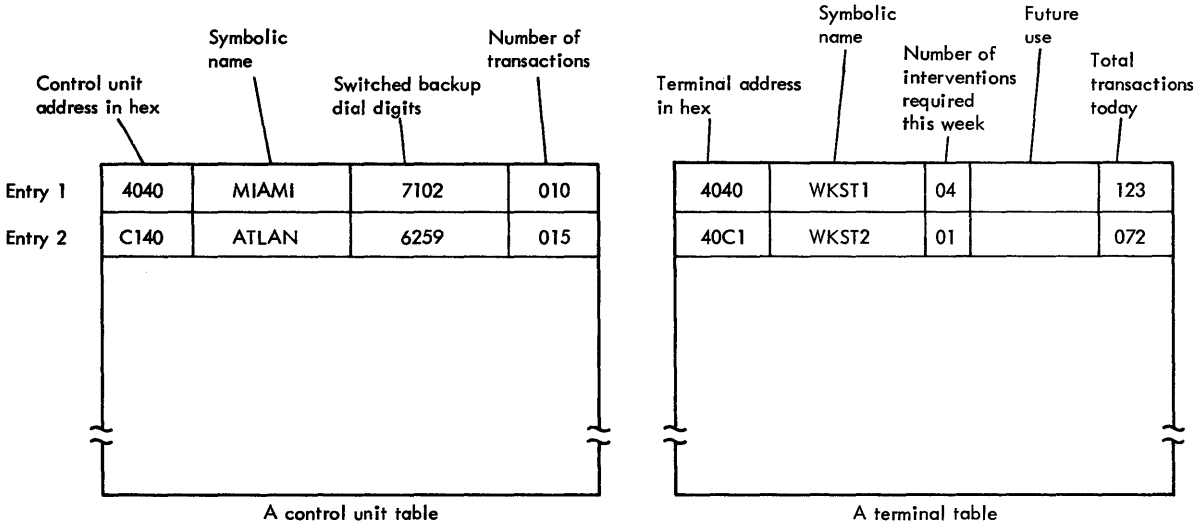


Figure 47. Table of Control Unit and Terminal Information

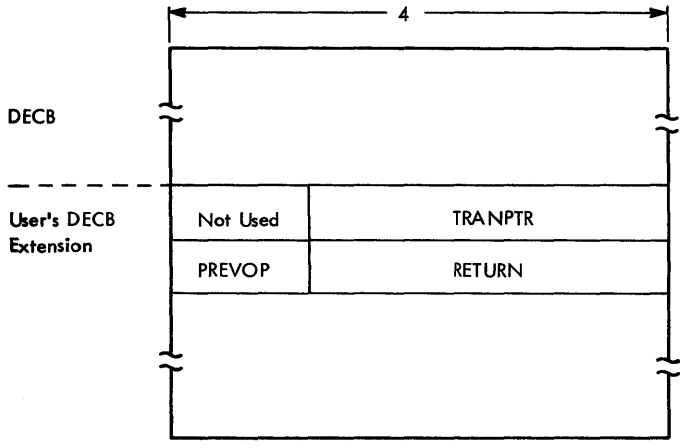


Figure 48. Example of a User-Built DECB Extension

One way to associate a table with line activity is to build your own extension to the DECB. Figure 48 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, DOS/VS coding technique for saving a return address with a DECB extension:

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

RETURN, shown in Figure 48, is a three-byte area in the DECB extension.

RELIABILITY AND ERROR RECOVERY

REMOTE LEASED LINE EVENT COMPLETION ANALYSIS

On completion of a 3270 I/O 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 BTAM.

The BSC 3270 remote leased line completion analysis is organized in six parts. Four of the parts are the flowcharts in Figures 49 through 52, 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 BTAM 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 IBM 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.

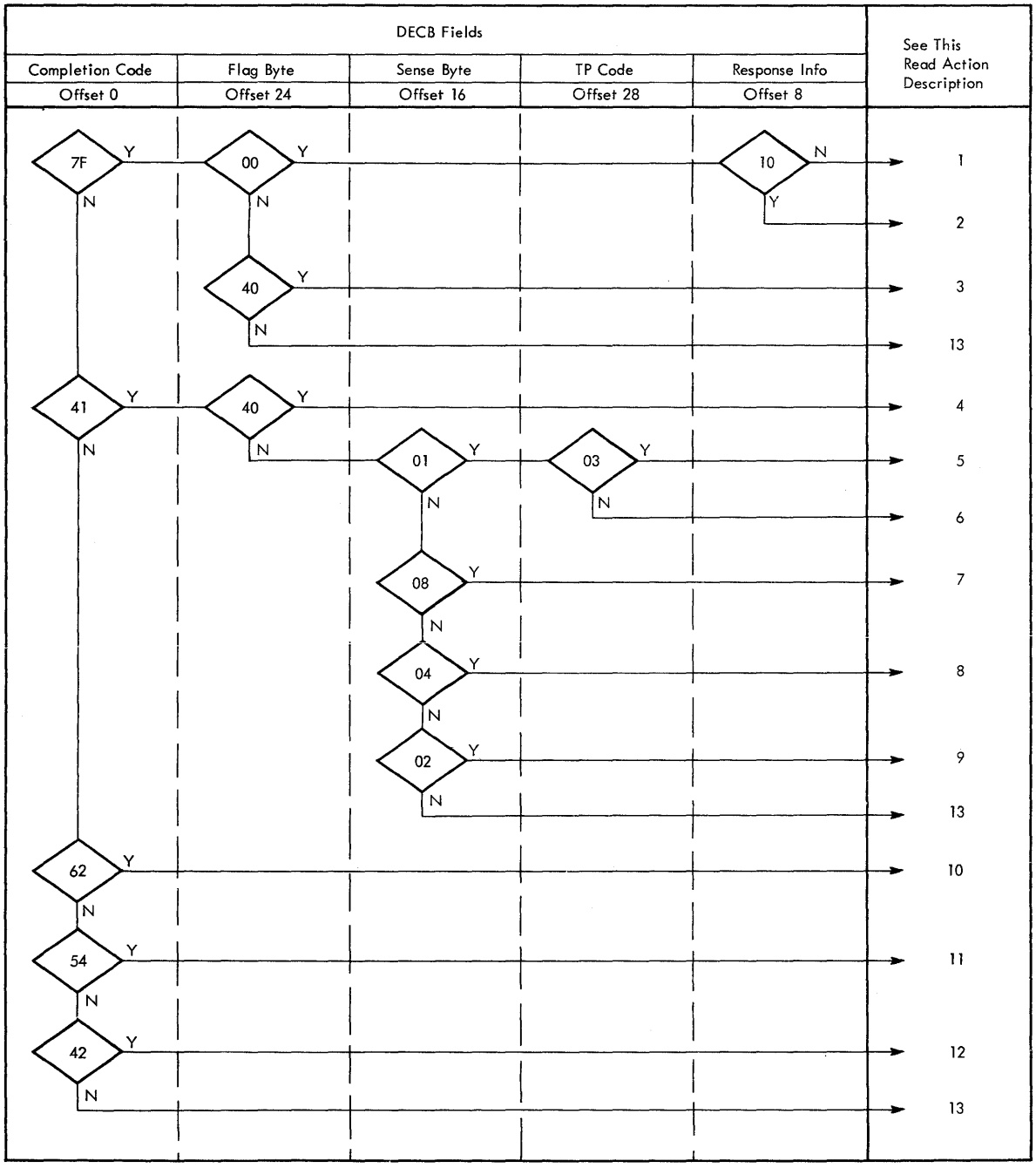


Figure 49. DOS, DOS/VS, ETAM Remote Nonswitched Line Read Completion Analysis

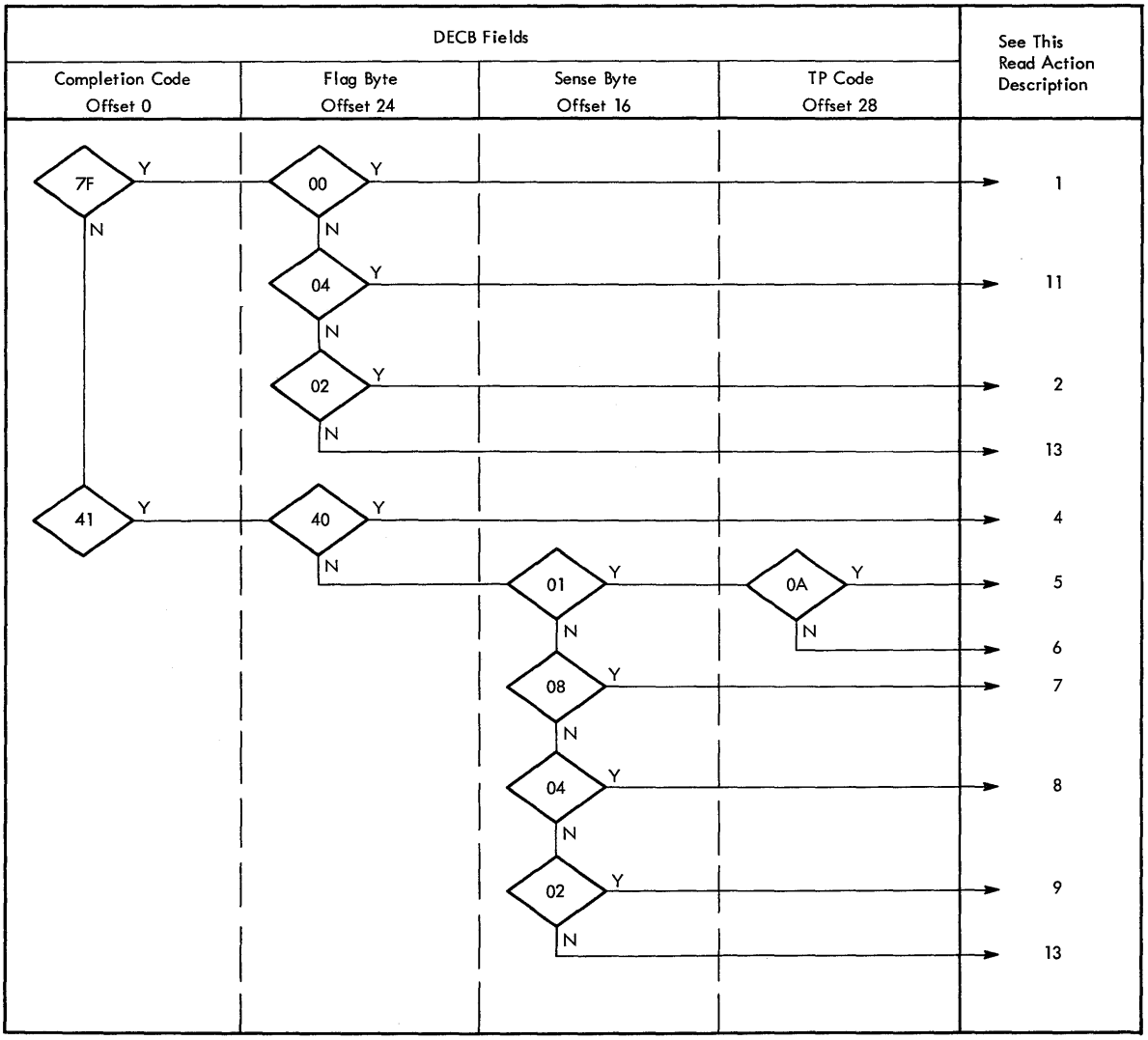


Figure 50. OS, OS/VS, BTAM Remote Nonswitched Line Read Completion Analysis

Read Action Description One

BTAM 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:

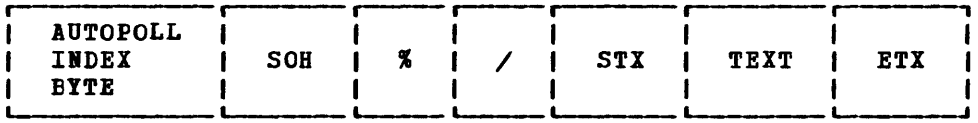
AUTOPOLL INDEX BYTE	STX	CU ADDR	DVC ADDR	AID	CURSOR ADDRESS	FIELD DATA	ETB / ETX

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, because the 3270 does not break an SBA sequence. 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 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.

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



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

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

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 and DOS/VS).

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 perfcrmed 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 OS 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.

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.

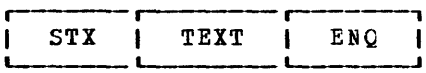
- A non-held line system does not hold the line open after receipt of a message. In this type of system the line control program might check and issue a Write Initial if output is available. If there is no output for the line, polling might be initiated with the Read Initial (TI) macro instruction.

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

Read Action Description Four

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

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 characters 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

BTAM 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 functicning.
- 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). BTAM 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 BTAM 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

BTAM 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 BTAM error recovery provides.
- If the problem is not corrected issue a Write Reset (TR) to reset the line with an EOT. Remove the control unit from the polling list as in Read Action Description Five.

Read Action Description Seven

BTAM 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

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

Explanation: An overrun condition has occurred. The I/O 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, take 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 DECB posted complete when the TWAIT is satisfied, and (R2) is loaded with the address of the DECB 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 PDUMP of the system and analyze it.
- Notify the operator of the condition.
- Issue a Read Initial (TI) to reset the line and resume polling.

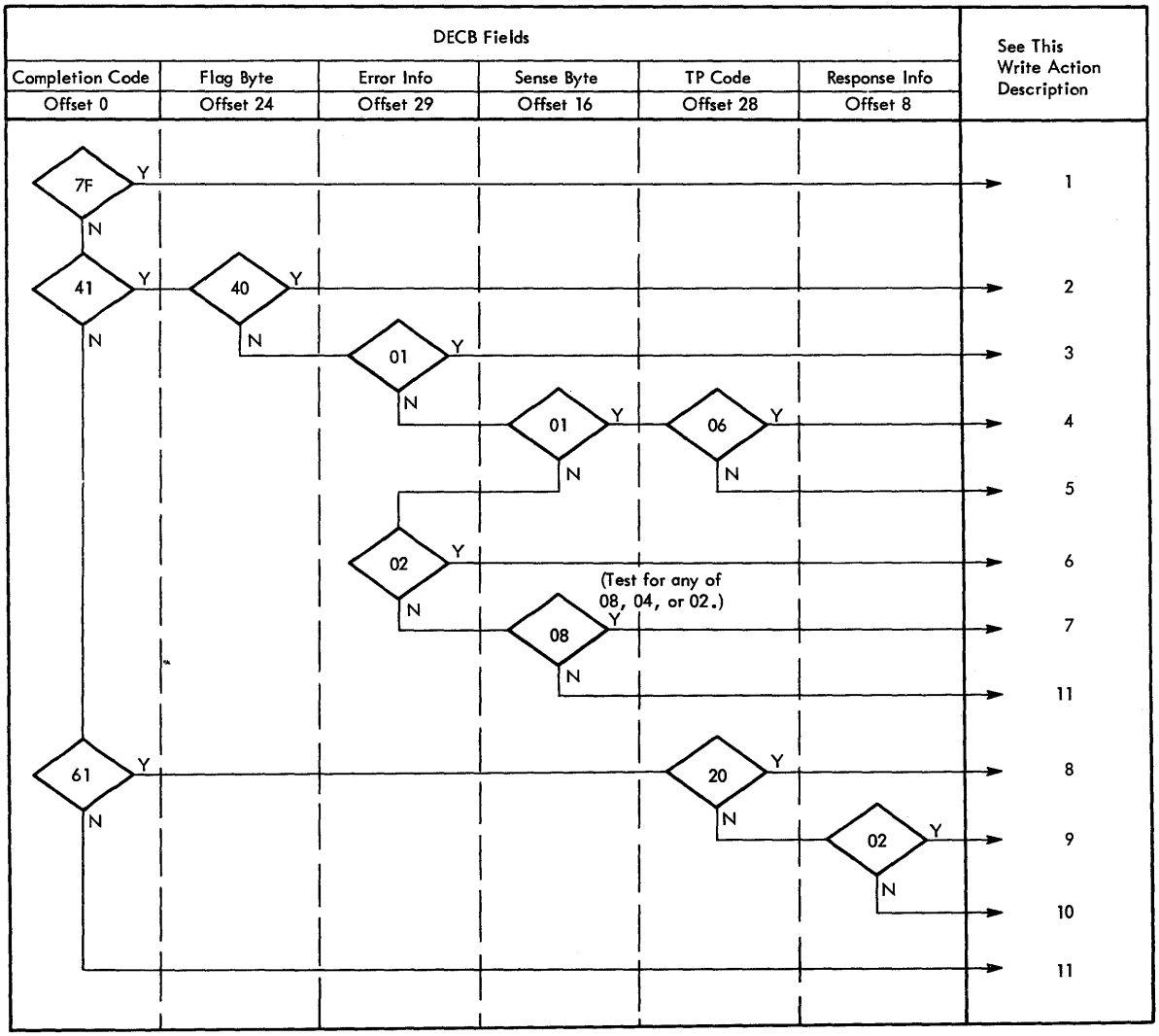


Figure 51. DOS, DOS/VS, ETAM Remote Nonswitched Line Write Completion Analysis

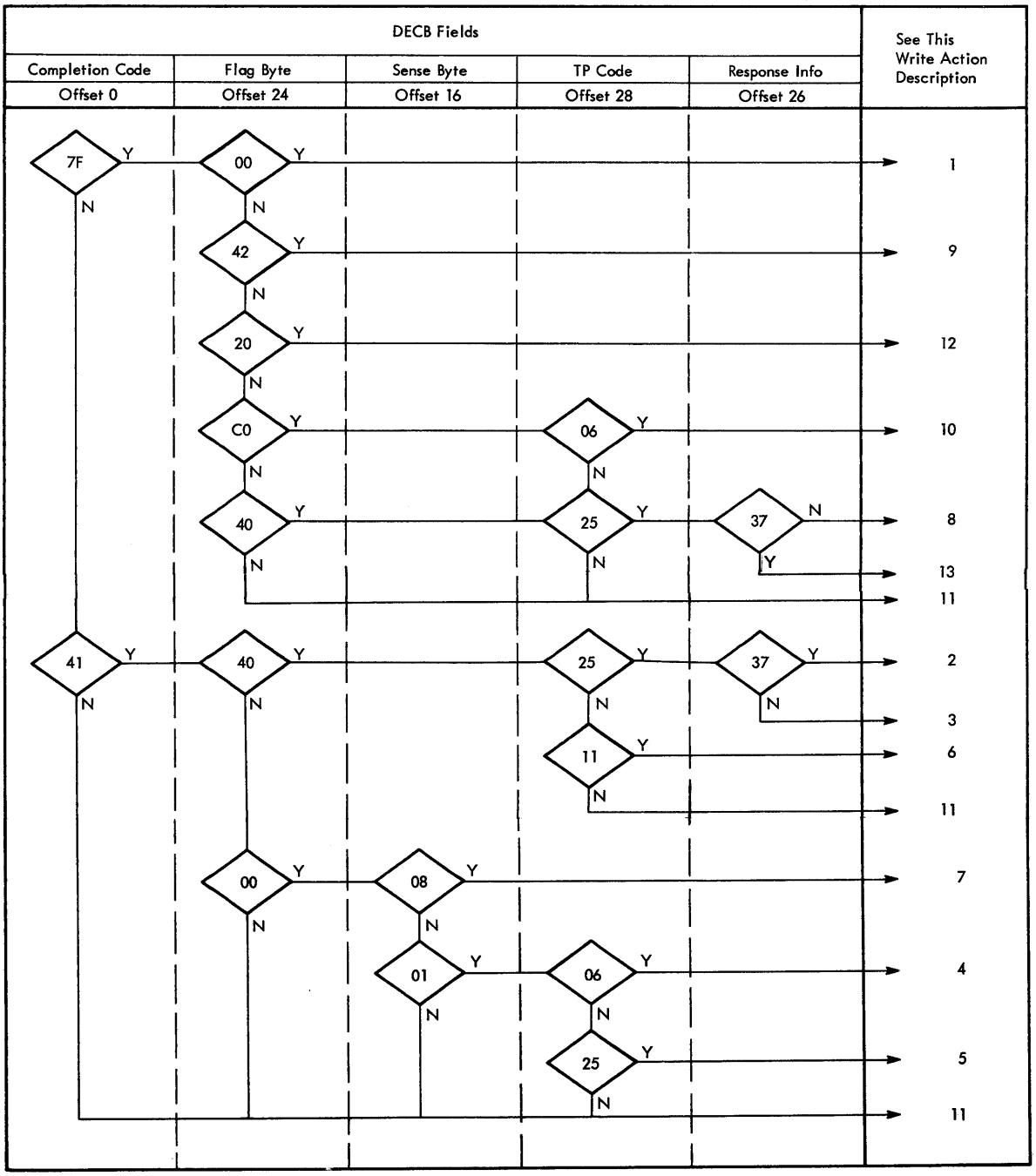


Figure 52. OS, OS/VS, BTAM Remote Nonswitched Line Write Completion Analysis

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.

- If the previous operation was Write Initial Conversational (TIV), Write Continue Conversational (TTV) or Write Conversational (TV), issue Read Continue (TT) tc 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 BCC. A 3275 sends an EOT and indicates an invalid BCC 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.

ETAM 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 section.

Write Action Description Five

ETAM 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).
- 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 permits, take the terminal out of service and proceed as in Write Action Description Four.

Write Action Description Six

ETAM 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

ETAM 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

ETAM 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

ETAM 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

ETAM 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

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

Explanation: This is an 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 validated 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 53 through 56. 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.

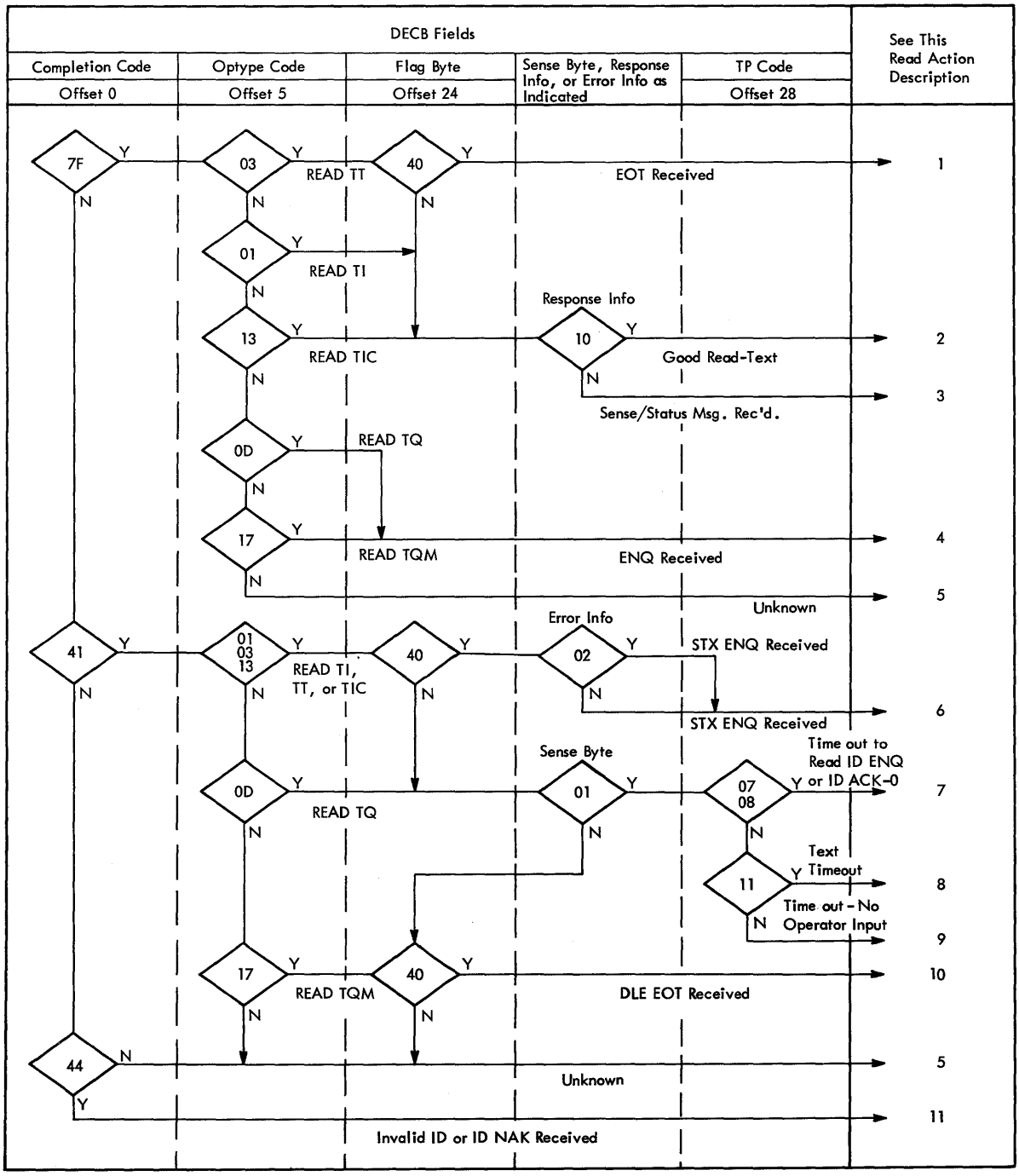


Figure 53. DOS, DOS/VS, BTAM Remote Dial Read Completion Analysis

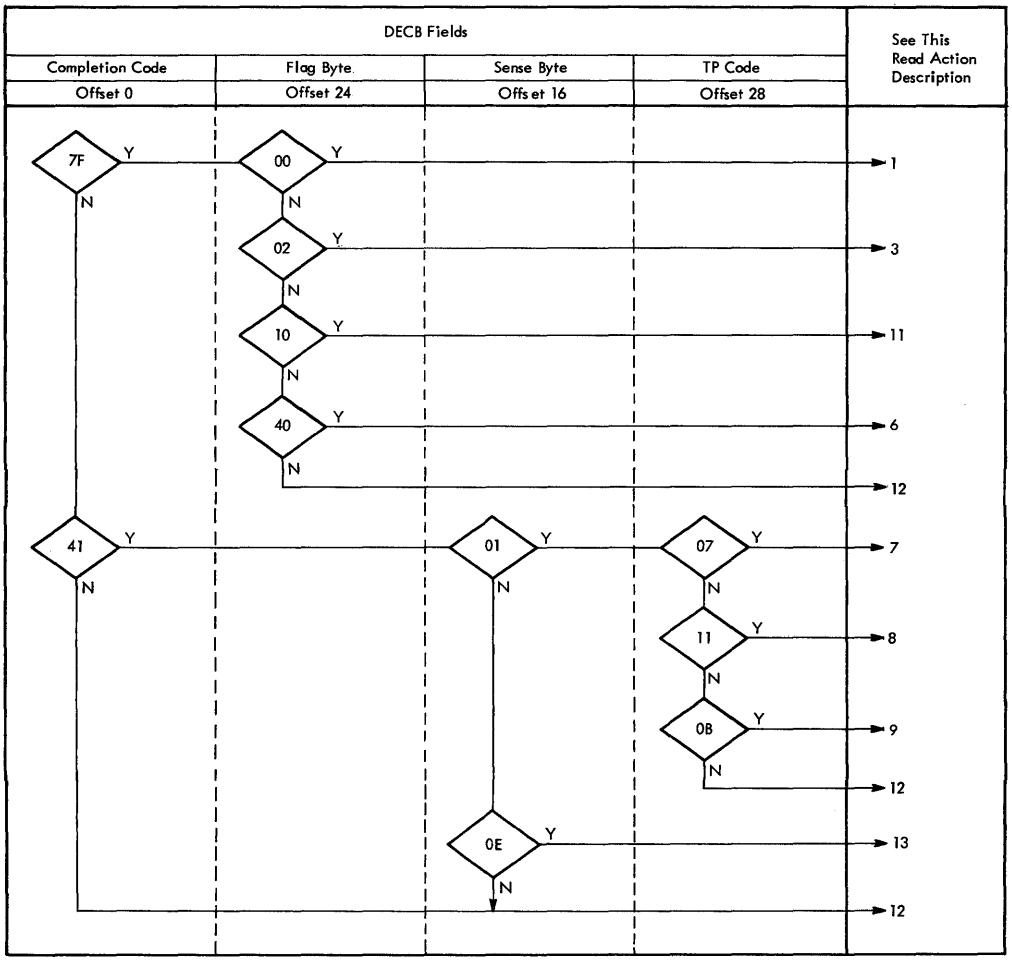


Figure 54. OS, OS/VS, BTAM Remote Dial Read Completion Analysis

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 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.

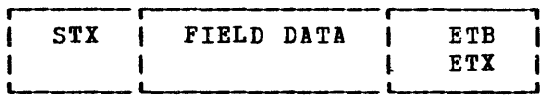
If an EOT was not received, then a text block has been successfully received from the 3275 in response to a Read Initial, Read Continue, or Read Connect instruction.

The input may have the following format:

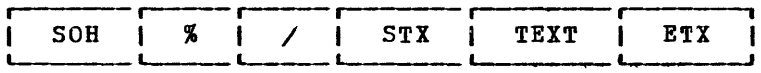
If the operator initiated the message other than with the Test Request key, the first block has the above general format. The maximum block length is 256 characters even if there are subsequent blocks because the 3275 does not break an SBA sequence (3 bytes represent a buffer address).

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.



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.



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

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 section.

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:

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 SBA 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.

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.

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

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 section.

Read Action Description Three

Explanation: A 3275 sense/status message has been received (see "Sense/Status Analysis" in this section), 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 section.

The message has the format:

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 instrucion 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.

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.

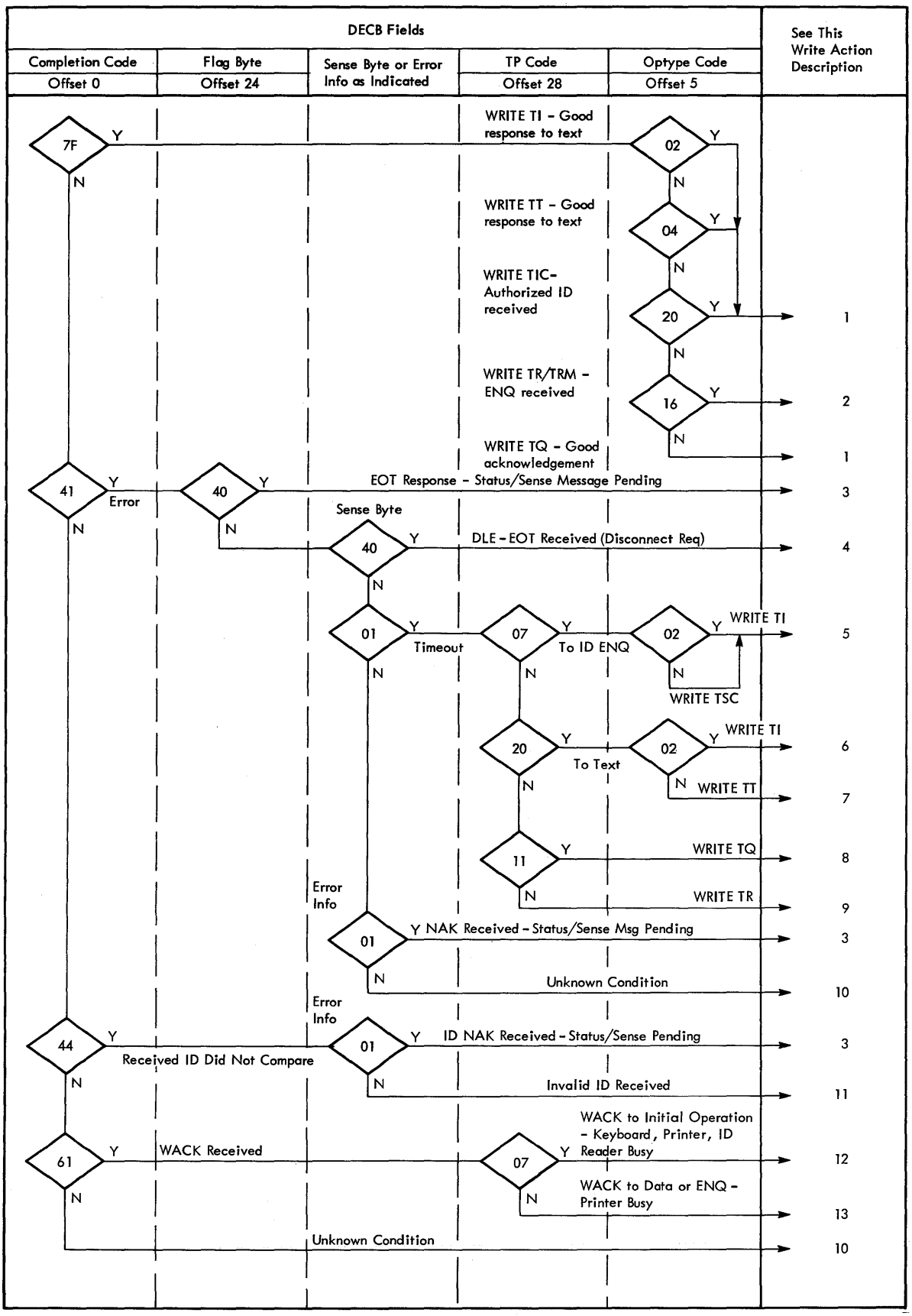


Figure 55. DOS, DOS/VS, BTAM Remote Dial Write Completion Analysis

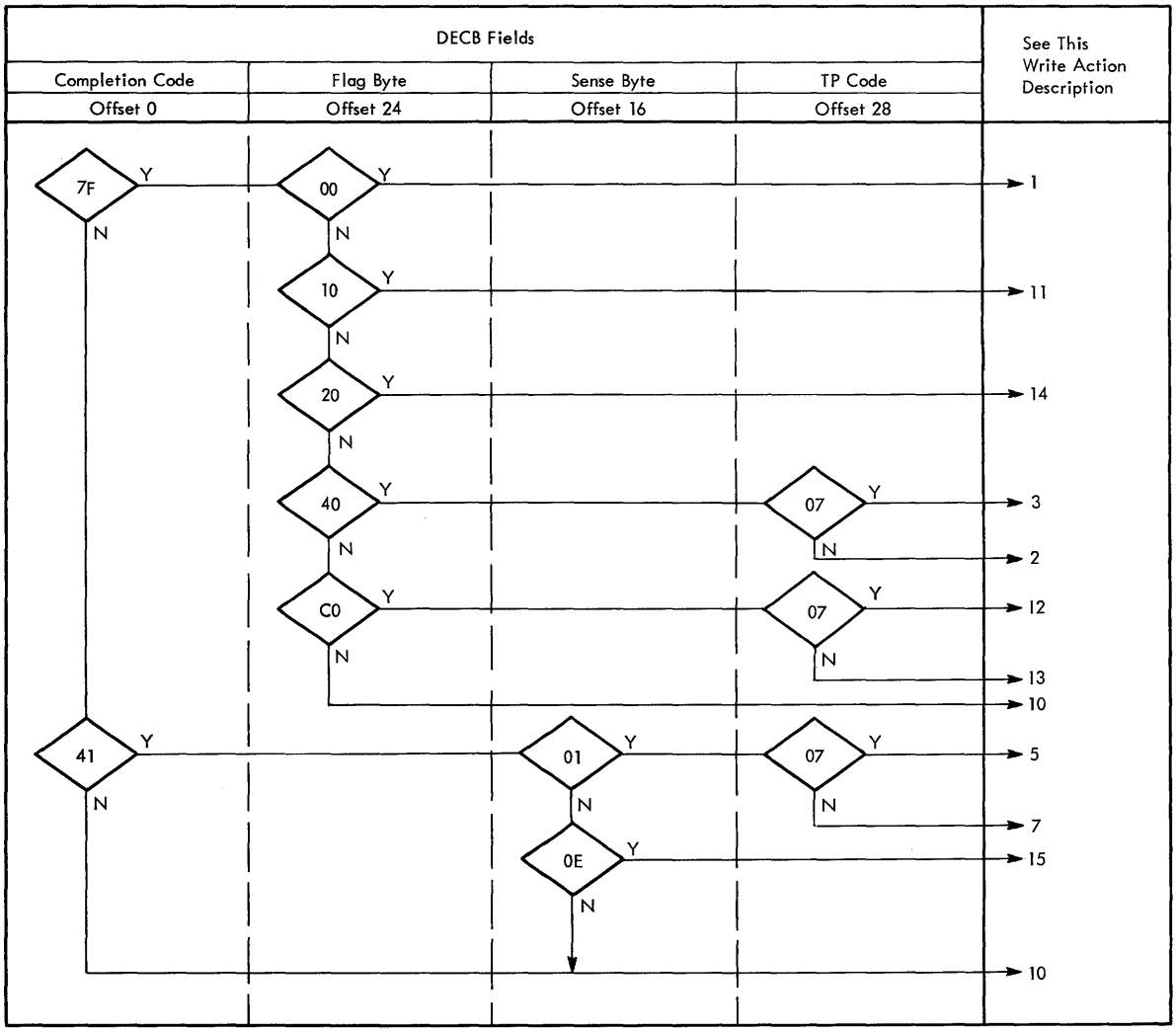


Figure 56. OS, OS/VS, BTAM Remote Dial Write Completion Analysis

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, 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.

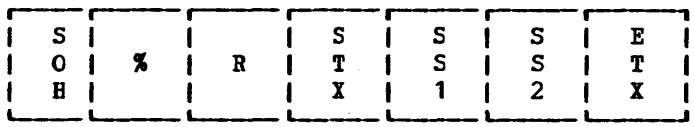
Write Action Description Three

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 EOT 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:



You should now analyze the two SS bytes, as described in the section "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 conneciton 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 BTMOD macro instruction DECBEXT parameter. If BTMOD specifies DECBEXT=YES, you must use the DECB extension technique; similarly, if you use the DECB extension technique, BTMOD must specify DECBEXT=YES. The BTMOD 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 DTFBT. 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 completed 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 acknowledgment (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 acknowledgments, 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 or 3274-1B 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 57 through 60. These flowcharts are a logical sequence in which completion information can be analyzed after a Read or Write or Erase/Write Alternate for the 3274-1B operation. The flowcharts refer to the action descriptions that immediately follow the flowcharts. The action descriptions are in the following format:

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

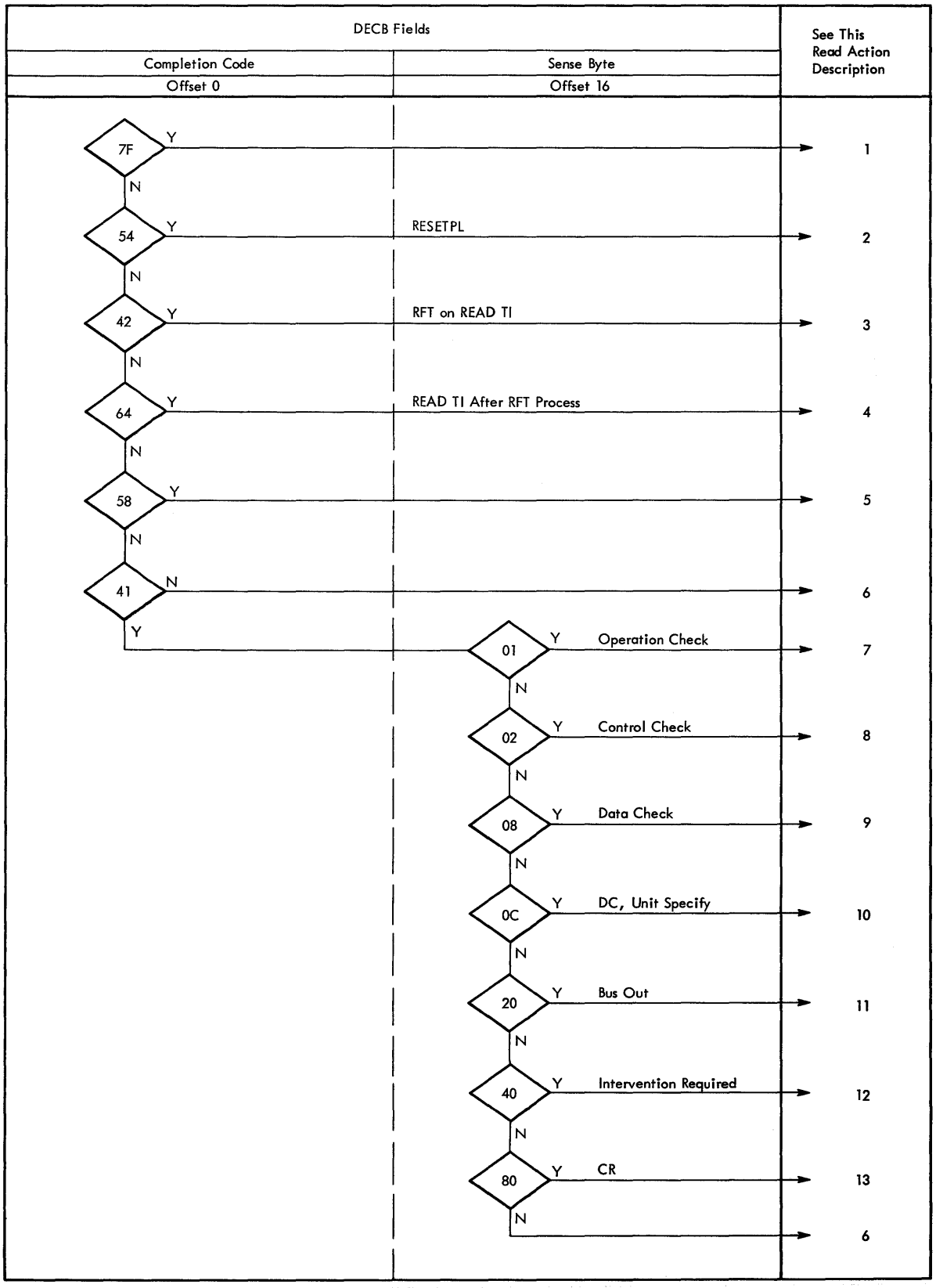


Figure 57. DOS, DOS/VS, BTAM Local Read Completion Analysis

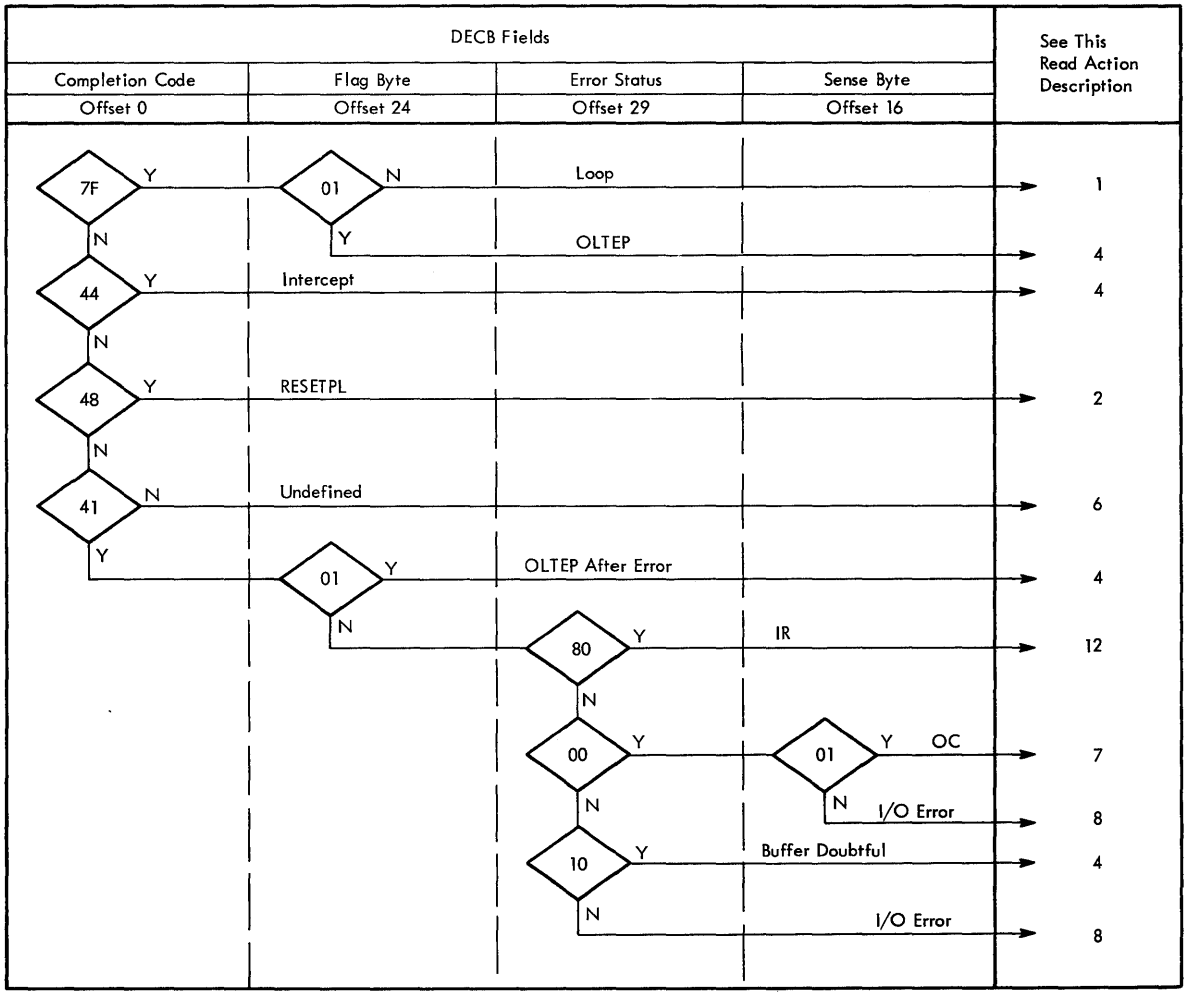


Figure 58. OS, OS/VS, BTAM Local Read Completion Analysis

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

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), or Erase/Write

Alternate (TSA), 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 DECB, which is posted as complete on the satisfaction of the TWAIT; (R2) is loaded with the address of the DECB, 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: BTAM 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 or 3274 Model 1B 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 or 3274 Model 1B 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 or 3274 Model 1B has detected a data check.

Action: Same as Read Action Description Eight.

Read Action Description Ten

Explanation: A printer cr 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/O operations.
- It may be preferable to issue an 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 or 3274 Model 1B 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 CHGNTRY macro instruction.
- Reissue the failing macro instruction if it was Read Initial (TI).

Read Action Description Thirteen

Explanation: The 3272 or 3274 Model 1B has detected an invalid command.

Action: Same as Read Action Description Seven.

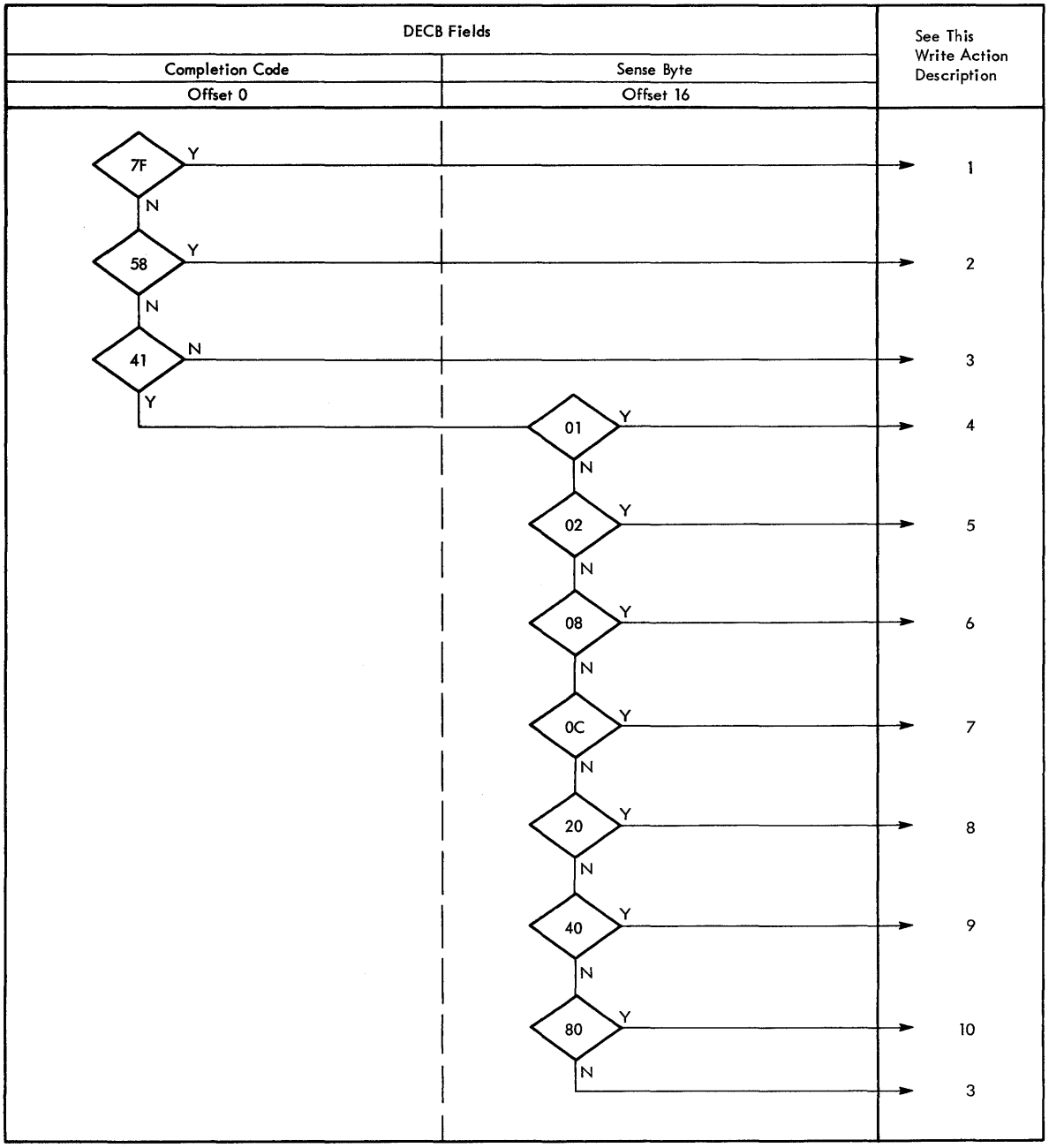


Figure 59. DOS, DOS/VS, BTAM Local Write Completion Analysis

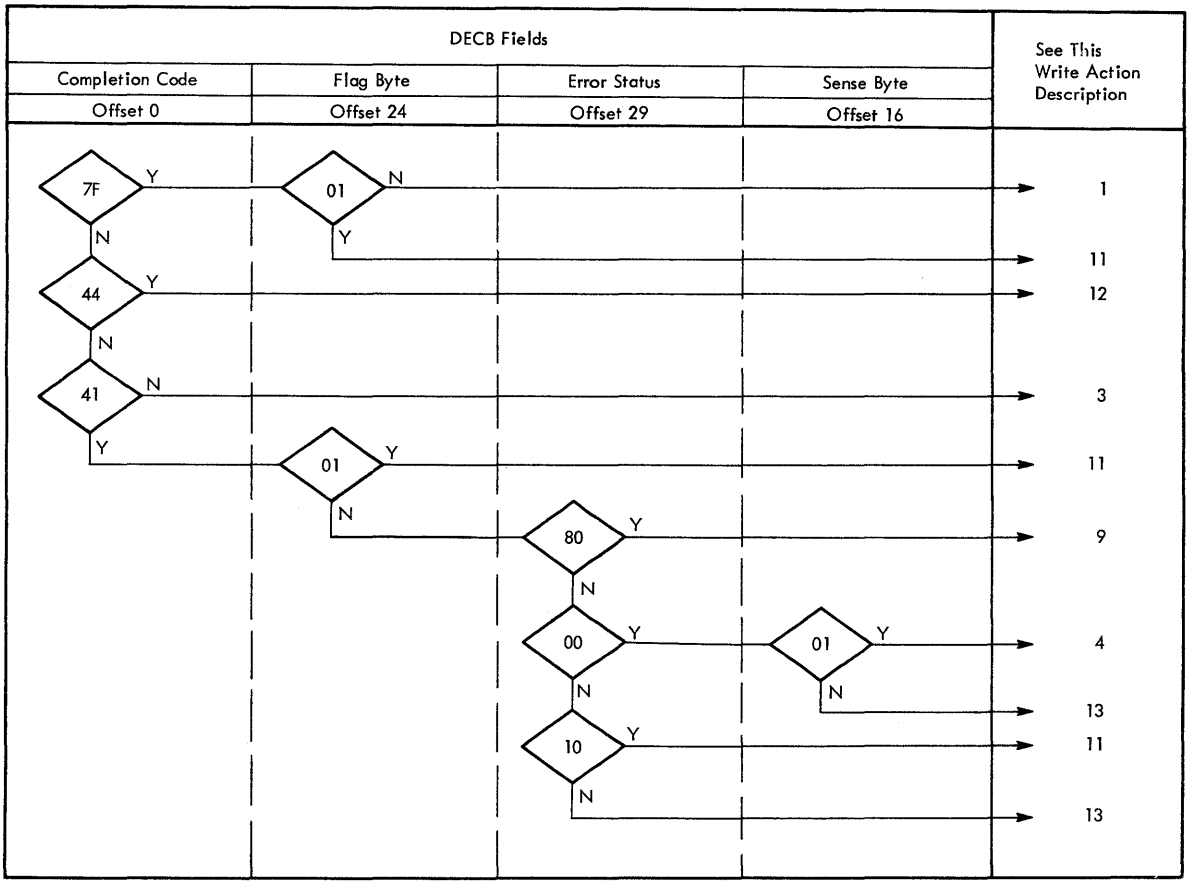


Figure 60. OS, OS/VS, BTAM Local Write Completion Analysis

Write Action Description One

Explanation: The Write operation has been posted complete without error. For a DOS or DOS/VS printer this only signals that the print operation has begun. Print errors that 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 (TS) 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 program error.

Action: Same as Read Action Description Six.

Write Action Description Four

Explanation: The 3272 or 3274 Model 1B 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 or 3274 Model 1B has detected a control check. The device failed to perform an operation or respond to the 3272 or 3274 in the amount of time allowed by the control unit.

Action: Same as Read Action Description Eight.

Write Action Description Six

Explanation: The 3272 or 3274 Model 1B 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 or 3274 Model 1B 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 or 3274 Model 1B has detected an invalid command.

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 cccurred following the previous operation or request-for-test processing.

Action: See Read Action Description Four.

Write Action Description Thirteen

Explanation: A permanent I/O 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, ccnditions 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.

CHAPTER 4. 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, OS/VS1, and OS/VS2, and supports the 3270 display system in areas such as device scheduling, diagnostic testing, and error handling and recording.

This chapter only suggests TCAM operations that might be used with the 3270; it does not give a complete description of TCAM support. The following TCAM publications provide detailed information about 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.

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; read buffer is not supported by TCAM.

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.

Under TCAM, for host-attached 3270 remote devices, the 3270 line configuration is always specified as BSC3 (leased multipoint) in the IODEVICE macro instruction, even if the devices are physically attached point-to-point. The switched 3275 is not supported by TCAM.

TERMINAL Macro Instruction

The TERMINAL macro instruction defines the cluster control unit and the devices attached to it (for BSC and locally attached terminals). It is required for 3270 control units if general polling is specified. Macro specifications differ for NCP and non-NCP attached 3270 devices. An example of NCP attachment is shown in the coding sample below. The format shown in Figure 61 for TERMINAL includes only those operands that apply to the 3270.

```
*Specification for an NCP Line Group for SDLC Lines

 SNA3270   GROUP        MH=SNAMH,TRANS=EBCF,BUFOUT=2
                        BUFMAX=5,BUFSIZE=120,PCI=(,N)

*Specification for an SNA Physical Unit (PU)

 PU3270    TERMINAL     TERM=PUNT,ACTIVE=YES

*Specification for an SNA Logical Unit (LU) (SDLC 3270)

 LU3270    TERMINAL     TERM=LUNT,RLN=1,GROUP=SNA3270
                        ACTIVE=(NO,YES),QBY=T,QUEUES=DR
                        SECTERM=YES,USS=3270,OPDATA=(18)
```

Note: The 3271, 3272 SDLC devices are LU.T0 and must adhere to SNA protocol for LU.T0 devices. The 3274, 3276 SNA devices are LU.T2 devices and must use SNA protocol for LU.T2.

The following restrictions apply to the TERMINAL macro instruction:

- LMD and MBR are not supported.
- The SCRSIZE operand on the TERMINAL macro has significance only for TSO 3270 terminals and non-SNA 3270 terminals which may enter into CROSS-DOMAIN sessions. The specifications for SCRSIZE provide support for the physically variable-sized screens available with the 3270. Instead of one (row, column) pair as in earlier releases of TCAM, there now may be two pairs for default and alternate sizes, respectively. The sample coding below shows the TERMINAL macro definition for a 3270 device capable of two physical screen sizes -- (24, 80) and (32, 80).

```
 Name      Operation   Operand

 TSO SNALU TERMINAL    GROUP=SNA3270, QBY=T, RLN=2,
                       TERM=LUNT, QUEUES=TS, UTERM=NO,
                       OPDATA=(81), ACTIVE=(YES,YES),
                       USS=SCS, FEATURE=(NOBREAK),
                       SCRSIZE=(24, 80, 32, 80)
```

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

INVLIST Macro Instruction

The INVLIST macro instruction generates the invitation list for a line and type of read operation for 3270 local. Each INVLIST macro defines the polling characters for one line, regardless of how many control units are on the line. TCAM requires a specific poll entry for every remote 3270 printer and display, whether polled or not. TCAM requires that every 3270 device be included in the invitation list for the line that the cluster is on. Figure 61 shows sample coding for a remote 3270 cluster, and Figure 62 shows the same for an invitation list for

a general poll. If general polling is not desired, remove the first entry in the invitation list (also remove the TERMINAL macro for the control unit).

The SNA 3270 must attach to TCAM via the NETWORK CONTROL PROGRAM (NCP) in the 3705, and the BSC 3270 may also. Resources attached to TCAM via the NCP require no invitation list; therefore, only non-NCP attached 3270 devices require an invitation list.

```
Name      Operation   Operand
-------------------------------------------------------------------
                      T
symbol    TERMINAL    QBY=  L  ,DCB=dcbname,RLN=integer

                      ,TERM=type,QUEUES=form[,ADDR=chars]

                      [,LEVEL=(integer,...)][,BUFSIZE=integer]

                      [,ALTDEST=entry][,OPDATA=(data,...)]

                      [,NTBLKSZ=(blocksize,subblocksize)]

                                          {YES}
                      [,RETRY=integer][,LMD={NO }]

                            YES           {YES}
                      [,MB=  NO  ][,SECTERM={NO }]
-------------------------------------------------------------------
```

Figure 61. TERMINAL Macro for the 3270

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

R3270CU   TERMINAL    QBY=T,RLN=1,TERM=327C,DCB=R3270DCB,          *
                      QUEUES=MR,OPDATA=(08)

R3270D1   TERMINAL    QBY=T,RLN=1,TERM=327R,DCB=R3270DCB,          *
                      QUEUES=DR,OPDATA=(81),SECTERM=YES,           *
                      ADDR=6060C1C12D

R3270D2   TERMINAL    QBY=T,RLN=1,TERM=327R,DCB=R3270DCB,          *
                      QUEUES=DR,OPDATA=(82),SECTERM=YES,           *
                      ADDR=606040402D

R3270D3   TERMINAL    QBY=T,RLN=1,TERM=327R,DCB=R3270DCB,          *
                      QUEUES=DR,OPDATA=(82),SECTERM=YES,           *
                      ADDR=6060C4C42D

R3270P1   TERMINAL    QBY=T,RLN=1,TERM=327R,DCB=R3270DCB,          *
                      QUEUES=DR,ADDR=6060C3C32D

IL3270R   INVLIST     ORDER=(R3270CU+40407F7F2D,                   *
                      R3270D1-4040C1C12D,R3270D2-404040402D,       *
                      R3270D3-4040C4C42D,R3270P1-4040C3C32D),      *
                      EOT=37

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

Figure 62. Invitation List for General Polling for a Remote 3270

DCB Macro Instruction

Figure 63 shows a sample DCB macro instruction for a line group of 3270 terminals. (The DCB macro for a line group is no different from that for any other type of terminal.) Dynamic buffering is supported for the remote 3270.

For NCP-attached 3270 devices, the DCB for the 3705 is the only requirement.

```
R3270DCB  DCB     DSORG=TX,MACRF=(G,P),DDNAME=R3270DD,        *
                  CPRI=S,BUFIN=4,BUFOUT=8,BUFSIZE=156,        *
                  BUFMAX=10,MH=MH3270,INVLIST=IL3270R,        *
                  TRANS=EBCD,PCI=(A,A)
```

Figure 63. Line Group DCB Macro for a Remote 3270

Defining the Local 3270 Cluster

When INVLIST, TERMINAL, and line group DCB macros are coded for a local 3270, each 3270 control unit is considered a line group and should have a DCB and a DD card. Each device attached to a control unit is considered a line on that line group and should have its own relative line number. Each local 3270 must be represented by a TERMINAL macro.

Dynamic buffering is not supported for the local 3270. PCI=(N,N) must be used. An invitation list must be coded for every local device. Figure 64 shows sample coding for a local 3270 cluster.

If you are defining a local 3270 that uses the 3278, 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).

INTRO Macro Instruction

The INTRO macro supplies the bulk of TCAM initialization information, including the type of lines. For the remote host-attached 3270, code the LINETYPE= operand on the INTRO macro as BISC or BOTH. Specify STSP or BOTH for the local 3270. If BISC is coded for a configuration that has BSC terminals and local 3270s, the local devices will not operate.

CONTROLLING THE 3270 NETWORK

After defining the 3270 system to TCAM you can use TCAM facilities to manage and control the subsystem. The following sections discuss some aspects of message handling for the 3270.

```
DCBL3270    DCB         BUFIN=1,BUFOUT=7,BUFMAX=7,                *
                        BUFSIZE=348,CPRI=S,DSORG=TX,              *
                        DDNAME=DD3270,MACRF=(G,P),                *
                        MH=MH3270,PCI=(N,N),TRANS=EBCD,           *
                        INVLIST=(IT1...IT2...IT3...Z)

L3270D1     TERMINAL    QBY=L,DCB=DCBL3270,TERM=327L,             *
                        QUEUES=DR,RLN=1,ALTDEST=L3270D1

L3270D2     TERMINAL    QBY=L,DCB=DCB3270,TERM=327L,              *
                        QUEUES=DR,RLN=2,ALTDEST=L3270D2

L3270D3     TERMINAL    QBY=L,DCB=DCB3270,TERM=327L,              *
                        QUEUES=DR,RLN=3,ALTDEST=L3270D3

L3270P1     TERMINAL    QBY=L,DCB=DCB3270,TERM=327L,              *
                        QUEUES=DR,RLN=4,ALTDEST=3270P1

                                    ***
IT1         INVLIST     ORDER=(L3270D1+06)

IT2         INVLIST     ORDER=(L3270D2+06)

IT3         INVLIST     ORDER=(L3270D3+06)

Z           INVLIST                              Output Only Printer

*** 06 causes TCAM to perform a Read Modified operation;
    02 causes a Read Buffer operation.
```

Figure 64. Defining a 3270

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. Figure 65 shows separation of the user's application into two modules.

Input from a remote 3270 is the result of execution of a Read Modified command. TCAM does only Read Modified for remote 3270s. When a 3270 successfully receives valid polling characters, its positive response is to transmit to the CPU all fields in which modified data tag bits have been set in the attribute bytes.

Output to a 3270 may originate either in an application program or in a message handler. For example, a message created in an application program may be augmented or edited by the MSGEDIT or MSGFORM macros in the message handler. The same message processing program (MPP) may contain these two modules.

MSGFORM Macro Instruction

The MSGFORM macro instruction supplies the framing characters (STX--ETX) for 3270 data streams. While there are no unique specifications of the MSGFORM macro for 3270, the following warnings should be noted:

- Do not issue both the MSGFORM macro and code STX--ETX in the data stream.

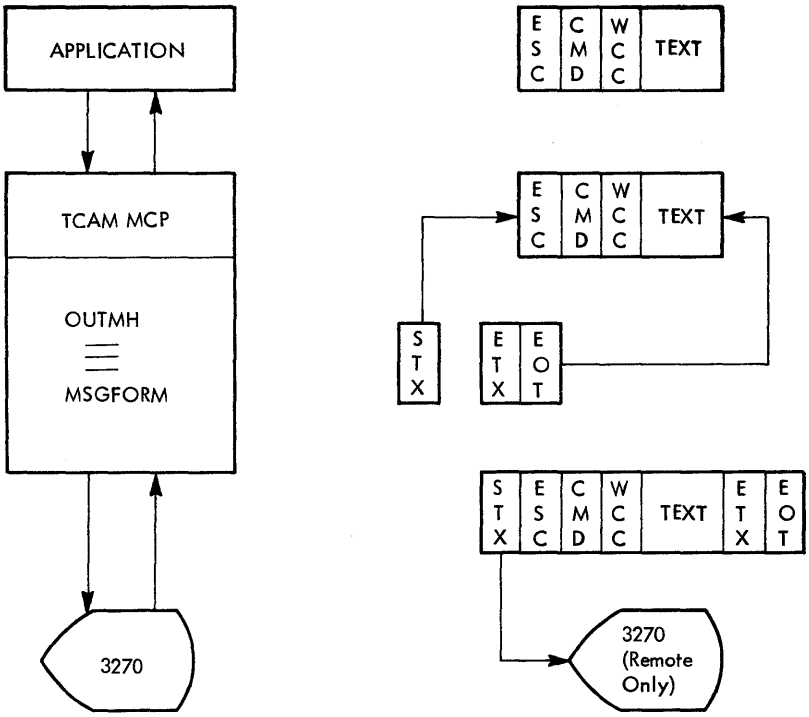


Figure 66. MSGFORM Operation

SCREEN Macro Instruction

The SCREEN macro instruction inserts the appropriate CMD character in the data stream immediately following an ESC character. The RETRV operand has no function for a remote 3270 data stream. If MSGFORM is to be used, however, it does allow you to code all 3270 data streams as if they were for a remote device. For a local 3270, RETRV specifies that TCAM search the remote data stream for the CMD byte and set the channel program for the local device. RETRV has no effect on the data stream, which must be edited before transmission in the outgoing message handler. Figure 67 shows the syntax of the SCREEN macro.

Figure 68 shows how the SCREEN macro instruction fits in an outgroup message handler program. In the example, SCREEN WRE scans the buffer and inserts the WRE command after the first ESC character. MSGFORM supplies the framing characters.

SNA 3270 devices do not allow the ESC character to be embedded in the data stream. Therefore, the ESC character must be the first data character in the message. Given this, the SCREEN macro operation is identical for SNA/SDLC, local channel, and BSC devices.

Name	Operation	Operand
[symbol]	SCREEN	{WDC | WRE | EAU | WEA} [,conchars,BLANK={char | NO | YES}] [,RETRV={YES | NO}]

Figure 67. Syntax of the SCREEN Macro Instruction

SCREEN Macro Instruction

The SCREEN macro instruction inserts the appropriate CMD character in the data stream immediately following an ESC character. The RETRV operand has no function for a remote 3270 data stream. If MSGFORM is to be used, however, it does allow you to code all 3270 data streams as if they were for a remote device. For a local 3270, RETRV specifies that TCAM search the remote data stream for the CMD byte and set the channel program for the local device. RETRV has no effect on the data stream, which must be edited before transmission in the outgoing message handler. Figure 67 shows the syntax of the SCREEN macro.

Figure 68 shows how the SCREEN macro instruction fits in an outgroup message handler program. In the example, SCREEN WRE scans the buffer and inserts the WRE command after the first ESC character. MSGFORM supplies the framing characters.

SNA 3270 devices do not allow the ESC character to be embedded in the data stream. For SNA 3270 devices the CMD is inserted as the first character in the message.

Name	Operation	Operand
[symbol]	SCREEN	{ WDC / WRE / EAU / WEA } [,conchars,BLANK= { char / NO / YES }] [,RETRV= { YES / NO }]

Figure 67. Syntax of the SCREEN Macro Instruction

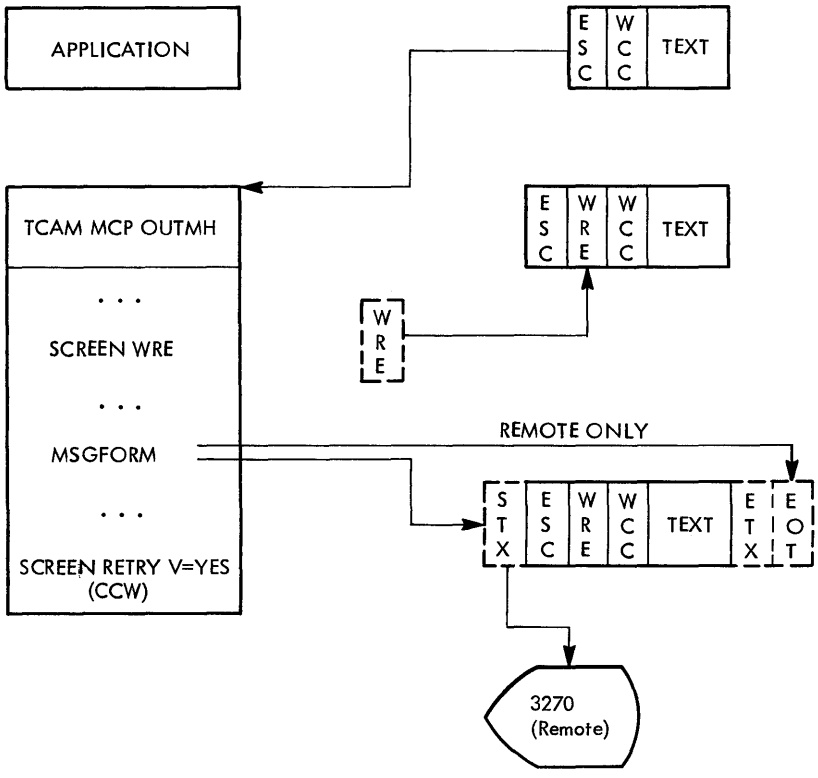


Figure 68. SCREEN in an Outgroup Message Handler Program

MSGLIMIT Macro Instruction

For non-NCP attached 3270 devices, code the MSGLIMIT macro for general or specific polling. 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, a TCAM message processor program sends a message for a remote printer to the 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 then scan the status messages for a printout error, and request that an application program perform a QRESET to resend the message.

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.

Only one application program should control the printer by holding the print queue until device end status is received.

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.

Figures 69 and 70 show examples of how a message handler or application program can test for the AID byte and take appropriate action.

HANDLING SENSE/STATUS CONDITIONS

Figure 71 shows the relation of the TCAM message control program to the sense/status handling. 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 text 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.

```
MH3270      STARTMH     LC=OUT
            INHDR
            CODE        TRAN3270
            MSGLIMIT    1
            SETSCAN     0
            LTR         15,15
            BM          OUT
            LR          10,15
            CLI         3(10),X'6D'     CLEAR KEY
            BNE         ONWARD1
            TERRSET
            B           ONWARD2
ONWARD1     CLI         3(10),X'7D'     ENTER KEY
            BE          NORMROUT
            CLI         3(10),X'F1'     PF1 KEY
            BE          DOTHING
                ...

            INMSG
            CANCELMG    X'0000080000'
            MSGGEN      X'0000080000'           *
                        X'0227F5C71DC11303'
                ...
```

Figure 69. Handling the AID Byte (Example 1)

```
MH3270      STARTMH     LC=OUT
            INHDR
            MSGLIMIT    1
            CODE
            MSGTYPE     X'6CD902'       SOH,%,R
*
*
*     Sense/Status Processing
*
*
            MSGTYPE                     ALL OTHER TYPES
            MSGEDIT     ((R,,SCAN,(2)))REMOVE CU AND DVC
            MSGTYPE     TABLE=AIDTABLE,EXIT=INVALID

              ...

AIDTABLE    TYPETABL    X'60',ROUTINE=KEYNOOP
            TYPETABL    X'E6',ROUTINE=CARDIN
            TYPETABL    X'E8',ROUTINE=NOPRTACT
            TYPETABL    X'6B',ROUTINE=PA3KEY
            TYPETABL    X'6C',ROUTINE=PA1KEY
            TYPETABL    X'6D',ROUTINE=CLEARKEY
            TYPETABL    X'6E',ROUTINE=PA2KEY
            TYPETABL    X'F1',ROUTINE=PF1KEY

              ...

            TYPETABL    X'F9',ROUTINE=PF9KEY
            TYPETABL    X'7D',ROUTINE=ENTERKEY
```

Figure 70. Handling the AID Byte (Example 2)

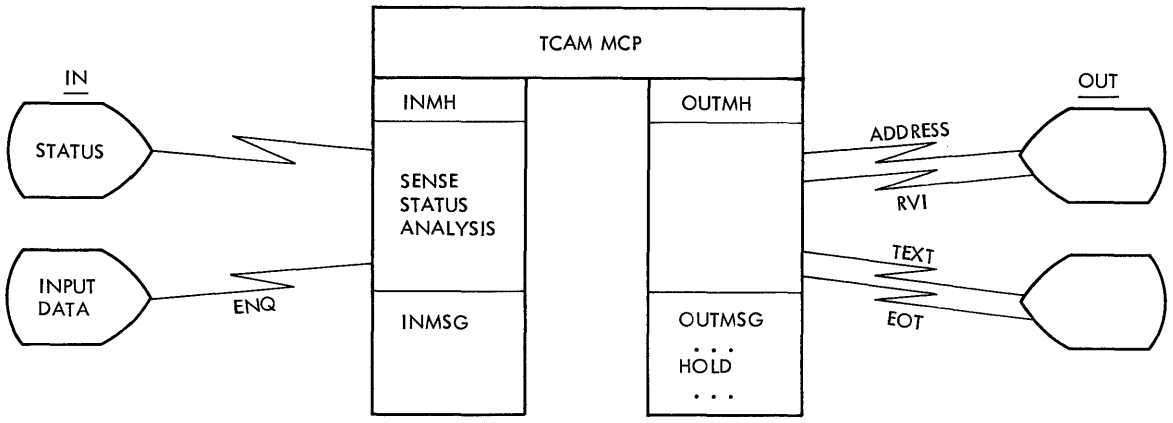


Figure 71. TCAM Sense/Status Problem

Figure 72 shows how TCAM and the message control program can handle sense/status messages. If TCAM addresses a device and receives a Reverse Interruption (RVI), or sends text and receives an EOT, sense/status is pending for the device. TCAM sets the appropriate bits for a permanent error and sends the message through message processing. The reason for the error message is not yet known, but you should issue a HOLD so that the message is not lost.

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 turn off the device until it can be 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 to minimize WACK responses.

Figure 73 shows a procedure for analyzing sense/status messages and methods for handling them. 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 INHDR. 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 HOLD. 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.

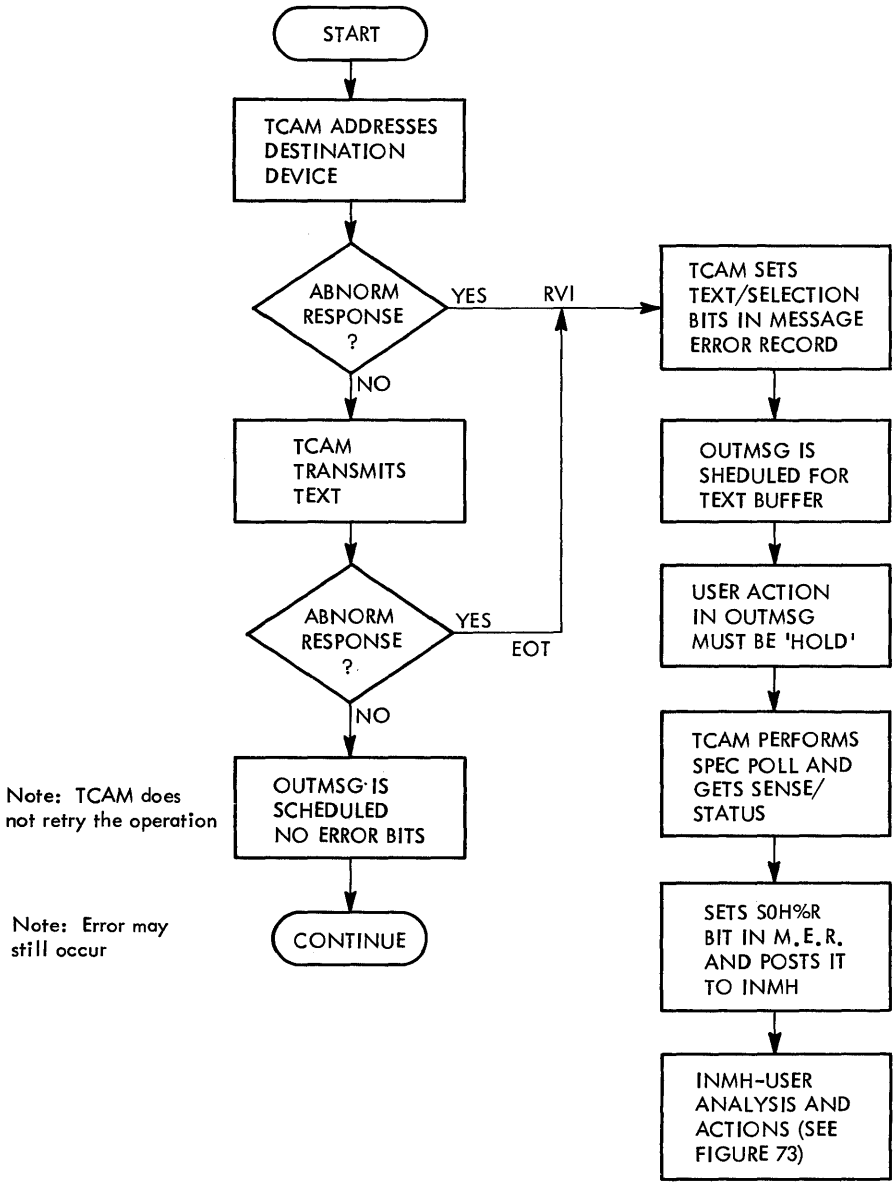


Figure 72. Handling Sense/Status Messages

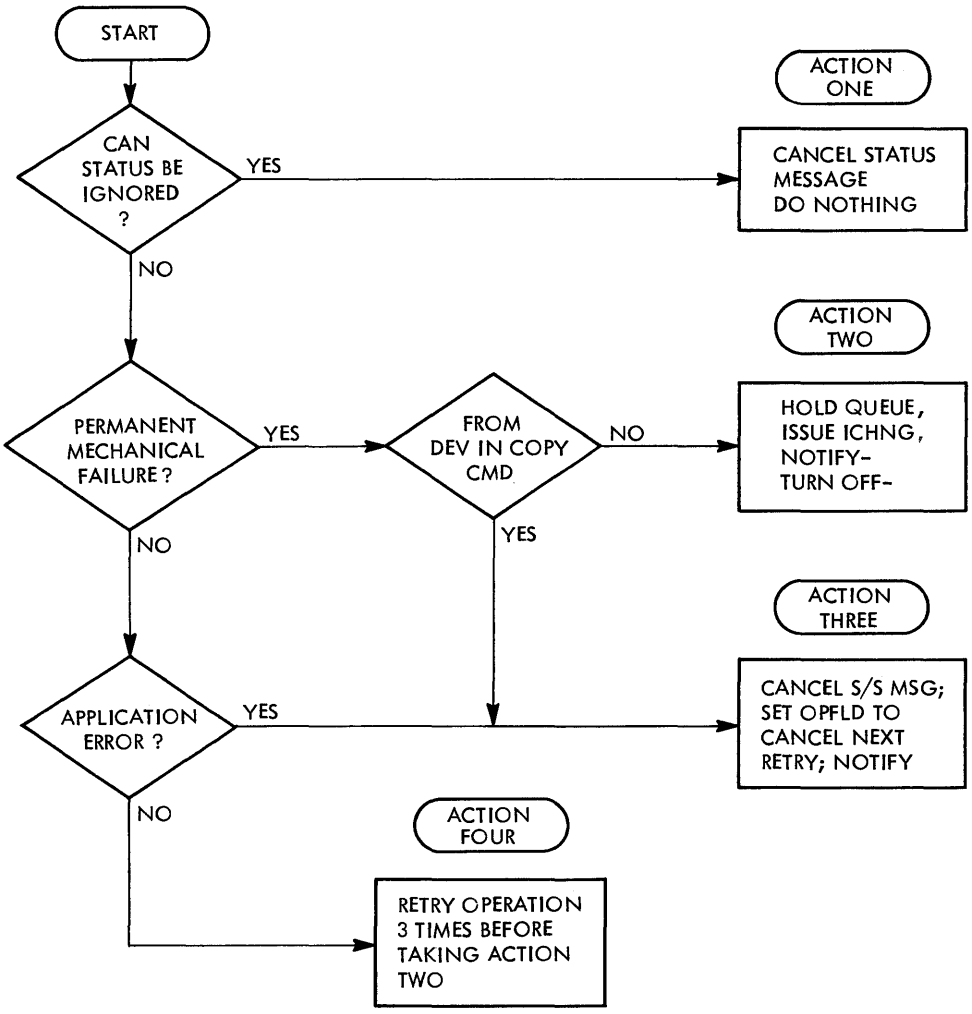


Figure 73. User Analysis of Sense/Status Messages

CHAPTER 5. VTAM SUPPORT

The Virtual Telecommunications Access Method (VTAM) and the Advanced Communication Function (ACF/VTAM) 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 provides large-screen support for the 3278 and 3276 displays plus the LOGMODE capability in record mode.

The VTAM and ACF/VTAM 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).
- An explanation of how to define a VTAM network that includes 3270s (the DOS/VS, OS/VS1, 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, and ACF/VTAM Macro Language Guide)

For explicit information on how to generate each 3270 device, see IBM 3704 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 BTAM under DOS/VS and with BTAM and TCAM under OS/VS. BTAM and TCAM 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 separate networks, provided that all the requirements of both access methods are met.

BTAM 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 74.) 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
- 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.)

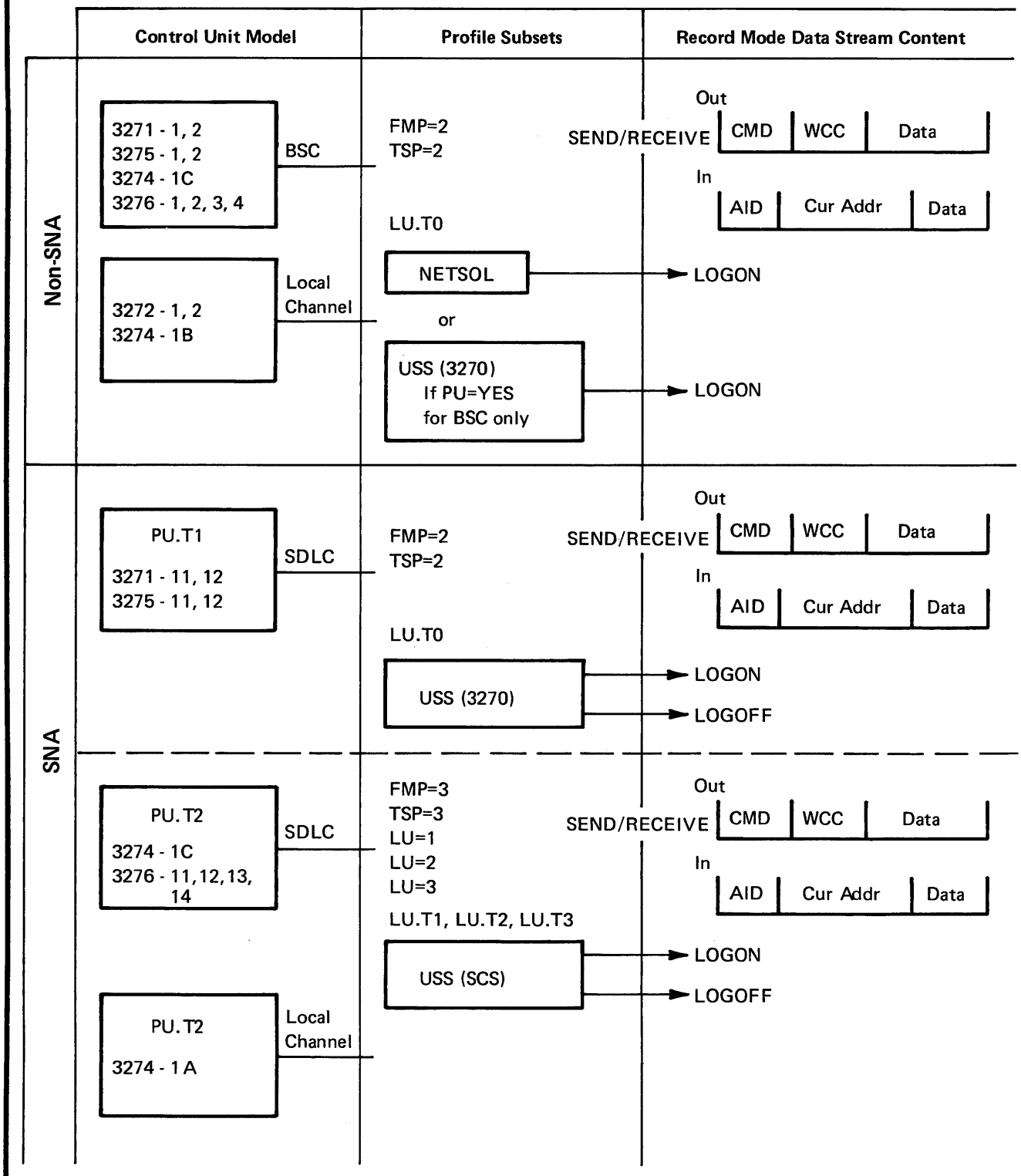


Figure 74. VTAM Support Summary

- 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 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 75 illustrates the defining of local non-SNA 3270s.

DEFINING THE BSC 3270

A CLUSTER and a TERMINAL definition statement define the BSC 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 buffer limit for the terminal
- 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 only)
- USS definition table requirements (ACF/VTAM 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 or 3276, substitute a 3271 as the operand.

LOGON REQUESTS (BSC 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 LOGAPPL operand of the TERMINAL statement for the BSC 3270. (NETSOL is the name of the IBM-supplied network solicitor.)

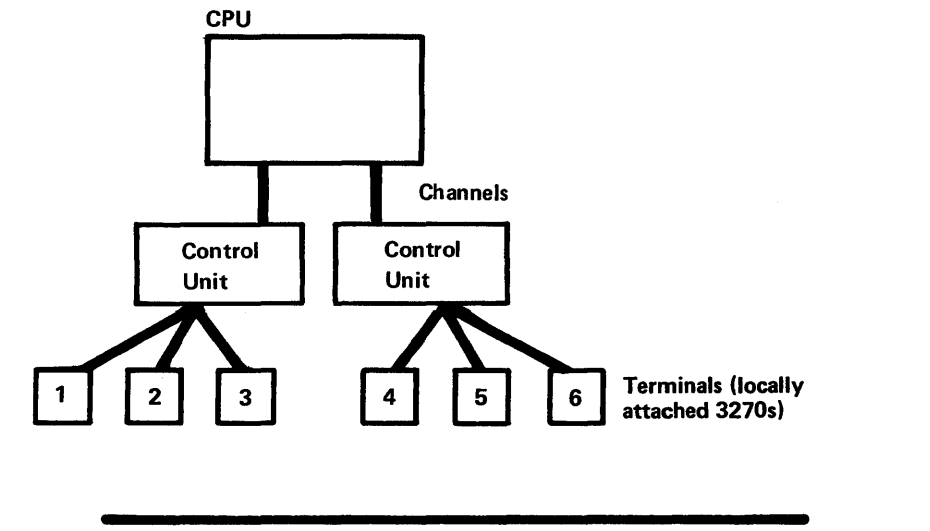


Figure 75. Grouping Locally Attached 3270s into Logical Sets

When the terminal operator enters a message at the 3270 terminal, the network solicitor determines whether the message is a valid logon 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: 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 logon, 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 logon has been rejected and invites the operator to enter another logon message.

Figure 76 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.

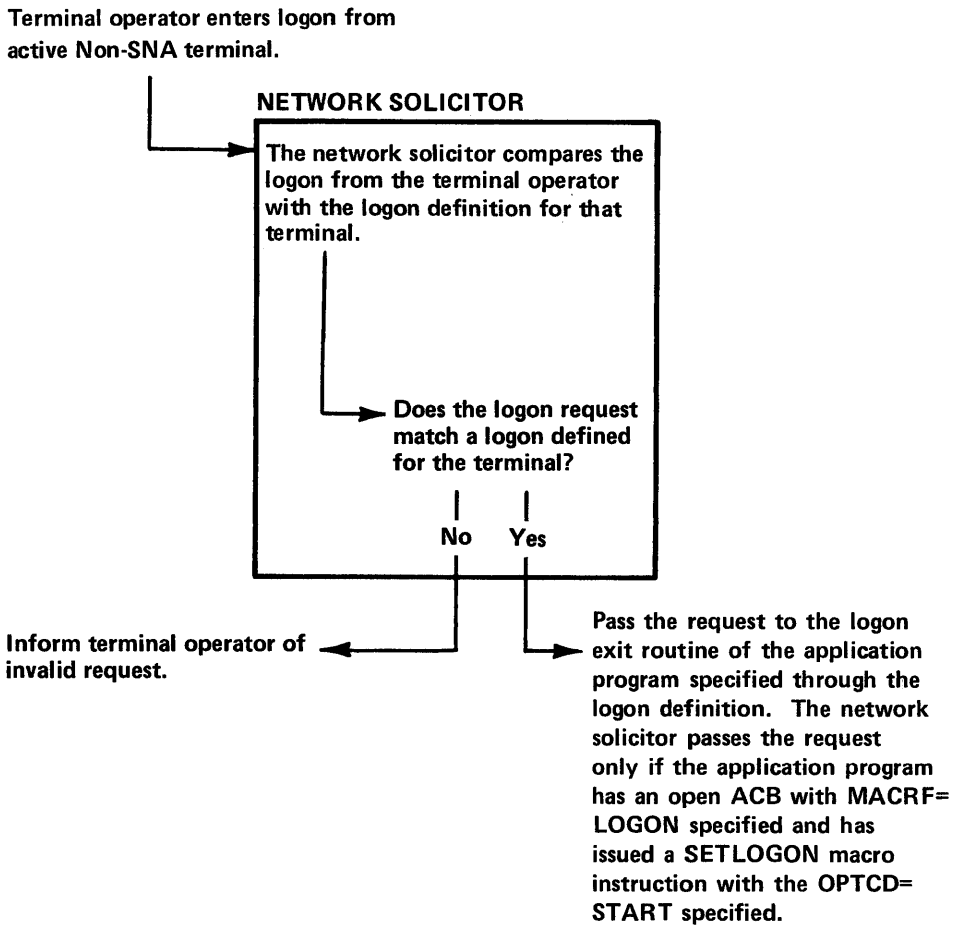


Figure 76. Processing a Terminal-Initiated Logon with the Network Solicitor

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 logon 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.

For the 3275 Models 11, 12 use PU parameters associated with PU.T1 support.

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 keyboard/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 1B 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 non-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 non-SNA 3270s, but a program can use record mode to communicate with both non-SNA and SNA 3270s. Using record mode with

non-SNA 3270s permits the VTAM (with some exceptions) and ACF/VTAM application program to communicate with non-SNA 3270s using the same VTAM instructions as those used to communicate with the SNA 3270s.

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 OPNDST. 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 non-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. Non-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 SESSIONC 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 non-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 LOGMODE 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.

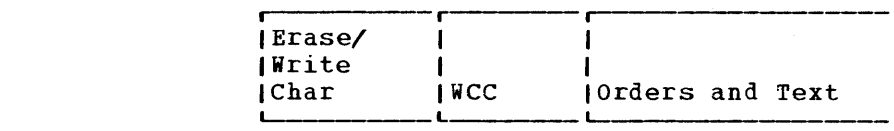
On completion of the RECEIVE macro instruction:

```
RECEIVE RPL=(2),RTYPE=DFSYN,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	SBA	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:



and be sent with this macro instruction:

```
SEND RPL=(2),STYPE=REQ,AREA=AREA1,RECLEN=50,OPTCD=ASY,
    POST=RESP,ECB=ECB1,RESPOND=(NEX,FME)
```

When VTAM determines that the message has arrived, it posts ECB1. (POST=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 RESP 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/O 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=(BB,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.

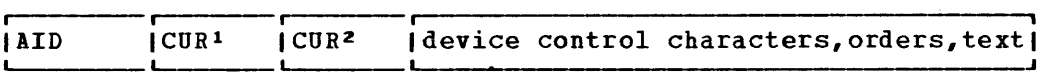
When the SEND data stream contains a 3270 Read command, the resulting input is received as a separate message, not as a response to the SEND operation. The application program should not begin or end a bracket when the data being sent contains a Read command. The application program must request a definite 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 3274/3276. 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.

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 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, Bid, 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 3270s

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 (OPTCD=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 TCAM 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 various print belts can be installed on this printer.

Network Solicitor Consideratons

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.

First byte	Second byte	Meaning
08	00	Device busy
04	00	Unit specify
02	00	Device end
01	00	Transmission check
00	20	Command reject
00	10	Intervention required
00	08	Equipment check
00	04	Data check
00	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 BASIC MODE

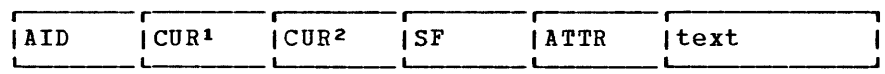
In basic mode, the MODE field of the NIB is set to BASIC when the 3270 terminal is connected. In basic mode, the VTAM application program exchanges data using READ and WRITE macro instructions; the program cannot use any record mode macro instructions with terminals connected in basic mode. It can, however, communicate with different devices on the same control unit in different modes at the same time. 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 FDBK 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 DO macro instruction is in the format:



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 RECLEN 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=CONV) 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 FDBK2 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 UNCOND 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:

First byte	Second byte	Meaning
C8	40	Device busy
C4	40	Unit specify
C2	40	Device end
C1	40	Transmission check
40	60	Command reject
40	50	Intervention required

First byte	Second byte	Meaning
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 (PU.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.

NON-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 BSC 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 non-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.

LOGON 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 77 shows SNA protocol differences between LU types 0, 1, 2, and 3 for the 3270.

LU Type 0	LU Types 1, 3	LU Type 2
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 Receives CANCEL, CHASE, BID, SHUTD, SIGNAL	Sends CANCEL, LUSTAT, SHUTC, SIGNAL Receives CANCEL, CHASE, BID, SHUTD, SIGNAL
Supports COPY command	Uses WRITE with start print bit set**	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*	Uses conditional bracket termination	Uses conditional bracket termination
Supports READ command to printers	Does not support READ command	Supports READ command

*PU.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.

**For LU.T1 only

Figure 77. 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 SEGNO 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 between 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

alphameric 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 alphameric input control; displayed, nondisplayed, display-intensified; selector-pen-detectable or -nondetectable; and modified or not modified.

attribute character: 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

basic mode: 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.

bracket: In VTAM, an exchange of data between an application program and a logical unit which accomplishes some task.

BSC: See Binary Synchronous Communications.

BTAM: 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 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 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 alphameric characters. The field continues to, but does not include, the next attribute character.

E

erase all unprotected (EAU) 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.

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 devices 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**.

MDT: 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.

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.

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 (PF) 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.

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 alphameric 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 definite 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 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 contains an attribute character and not a text character.

suppress index (SI) order: An order that generates the suppress index character, valid only for the 3288-2 printer (other printers receive |, an or 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.

INDEX

IBM **Technical Newsletter**

This Newsletter No. GN31-0755
Date January 6, 1978

Base Publication No. GC27-6999-2

File No. S360/S370-30

Previous Newsletters None

Introduction to Programming the 3270

This Technical Newsletter provides replacement pages for the subject publication. These replacement pages remain in effect for subsequent releases unless specifically altered. Pages to be inserted and/or removed are:

101–110
111–112 (text rearranged only)
113–128

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments
This newsletter provides reorganized and rewritten versions of Chapter 4, TCAM Support, and of Chapter 5, VTAM Support. New and changed material appears in both chapters to reflect more accurately the IBM 3274 and 3276 Control Units running with SNA protocol.

Note: *Please file this cover letter at the back of the manual to provide a record of the changes that have been made.*

IBM Corporation, Department 63T, Kingston, New York 12401

Printed in U.S.A.

Intro. to Programming the 3270 (File No. S360/S370-09) Printed in U.S.A. GC27-6999-2

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