

GC27-6980-4
File No. S370-30

Systems

**OS/VS
BTAM**

**OS/VS1 Release 7.0
OS/VS2 Release 3.8**

IBM

Appendix J of this publication lists the types of terminals that are supported by BTAM under OS/VS. Terminals that are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to IBM-supplied products or programs may have on such terminals.

Fifth Edition (March 1982)

This is a major revision of and makes obsolete GC27-6980-3 and associated newsletters GN27-1502, GN30-3061, GN30-3097 and GN30-3162.

This edition applies to release 7 of OS/VS1 and release 3.8 of OS/VS2 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM Systems, consult the IBM System/370 Bibliography, GC20-0001, and the current SRL Newsletter.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services which are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

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

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the content of this publication to: IBM Corporation, Publications Center, Department E01, P.O. Box 12195, Research Triangle Park, North Carolina 27709

This publication describes the BTAM facilities and macro instructions needed to write an application program that defines, activates, and controls a System/370-based teleprocessing system or local 3270 display system or both. Effective use of this publication does not presuppose a knowledge of teleprocessing techniques, but it does require a knowledge of the System/370 assembler language and of data management techniques. The publication does not contain detailed information on the terminal equipment and computers that may be used as remote or local stations in a BTAM-controlled system. For this information, see the lists of publications below.

This publication is organized as follows:

- "Teleprocessing and BTAM Concepts" explains concepts of teleprocessing in general and of BTAM in particular for the reader who is not already familiar with these concepts. This chapter also defines many terms used throughout the publication.
- "Defining the Teleprocessing System" tells how to define to OS/VS the characteristics of the communications lines and equipment making up the teleprocessing system. It includes information on the system generation procedure. The DCB macro instruction, among others, is described here.
- "Buffer Management" tells how to construct buffer pools and how to obtain and release buffers as needed to accommodate message data.
- "Code Translation" describes the facilities BTAM provides for accomplishing the necessary translation between the transmission code used on communications lines and EBCDIC, the internal code of System/370. The ASMTRTAB and TRANSLATE macro instructions are explained here.
- "Activating and Deactivating the Teleprocessing System" tells what procedures to follow in initializing the user program prior to message transmission and in deactivating the system upon conclusion of message transmission. The OPEN, LOPEN, and CLOSE macro instructions appear here.
- "Line Control and Message Transmission" briefly explains the techniques for

controlling communications lines of various kinds (switched, nonswitched, contention, polling) and presents the READ and WRITE macro instructions used in message transmission operation. Also given are the WAIT, TWAIT, and RESETPL macro instructions.

- "Start-Stop Read and Write Operations" lists the types of Read and Write operations applicable to each type of remote terminal and gives the channel program for each.
- "BSC Read and Write Operations" lists the types of Read and Write operations applicable to each kind of line configuration, gives the channel program for each, and indicates the types of remote stations for which each type can be used.
- "Local Read and Write Operations" lists the types of Read and Write operations applicable to the local 3270 display system and gives the channel program for each type.

Twelve chapters of programming considerations contain device-dependent information about remote computers, the 2780 data transmission terminal, the 2790 data communication system, the 2770 data communication system, the 2972 general banking terminal system, the 3270 information display system, 3650 retail store system, 3660 supermarket system, 3735 programmable buffered terminal, the 3741 model 2 data station, the 3750 switching system, the 3780 data communication terminal, and the 5275 direct numerical control station.

- "Error Recovery Procedures and Error Recording" explains the BTAM-provided facilities for diagnosing and attempting to recover from a variety of error conditions and for indicating and recording the occurrence of the errors.
- "Online Testing" describes the facilities available for diagnosing line and equipment troubles.
- Eighteen appendixes show control block, information table, macro instruction, and error message formats and contain code charts, sample programs, and storage estimates for VS2.

Before using this manual, the reader should be familiar with the following publications:

IBM Data Processing Glossary, GC20-1699

IBM System/370 Principles of Operation,
GA22-7000

OS/VS - DOS/VS-VM/370 Assembler
Language, GC33-4010

OS/VS Supervisor Services and Macro
Instructions, GC27-6979

OS/VS Data Management Services Guide,
GC26-3783

OS/VS Data Management Macro
Instructions, GC26-3793

For the transmission control units and terminals in his configuration, the BTAM user also needs to know the information in the following publications:

• **Transmission Control Units:**

IBM 2701 Data Adapter Unit, Component
Description, GA22-6864

IBM System/360 Component Description:
IBM 2702 Transmission Control,
GA22-6846

IBM System/360 Component Description:
IBM 2703 Transmission Control,
GA27-2703

• **Start-Stop Terminals:**

IBM 1030 Data Collection System,
GA24-3018

IBM 1050 Reference Digest, GA24-3020

IBM 1050 System Summary, GA24-3471

IBM 1050 Data Communications System,
Principles of Operation, GA24-3474

IBM 1050 Operator's Guide, GA24-3125

IBM 1060 Data Communications System,
GA24-3034

IBM System/360 Component Description:
IBM 2260 Display Station; IBM 2848
Display Control, GA27-2700

IBM System/360 Component Description:
IBM 2265 Display Station; IBM 2846
Display Control, GA27-2731

IBM 2740 Communications Terminal,
GA24-3403

IBM 2740/2741 Communications Terminal
Operator's Guide, GA27-3001

IBM 2760 Optical Image Unit, Component
Description, GA27-3011

• **BSC Stations:**

IBM SRL General Information -- Binary
Synchronous Communications,
GA27-3004

IBM System/3 RPG-II Telecommunications
Programming Reference Manual,
SC21-7507

IBM 1130 Functional Characteristics,
GA26-5881

IBM 1130 Synchronous Communications
Adapter Subroutines, GC26-3706

IBM 2770 System Components, GA27-3013

IBM 2780 Data Transmission Terminal,
Component Description, GA27-3005

IBM 2790 Data Communication System,
Component Description, GA27-3015

IBM 2972 Models 8 and 11 General
Banking Terminal System, Component
Description, GL27-3020

IBM 50 Magnetic Data Inscrber,
Component Description, GA27-2725

IBM 3270 Information Display System,
Library Users Guide, GA23-0058 for the
manuals applicable to the configuration.

Introduction to Programming the IBM
3270, GC27-6999

IBM 3735 Programmable Buffered
Terminal, Concept and Application,
GA27-3043

IBM 3735 Programmer's Guide, GC30-3001

IBM 3740 BTAM/TCAM Programmer's Guide,
GC21-5071

Component Information for the IBM 3780
Data Communication Terminal,
GA27-3063

• **Local 3270 Display System:**

IBM 3270 Information Display System,
Library Users Guide, GA23-0058 for the
manuals applicable to the configuration.

To assemble, link-edit and execute a BTAM program requires knowledge of the information in:

OS/VS System Generation Introduction,
GC26-3790

OS/VS1 System Generation Reference,
GC26-3791

OS/VS2 System Generation Reference,
GC26-3792

OS/VS Linkage Editor and Loader,
GC26-3813

OS/VS JCL Reference, GC28-0618

TELEPROCESSING AND BTAM CONCEPTS	1
Categories of Communications Lines	1
Line and Station Configurations	1
Duplex and Half-Duplex Transmission	2
Transmission Techniques	4
Transmission Codes	4
Line Control	4
Establishing Contact	5
Switched Lines	5
Terminal Lists	6
Positive and Negative Responses	6
Buffering	6
Dynamic Buffering	7
Conventions Used in this Publication	7
BTAM MACRO INSTRUCTION FORMATS	9
Positional Operands	9
Keyword Operands	9
Continuation Lines	9
Coding Aids	9
DEFINING THE TELEPROCESSING SYSTEM	11
Defining Communications Line Groups	11
DCB (Data Control Block) Macro Instruction	13
Defining and Modifying Terminal Lists	21
DFTRMLST (Define Terminal List) Macro Instruction	21
CHGNTRY (Change Terminal Entry) Macro Instruction	22
BUFFER MANAGEMENT	25
Constructing Buffer Pools	25
Using the BUILD Macro Instruction	25
Using the GETPOOL Macro Instruction	25
BTAM Construction of Buffer Pools	25
Programmer Buffering	25
Dynamic Buffering	27
Read Operations	27
Write Operations	28
Buffer Management Macro instructions	29
REQBUF (Request Buffer) Macro Instruction	29
RELBUF (Release Buffer) Macro Instruction	30
CODE TRANSLATION	31
ASMRTAB (Assemble Translation Table) Macro Instruction	31
TRNSLATE Macro Instruction	31
ACTIVATING AND DEACTIVATING THE TELEPROCESSING SYSTEM	34
Program Initialization	34
Opening and Closing Line Group Data Sets	34
OPEN Macro Instruction	35
LOPEN Macro Instruction	36
CLCSE Macro Instruction	36
LINE CONTROL AND MESSAGE TRANSMISSION	38
Line Control	38
Contention System	38
Centrally-Controlled System	38
Switched Systems	38
Error Detection and Message Blocking	39
Channel Programs	39
Message Transmission	39
User Program Analysis	40
Use of Line Control Characters	40
READ and WRITE Operations	41
The Reset Function	41
Terminal Lists	41
Data Event Control Block	41
READ and WRITE Macro Instructions	42
RESETPL (Reset Polling List or Reset Line) Macro Instruction	50

WAIT Macro Instruction	51
TWAIT Macro Instruction	51
Read Skip Operations	52
START-STOP READ AND WRITE OPERATIONS	53
IBM 1030 Data Collection System	53
Defining Terminal Lists	53
READ Macro Instructions	53
WRITE Macro Instructions	53
IBM 1050 Data Communication System -- Nonswitched Lines	55
Defining Terminal Lists	55
READ Macro Instructions	55
WRITE Macro Instructions	55
IBM 1050 Data Communication System -- Switched Lines	57
Defining Terminal Lists	57
READ Macro Instructions	57
WRITE Macro Instructions	58
IBM 1060 Data Communication System	60
Defining Terminal Lists	60
READ Macro Instructions	60
WRITE Macro Instructions	60
IBM 2260/2848 Display Complex (Remote) and IBM 2265/2845 Display Complex (Remote)	62
Defining Terminal Lists	62
READ Macro Instructions	62
WRITE macro instructions	63
IBM 2740 Communication Terminal -- General Information	66
IBM 2740 (Basic)	68
READ Macro Instructions	68
WRITE Macro Instructions	68
IBM 2740 with Checking Feature	69
READ Macro Instructions	69
WRITE Macro Instructions	69
IBM 2740 with Dial-up Feature	71
Defining Terminal Lists	71
READ Macro Instructions	71
WRITE Macro Instructions	71
IBM 2740 with Dial-up and Checking Features	72
Defining Terminal Lists	72
READ Macro Instructions	72
WRITE Macro Instructions	72
IBM 2740 with Dial-up and Transmit Control Features	74
Defining Terminal Lists	74
READ Macro Instructions	74
WRITE Macro Instructions	74
IBM 2740 with Dial-up, Transmit Control, and Checking Features	76
Defining Terminal Lists	76
READ Macro Instructions	76
WRITE Macro Instructions	76
IBM 2740 with Station Control Feature	78
Defining Terminal Lists	78
READ Macro Instructions	78
WRITE Macro Instructions	78
IBM 2740 with Station Control and Checking Features	79
Defining Terminal Lists	79
READ Macro Instructions	79
WRITE Macro Instructions	79
IBM 2760 Optical Image Unit -- General Information	81
Operation and Message Formats	81
Filmstrip Positioning	81
Modes of Operation	81
Response Points and Coordinates	82
Image Index Counter	82
Message Formats	82
Signals to Operator	84
Error Detection and Recovery	85
Online Testing	85
Programming Considerations	86
IBM 2740 with Checking Feature and 2760 Optical Image Unit	89
READ Macro Instructions	89
WRITE Macro Instructions	89
IBM 2740 with Dial-up and Checking Features and 2760 Optical Image Unit	92
Defining Terminal Lists	92

READ Macro Instructions	92
WRITE Macro Instructions	93
IBM 2741 Communication Terminal	96
General Information	96
READ Macro Instructions	97
WRITE Macro Instructions	97
Designing a Message Control Routine	98
Programming Considerations	98
AT&T 83B3 Selective Calling Stations	100
Defining Terminal Lists	100
READ Macro Instructions	100
WRITE Macro Instructions	100
Terminal-to-Terminal Operation	100
Western Union Model 33/35 TWX Terminals	101
Defining Terminal Lists	101
READ Macro Instructions	101
WRITE Macro Instructions	102
Western Union Plan 115A Outstations	103
Defining Terminal Lists	103
READ Macro Instructions	103
WRITE Macro Instructions	103
Terminal-to-Terminal Operation	103
World Trade Telegraph Terminals	104
General Information	104
Defining Terminal Lists	105
READ Macro Instructions	105
WRITE Macro Instructions	106
Programming Considerations for Use of Autopoll (Start-Stop)	107
Channel Programs	107
BSC READ AND WRITE OPERATIONS	109
General Information	109
Transmission Codes	109
Remote Station Compatibility and Intermixing	109
User Program Analysis	109
Line and Message Control Functions	109
ID Verification	109
Error Information Byte (EIB) Mode	110
Double Addressing (Multipoint Lines)	110
Message Formats	111
Use of Line Control Characters	111
Use of SOH and STX Characters	112
Coordinating BSC Central and Remote Programs	112
Responses	112
Responses to Message Text	112
Responses to ENQ	112
Responses to EOT (Switched Line Only)	112
BSC Nonswitched Point-to-Point Operation	114
READ Macro Instructions	114
WRITE Macro Instructions	115
BSC Nonswitched Multipoint Operation	118
Defining Terminal Lists	118
READ Macro Instructions	118
WRITE Macro Instructions	120
BSC Switched Point-to-Point Operation	125
Defining Terminal Lists	125
Defining Terminal List (SWLST) For Expanded ID Verification	128
READ Macro Instructions	130
WRITE Macro Instructions	133
LOCAL READ AND WRITE OPERATIONS	138
Local IBM 3270 Display System	138
READ Macro Instructions	138
WRITE Macro Instructions	138
PROGRAMMING CONSIDERATIONS FOR REMOTE COMPUTERS	141
System/3 Notes	142
System/7 Notes	143
IBM 2780 DATA TRANSMISSION TERMINAL -- PROGRAMMING CONSIDERATIONS	145
Initiating Transmission to an IBM 2780	145
End-to-End Control Characters	146
Component Selection	147
Horizontal Format Control	148

Multiple Record Transmission149
Programming Considerations150
IBM 2790 DATA COMMUNICATION SYSTEM -- PROGRAMMING CONSIDERATIONS152
2715 Pulse Count Feature152
2715 Tables155
Macro Instructions159
Programming Notes181
Message Format183
Control Transactions188
External Alarm Contact Feature190
IBM 2770 DATA COMMUNICATION SYSTEM -- PROGRAMMING CONSIDERATIONS192
System Configuration192
Transmission Codes192
Terminal Polling and Selection192
Temporary Transmission Delays193
Terminal Function Control194
Field-Control Operation195
Record Formats196
IBM 2972 GENERAL BANKING TERMINAL SYSTEM -- PROGRAMMING CONSIDERATIONS198
IBM 3270 INFORMATION DISPLAY SYSTEM -- PROGRAMMING CONSIDERATIONS199
3270 Devices Supported199
3270 Capabilities Supported199
Nonswitched Remote 3270 Display System199
Defining Communications Line Groups199
Defining and Modifying Terminal Lists200
Buffer Management203
Code Translation204
Activating and Deactivating the Teleprocessing System204
Line Control and Message Transmission204
Error Recovery Procedures and Error Recording211
Retry Options212
Online Testing212
Switched Remote 3270 Display System213
Defining communications Line Groups213
Defining and Modifying Terminal Lists213
Buffer Management215
Code Translation215
Activating and Deactivating the Teleprocessing System215
Line Control and Message Transmission215
Error Recovery Procedures and Error Recording221
Retry Options221
Online Testing221
Local 3270 Display System221
Defining the Local 3270 Display System222
Buffer Management223
Code Translation223
Activating and Deactivating the Local 3270 Display System223
Attention Interruptions and Read Initial Operations224
Read and Write Operations225
Prepare to Read Operations228
DEVICE END INTERRUPTS and BTAM READYQ229
Error Recovery Procedures and Error Recording230
Retry Options230
Online Testing230
System Generation230
Online Testing230
Conversion230
IBM 3650 RETAIL STORE SYSTEM231
IBM 3660 SUPERMARKET SYSTEM232
IBM 3735 PROGRAMMABLE BUFFERED TERMINAL233
IBM 3741 MODEL 2 DATA STATION, IBM 3747 DATA CONVERTER234
IBM 3750 SWITCHING SYSTEM (WORLD TRADE USERS ONLY)235
IBM 3780 DATA COMMUNICATION TERMINAL236

IBM 5275 DIRECT Numerical control station	237
ERROR RECOVERY PROCEDURES AND ERROR RECORDING	238
Basic Functions	239
Additional Functions	239
LERB (Line Error Recording Block) Macro Instruction	239
LERPRT (Line Error Recording Print) Macro Instruction	240
Error Detection and Analysis	241
User Program Analysis Procedure	241
BTAM ERP Error Analysis and Recovery Actions	243
Start-Stop Error Recovery Procedures	244
Error Recovery Actions for Start-Stop Operations	250
BSC Error Recovery Procedures	254
Error Recovery Actions for BSC Operations	254
Local 3270 Display System Error Recovery Procedures	255
Errors Detected by the Device or Control Unit and Channel Data	
Check Errors	255
Errors Detected by the Channel (Except Channel Data Check Errors)	256
SUGGESTED RETRY OPTIONS FOR READ AND WRITE OPERATIONS	263
ONLINE TESTING	272
Online Testing for Start-Stop Communications Lines	272
Test Type Codes	273
Terminal Test Restrictions	275
Online Testing for Binary Synchronous Communications Lines	276
Types of Tests	276
BTAM Responses to Request-for-Test Messages	277
BTAM Initiation of Request-for-Test Messages	278
ONLTST (Online Test) Macro Instruction	278
Formats of RFT, Test, and Console Messages	280
Request-For-Test Messages	280
Test Messages	281
Console Messages	284
Online Testing for Local 3270 Display System	284
Types of Tests	285
BTAM Response to Request-for-Test Messages	285
Formats of RFT, Test, and Console Messages	285
APPENDIX A: FORMAT OF TERMINAL LISTS	A-1
APPENDIX B: DATA EVENT CONTROL BLOCK	B-1
APPENDIX C: BTAM ERROR MESSAGES AND ABEND CODES	C-1
Assembly Errors	C-1
I/O Error Message	C-7
Line Error Recording Messages	C-8
Control Unit Or Device Inoperative Message	C-10
Remote BSC Station Error Messages	C-11
Error Status Messages (IBM 2770)	C-12
Terminal Error Status Message (IBM 2715)	C-13
Error Status Message (Remote IBM 3270)	C-14
BTAM Abend Codes	C-18
APPENDIX D: SYSTEM GENERATION MACRO INSTRUCTIONS	D-1
DATAMGT Macro Instruction	D-1
IODEVICE Macro Instruction	D-1
APPENDIX E: CODE CHARTS FOR BSC AND THE LOCAL 3270 DISPLAY SYSTEM	E-1
APPENDIX F: WORLD TRADE TELEGRAPH TERMINALS: TRANSLATION TABLE	
MODIFICATIONS	F-1
TRSLRCTW and TRSLRCT3 Macro Instructions	F-1
TRSLSCTW and TRSLSCT3 Macro Instructions	F-1
APPENDIX G: BTAM MACRO INSTRUCTION FORMAT CHARTS	G-1
APPENDIX H: BTAM CHARACTER SET AND CODE CORRESPONDENCE CHART	H-1
APPENDIX I: TRANSMISSION CODE CHART	I-1
APPENDIX J: LINE AND STATION CONFIGURATIONS SUPPORTED BY BTAM	J-1

APPENDIX K: IBM 2980 CHARACTER SET AND TRANSMISSION CODE CHART . . .	K-1
APPENDIX L: THE TPEDIT MACRO INSTRUCTION (IBM 50 MAGNETIC DATA INSCRIBER)	L-1
General Characteristics	L-1
TPEDIT Macro Instruction	L-1
Records Containing Errors	L-5
Examples of Records Containing Errors	L-6
Programming Considerations	L-8
End-of-Cartridge Code	L-8
APPENDIX M: SAMPLE 2715 TABLE LOAD MACRO ASSEMBLY	M-1
APPENDIX N: SAMPLE START-STOP AND BSC PROGRAMS	N-1
APPENDIX O: LOCAL 3270 SAMPLE PROGRAM	O-1
Defining the Local 3270 Display System	O-1
Operating the Sample Program	O-2
APPENDIX P: REMOTE 3270 SAMPLE PROGRAM	P-1
APPENDIX Q: USING SWITCHED NETWORK BACKUP WITH THE IBM 3872 OR 3875 MODEM	Q-1
APPENDIX R: BTAM STORAGE ESTIMATES FOR VS2.	R-1
ABBREVIATIONS AND ACRONYMS.	ABBRI
GLOSSARY.	GLOS-1
INDEX	INDEX-1

ILLUSTRATIONS

Figure 1. Basic Elements of a Teleprocessing System 2
Figure 2. Line and Station Configurations 3
Figure 3. Sample Line Addresses and Associated TCU and
Station Types 12
Figure 4. Sample Line Group Makeup 12
Figure 5. Relative Line Numbers for Example 1 13
Figure 6. Relative Line Numbers for Example 2 13
Figure 7. Keyword Operands for the BTAM DCB Macro Instruction . 14
Figure 8. Format of Data Control Block (DCB) 19
Figure 9. Contents of DCB Fields 20
Figure 10. Constructing Buffer Pools Using DS and BUILD 26
Figure 11. Constructing Buffer Pools Using GETMAIN and BUILD . . 26
Figure 12. Constructing Buffer Pools Using GETPOOL 26
Figure 13. Constructing Buffer Pools Automatically 27
Figure 14. Code Translation Tables Provided by BTAM 32
Figure 15. Initializing Your Program 34
Figure 16. Establishing Addressability for DCBs and DECBS 35
Figure 17. Formats of OPEN and CLOSE Macro Instructions. 37
Figure 18. READ and WRITE Options for Start-Stop 44
Figure 19. READ and WRITE Options for BSC 46
Figure 20. READ and WRITE Options for Local 3270 Display System 47
Figure 21. IBM 2260 AND 2265 Display Line Addresses 64
Figure 22. F (Function) Character Codes 81
Figure 23. A₁/I₁ and A₂/I₂ Character Codes 81
Figure 24. V & H (Vertical & Horizontal) Response Point
Coordinate Codes 83
Figure 25. Message Format 84
Figure 26. Examples of WRITE TIO and WRITE TCO Macro
Instructions (Nonswitched Line) 91
Figure 27. Examples of WRITE TIO, WRITE TVO, and WRITE TCO
Macro Instructions (Switched Line) 95
Figure 28. WT Telegraph Code 105
Figure 29. Example of a BSC Message Control Routine. 113
Figure 30. Summary of BSC Switched Line READ and WRITE Macro
and Terminal List Options 126
Figure 31. Multipoint Polling/Addressing Operations for 2780 . . 146
Figure 32. Examples of Component Selection for 2780 147
Figure 33. Vertical Forms Control Escape Sequences 148
Figure 34. Example of a 2780 Format Record 149
Figure 35. IBM 2780 Records: Number and Length 150
Figure 36. 2715 Macro Storage Site Estimates 162
Figure 37. 2798 GDU Guidance Panels 162
Figure 38. 2798 GDU Keyboard Character Conversion 163
Figure 39. AS or GDUAS Macro ID Parameter Decimal and
Hexadecimal Equivalents 164
Figure 40. Examples of Defining Transaction Group Tables for
Data Entry Units 165
Figure 41. ASLIST Operand Values for Guidance Lights by
Position on the Area Station 177
Figure 42. Message Formats 181
Figure 43. Transaction Control Byte Usage 185
Figure 44. Counter Control Operation 188
Figure 45. IBM 2213 Vertical Forms Control Escape Sequences . 194
Figure 46. IBM 2265 Erase and Write-at-Line Address Control
Characters 195
Figure 47. Control Unit Addresses for Polling List Entries and
for Identification in Input Messages (with Hexadecimal Repre-
sentations in EBCDIC and ASCII) 200
Figure 48. Control Unit Addresses for Selection List Entries
(with Hexadecimal Representations in EBCDIC and ASCII) 201

Figure 49. Device Addresses for Polling and Selection List Entries and for Identification in Input Messages (with Hexadecimal Representations in EBCDIC and ASCII)	.203
Figure 50. I/O Interface Code for Six-bit Structured Data	.204
Figure 51. Macro Instructions and Device Control Sequences for Nonswitched 3270 Function	.205
Figure 52. Sample Input Message (Showing Translation from EBCDIC to ASCII)	.207
Figure 53. Sample Output Message (Showing Translation from EBCDIC to ASCII)	.207
Figure 54. ID and Control Character Sequences for Calling or Answering Terminal Lists	.215
Figure 55. Macro Instruction Sequences to Perform Read and Write Functions at the Switched 3275	.216
Figure 56. Sample Local 3270 Display Systems Showing Device Addresses	.222
Figure 57. Line Group and Relative Line Numbers for Example 1	.223
Figure 58. Line Group and Relative Line Numbers for Example 2	.223
Figure 59. User Program Status Analysis	.242
Figure 60. User Program Sense Byte Analysis	.242
Figure 61. Should-Not-Occur Error Conditions Posted by BTAM ERP	.244
Figure 62. Status Analysis -- Start-Stop	.245
Figure 63. Sense Byte Analysis -- Start-Stop	.245
Figure 64. Unit Exception -- Start-Stop	.246
Figure 65. Lost Data -- Start-Stop	.246
Figure 66. Timeout -- Start-Stop	.247
Figure 67. Intervention Required -- Start-Stop	.248
Figure 68. Bus Out Check -- Start-Stop	.248
Figure 69. Data Check -- Start-Stop	.249
Figure 70. Overrun -- Start-Stop	.250
Figure 71. Command Reject -- Start-Stop	.250
Figure 72. Index to BSC ERP Tables	.254
Figure 73. Status Analysis -- BSC	.257
Figure 74. Channel Data Check -- BSC	.257
Figure 75. Equipment Check -- BSC	.258
Figure 76. Command Reject -- BSC	.258
Figure 77. Sense Byte Analysis -- BSC	.258
Figure 78. Bus Out -- BSC	.259
Figure 79. Overrun -- BSC	.259
Figure 80. Intervention Required -- BSC	.260
Figure 81. Data Check -- BSC	.261
Figure 82. Lost Data -- BSC	.262
Figure 83. Timeout -- BSC	.263
Figure 84. Unit Exception -- BSC	.264
Figure 85. Special Return Codes	.267
Figure 86. Error Post Actions	.272
Figure 87. Error Conditions (According to CSW and Sense Byte)	.273
Figure 88. Recovery Actions (By Error Condition and Failing Command)	.274
Figure 89. Error Conditions (According to ERPCODES Field in ERPIB)	.274
Figure 90. Recovery Actions (By Error Condition and Failing Command)	.275
Figure 91. Retry Options for Write Operations (Nonswitched Point-to-Point Line [BSC1])	.277
Figure 92. Retry Options for Read Operations -- Answering (Switched Point-to-Point [BSC2] except 3275)	.278
Figure 93. Retry Options for Read Operations -- Calling (Switched Point-to-Point [BSC2] except 3275)	.278
Figure 94. Retry Options for Write Operations (Switched Point-to-Point Line [BSC2] except 3275)	.279
Figure 95. Retry Options for Read Operations -- Answering (Switched Point-to-Point [BSC2] 3275 only)	.280
Figure 96. Retry Options for Read Operations -- Calling (Switched Point-to-Point [BSC2] 3275 only)	.281
Figure 97. Retry Options for Write Operations (Switched Point-to-Point [BSC2] 3275 only)	.282
Figure 98. Retry Options for Read Operations (Nonswitched Multipoint Lines [BSC3])	.283

	Figure 99. Retry Options for Write Operations (Nonswitched Multipoint Line [BSC3])	284
	Figure 100. Retry Options for Local 3270 Read and Write Operations	285
	Figure 101. Summary of BSC Online Test Options (Except for Remote 3270 Test Options)	300
	Figure 102. Summary of BSC Online Test Options for Remote 3270 Devices	301
	Figure 103. Summary of Online Test Options for Local 3270 Devices	301
	Figure 104. Format of Control Byte for OPENLST and WRAPLST Entries	A-1
	Figure 105. Open Polling or Addressing List (OPENLST): Format and Examples	A-2
	Figure 106. Wraparound Polling List (WRAPLST): Format and Example	A-3
	Figure 107. Dial List (DIALST): Format and Examples	A-3
	Figure 108. Identification List (IDLST): Format and Example	A-4
	Figure 109. Open and Wraparound Autopoll Lists for Start-Stop (SSALST, SSAWLST) and BSC (AUTOLST, AUTOWLST): Format	A-4
	Figure 110. BSC Dial List (BSCLST) (for System/370 to System/370): Format and Examples	A-6
	Figure 111. WT Terminal List (WTTALST): Format and Examples	A-6
	Figure 112. Calling and Answering Lists for Expanded ID Verification (SWLST): Format	A-7
	Figure 113. Format of Data Event Control Block	B-9
	Figure 114. Suggested Actions According to Remote 3270 Error Status Message	C-17
	Figure 115. World Trade Telegraph Codes ITA2 and ZSC3	F-1
	Figure 116. IBM 50 MDI Control Codes	L-2
	Figure 117. Format of Error Description Word	L-6
	Figure 118. Examples of Erroneous Records (IBM 50 MDI)	L-7
	Figure 119. Initial Format on Model 1 3277 Display Station	O-3
	Figure 120. Initial Format on Model 2 3277 Display Station	O-3
	Figure 121. Verification Format on Model 1 3277 Display Station	O-3
	Figure 122. Verification Format on Model 2 3277 Display Station	O-4
	Figure 123. Control Options Format on Model 1 3277 Display Station	O-4
	Figure 124. Control Options on Model 2 3277 Display Station	O-4
	Figure 125. BTAM Coding-Space Estimate	R-1
	Figure 126. BTAM Control-Information Area Estimate by Device type	R-2
VS2 Storage Estimates	Figure 127. BTAM Control Block Area for Each Line Group	R-4
	Figure 128. BTAM Control Block Space for Each Line	R-4
	Figure 129. BTAM Channel Program Space Estimate by Device per Line	R-5
	Figure 130. Storage Requirement for Code-Translation Tables for BTAM	R-6

SUMMARY OF AMENDMENTS (April 1975)

This Newsletter contains a discussion of an application program's response to a Request-for-Test message. It also includes a number of minor technical and editorial changes and corrections. If you are adding Release 4 of VS1 or Release 3 of VS2, you should, as with the addition of any new release, recompile your application programs.

Summary of Amendments Oct 1975

This Newsletter adds storage estimates for VS2, makes corrections to Autopoll information, and includes a number of technical corrections.

Summary of Amendments Sept 1976

This newsletter contains minor technical and editorial changes and corrections, updated information for READYQ and return codes for RESETPL and CHGNTRY macros, a new parameter HIO2740 for the RESETPL macro. Appendix B the DECB has been corrected and updated.

Summary of Changes Aug 1978

New Program Function Support of New 3270 Devices: The following summarizes the new devices that BTAM supports:

<u>NEW</u>	<u>ATTACHMENT</u>	<u>COMPATABLE WITH</u>
3274 1B	Local	3272
3274 1C	BSC	3271
3276 1, 2, 3, 4	BSC	3271

BTAM provides additional capabilities for applications to use the large (alternate) display or printer size of the devices attached to the 3274 1B and 1C and the 3276 1, 2, 3, 4 Control Units. For further description of these capabilities see the following two sections under "IBM 3270 Information Display System, Programming Considerations":

- * Nonswitched Remote 3270 Display System, Line Control and Message Transmission.
- * Local 3270 Display System, Read and Write Operations.

Summary of Changes Jan 1980

This newsletter contains minor technical and editorial changes; Prepare to Read support for a local 3274 Model 1D Control Unit; Write Structured Field support for a local 3274 Model 1D Control Unit and a remote 3274 Model 1C Control Unit.

BTAM provides Prepare to Read support and Write Structured Field support on OS/VS1 and OS/VS2 MVS.

For more information on these capabilities see the following sections under "IBM 3270 Information Display System, Programming Considerations:"

- * Nonswitched Remote 3270 Display System, Line Control and Message Transmission
- * Local 3270 Display System, Read and Write Operations
- * Local 3270 Display System, Prepare to Read Operations

This chapter explains some fundamental aspects of computer-based data communications systems (often called teleprocessing systems) of the kind accommodated by the OS/VS Basic Telecommunications Access Method (BTAM). It also explains some basic terminology used throughout the publication. Since this discussion is intended to explain teleprocessing (TP) systems for the BTAM user, it does not attempt to encompass all kinds of teleprocessing systems. Although some of the terms defined apply to all or most communications systems, other terms are limited to communications programming usage or specifically to BTAM. Moreover, concepts and terminology are presented from the programmer's viewpoint, rather than from the engineer's.

Viewed in its most elementary aspect, a teleprocessing system consists of (1) a central computer and associated transmission control equipment, (2) remote stations, and (3) the electrical circuits (called communication lines or data links) that connect the remote stations to the central computer (See Figure 1). For the purpose of this discussion, the central computer equipment comprises the central processing unit (CPU) and the equipment by which the CPU is connected to the communications lines. The generic name of this equipment is transmission control unit (TCU).

The equipment constituting a remote station can be either a terminal or another computer. A terminal consists of a control unit and one or more input and output devices, each of which is called a component of that terminal. Each input device and each output device is considered a separate component.

Remote stations in a BTAM-controlled teleprocessing system are usually separated from the central computer by a distance sufficient to require common carrier facilities and transmission techniques to accomplish communication between central computer and remote stations. (Communications common carriers are companies that furnish communications services to the public.) However, it is the method of connection to the central computer, rather than the distance from the computer, that determines whether a station is classed as remote. A station is considered remote if it is connected to the central computer through a transmission control unit (TCU). (A station connected directly to a computer data channel is termed a local station.) Except

for the local 3270 display system, OS/VS BTAM supports only remote stations, which must be connected to the central computer by means of an IBM 2701 Data Adapter Unit or an IBM 2702 or 2703 Transmission Control. Local 3270 display systems are connected directly to a selector, multiplexer, or block multiplexer channel of the central computer.

An operator's console is an input/output device whose function is to control the operations of the computer. The console and its terminal control unit make up a terminal that can communicate with the operating system and with problem programs but cannot communicate with other terminals. If the operating system includes the Multiple Console Support (MCS) option, BTAM can communicate with those operator's consoles that are connected to the central computer through a 2701, 2702, or 2703 transmission control unit.

Note: Appendix J describes the terminals and configurations supported by BTAM.

CATEGORIES OF COMMUNICATIONS LINES

Communications lines can be categorized by several sets of attributes, some of which are discussed below. Some attributes have significance for the user's BTAM program, others need only be specified during system generation, the way the attributes of local I/O devices are specified.

LINE AND STATION CONFIGURATIONS

A communications line can be classified according to whether it connects two or more than two stations and whether or not the electrical connection between the central computer and the station is continuously established. Figure 2 illustrates a teleprocessing system comprising several types of line and station configurations, the elements of which are explained below.

A nonswitched line is one that continuously links the stations associated with it, regardless of the amount of time it is in use for message traffic. This kind of line is usually furnished by a common carrier on a contractual basis, between specified locations for a continuous period, or regularly recurring periods, for the exclusive use of one customer.

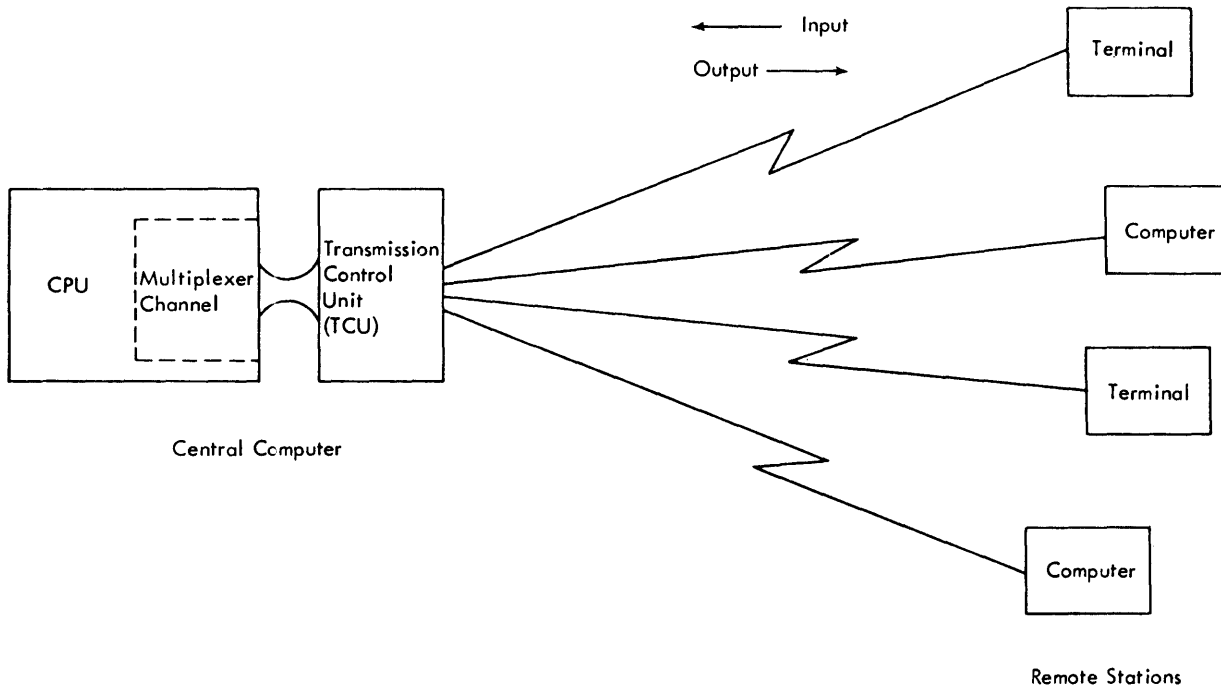


Figure 1. Basic Elements of a Teleprocessing System

A nonswitched line is called point-to-point if it connects the computer to a single remote station; it is called multi-point if several remote stations are connected to the line.

A switched line is one in which an electrical connection between the central computer and a remote station is established by dialing, the way ordinary telephone calls are made. As in the public telephone network, the actual communication path for a given transmission is not fixed but is automatically selected from a variety of possible paths by common carrier switching equipment.

Each remote station on a switched line is continuously connected to the common carrier switching center (exchange) by an access line, the way a telephone is. A telephone number is associated with the access line. Similarly, each TCU at the central computer is connected to the exchange by access lines. Usually, a TCU has several access lines, each with its own telephone number; multiple access lines permit simultaneous communication with several remote stations. Each connection of an access line at the TCU is called a switched line termination or line appearance.

Common carriers usually charge for switched lines on a time-used rather than a contractual basis.

A switched line is always considered point-to-point, as communication occurs

with only one remote station on a line during any call. Switched line connections are established by manual dialing or by automatic dialing under program control. The dialing operation may be performed at the central computer or at the remote station, and the called station answers manually or automatically. Not all of these options are available for all types of line configurations and remote stations. Within the limitations imposed by equipment or programming, the user chooses among them on the basis of the requirements of his application. For example, if the application involves collection of batched data from a number of remote stations after normal working hours, it would be appropriate to have each station equipped with an automatic answering facility and the central computer equipped with the calling facility. This would allow the program automatically to call the unattended stations and receive the batched data.

DUPLEX AND HALF-DUPLEX TRANSMISSION

The term duplex applies to a communications line that can accommodate data transmission in both directions at once. Half-duplex lines permit transmission in only one direction at a time. In a BTAM-controlled teleprocessing system, data transmission is always in half-duplex mode; messages are never transmitted in both directions at once.

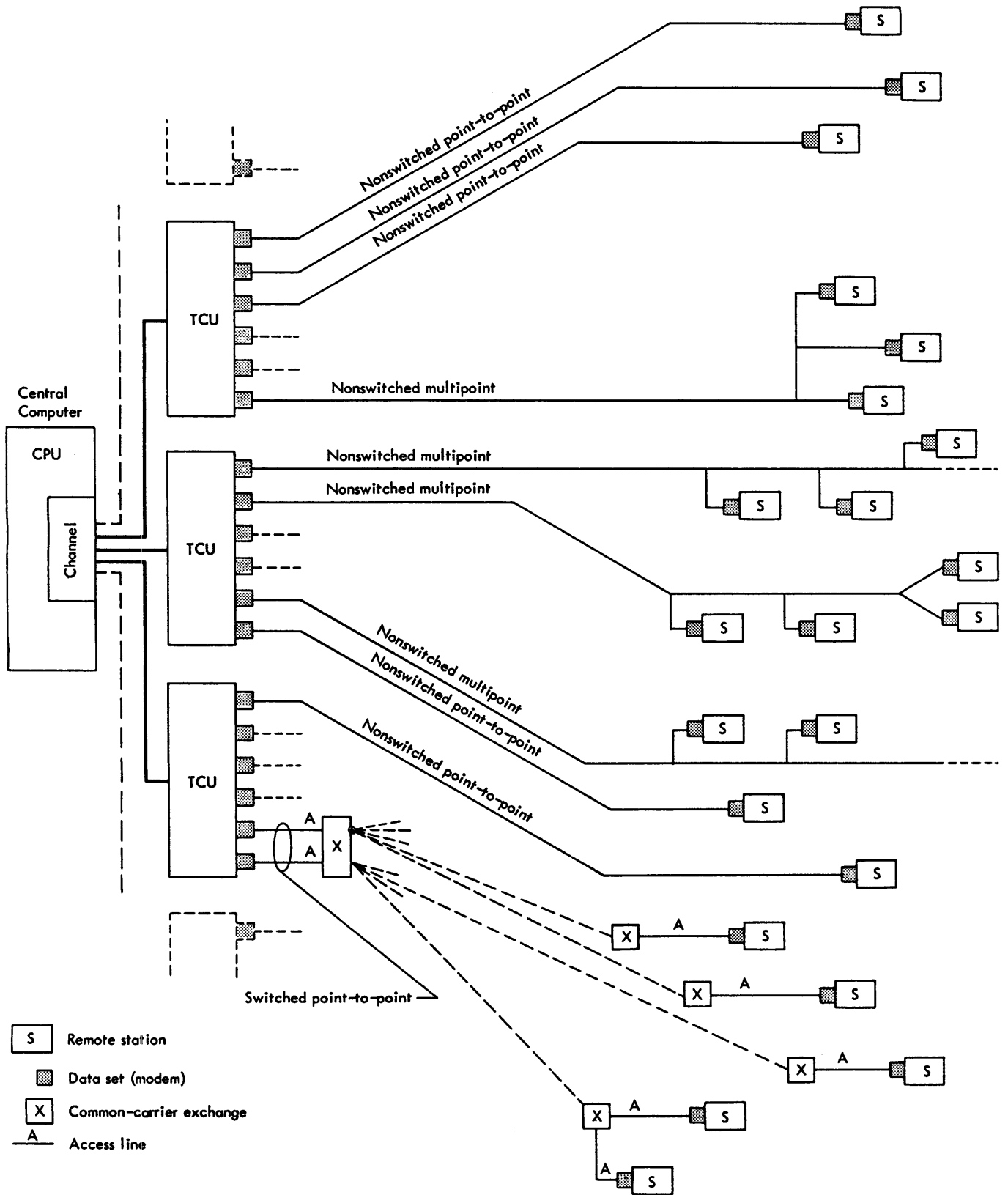


Figure 2. Line and Station Configurations

TRANSMISSION TECHNIQUES

Transmission technique is the way in which data characters are represented on the communications line. The two techniques used by computers and terminals supported by BTAM are start-stop and binary synchronous.

Detailed explanations of these techniques are not given here, since the programmer need not concern himself with them except to specify to BTAM which technique is used. Binary synchronous communication (BSC) is used for high-speed data transmission between the central computer and a remote computer or high-speed terminal. Start-stop transmission (also called asynchronous transmission) is used for data transmission at lower speeds between the central computer and remote terminals of various types.

TRANSMISSION CODES

Data can be represented on a communications line by any of several transmission codes. The code used on a given line is determined by the kind of station or the class of stations connected to the line. Some stations allow a choice of transmission codes. The BTAM programmer must be aware of the code used on a line, since he must sometimes specify, in the form of bit patterns, certain data characters to be transmitted by BTAM. At the back of this manual are charts giving the specific bit patterns of the characters contained in the character sets of the various transmission codes or station types.

LINE CONTROL

Just as a computing system, with its variety of peripheral input/output equipment, requires some means to coordinate the functioning of the various parts, the variety of I/O equipment making up a teleprocessing system requires a discipline to effectively manage the flow of message traffic. A significant difference should be noted, however. In a conventional computing system, the various I/O devices are at the service of the programmer; the requirements of his program and the characteristics of the data to be processed largely determine which input and output devices are to be activated and when. Moreover, the I/O devices are within reach of the computer operator; he can intervene when a device malfunctions to correct the condition or assign a different device. In a teleprocessing system, on the other hand, the central computer receives data at random from remote stations, and the operator at the central computer cannot exercise any direct control over remote stations. He cannot, for

example, correct a malfunctioning device at a remote station.

A further distinction between a computing system and a teleprocessing system lies in the handling of errors in data. With current techniques for transmitting data over long distances, errors are frequently introduced into message data by unavoidable transient line conditions such as crosstalk and lightning strikes. Transmission errors occur much less often in a computing system. A discipline for a teleprocessing system must accommodate the facility to detect transmission errors and, when possible, to correct them (as by retransmitting the message containing the errors). If the error is irrecoverable, its occurrence must be signaled to the user program, so that appropriate action can be taken.

The scheme of operating procedures and signals by which a teleprocessing system is controlled is called line control (for BSC, the term data link control is often used). A line control scheme must consider the functional characteristics and capabilities of the equipment and communication lines composing the system, as well as the operational requirements of the system. Some specific factors that line control must consider are: How is contact to be established between a sending and a receiving station? How is a message to be directed to a specific station on a multistation line? What happens if two stations try to send at the same time? What should be done if a station fails to respond to a message?

Line control can be classified in two ways. The first way is by the transmission technique (start-stop or binary synchronous) that is used for the line under consideration. With each of these techniques is associated a set of control characters and rules for their use to effect the needed functions. Some of the control characters are used for both start-stop and BSC, while others are peculiar to one or the other of the transmission techniques. The specific line control characters are explained under the discussions of these techniques in the section "Line Control and Message Transmission".

The second way in which line control can be classified is by the communication line configuration with which it is used. For example, line control for a switched line differs from that for a nonswitched line.

While the general capabilities and functions of a given line control scheme are identified in terms of transmission technique and line configuration, individual variations in capability and function arise from differences in the kind of stations to be controlled and from the presence or

absence in the stations of certain features. For example, a given line control scheme may include the control characters needed to indicate occurrence of a transmission error and to request automatic retransmission, but some types of station equipment that use that line control scheme may not be capable of error checking or automatic retransmission. Generally speaking, all stations connected to a given line must be designed to use the same line control scheme, and, if a certain capability is provided by some stations but not by others, the capability cannot be used.

It is not necessary for the BTAM programmer to specify the line control scheme to be used for a given line; this information is provided implicitly during system generation and during assembly in the DCB macro instruction for the line group of which the given line is a member. The programmer must, however, have a general understanding of line control concepts in order to correctly structure that portion of his program that performs message transmission. Also, the programmer must know the meanings of each of the line control characters, since he must regularly insert certain of them into output areas and arrange his program to look for them in input areas.

Line control functions can be considered in two categories: the functions needed to establish contact between central computer and remote stations and those needed to produce orderly flow of message traffic.

ESTABLISHING CONTACT

Contact may be established in several ways, depending in part upon the line configuration involved.

In some line control schemes one of the stations on a point-to-point nonswitched line can "bid" for use of the line, so that it can send a message to the other station. Occasionally both stations may simultaneously bid for use of the line. When this happens, the stations are said to contend with each other; a system in which this situation can occur is called a contention system. The line control scheme for a contention system must provide some means for resolving contention, that is, determining which of the contending stations is to be given the opportunity to send its message. After one station has been given control, the other is blocked from sending. A contention system is more frequently used for a point-to-point line configuration (that is, one involving only two stations) than for a multipoint configuration. BTAM currently provides contention line control only for point-to-point lines.

The alternative to a contention system is a system in which a control station (that is, the central computer) periodically contacts each of the remote stations in turn and allows it to send any input messages it has ready. ("Ready" means that the terminal operator is prepared to enter data from a keyboard or that some medium such as cards or paper tape has been placed in an input device so that the data can be transmitted automatically when the control station activates that device.) In this kind of system, each remote station has a unique identifier consisting typically of one or two characters, which, when sent over the line by the control station, causes that remote station and no other to respond. In a BTAM-controlled teleprocessing system only the control station, that is, the central computer, activates stations in this manner. The process of contacting in turn each of several stations on a line to determine whether any has input ready is called polling, and the station identifiers are called polling characters. Often, the first polling character identifies the station, and the second identifies a particular component from which data is solicited. A system in which stations are polled is called a polling system (in contrast to a contention system).

Although the term polling taken in its conceptual sense implies a nonswitched line to which are attached several stations, each of which is solicited in turn, the actual function of polling (that is, sending a station identifier) sometimes applies as well to a point-to-point nonswitched line or to a switched line. In the case of a switched line, the central computer may dial the telephone number of the station (or the station may dial the computer), and then the computer transmits the polling characters for that station.

In this discussion of contention and polling systems, the distinction between the two was based on establishing contact for the purpose of receiving input data from a remote station. The distinction is less clear in the case of output data. In either a contention or a polling system, the central computer must send a station identifier to select the specific station that is to receive an outgoing message. The station identifier in this case is called addressing characters (or selection characters), and the process is called addressing (or selection). As with polling characters, the first addressing character may identify the station, and the second, a particular component.

Switched Lines

It should be understood that, in the case of a switched line, the polling and ad-

dressings functions are independent of whether the central computer or a remote station initiated the telephone connection. Typically, the operator at a remote station dials the computer only when the remote station has data to send to the computer, and the computer therefore polls the station after the line connection is established. Similarly, the central computer may dial a remote station only when the computer has data to send and therefore addresses (or selects) the remote station. These conventions do not always prevail, however. For example, some applications require that certain stations be polled after working hours when the stations are unattended. With the proper common carrier equipment at the station, the computer can dial the station and then poll the input devices that the operator previously loaded with, for example, a deck of cards or a paper tape.

In establishing contact over a switched line, two situations should be avoided. First, dialing a wrong number can result in establishing contact with a station other than the one intended. Second, an unauthorized station, if provided with the telephone number of the central computer, could establish contact (assuming that the polling or addressing characters corresponded to the characters for authorized stations).

To prevent message transmission under either of these circumstances, identification verification may be used. (This is an optional facility available for certain kinds of stations.) In order to use this facility, each remote station that is permitted to call the computer over a specific switched line termination (that is, by calling a specific telephone number), must have an identification sequence that it automatically sends after the line connection has been established. The program compares the received sequence against a programmer-defined sequence. If they match, message transmission can proceed; if they differ, BTAM signals the fact by setting a flag bit and inhibits message transmission. The user's program must check the flag bit and take appropriate action, which ordinarily is to break the line connection.

If the remote station is a computer, the identification sequence is provided by the programmer, and each computer, central and remote, can check the identity of the other. If the remote station is a terminal, the sequence is mechanically or electrically established when the terminal is installed, and only the central computer can perform the checking function.

Because the central computer has no way of uniquely identifying a station that

calls it, all polling and addressing characters and identification sequences must be the same for any station that is to be permitted to call in over a given switched line termination.

The function of identification verification is not applicable to nonswitched lines, since the user determines, when the teleprocessing system is installed, which stations are to be connected to a specific nonswitched line.

Terminal Lists

When establishing contact with a remote station, the BTAM program gets the telephone numbers, polling or addressing characters, and identification sequences needed from a control table called a terminal list, which the programmer generates during assembly using a BTAM macro instruction provided for this purpose. The structure and contents of the terminal list vary according to the kind of line configuration and the remote station for which the list is being generated. (Terminal lists are not used for contention systems.)

Positive and Negative Responses

The discussion of how contact is established between stations has considered only the action taken by the originating station (that is, the station that initiates the contact). Before message transmission can proceed, the responding station (the station being contacted) must indicate to the originating station whether or not it is ready to receive or send a message. This indication is generally called a response or answerback, and it is termed positive if the station is ready, negative if it is not ready. The specific characters used for positive and negative responses vary with the type of station and the kind of line control (start-stop or BSC) under consideration.

BUFFERING

Buffering is a data management technique often used in conventional (nonteleprocessing) application, because, by permitting greater utilization of input/output areas, it minimizes the amount of main storage needed for these areas. This advantage is even more evident in a teleprocessing system, especially one involving many communication lines and varying message lengths.

Each Read or Write operation that involves transfer of text data between a central computer and a communication line requires that an input or output area be assigned to that line. However, to permanently assign main storage areas to each

communication line is wasteful, because these areas are idle except during the relative small proportion of time that text transfer to or from the communication line is in progress.

Because data transfer operations are virtually never in progress simultaneously on more than a small proportion of the lines in a system, only a relatively small number of main storage areas are needed to service many communication lines. Buffering permits these areas to be shared among the lines.

Buffering involves defining a group (or pool) of main storage areas called buffers, assigning buffers from this pool to Read and Write operations as needed, and then returning them to the pool when they are no longer needed, so they may be used for subsequent Read or Write operations.

When the buffer pool is formed, all buffers are chained together by placing a link field containing the address of the next buffer in the chain in the first fullword of each buffer. The link field of the last buffer contains zeros. Adjacent buffers in a chain are not necessarily in contiguous storage locations.

Buffers can be withdrawn from the pool singly or in chains. A buffer control block (BCB) associated with the pool always contains the address of the first available buffer of those remaining in the pool. When buffers are returned to the pool they are automatically inserted into the chain.

A control block associated with each Read and Write operation contains the address of the first buffer in the buffer chain that is assigned to the operation, so the programmer can always determine the address where the received data begins or where the data to be sent must be placed.

BTAM and OS/VS automatically perform the functions necessary to set up a buffer pool when the programmer provides certain information, such as the number of buffers he requires and the length of each. (All buffers in a pool have the same length). In addition, BTAM can automatically obtain buffers from the pool and provide them to the Read or Write operation; this is called dynamic buffer allocation or dynamic buffering. If the programmer does not specify the use of dynamic buffering, the program must request the required number of buffers before initiating the Read operation. This is called programmer buffering.

DYNAMIC BUFFERING

As mentioned earlier, buffering in general provides a significant increase in main storage utilization; dynamic buffering further increases the utilization. With programmer buffering, the programmer must anticipate the length of the message to be received; if messages can be of different lengths, he must request enough buffers to accommodate the longest message that can be expected, even though messages of this length are infrequently received. Furthermore, all buffers are provided in advance of the Read operation, even though they will not all be used at once.

When dynamic buffering is used, however, buffers are obtained singly (by means of program controlled interrupts) as the Read operation progresses. Only as many buffers as needed are obtained; when BTAM detects an ending character in a buffer, it does not get any more buffers. Besides allowing delayed acquisition of buffers, dynamic buffering allows buffers that are no longer needed to be progressively released to the buffer pool, instead of remaining idle until the end of the Read or Write operation and then being released as a group. This technique is possible, because, with dynamic buffering, BTAM sets a completion code in the high-order byte of each buffer when the Read or Write operation has finished filling or emptying the buffer. This is the same kind of completion code that is set in the event control block (ECB) at the end of the entire Read or Write operation. The program can check each buffer in turn for this completion code and release the buffer when the code is set.

Whether or not dynamic buffering is employed, BTAM does not release buffers that contain data. The programmer must do this himself, and failure to do so results in exhaustion of the buffer supply. Should this occur, no more Read or Write operations can be performed.

To summarize, dynamic buffering maximizes buffer utilization by (1) obtaining only as many buffers as are needed for an operation, (2) obtaining them just before actual use, and (3) allowing the programmer to release them one at a time immediately after use.

Further information on buffering will be found in the section "Buffer Management."

CONVENTIONS USED IN THIS PUBLICATION

To explain the use of BTAM, this publication must frequently express functional relationships between different parts of a

teleprocessing system, at varying levels of detail. To express these relationships clearly and concisely requires that certain conventions be observed.

Station, Computer, Terminal: The term station, when not qualified, refers to any of the computers and terminals, whether central or remote, connected to a BTAM-controlled communications line. Central computer means the computer in which the user program under consideration is running. Where the unqualified word computer appears, it means the central computer.

The general term remote station denotes a computer or terminal being controlled by the central computer. Where the context is appropriate, the specific term remote computer or remote terminal is used. For example, in discussions limited to start-stop communication lines, the phrase remote terminal is generally used, since start-stop lines do not accommodate remote computers. In discussions of binary synchronous lines, however, the more general phrase remote station is used, since BSC lines accommodate both computers and terminals.

Direction of Transmission: The terms input and output are always used relative to the computer in which the BTAM program under consideration is being run. Thus, whether BTAM is running in the central computer or a remote computer, input denotes data transmission from the remote station, and output denotes data transmission from the central computer.

In expressing a specific direction of transmission, the sending and receiving stations are always identified: as in "transmission from the central computer to a terminal." The phrase "transmission between the central computer and a terminal," on the other hand, implies transmission in either direction.

Data, Messages, Text, Control Characters: The term data is the most general of these terms; with respect to communication lines, it refers to any sequence of transmission code bit patterns, whether the patterns represent graphic characters, control characters, or binary information. Message means any sequence of data characters, considered as a unit, and includes any control characters necessary for transmission on a communications line. Text refers to the data characters comprising the information to be conveyed, such as plain language or binary data. Control characters are characters needed either to control transmission on the line (called line control or data link control characters) or to activ-

ate mechanical or formatting functions at a station (end-to-end control characters). Examples of line control characters are SOH, STX, and EOT (start of heading, start of text, end of transmission). Examples of end-to-end control characters are CR, LF, VT, and BEL (Carriage Return, Line Feed, Vertical Tab, Bell).

Usually, the name of a character and the function it performs are the same, for example, an EOT character indicates the end of transmission. In some cases, however, a particular function is effected by a different character or character sequence. For example, the EOA (end-of-address) character is sometimes used as a positive response signal, and for certain non-IBM terminals the characters FIGS H LTRS are employed as an end-of-transmission sequence. Where these disparities of function and character name occur, the intended meaning is made clear.

The text portion of an output message is given by the user to BTAM in a work area or buffer. The user also must provide certain line control characters in the buffer.

Read and Write Operations: The sequence of events by which data characters are sent or received is called a Read operation for input messages and a Write operation for output messages.

Each Read or Write operation is produced by a READ or WRITE macro instruction issued by the user's program (except for some operations performed automatically by error recovery procedures and online testing facilities). The term Read (or Write) operation may be qualified at several levels. For example, the phrase "Read operation" refers to any of several types of Read operation; the phrase "Read Initial operation" refers to any of several variations of Read Initial operations, and so on. Where a specific type is intended, the corresponding type code is usually given, as in "Read Initial Conversational (TIV) operation."

A Read or Write operation is performed by a sequence of commands executed by the channel to which the TCU is connected. These channel commands cause the TCU to transmit data characters and control signals on the line or, conversely, to respond to data characters and signals received from the line.

In discussions of Read and Write operations, the term command means a channel command, as represented in main storage by a channel command word (CCW).

BTAM macro instructions, like other operating system macro instructions, are written in the assembler language and accordingly are subject to the rules given in OS/VS and DOS/VS Assembler Language. BTAM macro instructions, like all assembler language macro instructions, are coded in this format:

Name	Operation	Operands
Symbol or Blank	Macro Name	One or more operands separated by commas

The operands are used to specify the facilities to be included, services to be performed, and various parameters needed by BTAM. Operands are coded according to the following rules.

POSITIONAL OPERANDS

Positional operands are shown as either small letters or capital letters. Small letters describe the kind of information to be coded; capital letters indicate the exact characters to be coded.

If the operand is shown as small letters (for example, inarea), substitute for it one of the values shown in the macro instruction format chart, Appendix G.

If the operand is shown as capital letters (for example, OPENLST), code it exactly as shown.

Code commas and parentheses exactly as shown. If an optional operand is omitted, a comma must still appear, except following the last operand coded.

KEYWORD OPERANDS

Keyword operands are shown as a word in capital letters, followed by an equal sign, followed by (1) a descriptive word or phrase in small letters or (2) a specific character or sequence of characters in capital letters.

If small letters follow the equal sign, code the keyword and equal sign exactly as shown, and substitute for the word or phrase in small letters one of the values shown in the macro instruction format chart.

If a specific character sequence follows the equal sign, code the entire operand exactly as shown.

Code commas and parentheses exactly as shown. Unlike positional operands, no comma need be coded in place of an omitted optional keyword operand.

CONTINUATION LINES

The operand field of a macro instruction can be continued on one or more additional lines as follows:

1. Enter a continuation character (any nonblank character that is not part of the operand coding) in column 72 of the line.
2. Continue the operand field on the next line, starting in column 16. All columns to the left of column 16 must be blank.

The operand field being continued may be coded in one of two ways. You may code the operand field through column 71, with no blanks, and continue the coding in column 16 of the next line, or you may truncate the operand field at the end of an operand (including the comma that follows the operand), and start the next operand in column 16 of the next line.

Examples:

Name	Operation	Operand	Col 72
NAME1	OP1	OPERAND1, OPERAND2, OPERAND3, OPERAND4, OPERAND5, OPERAND6	X
*		THIS IS ONE WAY	X
NAME2	OP2	OPERAND1, OPERAND2, OPERAND3, OPERAND4	X
*		THIS IS ANOTHER WAY	X

CODING AIDS

The symbols [] (brackets), { } (braces), and ... (ellipsis) are used to aid in defining macro instruction formats; they are never coded. Their meanings are as follows.

[] indicates that the enclosed operand is optional, or is coded under certain conditions. If more than one operand is stacked within brackets, one of the items, or none, may be coded. If one of the choices is underlined, the option associated with that choice is assumed to be specified if the operand is not coded.

indicates that one of the enclosed operands must be coded.

{ } defines the limits of a syntactical unit, if the unit consists of more than one operand, for example, ({operandone,operandtwo},...). The { } and ellipsis signifies that the two operands may be repeated in sets. The enclosing parentheses must be coded.

Defining a teleprocessing system means specifying to BTAM and OS/VS the characteristics of the communications lines, TCUs, and remote stations comprising the system. Using this information, OS/VS establishes the appropriate program interfaces between the user program and data management I/O routines.

During system generation, the programmer must specify the types of TCUs and remote stations making up the teleprocessing system and the features associated with each TCU and station. Appendix D explains how to code system generation macro instructions to provide this information. (See the OS/VS1 or OS/VS2 system generation reference manual for general information about the system generation procedure.)

During assembly and during program execution, the programmer must define communications line groups and terminal lists, as explained below.

For information about defining the local 3270 display system and about attention interruption handling, see the section "IBM 3270 Information Display System -- Programming Considerations."

DEFINING COMMUNICATIONS LINE GROUPS

A communications line group is a logical association of lines having characteristics similar enough that the same channel programs can be used for all lines in the group. These characteristics are as follows:

- All lines in a group must be start-stop, or all must be BSC; both types cannot be mixed in the same group.
- All lines must be of the same type. For start-stop lines, this means that all lines in the group must be non-switched, or all must be switched. For BSC lines, all lines in the group must be nonswitched point-to-point, non-switched multipoint, or switched point-to-point.
- All remote terminals connected to start-stop lines in a group must be of the same type, must have the same features, and must use the same transmission code.
- All remote stations connected to BSC lines in a line group must have the

same features and must use the same transmission code, but they need not all be of the same type. That is, more than one type (for example, System/370 Model 135, 1130, 2770) can be connected to the same nonswitched multipoint line, or more than one type can call or be called by the central computer over the same switched line termination (that is, the same telephone number at the central computer).

- Any optional functions that are specified apply to all lines in the group. For example, if dynamic buffering is specified for the line group, all lines must use dynamic buffering.
- Do not assign more than 60 lines to a DCB line group.

Communications line groups are treated as data sets, although they do not conform to the usual definition of data set (a named, organized collection of logically related records). Like other data sets, a communications line group is represented by a data control block (DCB), which is defined using the DCB macro instruction.

You may establish a line group in either of two ways. First, you may specify during system generation what lines are to constitute the group and symbolically identify the group during program execution by means of the UNIT parameter of a DD statement. Second, you may specify the makeup of the group during program execution, again using UNIT parameters on DD cards. Whichever method is used, during system generation you must identify to OS/VS (1) what kind of TCU is connected to each line and (2) what kind of terminals (start-stop) or what type of line (BSC) is associated with each line address.

A series of examples will make this clear. Assume that the teleprocessing network includes seven lines -- five start-stop and two BSC lines. Assume further that IBM 1050 terminals are connected to the five start-stop lines, and that any type of BTAM-supported BSC station is connected to the two BSC lines (see Figure 3). (The decision as to what types of remote stations are to be connected to which lines is part of installation planning; this example assumes that the decision has already been made.)

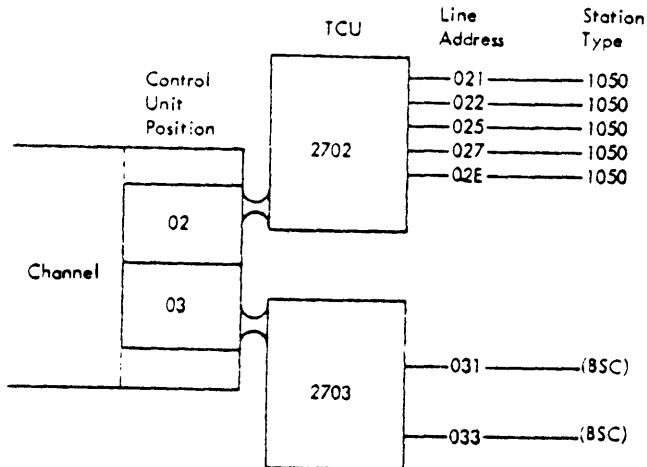


Figure 3. Sample Line Addresses and Associated TCU and Station Types

Before establishing the makeup of a line group, you must specify with IODEVICE macro instructions the type of TCU (2701, 2702, or 2703) and the type of terminal (for start-stop lines) or type of line (for BSC lines) associated with each line address.

```
IODEVICE UNIT=1050,ADDRESS=021,
TCU=2702,...
IODEVICE UNIT=1050,ADDRESS=022,
TCU=2702,...
IODEVICE UNIT=1050,ADDRESS=025,
TCU=2702,...
IODEVICE UNIT=1050,ADDRESS=027,
TCU=2702,...
IODEVICE UNIT=1050,ADDRESS=02E,
TCU=2702,...
IODEVICE UNIT=BSC3,ADDRESS=031,
TCU=2703,...
IODEVICE UNIT=BSC3,ADDRESS=033,
TCU=2703,...
```

Notice that the UNIT operands in the first five macro instructions specify the type of terminal, and those in the remaining two macro instructions specify "BSC3," a value that represents the type of line, in this case nonswitched multipoint. For a nonswitched point-to-point line you would specify BSC1; for a switched point-to-point line, BSC2. See the description of the IODEVICE macro instruction in Appendix D.

In these macro instructions, the three-digit addresses represent communications lines rather than specific devices, as would be the case in defining local I/O equipment. The ellipses represent other appropriate IODEVICE operands, including FEATURE, ADAPTER, and SETADDR. These are explained in Appendix D.

Assume now that you wish to establish three line groups from the seven lines, as

shown in Figure 4. Two examples show how to do this.

Example 1: To define the groups during system generation, you would code a UNITNAME macro instruction for each line group:

```
UNITNAME UNIT=(021,022,025,02E),NAME=GROUP1
UNITNAME UNIT=(027),NAME=GROUP2
UNITNAME UNIT=(031,033),NAME=GROUP3
```

During program execution, you would associate these line groups with specific data control blocks in your program by issuing for each group a DD statement identifying the group by its name, and indicating the number of lines in the group:

```
//DDGRP1 DD UNIT=(GROUP1,4),...
//DDGRP2 DD UNIT=(GROUP2,1),...
//DDGRP3 DD UNIT=(GROUP3,2),...
```

The name of the DD statement (for example, DDGRP1) must be the same as the DDNAME parameter in the data control block. In the foregoing DD statements, the number of lines indicated for each group is the same as the number of lines specified during system generation in the UNITNAME macros. You may, however, specify any lesser number of lines to be in the group when you issue a DD statement. Lines are always included beginning with the lowest line address.

For example, if you code

```
//DDGRP1 DD UNIT=(GROUP1,2),...
```

the two lines associated with the two lowest addresses in the original four-line group, 021 and 022, constitute the group.

Example 2: Alternatively, you may elect not to define the groups during system generation but to identify the lines in the group individually during program execution. To do this, code a separate DD statement for each line, specifying the

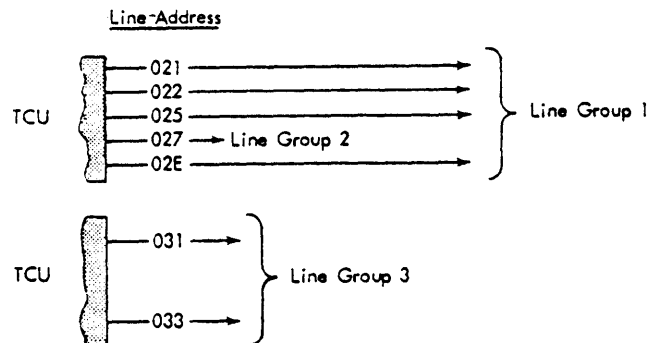


Figure 4. Sample Line Group Makeup

actual line address (instead of the name of a line group, as in example 1):

```
//DDGRP1 DD      UNIT=021,...
//      DD      UNIT=02E,...
//      DD      UNIT=027,...
```

These DD statements establish a group containing three lines.

Even if you have defined a line group by using a UNITNAME macro instruction during system generation, you may redefine it by using DD statements as shown in this example.

Relative Line Numbers: In READ and WRITE macro instructions, you must specify the line over which the Read or Write operation is to take place. Rather than specify an actual line address, you code in the macro a relative line number (RLN), which refers to the numerical position of the actual line address relative to other lines in the group. Thus, in Example 1, the relative line numbers are as shown in Figure 5; in Example 2, as shown in Figure 6. (In Example 2 four lines are not defined in any group.)

If you define a line group during system generation, using the UNITNAME macro instruction, the relative line numbers are assigned so that they represent an ascending numerical order of addresses, regardless of the actual sequence in which the addresses were arranged in the macro instruction. If, however, you define a line group by a sequence of DD statements, relative line numbers are applied in the same sequence in which the DD statements are coded.

See Appendix D for further information about system generation macro instructions involved in generating a BTAM system, and see the OS/VS1 or OS/VS2 system generation reference manual for complete information about system generation procedures.

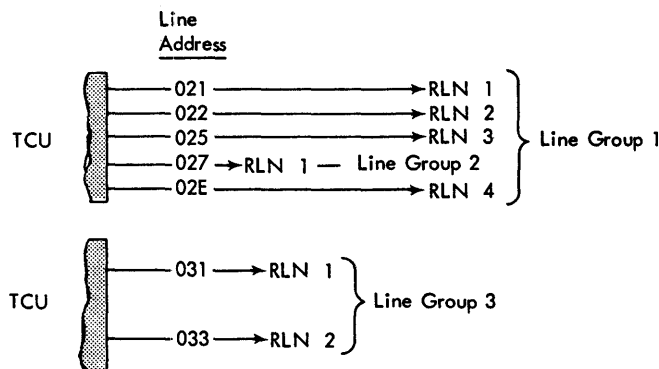


Figure 5. Relative Line Numbers for Example 1

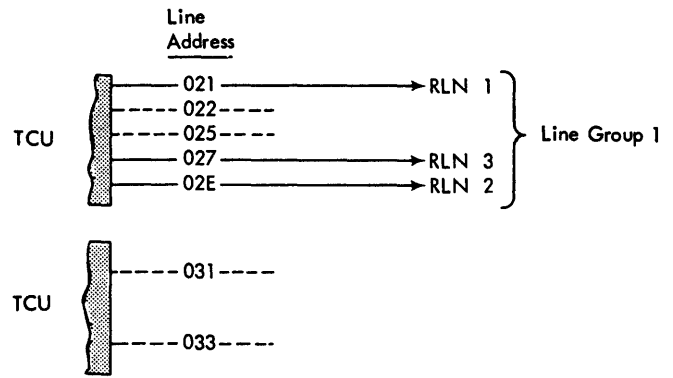


Figure 6. Relative Line Numbers for Example 2

DCB (Data Control Block) Macro Instruction

The DCB macro instruction defines the structure of a data control block and includes in it certain information that you have coded in the macro instruction. You must issue a separate DCB macro instruction for each line group data set.

Of the parameters that appear in the data control block, you must code certain ones in the macro instruction; others either may be coded in the macro instruction or may be supplied from an alternate source. The alternate source for an operand is indicated in the description of the operand, as follows:

- PP means you can enter the parameter into the data control block yourself during program execution at any time prior to opening the line group data set.
- OE means you can enter the parameter into the data control block yourself during program execution at any time up to and including the DCB exit taken during the opening process.

Name	Operation	Operands
symbol	DCB	keyword operands

symbol

is the name of the DCB macro instruction. It must be specified.

keyword operands

are the operands that can be included (see Figure 7).

The format of the data control block is shown in Figure 8. The contents of the fields in the DCB are described in Figure 9.

Keyword Operand and Description (part 1 of 5)

<code>DSORG=CX</code>	identifies the data set organization as that of a communications line group.
<code>MACRF=(R)</code> <code>MACRF=(W)</code> <code>MACRF=(R,W)</code>	specifies that access to the line group is to be gained with READ or WRITE macro instructions or both. Whichever option is coded, BTAM permits access with both READ and WRITE macro instructions. This operand is required.
<code>[DDNAME=ddname]</code>	(Alternate source: PP) is the name that appears in the DD statements associated with this data control block. If this operand is omitted, and no value is provided through an alternate source, the job is terminated.
<code>[BUFNO=number of buffers]</code>	(Alternate source: OE) is the number of buffers to be obtained by BTAM during open, if you wish BTAM to provide a buffer pool. Up to 225 buffers can be specified. You need not code this operand if BTAM is not to obtain a buffer pool.
<code>[BUFL=buffer length]</code>	(Alternate source: OE) is the length in bytes of the buffers making up the buffer pool, whether you provide the pool or BTAM provides it. The maximum value for BUFL is 32,760. A minimum limit on buffer length applies to BSC line groups under certain conditions (see "Programming Notes" under "READ and WRITE Macro Instructions." Specify this operand for all applications using buffers. BUFL must be a multiple of four.
<code>[BUFCB=buffer control block address]</code>	(Alternate source: OE) specifies the address of the buffer control block for a buffer pool you provide. If you wish BTAM to provide the buffer pool, omit this operand, and code the BUFNO and BUFL operands.
<code>[EXLST=exit list address]</code>	(Alternate source: PP) specifies the address of a BTAM program exit list, if you wish to provide one. Only the DCB exit may be used.
<code>[BFTEK=D]</code>	(Alternate source: OE) specifies that dynamic buffering is to be used for this line group. If dynamic buffering is specified, a buffer pool must be defined.
<code>[LERB=line error block address]</code>	(Alternate source: OE) specifies the address of the line error recording block. This operand is valid only if C is coded among the EROPT operand options.
<code>[EROPT=code]</code>	(Alternate source: OE) specifies the error recovery, error recording, and online test options to be provided for the line group. E specifies that the basic error recovery procedures (ERP) are to be provided for the line group. If EROPT is omitted, E is assumed. R specifies that text-read errors are to be retried in addition to the basic error recovery procedures. This option is valid only for the following terminals: 1050 terminals (valid for the card reader and paper tape reader only if line correction feature is installed), 2740 terminals with checking feature, and 2260 terminals. Do not specify EROPT=R if dynamic buffering is to be used (BFTEK=D), because the use of dynamic buffering precludes the retrying of text-read errors. (See the discussion under N, below, for considerations for AT&T 83B3 and WU 115A terminals.)

Figure 7 (Part 1 of 5). Keyword Operands for the BTAM DCB Macro Instruction

Keyword Operand and Description (part 2 of 5)

W

specifies text-write errors are to be retried in addition to basic error recovery procedures. This option is valid for all start-stop terminals, except World Trade telegraph terminals. It is invalid for BSC stations. It results in an additional copy of the message for each retry (except for the 2260 with the line address feature and the 1050 card punch and paper tape punch with the line correction feature). This parameter is ignored for BSC and World Trade telegraph terminals. Do not specify EROPT=W if dynamic buffering is to be used (BFTEK=D), because the use of dynamic buffering precludes the retrying of text-write errors.

C

specifies that threshold error counts and cumulative error counts are to be maintained in the line error recording block (LERB) for the line for data check, intervention required, and nontext time-out errors.

N

specifies that no error recovery procedures are to be provided for the line group. This parameter and E, R, W, and C are mutually exclusive. This parameter is invalid for BSC stations; if coded, it is ignored. It is recommended that EROPT=N be coded for AT&T 83B3 and WU 115A terminals if dynamic buffering is specified (BFTEK=D), because BTAM does not perform error retry either before or after start of text transfer when dynamic buffering is used for terminals of these types. If EROPT is omitted, or E, ER, or R is coded in the EROPT operand, ERP routines are unnecessarily loaded into the system, as they will remain unused.

T

specifies that the online test facility is to be used for the line group. This option is valid for all IBM stations with or without error recovery procedures. To receive standard IBM maintenance for a remote or local 3270 display system, this option must be specified.

Note: The parameters E, R, W, C, and T may appear in any combination. The parameter N may appear alone or with T. Commas must not be coded in this parameter. Example: EROPT=RECWT. When EROPT (any combination of E, R, W, and C) is coded in the DCB macro instruction, the user automatically gets the Outboard Recorder (OBR) and the Statistical Data Recorder (SDR) facilities for this line group. (These are facilities used by the customer engineer.) Error recovery procedures are required for BSC stations. For BSC line group data sets C and T are the only valid EROPT options; all other option codes are ignored. For World Trade telegraph terminals, E, C, and N are the only valid EROPT options; all other option codes are ignored. For the local 3270 display system, E and T are the only valid EROPT options; all other option codes are ignored.

DEVD=BS
DEVD=WT
DEVD=L D

LD

specifies that the line group contains locally attached devices.

BS

specifies that BSC is to be used and causes a 44-byte field to be added to the DCB. This field contains the line control characters in the transmission code to be used.

WT

must be coded if the line group contains World Trade telegraph terminals or if any of the keyword operands IAM,WRU,MON,MONDIY,ECM, and EOT are coded.

(MODE=((IBC),[CNTRL],[$\frac{A}{B}$],[$\frac{A}{B}$])) (BSC line group only)

IBC

specifies that the transmission control unit (TCU) at the central computer is to operate in EIB (Error Information Byte) mode. EIB mode is discussed in the General Information section of the BSC Read and Write Operations chapter.

CNTRL

should be coded if the central computer (this System/370) is to be given control when contention occurs on a point-to-point nonswitched line. It should be omitted if the remote station is to be given control.

Figure 7 (Part 2 of 5). Keyword Operands for the BTAM DCB Macro Instruction

Keyword Operand and Description (part 3 of 5)

A
specifies that communications are to be through the 2701 Data Adapter Unit's Dual Communication Interface A.

B
specifies that communications are to be through the 2701's Dual Communication Interface B. This parameter may not be coded if this feature is not present on the 2701.

A
specifies use of the transmission code designated by Code A for 2701 Data Adapter Unit Dual Code Feature.

B
specifies use of the transmission code designated by Code B for 2701 Dual Code Feature. This parameter may not be coded if this feature is not present on the 2701.

[CODE=transmission code]

(BSC line group only)

EBCDIC
specifies transmission in Extended Binary Coded Decimal Interchange Code.

USASCII
specifies transmission in United States of America Standard Code for Information Interchange.

TRANSC
specifies transmission in six-bit Transcode.

[,READYQ={ 0
 {relexp} }]

specifies the action that BTAM takes when polling a local 3270 device on a Read Initial operation, if a device previously not ready is turned on.

0
specifies that BTAM issues a RESETPL for the user when it discovers that a device previously not ready has become ready during Read Initial processing by sending Device End to the channel. After a Read Initial is issued, BTAM checks for attention interruptions before checking for devices that have become ready. The relative line number of the device that became ready is in the DECB at location DEPOLT+3.

relexp
specifies the address of a user routine that is asynchronously scheduled when a device that was previously not ready sends Device End to the channel. The user routine might, for example, Write Erase to clear the device's buffer. The routine receives control in problem program state, operating under the protection key of the user's task. Upon entry, register one contains in the high-order byte the relative line number of the device that became ready; the low-order three bytes contain the address of the DEB that defines the device's line group. Standard register save conventions are expected.

Note: If READYQ is omitted, an Intervention Required is posted as a permanent I/O error. If READYQ is specified, DEVD and LERB cannot be specified.

Figure 7 (Part 3 of 5). Keyword Operands for the BTAM DCB Macro Instruction

Keyword Operand and Description (part 4 of 5)

The following six operands apply only to line groups for World Trade telegraph terminals:

[MON=YES]
[MON=NO]

YES

specifies that each terminal of the line group is equipped with the optional Motor-On feature.

NO

specifies that the terminals are not equipped with the Motor-On feature. NO is assumed if this operand is omitted.

[MONDLY=nn]
[MONDLY=15]

nn

specifies the number of Mark Characters corresponding to a 1.5-second time-out when the terminal is not equipped with the optional Motor-On feature. MONDLY=10 corresponds to 50-baud service, MONDLY=15 corresponds to 75-baud service, and MONDLY=20 corresponds to 100-baud service. When this operand is omitted or nn exceeds 20, MONDLY=15 is assumed.

[IAM=YES]
[IAM=NO]

YES

specifies that the terminal can ask for the computer identification sequence by sending FIGS D.

NO

specifies that the terminal cannot ask for the identification sequence of the computer. NO is assumed if this operand is omitted.

Figure 7 (Part 4 of 5). Keyword Operands for the BTAM DCB Macro Instruction

Keyword Operand and Description (part 5 of 5)

WRU=YES

WRU=NO

YES

specifies that by sending FIGS D, either the computer or the terminal can ask for the identification sequence of the other. When WRU=YES is specified, IAM=YES is assumed.

NO

specifies that the computer cannot ask for the identification sequence of the terminal. NO is assumed if this operand is omitted.

EOM=WRU

EOM=X'hh'

EOM=X'hhlF'

WRU

specifies that the end-of-message signal is the WRU signal.

X'hh'

specifies that FIGS x is used as the EOM signal.¹ hh is the hexadecimal representation of FIGS x set in the adapter.

X'hhlF'

specifies that FIGS y LTRS is used as the EOM signal.¹ hh is the hexadecimal representation of FIGS y set in the adapter. WRU is assumed if this operand is omitted.

EOT=2EOM

EOT=X'hhlF'

2EOM

specifies that two consecutive EOM signals will be recognized by BTAM as end-of-transmission, except when IAM=YES and EOM=WRU are specified.

X'hhlF'

specifies that FIGS y LTRS is used as the EOT signal.¹ Therefore, EOM=X'hhlF' cannot be specified for the EOM signal.

Note: A time-out is also recognized as EOT. Moreover, two consecutive EOM signals are always recognized as an EOT signal, except when IAM=YES and EOM=WRU are specified.

¹x and y are the values assigned by the user and set in the adapter at the time of installation of the equipment.

Figure 7 (Part 5 of 5). Keyword Operands for the BTAM DCB Macro Instruction

Displacement						
Hex	Dec					
10	16	DCBBQFLG	DCBWTEOM	DCBWTEOT	DCBWTPAD	World Trade Telegraph Interface (before and after Open)
14	20	DCBBUFNO	DCBBUFCB			
18	24	DCBBUFL		DCBDSORG		Common Interface
1C	28	DCBDEVTP	DCBIOBAD			
20	32	DCBBFTEK	DCBERROP	DCBBUFCT		Foundation Extension
24	36	DCBEIOBX	DCBEXLST			
28	40	DCBDDNAM				Foundation before Open
2C	44					
30	48	DCBOFLGS	DCBIFLG	DCBMACR		
28	40	DCBTIOT		DCBMACRF		Foundation After Open
2C	44	DCBIFLGS	DCBDEBAD			
30	48	DCBOFLGS				
30	48		DCBREAD, DCBWRITE			BTAM Interface
34	52	DCBLERB				
38	56		DCBXCDE			BSC Interface (before Open)
3C	60	DCBBSTSX				
40	64	(reserved)				
38	56	DCBXMODE	DCBXCDE	DCBBSRSV	DCBBSWBT	
3C	60	DCBBSTSX	DCBBSSTX	DCBBSTEX	DCBBSETX	
40	64	DCBBSAK0		DCBBSAK1		
44	68	DCBBSENQ	DCBBSNAK	DCBBSETB	DCBBSDL	BSC Interface (after Open)
48	72	DCBBSEOT	DCBBSSYN	DCBBSTBE	DCBBSTEB	
4C	76	DCBBSONL		DCBBSSAK		
50	80	DCBBSRVI		(reserved)		
54	84	(reserved)				
60	96	(reserved)				

Figure 8. Format of Data Control Block (DCB)

Field	Contents
DCBBQFLG	World Trade telegraph flag byte.
DCBWTEOM	The EOM character (WT terminals).
DCBWTEOT	The EOT character (WT terminals).
DCBTPAD	Number of pad (LTRS) characters required for motor-on delay (WT terminals).
DCBBUFNO	Number of buffers, obtained during open for this DCB.
DCBBUFCEB	Address of buffer control block.
DCBBUFL	Buffer length (length of buffers to be obtained during open for a BTAM-provided buffer pool or the buffer length to be used if the length parameter of a READ or WRITE macro instruction is coded as 'S' or both).
DCBDSORG	Data set organization (bit 3=1 for BTAM).
DCBDEVTP	Index to the device entry in Device I/O directory.
DCBIOBAD	Input/output block (IOB) address.
DCBBFTEK	Buffering technique (bit 4=1 indicates dynamic buffering).
DCBERROP	Error recovery procedures defined by DCB EROPT operand.
DCBBUFCT	Maximum number of buffers to be obtained by BTAM for a Read or Write operation (dynamic buffering).
DCBEIOBX	Extended IOB index.
DCBEXLST	Address of a user-provided exit list.
DCBDDNAM	DD name of the line group data set.
DCBOFLGS	Flags used during open and checked by programmer to determine whether the data set has been opened.
DCBIFLG	Flags used by the Input/Output Supervisor (IOS).
DCBMACR	Macro instruction reference.
DCBTIOT	Pointer to the DD entry in task I/O table.
DCBMACRF	Same as DCBMACR.
DCBIFLGS	Same as DCBIFLG.
DCBDEBAD	Address of the associated Data Extent Block (DEB).
DCBOFLGS	Same as DCBOFLGS above.
DCBREAD/DCBWRITE	Address of the Read/Write module.
DCBLERB	Address of the line error recording block (LERB).
DCBXMODE	Transmission mode for BSC lines.
DCBXCODE	Transmission code for BSC lines.
DCBBSRSV	The DLE character.
DCBBSWBT	(reserved)
DCBBSTSX	DLE character.
DCBBSSTX	STX character.
DCBBSTEX	DLE character.
DCBBSETX	ETX character.
DCBBSAK0	ACK-0 sequence ¹ .
DCBBSAK1	ACK-1 sequence ² .
DCBBSENQ	ENQ character.
DCBBSNAK	NAK character.
DCBBSETB	ETB character.
DCBBSdle	DLE character.
DCBBS EOT	EOT character.
DCBSSYN	SYN character.
DCBBSTBE	DLE character.
DCBBSTEB	ETB character.
DCBBSONL	SOH % characters.
DCBBSSAK	WACK sequence ³ .
DCBBSRVI	RVI sequence ⁴ .

Hexadecimal representation
of
transmission
code

¹ACK-0 is two characters: DLE X'70' (EBCDIC), DLE 0 (USASCII), or DLE - (TRANSCODE).
²ACK-1 is two characters: DLE / (EBCDIC), DLE 1 (USASCII), or DLE T (TRANSCODE).
³WACK is two characters: X'106B' (EBCDIC), X'103B' (USASCII).
⁴RVI is two characters: X'107C' (EBCDIC), X'103C' (USASCII).

Figure 9. Contents of DCB Fields

DEFINING AND MODIFYING TERMINAL LISTS

A terminal list is a table from which BTAM obtains the information it needs to establish contact with a remote station when you issue a READ Initial or WRITE Initial macro instruction (and occasionally other types of READ and WRITE macro instructions). This information consists of telephone numbers (dial digits), polling and addressing sequences, and identification sequences which can be sent to remote stations or against which an incoming sequence can be checked to ensure that contact has been established with a valid station.

There are several kinds of terminal lists, having different names and formats. For example, a polling list is one kind of terminal list; it is used for supplying the polling sequences BTAM needs to activate certain kinds of remote stations. Another kind is a dial list, used in operations over switched lines. Appendix A shows the formats of terminal lists and gives examples of what they contain.

Two macro instructions, DFTRMLST and CHGNTY, provide the ability to define terminal lists and to modify existing lists.

DFTRMLST (Define Terminal List) Macro Instruction

DFTRMLST generates a terminal list having the format and contents required by the type of station and kind of communication line involved in the Read or Write operation that uses the list. The macro instruction specifies the format and provides telephone numbers, polling or addressing characters, and identification sequences, as required by the Read or Write operation.

Described below are each of the operands that may be coded in a DFTRMLST macro instruction; only a few of these are coded for a particular list. To determine which ones to code for a particular Read or Write operation, see the discussion on terminal lists for the particular type of remote station or line configuration for which the terminal list is required in the sections "Start-Stop Read and Write Operations" and "BSC Read and Write Operations."

A separate DFTRMLST macro instruction must be issued for each list to be defined. Appendix A illustrates the formats of various kinds of lists, with examples.

Notes: The DFTRMLST macro instruction is not used for the local 3270 display system.

For more information about the DFTRMLST macro instruction for the remote 3270 display system, see "Defining Terminal Lists"

under the heading "Defining and Modifying Terminal Lists" in the section "IBM 3270 Information Display System -- Programming Considerations."

Name	Operation	Operand
symbol	DFTRMLST	list type, device-dependent operands

list type
specifies the format of the terminal list. Code one of the following, as required by the remote station or line configuration involved:

OPENLST (start-stop, BSC; multipoint line)
generates an open polling list (for programmed polling of start-stop terminals) or an addressing list (for addressing of start-stop or BSC stations).

WRAPLST (start-stop; multipoint line)
generates a wraparound polling list (for programmed polling).

SSALST (start-stop; multipoint line)
generates an open polling list for Autopoll operations.

SSAWLST (start-stop; multipoint line)
generates a wraparound polling list for Autopoll operations.

DIALST (start-stop, BSC; switched line except for switched 3275)
generates a calling list or an answering list.

IDLST (start-stop [TWX only]; switched line)
generates a calling list or an answering list with ID verification.

BSCLST (BSC; switched line)
generates a calling or an answering list (for switched lines), with ID verification. This kind of list specifies an identification sequence to be sent to a remote BSC station and specifies what identification sequence will be accepted from a remote BSC station.

AUTOLST (BSC; multipoint line)
generates an open polling list for Autopoll operations.

AUTOWLST (BSC; multipoint line)
generates a wraparound polling list for Autopoll operations.

WTTALST (start-stop [WT telegraph only]; nonswitched point-to-point line)

generates a World Trade telegraph terminal list containing the identification sequence expected from a remote station and the identification to be sent to the remote station when transmission begins.

WTLIST (BSC; switched line)

generates a terminal list to be used for Read and Write operations involving manual dialing of a remote station or manual answering of calls from remote stations, where the expanded ID verification facility is not to be used (that is, when only one unique ID sequence is to be accepted from any remote station that calls or is called by the central computer).

SWLST (BSC; switched line)

generates a terminal list to be used for Read and Write operations involving automatic or manual dialing of a remote BSC station or automatic answering of calls from remote BSC stations, where the expanded ID verification facility is to be used (that is, when any of several authorized ID sequences is to be accepted from a remote station).

device-dependent operands

specify the information to be placed in the list. Parenthesized sublists separated by commas are required if more than 255 characters.

xx

two hexadecimal digits representing the transmission bit pattern of a single polling or addressing character. Example: 62 (representing the polling character A in transmission code [1030]).

xxyy

four hexadecimal digits representing the transmission code bit patterns of a two-character polling or addressing sequence. Example: E202 (representing the polling characters A1 in transmission code [1050]).

dialcount

one or two decimal digits representing the number of dial digits in the telephone number of the remote station to be called. Example: 7.

dialchars (numeric only)

the digits of the telephone number to be dialed. Example: 5672022.

numrec

one or two decimal digits representing the number of characters in an identification expected from a remote station.

ridseq

hexadecimal digits representing the transmission code bit patterns of the identification sequence to be received.

numsent

one or two decimal digits representing the number of characters in the identification sequence to be sent to a remote station.

tidseq

hexadecimal digits representing the transmission code bit patterns of the identification sequence to be sent.

numcnsent

one or two decimal digits representing the number of characters in a terminal control sequence to be sent to a TWX station.

cntrlseq

hexadecimal digits representing the transmission code bit patterns of the terminal control sequence to be sent.

length

the number of characters composing a "data tone" (an audible signal to be sent to a remote station that calls the central computer). Code this operand only for lists of the WTLIST type. (A sequence of X'FF' characters is recommended for the data tone.)

area

the address of the area containing the data tone character sequence. Code this operand only if you code the length operand.

faaseq (2760 only)

hexadecimal digits representing the transmission code bit patterns of the three-character frame change sequence (F,A₁,A₂ characters).

Programming Note: The DFTRMLST macro instruction cannot define open or wrap-around lists of the OPENLST or WRAPLST types having more than 31 entries. If a larger list is required, you must define it yourself; see Appendix A for the required format.

CHGNTRY (Change Terminal Entry) Macro Instruction

CHGNTRY is used to cause BTAM to suspend or resume polling or addressing of a specific remote station or component represented by a terminal list entry or to change the value of a control byte in an answering list of the SWLST form. (CHGNTRY cannot be used to change the control byte value of a calling list of the SWLST form.) For a

programmed polling list or an addressing list, CHGNTRY sets the skip bit of the entry to one, if polling or addressing is to be skipped; or sets the bit to zero, if polling or addressing is to be resumed. For an Autopoll polling list, CHGNTRY moves an entry to be skipped to the end of the list, so that all active entries appear at the beginning of the list and all entries to be skipped appear at the end of the list. CHGNTRY moves an entry to be reactivated back to its original position in the list.

You must issue a separate CHGNTRY macro instruction for each list entry you wish to skip or activate or for each SWLST control byte value to be changed.

You can change a terminal list entry only if the list is not currently in use by a Read or Write operation. You should therefore issue CHGNTRY only after making sure that no Read or Write operation is in progress on the line to which the list applies. If you wish to change the list while wraparound polling is in progress, first issue a RESETPL macro instruction to terminate polling, then issue a CHGNTRY macro instruction for the entry to be changed.

CHGNTRY cannot be used to modify a terminal list of the IDLST or BSCLST format.

Note: A special form of the CHGNTRY macro instruction is used for the local 3270 display system. See "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Information Display System -- Programming Considerations."

Name	Operation	Operand
[symbol]	CHGNTRY	listaddr, listype, listposition, numchars, action

listaddr

specifies the address of the first entry of the terminal list containing (1) the entry to be skipped or activated or (2) the SWLST entry the control byte value of which is to be changed.

listype

specifies the type of list, as coded in the DFTRMLST macro instruction that defined the list: OPENLST, WRAPLST, DIALST, SSALST, SSAWLST, AUTOLST, AUTOWLST, or SWLST. (IDLST, BSCLST and WTTALST are invalid operands.)

listposition

specifies the original relative posi-

tion in the list of the entry to be changed. Code a one if the first entry is to be changed, two if the second entry, etc.

numchars

specifies the number of polling or addressing characters in each entry of the list. This operand may be omitted if listype is SSALST, SSAWLST, or SWLST.

action

specifies the action to be performed on the entry:

- For listypes other than SWLST:

SKIP

indicates that polling or addressing is to be suspended.

ACTIVATE

indicates that polling or addressing is to be resumed.

- For a listype of SWLST:

The following operands specify the action to be performed when the ID ENQ sequence is received from a remote station on a Read Connect operation.

ACTIVATE

specifies that BTAM is to send the ID ACK-0 sequence contained in the idsent field of the answering list and then read a message block, if any. ACTIVATE sets the control byte to X'00'.

DISC

specifies that BTAM is to send the disconnect signal (DLE EOT) and then break the line connection. (The two commands that perform this function are part of the Read Connect channel program.) BTAM then restarts the channel program at the Enable command to await a new call. DISC sets the control byte to X'01'.

POST

specifies that BTAM is to post the Read Connect operation complete. The user program then must take the appropriate action. POST sets the control byte to X'02'.

Return Codes: After you issue a CHGNTRY macro instruction with a listype of SSALST, SSAWLST, AUTOLST, or AUTOWLST only, BTAM indicates the result of the operation, by means of a return code in register 15:

<u>Code</u>	<u>Meaning</u>
X'00'	The requested action was performed, or it was already performed (that is, polling or addressing was already suspended or resumed).
X'04'	The requested action was <u>not</u> performed, because the terminal list is in use by a Read or Write operation.
X'08'	The requested action was <u>not</u> performed, because the value coded in the listposition operand exceeded the number of entries in the list (that is, no such entry exists).

This chapter describes how to construct buffer pools, obtain buffers through both programmer buffering and dynamic buffering, and release buffers after use. Dynamic buffering for Read and for Write operations is differentiated.

CONSTRUCTING BUFFER POOLS

If you intend to use buffers for holding input and output messages, a buffer pool must be constructed in one of several ways, as illustrated by Figures 10 through 13. (Only those operands of concern in constructing buffer pools are shown.) The BUILD, GETMAIN and GETPOOL macro instructions mentioned below are fully explained in OS/VS Data Management Services Guide and OS/VS Supervisor Services and Macro Instructions.

Using the BUILD Macro Instruction

First, reserve a storage area during assembly using DC or DS instructions, or issue a GETMAIN macro instruction to obtain the space. The area must begin on a fullword or doubleword boundary and must contain enough space for an eight-byte buffer control block and the number of buffers needed.

Then issue a BUILD macro instruction specifying the number of buffers, their length, and the address of the area reserved during assembly or obtained by the GETMAIN macro (GETMAIN provides the address, in a register, of the area it has obtained). The BUILD macro constructs the buffer control block and the buffer chain.

The length of each buffer must be four bytes longer than the length of the data to be placed in the buffer, because BTAM uses the first four bytes of each buffer as a link field containing the address of the next buffer. If this caution is not observed, the data, when placed in the buffers, may overlay the link field, which destroys the link addresses and thus causes loss of data.

In the DCB macro instruction for each line group that is to use this buffer pool, specify the address of the buffer control block (BUFCB operand).

Figures 10 and 11 show examples that use DS statements and the GETMAIN macro.

Using the GETPOOL Macro Instruction

You may issue a GETPOOL macro instruction either before opening the data control block to be associated with this buffer pool or during the DCB exit routine. In the GETPOOL macro instruction specify the address of the data control block and the number and length of the buffers you need. You must also specify the buffer length in the DCB macro instruction (BUFL operand).

GETPOOL obtains sufficient storage to accommodate the pool, structures the buffer control block and the buffer chain, and places the buffer control block address in the data control block (see Figure 12).

BTAM Construction of Buffer Pools

If you wish BTAM to provide the buffer pool automatically, you simply specify the number of buffers (BUFNO) and their length (BUFL) in the DCB macro instruction for the line group that is to use the buffer pool. During the opening of the data control block, BTAM uses OS/VS data management facilities to obtain storage for the buffer pool and then structures it (see Figure 13).

After a buffer pool has been constructed, you can either request buffers yourself before the Read or Write operation that uses them (programmer buffering), or let BTAM obtain them automatically (dynamic buffering).

PROGRAMMER BUFFERING

To obtain buffers yourself, issue a REQBUF macro instruction, specifying how many you need. Then check the return code in register 15 to determine whether all of the buffers you requested, some of them, or none of them are available to you. If any are available, REQBUF provides, in a register you have designated, the address of the first buffer. Simply specify this address in the READ or WRITE macro instruction. In the case of a WRITE macro instruction, you move the message to be written into the buffers, beginning at the address of the first buffer.

In moving an output message into a buffer chain, remember that each buffer begins with a fullword link field. You must fill each buffer individually, inspecting the link field each time to learn the location of the next buffer. The address of the

```

...
BUILD      BFRPOOL, 20, 100          BUILD BUFFER POOL
...
OPEN      (LINEGP1,,LINEGP2)        OPEN LINE GROUPS
...
ENDJOB    ...
          CLOSE      (LINEGP1,,LINEGP2)  CLOSE LINE GROUPS
          ...
          RETURN
LINEGP1   DCB        BUFL=100,BUFCB=BFRPOOL,...
LINEGP2   DCB        BUFL=100,BUFCB=BFRPOOL,...
BFRPOOL   DS         D                BUFFER CTL BLOCK SPACE
          DS         500F             2000-BYTE BUFFER AREA

```

Figure 10. Constructing Buffer Pools Using DS and BUILD

```

...
USING     IHADCB, DCBREG             ESTABLISH DCB ADDRESSABILITY
...
GETMAIN   R, LV=2008                OBTAIN STORAGE FOR POOL
LR        POOLREG, 1                OBTAIN ADDRESS OF POOL
...
BUILD     (POOLREG), 20, 100        BUILD BUFFER POOL
...
LA        DCBREG, LINEGP1           PLACE ADDRESS OF
ST        POOLREG, DCBBUFCB         BUFFER POOL IN
LA        DCBREG, LINEGP2           LINE GROUP
ST        POOLREG, DCBBUFCB         DCB'S
...
OPEN      (LINEGP1,,LINEGP2)        OPEN LINE GROUPS
...
ENDJOB    ...
          CLOSE      (LINEGP1,,LINEGP2)  CLOSE LINE GROUPS
          LR         1, POOLREG          PROVIDE ADDRESS OF POOL
          FREEMAIN   R, LV=2008, A=(1)   RELEASE STORAGE
          ...
          RETURN
LINEGP1   DCB        BUFNO=10,...
LINEGP2   DCB        BUFNO=10,...
          DCBD       DSORG=BK

```

Figure 11. Constructing Buffer Pools Using GETMAIN and BUILD

```

...
GETPOOL   LINEGP1, 10, 100          BUILD BUFFER POOL
...
GETPOOL   LINEGP2, 8, 120          BUILD BUFFER POOL
...
OPEN      (LINEGP1,,LINEGP2)        OPEN LINE GROUPS
...
ENDJOB    ...
          CLOSE      (LINEGP1,,LINEGP2)  CLOSE LINE GROUPS
          ...
          FREEPOOL   LINEGP1            RELEASE BUFFER POOLS
          FREEPOOL   LINEGP2
          ...
          RETURN
LINEGP1   DCB        BUFL=100,...     SPECIFY BUFFER LENGTH
LINEGP2   DCB        BUFL=120,...

```

Figure 12. Constructing Buffer Pools Using GETPOOL

	...		
	OPEN	(LINEGP1,,LINEGP2)	OPEN LINE GROUP AND BUILD POOLS
	...		
ENDJOB	...		
	CLOSE	(LINEGP1,,LINEGP2)	CLOSE LINE GROUPS
	...		
	RETURN		
LINEGP1	DCB	BUFL=100,BUFNO=10	SPECIFY BUFFER LENGTH
LINEGP2	DCB	BUFL=120,BUFNO=8	

Figure 13. Constructing Buffer Pools Automatically

buffer, plus four bytes, yields the address where the message data should begin.

After you issue a REQBUF macro instruction, the return code in register 15 may indicate that only some of the buffers you requested are available or that none of them are available.

If some buffers are available, they are assigned to you. The address of the first one is in the register you designate, and register 0 indicates how many of the buffers that you requested were unavailable.

The action you take when the full number of buffers is not available depends on your application. Either use the number of buffers supplied (if any) and issue another REQBUF for the remainder; or, issue a RELBUF macro instruction to release the ones supplied to you and reissue the REQBUF for the original number of buffers you requested. If the insufficient-buffer condition occurs infrequently, the cause is probably a momentary peak of activity on several lines at once. In this case, you will most likely obtain the buffers you need the next time you issue the REQBUF macro instruction. On the other hand, frequent recurrence of this condition indicates that you should increase the number of buffers in the pool, as the amount of transmission activity on the lines using the pool exceeds the present capacity of the pool.

DYNAMIC BUFFERING

To be able to use dynamic buffering for a line group, you must specify BFTEK=D in the DCB macro instruction for the line group. Because channel programs differ for dynamic buffering and programmer buffering, and all lines in a line group use the same channel programs, you must use either dynamic buffering or programmer buffering for all lines in the group; you cannot use dynamic buffering for some lines, and programmer buffering for others.

Note: Dynamic buffering cannot be used for the local 3270 display system. If dynamic buffering is specified, the specification is ignored.

For VS2 Release 2, dynamic buffering requires running in real mode (ADDRSPC=REAL on the JOB or EXEC statement). If you try to run in virtual mode with dynamic buffering, abend code 99 is issued.

Read operations and Write operations employ dynamic buffering somewhat differently.

READ OPERATIONS

The first buffer for a Read operation may be obtained in one of two ways: either you supply the buffer yourself, by giving its address in the area operand of the READ macro instruction, or you let BTAM provide the first buffer by coding 'S' as the area operand. BTAM places the address of the first buffer it obtains in the DECAREA field of the DECB for the line. This tells you where the received message begins. Regardless of which method you choose, BTAM automatically obtains all subsequent buffers needed to contain the data being received. If you provide the first buffer yourself, BTAM automatically places the address of the first buffer it provides in the first fullword of your buffer and reads data into your buffer beginning at the second fullword.

An advantage of supplying the first buffer yourself is that it need not be a buffer from the buffer pool; it can be an area you have defined in your program as the place where all incoming messages begin; this affords you the convenience of always beginning your message processing at the same main storage address. Another advantage is that this area can be small compared to the size of your buffers, allowing short messages to be read into this small area rather than into a regular buffer. Improved buffer utilization results, especially when the pool consists of a small number of large buffers.

After each buffer is full, it is posted complete. The first word of each buffer is treated as an event control block (ECB). A completion code is set in the high-order byte of the ECB, and the address of the

next buffer is placed in the three low-order bytes.

The user program may wait for the entire message block to be read by issuing a WAIT macro instruction for the primary ECB, in the same manner as is done without dynamic buffer allocation. Alternatively, the user program may wait for each buffer to be posted complete. This is accomplished by obtaining the address of the first buffer from the DECAREA field of the DECB and using that address as the ECB address in a WAIT macro instruction. After the first wait completes, the user program may obtain the address of the second buffer from the chain address field of the first buffer and issue a WAIT macro instruction for the second buffer. Succeeding buffers are waited for in a similar manner. After each buffer completes, the user program must check for a zero chain address, which indicates that it is the last buffer in the chain.

As the Read operation progresses, BTAM obtains buffers successively until it detects the receipt of an ending character such as ETB, ETX, or EOT. When this occurs, BTAM does not obtain any more buffers. If by the time the ending character is received BTAM has obtained another buffer, BTAM releases that buffer automatically, unless the ending character is in the last byte of the current buffer. In this event, you must release the extra buffer yourself. You can check for this condition in one of two ways.

1. Compare the residual count in the DECCOUNT field against the buffer length in the DCBBUFL field minus four. If count and length-minus-four are equal, the last buffer BTAM obtained for the Read operation is unused. (This method cannot be used if the Read operation includes the Reset function, for example, the Read Initial and Reset (TIR) option.)
2. Test the last byte of the next-to-last buffer for an appropriate ending character. If one is present, the last buffer is unused.

When you detect an unused buffer, release it with a RELBUF macro instruction and place zeros in the low-order three bytes of the high-order word of the next-to-last buffer (that is, the one containing the ending character), to indicate that this buffer is the last one in the chain. If you are waiting for buffers, do not release the unused buffer until it is posted.

In the channel programs for Read operations using dynamic buffering, each Read Text command is followed by a Read Skip

command. When the Read Text command is executed, a program controlled interruption (PCI) occurs. This causes BTAM to obtain another buffer, place its address in the next Read Text command, and change the Read Skip command to a transfer-in-channel (TIC) command pointing to the next Read Text command. When the first buffer is filled, incoming data begins filling the buffer just obtained. The same action occurs as each Read Text command is executed.

The action just described represents the normal case in which BTAM is able to obtain the next buffer in time to receive data from the line. Occasionally, however, BTAM may be unable to obtain the next buffer in time. Should this occur, the Read Skip command following the Read Text command remains unchanged. The Read Skip receives, but does not place in main storage, all data received from the line after the current buffer is full. In this way, the line is cleared of incoming data. The Read operation ends when BTAM detects an ending character, posts the operation as normally completed (X'7F') in the event control block (DECSDECB), and turns on bit 4 of DECFLAGS to indicate that part of the incoming message has been lost in the manner described. By checking this bit after each Read operation using dynamic buffering, you can detect the condition and take appropriate action; normally, you would release the buffers and send a negative response in reply to the message, causing the remote station to resend it.

As indicated under "Programmer Buffering," frequent unavailability of buffers may be caused by a buffer pool that is too small to satisfy the demands made upon it. Increasing the number of buffers should solve the problem.

Caution: If the CPU is stopped while operations involving dynamic buffering are in progress, message data may be lost, as the program-controlled interruptions (PCI) required to obtain successive buffers are not handled when the CPU is stopped.

WRITE OPERATIONS

Whereas in Read operations the main storage locations of individual buffers are unknown to the programmer until BTAM links them into a chain, in Write operations the chain must already have been formed from buffers whose locations and contents are known to the programmer. You must, therefore, always specify in the Write operation the address of the first buffer in the chain whose contents are to be transmitted. As the Write operation progresses, BTAM provides to the operation the address of each of the remaining buffers in the chain.

Normally, you will have obtained the buffers for the Write operation by means of a REQBUF macro instruction; or you will have obtained them dynamically during a preceding Read operation, when you wish to send the same data you received during the Read. A Write operation ends when BTAM detects an ending character or when all the data in the last buffer has been transmitted, whichever occurs first. The length you specify in the WRITE macro instruction must be great enough to encompass the number of characters in the last buffer, including the ending character or character sequence. For write operations using dynamic buffering, the ending character must be in the last buffer in the chain.

After each buffer is transmitted, it is posted complete, in the same manner as for Read operations. The user program may wait either for the entire message block to be transmitted or for each buffer, in the same manner as for Read operations. After the message is successfully transmitted, the buffer chain can be returned to the pool with the RELBUF macro instruction.

BUFFER MANAGEMENT MACRO INSTRUCTIONS

REQBUF (Request Buffer) Macro Instruction

REQBUF is used to obtain one or more buffers from a buffer pool that has been constructed before or during opening of a line group data set.

When you are using programmer buffering, you may issue a REQBUF macro instruction to obtain one or more buffers in which data can be received from a line (Read operations) or in which to build or move an output message (Write operations).

When you are using dynamic buffering, BTAM automatically obtains buffers for Read operations, so you do not issue a REQBUF macro instruction to obtain them. For Write operations, however, use of REQBUF is the same as for programmer buffering.

The buffers provided are not necessarily in consecutive storage locations. They are chained together, the link field of each containing the address of the next. The link field of the last buffer in the chain contains zeros.

Name	Operation	Operand
{symbol}	REQBUF	dcbaddr, returnreg, {count}

dcbaddr
specifies the address of the data control block with which the buffer pool is associated.

returnreg
specifies a general register (2 through 12) into which you wish BTAM to return the address of the first buffer to be provided.

count
specifies the number of buffers you are requesting.

If you specify one of the registers 2 through 12, you must previously have loaded the count into the low-order byte of that register; the high-order bytes are ignored.

If you specify register 0, you must previously have loaded the count into the high-order byte of the register; the low-order bytes must contain zero.

If you omit this operand, BTAM provides one buffer, that is, the link field contains zero.

Return Codes: After you issue a REQBUF macro instruction, the low-order byte of register 15 contains a return code indicating the result of the buffer request. (The three high-order bytes of the register contain zeros.) The return code, in hexadecimal notation, is one of the following:

- 00 Normal return. BTAM has provided the total number of buffers you requested. The return register contains the address of the first one.
- 04 Partial fulfillment of request. You requested more buffers than are currently available in the pool. All those available were provided. The return register contains the address of the first one.
- 08 No buffers available. The buffer pool had been exhausted at the moment of your request. The return register and register 0 contain zero.
- 0C No buffer pool. The request cannot be filled, because no buffer pool is associated with the data control block you have specified in the macro instruction.
- 10 No buffer routine. The request cannot be filled because the BTAM buffer management routine has not been included in your program. (The routine is automatically included if you have specified BFTEK=D in the DCB macro or if the data control block

contains the address of a buffer control block.)

Programming Notes: If the buffer request was partially filled (return code is 04), the low-order byte of register 0 contains the count of the number of buffers not provided. (The three high-order bytes contain zeros.)

When the REQBUF macro instruction is used for the local 3270 display system, an entire message must fit into one buffer.

RELBUF (Release Buffer) Macro Instruction

RELBUF is used to return to the buffer pool one or more buffers obtained by a REQBUF macro instruction or automatically during dynamic buffering. Failure to issue this macro instruction following Read and Write operations for which buffers have been obtained will ultimately result in exhaustion of the buffer pool. RELBUF releases each buffer in the chain, beginning with the one whose address you specify and ending with the one whose link field contains zero (that is, the last buffer).

Name	Operation	Operand
[symbol]	RELBUF	dcbaddr,bufferaddr

dcbaddr

specifies the address of the data control block associated with the buffer pool to which the buffers are to be released.

bufferaddr

specifies a general register (2 through 12) into which you must pre-

viously have placed the address of the first buffer to be released.

Return Codes: After you issue a RELBUF macro instruction, the low-order byte of register 15 contains a return code indicating the result of the operation. (The three high-order bytes of the register contain zeros.) The return code, in hexadecimal notation is one of the following:

- 00 Normal return: The specified buffers have been returned to the pool.
- 04 Already returned: The first buffer of the chain to be released has already been returned to the pool (or has never been obtained from the pool).
- 0C No buffer pool: The buffer release cannot be accomplished because no buffer pool is associated with the data control blocks you have specified in the macro instruction.
- 10 No buffer routine: The buffer release cannot be effected, because the BTAM buffer management routine has not been included in your program.

Programming Note: If you wish to release a different number of buffers than you obtained by a REQBUF macro or by dynamic buffering (assuming the first buffer to be released is the same as the first buffer that was obtained), you will have to place zeros in the link field of the last buffer you wish returned. Be sure to retain the address of the buffer that follows the last one you return, since it will become the first of the remaining buffers in the original chain.

As pointed out in the first chapter in the discussion of how information is represented in various parts of a teleprocessing system, it is the programmer's responsibility to perform code conversion between transmission code and the internal code of the central computer, if the application requires it.

BTAM provides a translation routine and a set of translation tables that convert between EBCDIC and the transmission code or codes employed by the types of remote stations supported by BTAM. Some terminal types can be furnished with any of several character sets; BTAM provides translation tables for the more common sets. (In most cases the sets vary by only a few characters.) When a remote station in your configuration uses a character set not directly supported by a BTAM-provided translation table, you can easily modify an existing table to accommodate that station. Alternatively, you can define an entirely new table (but you must not give it the same name as a BTAM-provided table). You must format any table you define according to the requirements of the System/370 translate (TR) instruction.

If you wish to refer to a BTAM-provided translation table after assembling it into your program (for example, to modify the table by means of an MVC instruction or to use it in conjunction with the TR instruction), you must refer to the table by the name IECTxxxx, where the x's represent the four-character table name as shown in Figure 14. In referring to the table with the TRNSLATE macro instruction, however, you need specify only the four-character table name. Figure 14 lists the translation tables provided by BTAM.

At the end of this publication are two sets of code tables. Appendix H is a code correspondence chart that shows for each of the 256 EBCDIC bit patterns the corresponding character (and its transmission code bit pattern) to or from which the BTAM-provided translation tables convert the EBCDIC character. Full understanding of this chart requires that you read the explanatory material preceding it.

Appendix I shows for each of the 256 possible bit patterns in a System/370 byte the character represented by that pattern in each of the transmission codes and in EBCDIC. This chart is useful in interpreting the contents of main storage locations.

ASMTRTAB (Assemble Translation Table) Macro Instruction

ASMTRTAB assembles into a program one or more BTAM-provided translation tables. You may code all table names in one ASMTRTAB, and you need code only one ASMTRTAB regardless of the number of lines and line groups for which the table is needed. Code the macro instruction among the program constants, not in the middle of executable code.

Note: The ASMTRTAB macro instruction is not used for the local 3270 display system.

Name	Operation	Operand
(Omit)	ASMTRTAB	tablename,...

tablename

specifies the BTAM-provided translation table or the table you wish to assemble into your program. Code any table name listed in Figure 14. Table names may be coded in any sequence.

Example: If you wish to perform code translation between EBCDIC and 1030 code, and between EBCDIC and TRANSCODE, code:

```
ASMTRTAB RC30,SD30,RC80,SD80
```

TRNSLATE Macro Instruction

TRNSLATE translates data in main storage from transmission code to EBCDIC (for received data) or from EBCDIC to transmission code (for data to be transmitted). Code TRNSLATE at each point in your program where translation is required.

Note: The TRNSLATE macro instruction is not used to translate between transmission code and EBCDIC for the local 3270 display system.

Name	Operation	Operand
[symbol]	TRNSLATE	[dcbaddr], tablename, area, length

dcbaddr

specifies the address of the data control block for the line group. This operand is required if you code the length operand as 'S'; otherwise, it may be omitted.

tablename
 specifies the four-character name (for example, RC50) of the BTAM-provided translation table to be used, or the name of your own translation table. (You must have previously assembled the indicated table into your program.)

buffers must be formatted and chained just like BTAM-provided buffers: the first fullword of each buffer contains the address of the next buffer (except that the first fullword of the last buffer contains zeros). Each buffer must begin on a fullword boundary.

area
 specifies the address of the storage area in which the data to be translated is located. If dynamic buffering is used for the line group involved, the address specified by area must be a fullword boundary. If you use TRNSLATE to translate a chain of buffers you have defined, those

length
 specifies the number of bytes to be translated, from 1 to 32,767. If you wish to translate the contents of a chain of buffers, code 'S' as the length operand. This causes the translate routine to use the buffer length given in the data control block.

Type of Remote Station	Transmission Code	Table Name
For incoming messages: (Translation from transmission code to EBCDIC):		
IBM 1030	EBCD/PTTC	RC30
IBM 1050	EBCD/PTTC	RC50 RF50*
IBM 1060	BCD/PTTC	RC60
IBM 2260	USASCII	RSCI
IBM 2740,2741	BCD/PTTC	RB40 RU40*
	EBCD/PTTC	RC40 RF40*
	Correspondence code	RC41 RF41*
IBM S/370	USASCII	RASA
IBM System/3	USASCII	RASA
IBM 2770	USASCII	RASA
IBM 2780	{USASCII Six-bit Transcode	RASA RC80
Remote IBM 3270	USASCII	RASA
AT&T 83B3, WU 115A	Baudot code	RCT1
WU TWX (Models 33,35)	TWX Code	RCT2
World Trade Telegraph Terminals	{ZSC3 code ITA2 code	RCT3 RCTW

Figure 14 (Part 1 of 2). Code Translation Tables Provided by BTAM

Type of Remote Station	Transmission Code	Table Name
For outgoing messages (translation from EBCDIC to transmission code):		
IBM 1030	EBCD/PTTC	SD30
IBM 1050	EBCD/PTTC	SD50
IBM 1060	BCD/PTTC	SD60
IBM 2260	USASCII	SSCI
IBM 2740,2741	{ BCD/PTTC	SB40
	{ EBCD/PTTC	SD40
	{ Correspondence code	SD41
IBM S/370	USASCII	SASA
IBM System/3	USASCII	SASA
IBM 2770	USASCII	SASA
IBM 2780	USASCII	SASA
	Six-bit Transcode	SD80
Remote IBM 3270	USASCII	SASA
AT&T 83B3, WU 115A	Baudot code	SCT1
WU TWX (Models 33,35)	TWX code (even-parity)	SCT2
World Trade Telegraph Terminals	{ ZSC3	SCT3
	{ ITA2	SCTW
<p>1. Translation tables marked * convert both uppercase and lowercase alphabetic characters to uppercase EBCDIC equivalents (for example, both A and a are converted to A); tables not so marked convert uppercase to uppercase and lowercase to lowercase (for example, A to A and a to a).</p> <p>2. Transmission code abbreviations used above: BCD = binary coded decimal EBCD = extended binary coded decimal PTTC = perforated tape and transmission code USASCII = USA Standard Code for Information Interchange ZSC3 = Figure Protected Code ITA2 = International Telegraph Alphabet Number 2</p> <p>3. See "General Notes" in Appendix H for discussion of TWX Code parity.</p>		

Figure 14 (Part 2 of 2). Code Translation Tables Provided by BTAM

ACTIVATING AND DEACTIVATING THE TELEPROCESSING SYSTEM

The operations performed by a user's teleprocessing program preparatory to data transmission is called activating the system. Similarly, deactivating the system refers to the operations performed after all transmission has ceased. These operations largely consist of opening (activating) and closing (deactivating) the communications line group data sets.

PROGRAM INITIALIZATION

Before activating the teleprocessing system you must first perform the usual initialization steps required of any program that runs under OS/V.S. These are as follows:

1. Using a SAVE macro instruction (or a Store Multiple instruction), store the contents of the general registers you will use in your program in a register save area, the address of which is in register 13 upon entry to your program.
2. Store the contents of register 13 in the second fullword of a save area you have defined in your program.
3. Load the address of your program's save area into register 13. (Save areas are required by most system macro instructions.) Unless you require register 13 for other purposes, you need to load it only at the beginning of your program.

See Figure 15 for an example of the foregoing linkage. More detailed information on the use and format of register save areas and on linkage conventions is contained in OS/V.S Supervisor Services and Macro Instructions and OS/V.S Data Management Services Guide.

In addition to these initialization steps, you should create dummy control sections (DSECTs) for the data control blocks and data event control blocks in your program, to allow you to refer symbolically to fields in these control blocks. To create these DSECTs, use the DCBD and IECTDECB macro instructions as shown in Figure 16. If the teleprocessing system includes BSC stations, a second operand, DEVD=BS, must appear in the DCBD macro; similarly, if the system includes World Trade telegraph terminals, code the second operand as DEVD=WT. If the system includes both kinds of stations, code DEVD=(BS,WT).

Code the DCBD and IECTDECB macros at the end of the control section (CSECT) in which they appear.

OPENING AND CLOSING LINE GROUP DATA SETS

Before you can perform data transmission operations over a line, you must open, or activate, the line group data set encompassing that line, by means of an OPEN macro instruction. When you issue an OPEN macro instruction, an open routine establishes and initializes various internal control blocks and loads from the system

YOURPROG	CSECT		
*	SAVE	(14,12)	1. SAVE REGISTERS IN CALLING PROGRAM'S SAVE AREA
	LR	BASEREG,15	
	USING	YOURPROG,BASEREG	
	ST	SAVEREG,SAVEAREA+4	2. SAVE REG 13 IN 2ND FULLWORD
	LA	SAVEREG,SAVEAREA	3. LOAD YOURPROG SAVEAREA ADDRESS
*			
BASEREG	EQU	12	
SAVEREG	EQU	13	
	.		
	.		
BEGIN	EQU	*	
	.		
	.		
SAVEAREA	DS	18F	

Figure 15. Initializing Your Program


```

YOURPROG CSECT
.
.
USING    IHADCB,DCBREG
USING    IECTDECB,DECBREG
.
.
DCBD     DSORG=BX
IECTDECB

```

Figure 16. Establishing Addressability for DCBs and DECBS

library those routines and tables needed for BTAM to construct the channel programs required by subsequent READ and WRITE macro instructions. The Open routine also "conditions" the communications line adapters within each TCU associated with the line group. Conditioning a line adapter makes the line attached to it ready for data transmission.

The fact that you have issued an OPEN macro instruction does not guarantee that the line group is open. The DCB for the line group has a bit, called the Open flag, that you can check to determine whether the line group is open. The Open flag is bit 3 of the DCBOFLGS field; if it equals one, the line group is open.

If after you issue the OPEN macro instruction, the Open flag is still zero, there is probably a coding error; most likely, the DD (data definition) card for the line group contains the wrong line group name.

Even if the open flag is 1, one or more lines in the line group may not be ready for transmission because: the local control unit was powered off; a device attached to the control unit was inoperable; or the line adapter was not successfully conditioned.

If this occurs because the TCU power is off or if the TCU is offline (remotes only); or the 3270 CU or device is inoperable, powered off or is offline the operating system prints, on the console, error message IEC804A, and enters wait state. This message identifies the condition and requests a response from the console operator.

He replies; CONT. to retry this I/O operation possibly after correcting the error condition; POST to proceed with OPEN processing; DROP (Local 3270 ONLY) to proceed with OPEN processing without issuing the message for other devices in the line group.

The DROP or POST option will allow the user application to proceed with operations on unaffected lines. IF unsuccessful

conditioning occurs for some other reason (for example, TCU malfunction), the fact that the line has not been opened becomes evident when the first READ or WRITE macro instruction issued for that line results in a return code of X'14'. For this reason, the user program should check for this return code after the first READ or WRITE macro instruction following opening of the line.

BTAM provides the LOPEN (Line Open) macro instruction for use in opening (that is, conditioning the line adapter for) a single line in a line group. LOPEN is intended for use following a return code of X'14'.

Depending on your application, you may wish to open all line groups at once or to open different groups at successive intervals during the day. Opening line groups at different times would be appropriate, for instance, when the remote stations connected to one group are located in a different time zone from those connected to another group.

After completion of data transmission over all lines in a line group, you may close the line group by means of a CLOSE macro instruction. If BTAM provided a buffer pool during opening of the line group (see the section "Buffer Management"), you must issue the CLOSE macro instruction only after you have no further use for the contents of any of the buffers in that pool. This is necessary because when you close the line group, BTAM relinquishes the main storage area occupied by the pool and various pointers to buffers no longer exist.

OPEN Macro Instruction

OPEN completes the initialization of the data control block representing the line group data set, builds a buffer pool if you specify in the DCB macro that this be done, and loads from the system library those routines and tables necessary for BTAM to construct the appropriate channel programs. As explained earlier, the Open routine also conditions each TCU line adapter connected to a line in the group.

See Figure 17 for the format of the OPEN macro instruction.

A single OPEN macro instruction can activate any number of line groups and any other data sets defined in your program, including those for other access methods.

Example: To open two line group data sets and three BSAM data sets (one on magnetic tape, two on direct access devices), you could code a single OPEN macro instruction as follows:

OPEN1 OPEN (LG1050,,LG2740,,TAPELOG,
(OUTPUT),MSGFILE1,
(INOUT,LEAVE),MSGFILE2,
(OUTPUT))

LG1050 and LG2740 are the two line group data sets; the second comma following each of these operands indicates the absence of volume-positioning option parameters, which are not appropriate for communications line groups. The remaining operands are representative of data set addresses and volume-positioning options for the three BSAM data sets. (See OS/VS Data Management Macro Instructions for information on coding OPEN macro instructions for nonline-group data sets.)

No return code is provided following an OPEN macro instruction; as explained earlier, you should check the Open flag in the DCB to see whether the line group was successfully opened.

LOPEN Macro Instruction

LOPEN causes BTAM to issue commands that condition the TCU line adapter for a specific line, when conditioning of the adapter was not successful during opening of the line group. It is appropriate to issue LOPEN after receiving a return code X'14' following issuance of a READ or WRITE macro instruction for the line. LOPEN causes the appropriate command (Set Address, Set Mode, or Enable) to be sent to the line adapter.

(LOPEN may also be used to reestablish data set synchronism for a line using an IBM 3977 Model 2 modem (data set), as follows. When the modem loses synchronism, transmission errors (that is, a NAK response from the remote station or a time-out error) occurs during Write operations. When errors of these kinds occur, it is appropriate to issue an LOPEN macro instruction, which, by disabling the line and then enabling it or setting the mode, causes the modem to regain synchronism.)

Name	Operation	Operand
[[symbol]]	LOPEN	decbaddr

decbaddr

specifies the address of the data event control block associated with the line.

Programming Note: You should not issue an LOPEN macro instruction from within a timer exit, since LOPEN uses the STIMER macro.

Return Codes: Upon return of control to your program, the low-order byte of register 15 contains a return code. Normal completion is indicated by X'00'. Abnormal completion is indicated by the following hexadecimal codes:

- 04 The line was not successfully opened.
- 08 The specified line is busy.
- 0C The relative line number specified in the data event control block is larger than the number of lines in the line group.
- 10 The DCB for the line group is not open.
- 14 The request was rejected, because OLTEP was using the local 3270 device.
- 20 Local device is not 3270 device type

Only the abnormal return codes, X'04' and X'14', can be encountered in a debugged user program; the other four result from program errors. If a code of X'04' is returned after you issue an LOPEN macro instruction, you may wish to notify the console operator that he should check the condition of the affected TCU.

CLOSE Macro Instruction

CLOSE terminates the availability of a line group data set, frees the storage space occupied by the buffer pool if the pool was constructed by the Open routine, and frees the storage space obtained by the Open routine for control blocks. CLOSE also causes the fields in the data control blocks to be restored to the condition they were in before the DCB was opened. Just as OPEN causes the TCU line adapters associated with the line group to be conditioned for use, CLOSE cancels the conditioning. For this reason, if you issue a CLOSE macro instruction while data transfer is still in progress over one or more lines in the line group, unpredictable loss of data can result. You should therefore close the line group only after all message traffic has ceased. See Figure 17 for the format of the CLOSE macro.

A single CLOSE macro instruction can deactivate any number of line groups and any other data sets defined in your program (including those for other access methods), in the same way an OPEN macro instruction can activate them.

Name	Operation	Operand
[symbol]	{ OPEN CLOSE }	{{dcb,,}...}, [MF=L MF=(E,listname)]
symbol	specifies:	<ul style="list-style-type: none"> • For standard or execute macro instruction format, the name of the first instruction generated by the macro instruction. For these formats the use of symbol is optional. • For list format, the name of the parameter list created by the macro. For this format, you must specify a name.
dcb	specifies the name of the line group data set you wish to open or close.	
MF=L	(List format)	specifies that a parameter list is to be created, containing the names of the data control blocks to be opened or closed. The function is not performed until you issue an OPEN or CLOSE macro instruction of the execute format specifying the name of the parameter list.
MF=(E,listname)	(Execute format)	specifies that the open or close function is to be executed for the data sets contained in the parameter list specified by listname. You must previously have created the list with an OPEN or CLOSE macro instruction of the list format (MF=L). If you wish to override certain parameters in the list, specify replacement parameters in the macro instruction having the execute format. Code the replacement parameters in the positions corresponding to the locations of the parameters to be overridden.
(Standard format - MF operand omitted)		specifies that both (1) a parameter list is to be created, containing the names of the data control blocks to be opened or closed, and (2) the open or close function is to be executed for the data sets contained in the created parameter list.
<u>Example:</u>		<pre> OPENLIST OPEN (LG1050,,LG2740,,LG1130),MF=L . . . OPEN (,,LG2260,,),MF=(E,OPENLIST) </pre>
		The first macro creates a list; the second executes the Open function for data sets LG1050, LG2260, and LG1130.
		After you have defined a parameter list by either an OPEN or a CLOSE macro instruction of the list or standard format, you may subsequently specify that list by both OPEN and CLOSE macro instructions of the execute format.

Figure 17. Formats of OPEN and CLOSE Macro Instructions

LINE CONTROL

Communication between the central computer and remote stations requires a discipline called line control, as mentioned earlier in this publication. Given here is a summary of the control scheme used for various line configurations and types of remote stations.

Line control does not apply to the local 3270 display system, which uses attention interruptions to regulate communications between the central computer and local display stations. For more information, see "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Information Display System -- Programming Considerations."

Contention System

In the most elementary form of line control each of the two stations at the ends of a point-to-point communications line gains use of the line by sending to the other station a special control character signifying the station's intention to begin transmission. The first station to initiate contact in this manner "seizes" the line and prevents its use by the other station until the first station has concluded its message transmission. If both stations should simultaneously try to initiate transmission, they are said to be contending for use of the line, hence the name contention system. In this kind of system some method is required for resolving a contention situation.

The action of requesting use of the line is sometimes called bidding for the line.

Centrally-Controlled System

In this kind of system, the central computer acts as a control station. That is, it initiates all contacts between all stations on a multistation (multipoint) line. It does this by periodically sending on the line a series of station identifiers, called polling characters or polling sequences. Each station on the line has a different polling sequence. Thus, although all stations receive all polling sequences, each station responds only to its own. This response indicates to the control station (the central computer) whether or not that remote station is ready at that moment to send a message. It sends a positive response if it is ready, a negative response if it is not. For some types of

stations the polling sequence identifies a specific component of the station, as well as the station itself. In this case, the response indicates whether or not that particular component is ready to send a message.

Similarly, when the control station wishes to send a message to a remote station, it transmits an identifier sequence on the line. This is called addressing, or selection. Again, all stations receive the addressing characters, but only one responds. The addressed station returns to the computer a positive response if the station (and perhaps a specific component) is ready to receive a message.

In a system of this kind, the stations can be in one of two modes: control mode and text mode. The stations are all in control mode before a transmission begins, and in this mode they monitor the line for polling and addressing sequences. When a polled or addressed station responds positively, message transmission between the central computer and the remote station can begin. At this point, it is necessary to place all stations in text mode, so that any characters received by any station except the polled or addressed station are ignored. (If the other stations remained in control mode, any sequence of message characters that happened to constitute a polling or addressing sequence for one of the stations would activate that station.) Accordingly, each message begins with a special control character whose purpose is to cause the stations to enter text mode. Two characters used for this purpose are EOA (end-of-address) and STX (start-of-text). The type of station on the line determines which character is used.

At the end of a transmission, all stations on the line must be returned to control mode, so that they can again respond to polling and addressing sequences. Another character or character sequence, called end-of-transmission (EOT), performs this function.

The function of returning the stations to control mode is often called resetting the line.

Switched Systems

In a switched system, contact must be established by one or the other of two stations: the central computer or the remote station. In some switched configurations,

either the computer or a remote station can call the other station; in others, only one or the other of these can make the call. User requirements determine which case applies.

When the computer initiates contact with the remote station, it performs the calling function, when it answers a call from a remote station, it performs the answering function.

Although a remote station can call the central computer at any time, the computer, to fulfill its function as control station, must be able to accept or reject a call. If it wishes to accept calls, it "enables the line," that is, conditions the TCU to respond to calls over the given switched line termination. The user program determines which lines are to be enabled at any given moment. Conversely, to return the TCU to the state in which it will not respond to (that is, answer) calls is called disabling the line.

If a remote station calls in on a line that is not currently enabled or that is enabled but is occupied with another remote station, the calling station receives a busy signal, and contact is not established. The station must try again later.

After the line connection is established, one of the preceding line control schemes, contention or centrally controlled, takes effect just as on a non-switched line. The scheme used is the same as that used for a nonswitched line for the particular type of stations involved.

ERROR DETECTION AND MESSAGE BLOCKING

Line control may also involve detection of transmission errors. For the types of remote stations for which this is possible, a character called end-of-transmission-block (ETB), also called end-of-block (EOB), is sent following a sequence of text characters; this sequence is then called a message block. Whenever the sending station senses an ETB in the data it is sending, it follows that ETB with a check accumulation (VRC, LRC, or cyclic) and awaits a response from the receiving station. The receiving station compares the check character with the check character it has accumulated. If they match, indicating that it received the text without error, it sends a positive response (or acknowledgment) to the sending station. If they do not match, indicating that a transmission error has occurred, it sends a negative response (acknowledgment) to the sending station. A positive response indicates that the sending station may continue with the next mes-

sage block; a negative response tells it to resend the erroneous block.

CHANNEL PROGRAMS

The various line control functions are achieved by the central computer through a combination of equipment and programming. Generally, each discrete function, such as enabling or disabling the line and reading and writing message text and responses is effected by separate channel commands that, when combined in appropriate sequences in a channel program, perform the overall line control actions needed to establish contact, transmit messages and check for errors. Channel programs are generated by BTAM as directed by the READ and WRITE macro instructions issued in the user program.

MESSAGE TRANSMISSION

All message transmission is effected by Read and Write operations of various kinds, which in turn are produced by coding equivalently named macro instructions in the user program. (For information about Read and Write operations for the local 3270 display system, see the section "IBM 3270 Information Display System -- Programming Considerations.") Thus a Read Initial operation is produced by a READ Initial macro instruction. These operations are as follows.

A Read or Write Initial operation establishes contact with the remote station and receives or sends the first message block. In establishing contact, the operation performs whatever functions are appropriate. That is, for a nonswitched line in a contention system, Read Initial first sends the character that signifies to the receiving station that the line is being seized by the sending station. In a centrally controlled system, the first function is to send a character or sequence that places all stations in control mode, as explained earlier. For a switched line, Read Initial either enables the line, if the operation is to continue when a remote station calls in, or it dials the remote station.

Following execution of whichever of the foregoing functions is appropriate, polling may take place, if required by the type of station involved. Then the first block of the message is read or written.

After a Read or Write Initial operation has concluded, you generally issue as many READ or WRITE Continue macro instructions as necessary to receive or send the remaining blocks of the message.

If a Read operation receives an erroneous message block, you may undertake Read Repeat operation; the negative response sent by Read Repeat signifies to the remote terminal operator or to the remote computer program that he or it should resend the block in error.

Sometimes it is desirable to reverse the direction of message transmission during one transaction or to exchange the roles of the receiving and sending stations. Read and Write Conversational operations permit this.

In binary synchronous communications, it is sometimes desirable to send data in transparent mode. This means that any transmission code bit pattern can be sent as data; whereas in normal transmission certain patterns are recognized and responded to as line control characters. Read and Write transparent operations are available for this purpose.

These various operations can be combined in several ways. Inspection of the Read and Write operations for a specific type of remote station and line configuration will illustrate some of these ways.

Although in coding a user program, it is not usually necessary to understand all the details of the various commands that make up a channel program, each command is explained fully in the publications pertaining to TCUs. These publications are listed in the Preface of this manual.

User Program Analysis

Upon completion of each Read or Write operation, the user program must analyze the results of the operation to determine which Read or Write operation to perform next. If the operation was successful and either message text or some expected response was received, the decision about the next operation depends largely on the kind of application. Sometimes, it may depend on the content of the received text. For example, in an application that involves transmission of fairly long messages, it is common practice to break the message into sequences of message blocks. It is then appropriate to send or receive the first block using a WRITE or READ Initial macro instruction or one of the variants, such as WRITE Initial Transparent for BSC, and then send or receive the rest of the blocks with WRITE or READ Continue macros or variants.

An operation may end successfully, but with some exceptional condition. For example, a sequence of Read operations will end when a remote station sends an EOT after having sent a number of blocks of

text. Since the user program probably does not know when to expect the last block of text, if message lengths vary, it should check after every Read operation for receipt of an EOT, which is considered an exceptional condition.

Some operations will end unsuccessfully, with an error condition of some kind, such as a parity error (data checks) in text or an invalid response. Again, the user program must analyze the results of each Read or Write operation to see if an error condition has occurred.

BTAM provides error recovery procedures (ERP) for automatically attempting to recover from errors. These are optional for start-stop lines, mandatory for BSC lines. It is only after BTAM ERP has attempted recovery and failed that the error condition is indicated to the user program. If ERP is successful in clearing the condition, BTAM posts the operation complete-without-error, and the user program is unaware that the error has occurred.

The section "Error Recovery Procedures and Error Recording," discusses the BTAM ERP facilities and suggested user analysis procedures.

Use of Line Control Characters

To achieve successful communication with any given type of remote station requires that the data stream between the central computer and the remote station contain the appropriate line control (also called data link control) characters and character sequences. A BTAM programmer must be concerned with the proper use of these characters. In message data received from a remote station, you may need to scan the input areas to determine the locations of control characters and perhaps to remove them. In message data to be sent to a remote station, however, you must assure yourself that these characters are sent at the appropriate point in the transmission. Some control characters are sent automatically by BTAM in a separate command within a channel program. Others you must place in the message output area. For example, when using a transparent-type Write operation to send data in transparent mode (that is, to prevent the control units at the central computer and remote station from reacting to bit patterns that correspond to line control characters), you must place the DLE STX character sequence in the output area at the point where transparent transmission is to begin. You do not, however, place the ending sequence, DLE ETX (or DLE ETB), in the output area, because, as inspection of the channel program shows, the command following the Write Text command sends these characters.

It is most important to be familiar with the usage of line control characters for the type of remote station for which you are coding Read and Write operations. The line control characters and their proper usage are defined in the publications pertaining to the various types of stations, and, in the case of binary synchronous communications, in the general information publication for BSC. (These publications are listed in the Preface of this manual.) Line control character usage may vary depending on particular features or combination of features with which the stations are equipped.

READ AND WRITE OPERATIONS

The next two chapters of this publication contain descriptions of the READ and WRITE macro instruction options available for each of the types of remote stations with which the central computer can communicate under BTAM control. In the section "Start-Stop Read and Write Operations," these descriptions are arranged by type of station. In the section "BSC Read and Write Operations," they are arranged by type of line configuration. This is done because the channel program for each type of operation is the same for any type of station (for a given line configuration).

The Reset Function

For many of the Read and Write operations listed there is an optional reset function. This simply means that if the Read or Write operation has progressed satisfactorily up to that point (that is, message text was received or sent without error), one or two additional commands are executed that reset the station to control mode, and, for switched line operations, that break the line connection. This is the only difference between a reset and a nonreset operation, and for this reason is not stated explicitly in each description. The reset function is not performed if a permanent error occurred during the operation.

Terminal Lists

The description of the DFTRMLST macro instruction earlier in this manual explains all of the operands of that macro instruction. In the next two chapters, each section covering a type of remote station or a line configuration indicates which type of terminal list you must define for Read and Write operations and shows what operands to code in the DFTRMLST operand field to obtain that list. See the explanation of the DFTRMLST macro instruction for the meanings of the operands, and see Appendix A for format illustrations and examples.

Data Event Control Block

The parameters BTAM needs to perform a Read or Write operation are contained in a data event control block (DECB). Some of these parameters are:

- The type of Read or Write operation (for example, Initial, Continue Conversational)
- The address of the data control block (DCB) for the line group encompassing the line over which the operation is to take place
- The relative line number of the line involved
- The address of the terminal list entry containing the information necessary to establish contact with the remote station
- The addresses of the input or output areas to contain the message text.

The DECB also contains fields in which the results of the Read or Write operation are indicated. Among these fields are:

- An event control block (ECB), in which a standard completion code is placed upon conclusion of the Read or Write operation.
- A response field (DECRESPN), which receives addressing responses from the remote station.
- Fields containing specific indicators of the results of the operations: DECSNS0 (sense information); DECFLAGS (condition flags); DECERRST (error status) and DECCSWST (channel status word status byte).

The format of the DECB and the contents of its fields are given in Appendix B.

One DECB is required for each communications line; more than one can be provided, if desired.

DECBS are created by READ and WRITE macro instructions as follows. A macro of the list form (specified by the keyword operand MF=L) reserves space for a DECB and fills in certain of its fields with the parameters provided by the macro. This is done during assembly, and is the sole function of the list form macro instruction; that is, the macro instruction does not perform a Read or Write operation. If you define a DECB in this way, you must code the macro instruction among the program constants (or create your own linkage around it), since a macro instruction of

the list form does not generate executable code.

In order to perform a Read or Write operation using a DECB created by the list form of the macro instruction, you issue a READ or WRITE macro instruction of the execute form, specified by the MF=E keyword operand. This form of macro instruction does not establish a DECB; it executes the Read or Write operation using an existing list. In this macro instruction you may specify which, if any, of the parameters in the original DECB you wish to change. For example, if you wish to issue a series of WRITE macro instructions, all of which require the same DECB parameters except for the entry parameter, it would be appropriate to issue one WRITE (or READ) macro instruction of the list form to establish the DECB. Then you would code the other WRITE macro instructions in the execute form and in each one specify only the entry operand, of those operands that are optional.

An alternate method is to code the standard form of the READ or WRITE macro instruction, by omitting the MF keyword operand. A macro instruction of this type generates both a DECB and the executable code required to perform the Read or Write operation.

Just as you issue a macro instruction of the execute form referring to a DECB defined by a macro instruction of the list form, you may issue an execute-form macro instruction that refers to a DECB generated by a previous macro instruction of the standard form.

An important point to remember in using the same DECB for a sequence of Read or Write operations is that the contents of many of the fields change with each issuance of a macro instruction or execution of a Read or Write operation. This means that at the conclusion of each Read or Write operation you should do whatever checking of DECB fields is necessary before you issue the next macro instruction that will refer to the same DECB.

Not all of the READ and WRITE macro instruction operands are optional. Regardless of the macro instruction form, you must provide the address of the DECB and the operation type. In the standard form, you must always code the DCB address and the relative line number.

READ and WRITE Macro Instructions

READ and WRITE macro instructions produce the Read and Write operations that achieve message transmission. You issue one of these macro instructions each time you wish to receive a message from a remote station, send a message to a remote station, or per-

form any of several other functions related to message transmission, such as sending and receiving responses, disabling or disconnecting a switched line, etc.

In the READ or WRITE macro instruction you specify:

- The line group and specific line within that group over which the operation is to occur.
- The address of a terminal list or an entry in that list that contains the information BTAM needs to establish contact with a station. Examples of this kind of information are telephone numbers, polling and addressing sequences, and identification sequences.
- The type of Read or Write operation to be performed (Read Initial, Write Continue, etc.)
- The address of the data event control block (DECB) that the READ or WRITE macro instruction is to define or the address of an existing DECB that the operation is to use.
- The addresses of input and output areas into which or out of which message text is to be received or sent.

Each of these parameters is discussed in the explanation of the operands.

Name	Operation	Operands
{symbol}	{READ } {WRITE }	decbaddr, optype, decbaddr, {(inoutareal)} {({inareal}, {outareal})}, {(inoutlength)} {({inlength}, {outlength})}, {entry}, {rln} [,MF=L] [,MF=E]

decbaddr

specifies the address of the DECB associated with the line. You can use register notation only if the macro instruction is of the execute form (MF=E).

optype

specifies one of the operation-type codes listed in Figures 18, 19, and 20. The channel program generated for each type of Read and Write operation differs depending on the particular terminal and network configuration.

The available types for a given type of remote station or line configuration are given in the next two chapters. The available types for the local 3270 display system are given in the section "Local Read and Write Operations." In all cases, if the single letter T is coded, no type code is set in the DECB. The T can be used:

1. With a list form, to create a DECB with no type code. The type code would be furnished by a subsequent READ or WRITE macro instruction of execute form.
2. With an execute form, when the type code already in the DECB is to be used.

dcbaddr
specifies the address of the DCB for the line group.

inoutarea
specifies the address of the first byte of the input area (Read operations) or the first byte of the output area (Write operations). In a READ macro instruction, you may code this operand as 'S' if you are using dynamic buffering and wish BTAM to provide the needed buffers. If the inoutarea operand is specified as 'S', the inoutlength operand is ignored and BTAM obtains the buffer length from the DCB. This operand may be omitted for read types TIQ and TQ and write types TR, TQ, TN, TA, TD, and TW.

For write type TQ, if inoutarea is omitted or inoutlength is equal to or less than two, the response is read into the DECRESFN field of the DECB.

inarea and outarea (BSC only)
are for use in READ macro instructions of the TCW, TTL and TPL types and WRITE macro instructions of the TIV, TIVX, TTV, and TTVX types.

For read types TTL and TPL, outarea contains the leading-graphics characters to be sent to the remote station, and inarea receives the text from the station. For read type TCW, outarea specifies the address of the tone characters to be sent to the remote station, and inarea receives the text from the station. For write types TIV, TIVX, TTV, and TTVX, outarea contains the text to be sent to the remote stations, and inarea receives the text transmitted from the remote station.

For either READ or WRITE macro instructions, you may code inarea (but not outarea) as 'S' if you are using

dynamic buffering and wish BTAM to provide the needed buffers.

For more information about using the inarea and outarea operands for the remote 3270 display system, see "Read Operations" and "Write Operations" under the heading "Line Control and Message Transmission" in the section "IBM 3270 Information Display System -- Programming Considerations."

inoutlength
specifies the number of bytes in the input or output area defined by the inoutarea operand. In a WRITE macro instruction, you may code this operand as 'S', to cause BTAM to obtain the buffer length from the DCB. The inoutlength operand need not be coded for read types TIQ and TQ and write types TR, TO, TN, TA, TD, and TW.

For write type TQ, if inoutlength is omitted or is equal to or less than two, the response is read into the DECRESFN field of the DECB.

CAUTION: Specify 'S' only if the last buffer to be sent contains the ending character.

inlength and outlength (BSC only)
are for use in the same types of macro instructions indicated under "inarea and outarea," and specify the length of these areas. In a WRITE macro instruction, you may code outlength as 'S', to cause BTAM to obtain the buffer length from the DCB. The same caution indicated for inoutlength applies to outlength.

Programming Notes:

1. The value specified for inoutlength, inlength, or outlength must include (a) all control characters that are to be sent or received if they will be sent from or received into the area (that is, the length should not include any control characters sent automatically by BTAM or received into other than the input area) and (b) the four-byte link field, if dynamic buffering is used.
2. Any macro instruction of the execute form that specifies inarea and outarea rather than inoutarea must refer to a DECB that has been defined with a list or standard-form macro instruction that also specified inarea and outarea, because the DECB required for leading-graphics and conversational operations is longer than that for operations not requiring both input and output areas.

OPTION	TYPE CODE	1030	1050 (nonsw)	1050 (switched)	1060	2260	8383 115A	TW'X 33/35	WT Teleg
READ Initial	TI	X Note	X Note	X	X Note	X	X	X	X
READ Initial with Reset	TIR	X Note	X Note	X	X Note	X		X	
READ Continue	TT	X Note	X	X	X Note	X			X
READ Continue with Reset	TTR	X Note	X	X	X Note	X			
READ Continue with Leading Acknowledgment	TTA								
READ Continue with ID Exchange	TE								X
READ Conversational	TV			X				X	
READ Conversational with Reset	TVR			X				X	
READ Repeat	TP	X Note	X	X	X Note	X			
READ Repeat with Reset	TPR	X Note	X	X	X Note	X			
READ Buffer	TB					X			
READ Skip	TS	X	X	X	X	X	X	X	
WRITE Initial	TI	X	X	X	X	X	X	X	X
WRITE Initial with Reset	TIR	X	X	X	X	X	X	X	
WRITE Initial Optical	TIO								
WRITE Invitational Optical	TCO								
WRITE Continue	TT	X	X	X		X			X
WRITE Continue with Reset	TTR	X	X	X		X			
WRITE Continue Conversational	TTV		X*	X					
WRITE Conversational	TV			X				X	
WRITE Conversational with Reset	TVR			X				X	
WRITE Conversational Optical	TVO								
WRITE at Line Address	TL			X		X			
WRITE at Line Address with Reset	TLR					X			
WRITE Erase	TS					X			
WRITE Erase and Reset	TSR					X			
WRITE Break	TB						X		
WRITE Positive Acknowledgment	TA	X	X	X	X	X			
WRITE Negative Acknowledgment	TN	X	X	X	X	X		X	
WRITE Disconnect	TN								

Note: Options for which Auto Poll channel programs are generated if the IODEVICE system generation macro instruction for the time specified FEATURE= AUTO POLL.

*Write TTV cannot be used if Auto Poll is specified (i.e., FEATURE=AUTO POLL in IODEVICE macro for the line).

Figure 18 (Part 1 of 2). READ and WRITE Options for Start-Stop

OPTION	TYPE CODE	2740	2740C*	2740D*	2740 DC*	2740 DT*	2740 DTC*	2740 S*	2740 SC*	2740 CO*	2740 DCO*	2741 NS*	2741 SW*
READ Initial	TI	X	X	X	X	X	X	X Note	X Note	X	X	X	X
READ Initial with Reset	TIR		X	X	X	X	X		X Note	X	X		
READ Continue	TT		X		X		X		X	X	X		X
READ Continue with Reset	TTR		X		X		X		X	X	X		
READ Continue with Leading Acknowledgment	TTA									X	X		
READ Continue with ID Exchange	TE												
READ Conversational	TV			X	X	X	X				X		X
READ Conversational with Reset	TVR			X	X	X	X				X		
READ Repeat	TP		X		X		X		X	X	X		
READ Repeat with Reset	TPR		X		X		X		X	X	X		
READ Buffer	TB												
READ Skip	TS	X	X	X	X	X	X	X	X	X	X	X	X
WRITE Initial	TI	X	X	X	X	X	X	X	X	X	X		
WRITE Initial with Reset	TIR	X	X	X	X	X	X	X	X	X	X		
WRITE Initial Optical	TIO									X	X		
WRITE Invitational Optical	TCO									X	X		
WRITE Continue	TT		X		X		X		X	X	X	X	X
WRITE Continue with Reset	TTR		X		X		X		X	X	X		
WRITE Continue Conversational	TTV		X		X							X	X
WRITE Conversational	TV		X	X	X	X	X			X	X	X	X
WRITE Conversational with Reset	TVR		X	X	X	X	X			X	X		
WRITE Conversational Optical	TVO									X	X		
WRITE at Line Address	TL												
WRITE at Line Address with Reset	TLR												
WRITE Erase	TS												
WRITE Erase with Reset	TSR												
WRITE Break	TB												
WRITE Positive Acknowledgment	TA		X		X		X		X	X	X		
WRITE Negative Acknowledgment	TN		X	X	X	X	X		X	X	X		
WRITE Disconnect	TN												X

*C Checking feature
D Dial-up feature
T Transmit Control feature
S Station Control feature
O Optical Image Unit feature
NS Nonswitched
SW Switched

Note: Options for which Auto Poll channel programs are generated in the IODEVICE system generation macro instruction for the line specified FEATURE = AUTOPOLL

Figure 18 (Part 2 of 2). READ and WRITE Options for Start-Stop

Option	Type Code	Nonswitched Point-to-Point	Multipoint	Switched Point-to-Point
READ Initial	TI	X	X	X
READ Connect	TC			X
READ Connect with Tone	TCW			X
READ Continue	TT	X	X	X
READ Continue with Leading Graphics ^{1,4,8}	TTL	X	X	X
READ Repeat	TP	X	X	X
READ Repeat with Leading Graphics ^{1,4,8}	TPL	X	X	X
READ Initial Inquiry	TIQ	X		
READ Inquiry	TQ	X	X	X
READ Inquiry Monitor ¹⁰	TQM			X
READ Interrupt ⁷	TRV	X	X	X
WRITE Initial ²	TI	X	X	X
WRITE Initial and Reset ²	TIR	X	X	
WRITE Continue ²	TT	X	X	X
WRITE Continue and Reset ²	TTR	X	X	
WRITE Reset	TR	X	X	X
WRITE Inquiry	TQ	X	X	X
WRITE Reset Monitor ¹⁰	TRM			X
WRITE Disconnect	TD			X
WRITE Wait Before Transmit ¹	TW	X	X	X
WRITE Initial Conversational ^{2,5}	TIV	X	X	X
WRITE Continue Conversational ^{2,5}	TTV	X	X	X
WRITE Initial Transparent ^{3,9}	TIX	X	X	X
WRITE Initial Transparent and Reset ^{3,9}	TIXR	X	X	
WRITE Initial Transparent Block ⁹	TIE	X	X	X

Figure 19 (Part 1 of 2). READ and WRITE Options for BSC

Option	Type Code	Nonswitched Point-to-Point	Multipoint	Switched Point-to-Point
WRITE Continue Transparent ^{3,9}	TTX	X	X	X
WRITE Continue Transparent and Reset ^{3,9}	TTXR	X	X	
WRITE Continue Transparent Block ⁹	TTE	X	X	X
WRITE Initial Conversational Transparent ^{3,6,9}	TIVX	X	X	X
WRITE Continue Conversational Transparent ^{3,6,9}	TTVX	X	X	X
WRITE Break ¹¹	TB			X
WRITE Connect	TC			X

¹This macro cannot be used for a 2780 with which the central computer communicates using 6-bit Transcode.
²This macro cannot be used for a 2715 because text transmission to this type of station is always in transparent mode.
³This macro cannot be used for a 2972 because text transmission to this type of station is always in nontransparent mode.
⁴The 1800, 2715, and 2770 ignore leading graphics characters sent to them. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.
⁵The 1800, 2770 and 2972 do not transmit text as a response to text received from the central computer, the return
⁶The 1800, 2715 and 2770 the usual alternating acknowledgment (ACK-0 or ACK-1).
⁷When this macro is used for the remote 3270, the response is always EOT.
⁸This macro is not applicable for the remote 3270, which cannot receive leading graphics.
⁹This macro is applicable for the remote 3270 only when that device has structure field capability.
¹⁰This macro is applicable only to the switched 3275.
¹¹This macro is not applicable to the switched 3275.

Figure 19 (Part 2 of 2). READ and WRITE Options for BSC

Option	Type Code
READ Initial	TI
READ Modified	TM
READ Modified from Position	TMP
READ Buffer	TB
READ Buffer from Position	TBP
WRITE Initial	TI
WRITE Erase	TS
WRITE Erase Alternate ¹	TSA
WRITE Structured Field ²	TSF
WRITE Unprotected Erase	TUS

¹Does not apply to 3272 devices.

²Does not apply to 3272 or 3274 Model 1B devices.

Figure 20. READ and WRITE Options for Local 3270 Display System

- If online testing is made available for a BSC line (by coding T among the EROPT options in the DCB macro instruction for the line group), all READ Initial macro instructions issued for the line must specify a length of no less than 300 bytes. Online test RFT messages may be received into this area at any time. If 'T' is coded as the optype operand in a WRITE macro instruction of the list form (MF=L), and the '(inarea,outarea)' and '(inlength,outlength)' operands are not coded, the resultant DECB does not allow space for the DECWLNG and DECWAREA fields.

Note: For devices attached to the 3274 or 3276 choose the test type that corresponds to the supported buffer size.

entry

specifies the address of the terminal list or an entry therein, as follows:

1. For a nonswitched line (OPENLST, AUTOLST, or SSALST), it specifies the address of an entry within the terminal list.
2. For a nonswitched line (WRAPLST, AUTOWLST, or SSAWLST), it specifies either the address of any entry within the terminal list or 'S'.

Note: If 'S' is specified, the system provides the address of an entry in the polling list as follows:

- a. If the previous polling operation terminated with a negative response as a result of a RESETPL macro instruction, the address of the next entry is provided.
 - b. Otherwise, the address of the entry that was last polled is provided.
3. For a switched line (DIALST, SWLST, BSCLST, IDLST, or WTTALST), it must specify the address of the beginning of the terminal list; it cannot be coded as 'S'.

For read types TMP and TBP for the local 3270 display system, entry specifies the address of a four-byte area that gives the position from which the read operation is to begin. This operand may be omitted for all other read and write types for the local 3270 display system, since the operand is ignored.

rln

specifies in decimal the relative line number within the line group (range 1-255 inclusive). This value is placed in the DECRLN field of the DECB in binary form.

For read type TI for the local 3270 display system, rln specifies the first display station that is to be checked for an attention interruption. For all other read and write types for the local 3270 display system, this operand specifies the device from which or to which a message is to be read or written.

MF=L

specifies that this macro instruction causes only the creation of a data event control block the name of which

is specified by the decbaddr operand. Specify this when you wish to create a data event control block that will be referred to subsequently by one or more READ or WRITE macro instructions (each of which will specify the MF=E operand and decbaddr operand of which will specify the address of the data event control block created by this macro instruction).

MF=E

specifies that this macro instruction causes execution of the Read or Write operation, using a data event control block created by a READ or WRITE macro instruction of the list or standard form.

Return Codes: After a READ or WRITE macro instruction, BTAM sets register 15 to zero if no error has been detected. If an abnormal condition is detected, the operation is not started, and control is returned to your program at the instruction following the READ or WRITE macro instruction. A return code in register 15 indicates the error. Bits 24 through 31 will contain one of the following error codes in hexadecimal notation:

- 04 Busy: The specified line is busy with a previously requested Read or Write operation.
- 08 Invalid RLN: The relative line number specified in the operand field of the READ or WRITE macro instruction is zero or is larger than the number of lines in the line group.
- 0C (1) Invalid "optype" code: The READ or WRITE macro instruction specified an "optype" that is invalid for the kind of remote station for which you issued the macro instruction.

(2) An initial-type WRITE macro instruction (for example, WRITE TI, TIX, TIV) erroneously specified an answering list instead of a calling list (that is, no dial digits are present in the list).
- 10 All skip bits on (programmed polling): The skip bit is on in all of the entries in the polling or addressing list.

For local 3270, all skip bits are on by means of the CHGNTRY macro.

Usage Count too large (Autopoll): The Usage Count is larger than its maximum value of 15.
- 14 Line Error during Open: SAD or Enable command (issued during Open) resulted in a permanent I/O error. The error status in the DECB may be inspected to determine the cause of the error. For local 3270 devices, the Write Erase

command issued during open processing did not complete successfully; other devices in the line group which were successfully opened will continue to operate when a read or write macro is issued specifying the rln of any of those devices.

- 18 **Buffers Not Available:** The buffer pool does not contain enough buffers to satisfy the Read operation (area coded 'S').
- 1C **No Buffer Pool:** No buffer pool was defined in the DCB macro instruction, or there was no indication that BTAM was to provide the pool associated with the line group prior to Open.
- 20 **No Buffer Routine:** You did not indicate you wanted the buffer routine prior to open, so it was not loaded with the system. The return code occurs on a Read operation.
- 24 **Invalid Order:** The second byte of the area specified by the entry operand of a READ TMP or TBP macro instruction (for a local 3270 display system) is not an SBA order.
- 28 **Invalid Control Block:** An invalid control block was encountered during a read or write operation for the local 3270 display system.
- 2C **Device Not Available:** A request for a read or write operation was rejected, because OLTEP is using the local 3270 device. If the read operation requested was a READ INITIAL, processing does not continue for the rest of the line group.

Note: All nonzero return codes indicate that no I/O operation was initiated; therefore, the program must not issue a WAIT or TWAIT macro instruction for a READ or WRITE macro instruction that resulted in a nonzero return code, because the task would enter a permanent wait state.

Programming Note: Execution of a READ or WRITE macro instruction causes control to be passed to a BTAM routine which constructs channel programs. If no invalid conditions are detected, a channel program is generated for the requested I/O operation. After the channel program has been started, control is returned to your program with a return code of zero. The I/O operation proceeds asynchronously with respect to program execution. When you wish to determine whether the I/O operation has completed, issue a WAIT or TWAIT macro instruction, or check the DECSDECB field (the event control block).

If you intend to make use of the 'S' option for the entry operand in READ Initial macro instructions, ensure that the polling list address is placed in the data event control block before the first execu-

tion of the READ Initial macro instruction. This may be done by defining the polling list address in a READ macro instruction of the list form and then using the 'S' in a READ Initial macro instruction of the execute form. However, if a WRITE Initial macro instruction is issued (using the same DECB) before the first READ Initial macro instruction, steps must be taken to replace the polling list address in the DECB. After the first READ Initial macro instruction, BTAM maintains the polling restart address in the DECB for the line. Thus, by using the 'S' option, the polling list address is preserved across write operations (even though they utilize the same DECB field for addressing list pointers).

When a READ or WRITE macro instruction specifies "reset at completion" for a switched line, BTAM disconnects the line only if no error condition occurs during the execution of the basic channel program. Thus, the program may attempt retransmission without reestablishing the line connection. If the program elects not to attempt retransmission, the WRITE TN macro instruction may be executed to perform the disconnect function.

When a READ or WRITE macro instruction specifies "reset at completion" for a non-switched line, the EOT character (or sequence of characters) is transmitted only if no error condition occurs during execution of the basic channel program.

When a polling function is performed in a Read Initial operation, the terminal list address field in the DECB (DECENTRY) contains the address of the entry in the polling list that was last polled. Thus, the program may determine the source of the message (if one was received) by inspecting the contents of the polling list entry at that address. Note that the terminal list address field in the DECB is not modified for addressing operations.

With Autopoll an index byte is provided in the first byte of DECPOLPT. The index byte contains the number of the polling entry for the terminal from which the message was read (that is, it contains one for the first entry in the polling list, two for the second entry, etc.). The program may obtain the index byte from DECPOLPT to identify the originating terminal following any Read operation.

When a READ macro instruction is used for the local 3270 display system, the relative line number of the device from which the message is read is placed into the DECPOLPT field of the DECB.

User program error routines that operate synchronously with respect to the comple-

tion of the I/O operation may retry a macro instruction with the knowledge that the proper parameters are in the DECB. The error routine may use a READ or WRITE macro instruction of the execute form with only the decbaddr and the otype operands specified.

RESETPL (Reset Polling List or Reset Line) Macro Instruction

RESETPL may be issued whenever you wish to cancel a Read operation that is currently in progress but has not yet received a positive response to polling from a remote station (nonswitched multipoint line), or has not yet received an ENC character indicating the remote station's intention to transmit (nonswitched point-to-point line), or has not yet received a call from a remote station (switched line). If at the time the RESETPL is issued, a positive response or a call has been received or message transmission has taken place, the reset function has no effect; the Read operation proceeds as usual.

RESETPL functions with a nonswitched multipoint line in the following manner. If a programmed polling operation is currently in progress, and if it elicits a negative response, polling is terminated, the polling list pointer (DECPOLPT) is incremented, and the operation is posted complete. If an Autopoll polling operation is currently in progress, and if it elicits negative responses to all entries in the list, the operation is posted complete and the index byte identifying the last active entry is stored in the first byte of DECPOLPT. In both cases (programmed and Autopoll), the negative response bit is set in DECFLAGS. If the polling operation elicits a positive response or a time-out, the polling list pointer is not incremented, and the operation proceeds to its normal conclusion (normal conclusion for a time-out is to post it complete-with-error).

For a nonswitched point-to-point line, if a Prepare operation is currently in progress, and the Prepare has not been completed, a Halt I/O command is issued for that line. If an operation other than the Prepare is currently in progress (for example, message reception, message transmission, addressing), it proceeds to its normal completion.

The RESETPL macro instruction functions with a switched line in the following manner. If an Enable command has been issued to a line (to allow a terminal to dial the computer), and a call has not been received (the Enable has not been completed), BTAM stops line activity by means of a Halt I/O command. If the Enable has already been completed and a polling operation (start-stop only) is currently in progress, the

function described above for programmed polling or multipoint lines is performed. If the Enable has already been completed and no polling operation (Start-Stop only) is in progress, the Read operation proceeds unaffected.

Note: A special form of the RESETPL macro instruction is used for the local 3270 display system. See "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Information Display System -- Programming Considerations." If only the decbaddr operand is specified, the instructions that are generated include support for the local 3270 display system.

Name	Operation	Operand
[symbol]	RESETPL	decbaddr [, , H102740 , POLLING , ANSRING, H102740]

decbaddr

specifies the address of the data event control block for the line for which the reset operation is to be performed.

POLLING

specifies that only the instructions required to terminate polling on a nonswitched line will be generated by this macro instruction.

ANSRING

specifies that only the instructions required to terminate an answering operation on a switched line will be generated by this macro instruction.

H102740

specifies that a H10 will be done to a 2740 line.

If no second operand is specified, instructions are generated to determine at execution time which function is to be performed. For World Trade telegraph terminal lines, omit POLLING and ANSRING.

Programming Note: No further READ or WRITE macro should be issued for a line for which a RESETPL macro has been issued until the operation in progress has been posted complete. That is, provided that the RESETPL macro instruction gave a return code of X'00' or X'04', a WAIT macro instruction should be coded between the RESETPL macro and the next READ or WRITE macro.

Return Codes: After execution of a RESETPL macro instruction, bits 24 through 31 of register 15 contain a return code indicating the status of the operation. Bits 0 through 23 contain zeros. The code is one of the following, in hexadecimal notation:

- 00 Normal Return: This code is set if an Enable or Prepare command was outstanding and a Halt I/O instruction was successfully executed.
- 04 Complete: This code is set if the Enable command was already completed or the Post flag in the UCB is not on.
- 08 Illegal Request: This code is set if the unit control block (UCB), an internal OS/VS control block, specifies a nonteleprocessing device.
- 0C Unsuccessful: This code is set if an invalid unit control block address has been passed to the IOHALT system macro instruction, or if the Halt I/O instruction has terminated in error (for example, Channel Data Check, or a Not Operational condition code). This code is also set if the DCB has not been opened.
- 10 Not Issued: this code is set if no Enable command had been issued. This is not used in point-to-point contention.

Note: These return codes have different meanings for the form of the RESETPL macro instruction used for the local 3270 display system. See "Attention Handling and Read Initial Operations" in the section "IBM 3270 Information Display System -- Programming Considerations."

WAIT Macro Instruction

The WAIT macro instruction relinquishes control of the CPU when the user program has no further processing to do and must wait for the completion of one or more Read/Write operations. See OS/VS Supervisor Services and Macro Instructions for complete information about this macro instruction.

Name	Operation	Operand
[symbol]	WAIT	[count], ECB=ecb address ECBLIST=ecb list addr

count

specifies the number of events among the events referred to by the ECB or ECBLIST operand that must be posted complete before the WAIT macro instruction is satisfied. If the count operand is omitted, one is assumed.

ECB

specifies the address of an event control block (ECB) representing a single event to be posted complete before processing by the user program can continue.

ECBLIST

specifies the address of a variable-length list containing fullword entries with each fullword entry containing the address of an event control block (ECB) in the low-order three bytes. Each event control block pointed to represents an event awaiting completion. In this list of ECB addresses, the high-order bit (0-bit) of each fullword entry except the last in the list must be zero. In the last entry in the list, you must set the 0-bit in the high-order byte of the entry to one.

TWAIT Macro Instruction

The TWAIT macro instruction relinquishes control of the CPU when the user program must wait for the completion of one of a number of events before further processing can be done.

Name	Operation	Operands
[symbol]	TWAIT	(returnreg), ECBLIST=ecb list addr

returnreg

contains the address of the ECB representing the event posted complete.

ECBLIST

specifies the address of the user-created list of ECB addresses representing events awaiting completion. Each entry in the list is a fullword containing an address in the low-order three bytes. Because the list is of variable length, the high-order bit (0-bit) of each fullword entry (except the last) must be set to zero except that of the last entry. The high-order bit of the last fullword entry must be set to one to identify the entry as the last in the list.

If TWAIT is issued for any event other than a Read or Write operation, your program should clear the ECB. The TWAIT macro instruction is similar to the WAIT macro instruction except that:

- TWAIT requires the completion of one event only before returning control to the problem.
- The ECB keyword is not used in TWAIT.
- The address of the ECB that was posted complete is returned to you in the register specified (as the first operand of TWAIT).

- The displacement of the ECB address from the beginning of the ECB list (as specified in the ECBLIST operand) is returned in register 15.

The last point above simplifies branching to a routine associated with a particular ECB. Set up a where-to-go list of four-byte entries, each entry containing the address of a routine to be associated with the ECB whose address is in the corresponding entry of the ECB list. Then you may code, following the WAIT macro instruction,

```
L      15,WTGLIST(15)
BALR   14,15
```

or the equivalent, to branch and link to the appropriate routine.

Read Skip Operations

One of the types of Read operations that may be performed for terminals on a start-

stop communications line is Read Skip. In this operation, effected by the READ Skip macro instruction, any data being received from a terminal is discarded, instead of being placed in main storage. This action "clears the line" of any unwanted data, so that normal Read and Write operations can be resumed. A Read Skip operation is intended for use in user-written recovery routines when a lost-data error has occurred. (It may be used for any start-stop terminal.)

Because it is not used in normal message transmission operations, Read Skip is not discussed in the section "Start-Stop Read and Write Operations."

In a READ Skip macro instruction you need not specify the address of an input area, since the data it receives does not enter main storage; however, you must specify for the length a value exceeding the maximum amount of text data to be received by the Read Skip operation.

IBM 1030 DATA COLLECTION SYSTEM

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
[ OPENLST  
  WRAPLST ], (xx,...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
[ OPENLST,xx
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response, successively polls terminals in the polling list, beginning with the terminal to which it sent the response, and upon receiving a positive response to polling reads a message block. This macro is for use following a successful READ Initial (TI) or another READ Continue to receive another message block.

1. Write Circle Y EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TPR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response, successively polls the terminals in the polling list, beginning with the terminal to which it sent the negative response, and upon receiving a positive response to polling, reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT) or READ Repeat (TP), to read the same message block received by the previous operation.

1. Write Circle N EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. If an EOA is the first character of a message block (as it will be if the block has not been modified since it was received from a 1030 terminal), it will print at the terminal as #. You should therefore overlay with an idle character the first character of each block received from a 1030.
2. Each outgoing message block must end with ETB.
3. Insert three idle (EBCDIC) or Write Marks (transmission code) characters between adjacent message text characters. (This is required of all messages sent to a 1033 printer.)

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads a response from the terminal.

1. Write EOT EOT EOT Circle S
2. Write Addressing Character
3. Write "1"
4. Read Response
5. Write EOA

6. Write Text
7. Read Response
8. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the com-

puter received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write Circle Y EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 1050 DATA COMMUNICATION SYSTEM --
NONSWITCHED LINES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character identifies the terminal; the second identifies the specific component from which an input message is solicited. (If the second character is the common polling character, 0, input messages are read from any ready component. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{ OPENLST  
{ WRAPLST }, (xxyy, ...)
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence. The first character identifies the terminal; the second identifies the specific component that is to receive the output message. (If the second character is the common addressing character, 9, the output message is sent to all ready components.) To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST, (xxyy, ...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is

for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. The first block of a message received from a 1050 on a Read Initial operation will begin with an EOA character. If the same message block is then sent to a 1050, it will be printed as #. This may be avoided by overlaying the EOA with an Idle character before sending the message block.
2. Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the terminal sends a negative response to addressing, the operation is posted complete.

1. Write EOT EOT EOT
2. Write Addressing Sequence
3. Read Response
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

2. You must specify a polling list entry in the WRITE TTV macro instruction.

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets the terminals on the line to control mode, successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Write Polling Sequence
5. Read Response
6. Read Text

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

Programming Notes:

1. WRITE TTV performs exactly the same functions as would be performed by a WRITE Continue (TT) followed by a READ Initial (TI), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

IBM 1050 DATA COMMUNICATION SYSTEM --
SWITCHED LINES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation that answers a call from a terminal requires an answering-polling list; a Read Initial operation that calls a terminal requires a calling-polling list. Either type of list may have one or more terminal entries (all representing the same terminal), each containing a two-character polling sequence. The first character identifies the terminal, and must be the same character for all entries in the list; the second character identifies the specific component from which an input message is solicited. (If the second character is the common polling character, 0, input messages are read from any ready component.)

To define an answering-polling list, code the operand field of a DFTRMLST macro like this:

```
DIALST,0,(xxyy,...)
```

To define a calling-polling list, code the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars,(xxyy,...)
```

Write Operations

A Write Initial operation that calls a terminal requires a calling-addressing list; a Write Initial operation that answers a call from a terminal requires an answering-addressing list. Either type of list may have one or more terminal entries (all representing the same terminal), each containing a two-character addressing sequence. The first character identifies the terminal, and must be the same for all entries in the list; the second character identifies the specific component that is to receive the output message. (If the second character is the common addressing character, 9, the output message is sent to all ready components.)

To define a calling-addressing list, code the operand field of a DFTRMLST macro like this:

```
DIALST,dialcount,dialchars,(xxyy,...)
```

To define an answering-addressing list, code the operand field of a DFTRMLST macro like this:

```
DIALST,0,(xxyy,...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering-Polling List)

READ Initial answers a call from a terminal, polls it, reads the response, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Polling sequence
6. Read Response
7. Read Text
8. Write EOA EOT (TIR only)
9. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Calling-Polling List)

READ Initial dials the terminal, polls it, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial Digits
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Polling sequence
6. Read Response
7. Read Text
8. Write EOA EOT (TIR only)
9. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and - sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP)

READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)
4. Disable (TPR only)

Programming Note: In order to be able to issue READ Repeat for the paper tape reader or card reader, the reader must be equipped with the Line Correction feature. For either of these components you may issue it only twice in succession. Furthermore, you should use this macro for the paper tape reader only if the message block being read is less than 312 characters. (Otherwise, the time required to back up the tape for retransmission exceeds the time-out interval of the terminal.)

READ Conversational (TV)

READ Conversational and Reset (TVR)

READ Conversational polls the terminal, and if the response to polling is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for polling and reading a message block from a terminal with which a previous READ or WRITE macro has already established the line connection. Its main purpose is to allow you to change from sending message blocks to receiving them, without having to reestablish the line connection. READ Conversational can follow a READ Initial (TI), READ Continue (TT), or READ Repeat (TP), or a WRITE Initial (TI), WRITE Continue (TT), or WRITE Conversational (TV).

The terminal list used by the READ Conversation must be an open polling list (calling or answering), of the DIALST format.

1. Write EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write EOA EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. The first block of a message received from a 1050 on a Read Initial or Read Conversational operation will begin

with an EOA character. If the same message block is then sent to a 1050, it will print as #. This may be avoided by overlaying the EOA with an Idle character before sending the block.

2. Each outgoing message block must end with EOB.

WRITE Initial (TI)

WRITE Initial and Reset (TIR) (Using Calling-Addressing List)

WRITE Initial dials a terminal, addresses it, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Addressing sequence
6. Read Response
7. Write EOA
8. Write Text
9. Read Response
10. Write EOT (TIR only)
11. Disable (TIR only)

WRITE Initial (TI)

WRITE Initial and Reset (TIR) (Using Answering-Addressing List)

WRITE Initial answers a call from a terminal, addresses it, and if the response is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Addressing sequence
6. Read Response
7. Write EOA
8. Write Text
9. Read Response
10. Write EOT (TIR only)
11. Disable (TIR only)

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response

3. Write EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes a positive response to text (the EOA character is the positive response), addresses the terminal, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

The terminal list used by the WRITE Conversational must be an addressing list (calling or answering) of the DIALST format.

1. Write EOA EOT EOT EOT
2. Write Addressing sequence
3. Read Response
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT (TVR only)
8. Disable (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, resets it to control mode, polls it, and upon receiving a positive response to polling, reads a message block.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Write Polling Sequence
5. Read Response
6. Read Text

Programming Notes:

1. WRITE TTV performs exactly the same functions as would be performed by a

WRITE Continue (TT) followed by a READ Conversational (TV), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (that is, by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

2. You must specify a polling list entry in the WRITE TTV macro instruction.

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write EOT
2. Disable

IBM 1060 DATA COMMUNICATION SYSTEM

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character identifies the control unit, the second identifies the teller terminal that is to be polled.

To define a polling list, code the operand field of a DFTRMLST macro like this

```
{ OPENLST }
{ WRAPLST }, (xxyy, ...)
```

Write Operations

A Write Initial operation requires an addressing list having one terminal entry that contains a two-character addressing sequence. The first character identifies the control unit, the second identifies the teller terminal to which the message is to be sent.

To define an addressing list, code the DFTRMLST operand field like this:

```
OPENLST, xxyy
```

The list must be defined as an open list.

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response, successively polls terminals in the polling list, beginning with the terminal to which it sent the response, and upon receiving a positive response to polling, reads a message block. This macro is for use following a successful READ Initial (TI) or

another READ Continue to receive another message block.

1. Write Circle Y EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response, successively polls the terminals in the polling list, beginning with the terminal to which it sent the negative response, and upon receiving a positive response to polling reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT) or READ Repeat (TP), to read the same message block received by the previous operation.

1. Write Circle N EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write Circle Y EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with ETB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to address is positive, writes an EOA followed by message text and reads the response to text. If the terminal sends a negative response to addressing, the operation is posted complete.

1. Write EOT EOT EOT
2. Write Addressing sequence
3. Read Response
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT EOT EOT (TIR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write Circle Y EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 2260/2848 DISPLAY COMPLEX (REMOTE) AND
IBM 2265/2845 DISPLAY COMPLEX (REMOTE)

The information in this section applies equally to the IBM 2260/2848 display complex and the IBM 2265/2845 display complex, except that references to multiple display stations and the general polling function do not apply to the 2265/2845 display complex (only one 2265 display station can be attached to a 2845 display control).

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character identifies the 2848 or 2845 Display Control, the second identifies the 2260 or 2265 Display Station from which an input message is solicited. (If, for a 2260/2848, the second character is coded as X'FF', a general poll is performed.)

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{ { OPENLST }, (xxyy, ...)
  { WRAPLST }
```

Write Operations

A Write Initial operation requires an addressing list. The list may have one or more terminal entries, each containing a two-character addressing sequence. The first character identifies the 2848 or 2845 Display Control, the second identifies the 2260 or 2265 Display Station or 1053 printer that is to receive the output message.

To define an addressing list, code the DFTRMLST operand field like this:

```
{ OPENLST, (xxyy, ...) }
```

READ MACRO INSTRUCTIONS

The format of a message received through a READ macro instruction is:

```
[STX|device address| [text] | [CAN] | ETX]
```

STX

specifies the start of text characters.

device address

identifies the sending unit (display station or printer).

text

is the message text.

CAN

is the cancel character, sent only if the display control detects an internal operation error when transmitting the message.

ETX

is the end-of-text character.

READ Initial (TI)

READ Initial and Reset (TIR)

READ Initial successively polls the display stations and printers in the polling list, and upon receiving a positive response to polling, reads a message block or a printer status message.

1. Write STX and 15 EOT's
2. Write Polling Sequence
3. Write READ MI code
4. Read Response
5. Read Text
6. Write STX EOT EOT EOT (TIR only)

Functions of the Read Initial Operation

The Read Initial operation can have one of three functions:

Specific Polling of One or More Display Stations:

A Read Initial operation executed for this purpose is similar to the Read Initial for other types of terminals that use the polling scheme. That is, the stations to be polled are individually represented in a terminal list and polling proceeds until the end of the list is reached (open list) or until a station returns a positive response and a message block (open or wraparound list).

Requesting Printer Status:

The function of polling ordinarily refers to contacting a terminal or terminal component to determine whether it has any message to send to the computer. With reference to the 1053 printer attached to an IBM 2848 Display Control, the term polling means contacting the printer to see if it is ready to receive a message from the computer. (The printer may not be ready because the terminal operator is using it locally or because its power is off or it is out of paper.) In order to be ready, the printer mechanism must be ready and the printer buffer must not be in use. (The printer buffer is in

use when it is being filled from the keyboard buffer or the line, or if its contents are currently being printed.) If the printer, when polled, is ready, it returns to the computer a positive response, the format of which is STX (device address) ETX.

If the printer is not ready, it returns a NAK; if the printer mechanism is ready but the printer buffer is in use, the printer returns an EOT. Either NAK or EOT is considered a negative response. Once the printer status is requested, whether the status is positive or negative the printer is in the "printer-request" condition, which means that it is available only for a message sent by a Write operation.

If a general poll operation (see below) is being executed after the status is requested, the printer returns the positive response when its status changes from not-ready to ready. The response is given the first time the printer is polled following the change to the ready condition.

General Polling of a 2848 Display Control:
In this form of operation, which is achieved by coding X'FF' as the second byte of a single polling list entry, all the display stations and printers connected to the display control identified by the first character of the entry are polled in wrap-around fashion, i.e., continuously, until one of the stations returns a positive response or until the printer responds with a status indication. The printer is always polled first, then the display stations, in each polling "pass". (The printer status is returned when it is polled, either if a previous Read Initial operation requested the status indication or if a previous Write Initial operation for the printer was not possible because the printer was not free.)

When a display station sends a response, the Read operation receives the message block into the input area specified in the READ macro. When a printer returns a response, the Read operation receives the status indication in the input area.

Programming Note: It is advisable to turn on the end-of-list bit of the polling list entry for the printer when executing a Read Initial operation to request printer status; then if the printer returns a negative response the operation is posted complete and the negative response condition can be tested for in the DECB. If the end-of-list bit is not on, the negative response is treated just like a negative response from a display station; polling continues, with no response indication given in the DECB.

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same display station that sent the previous block.

1. Write ACK
2. Read Text
3. Write STX EOT (TTR only)

READ Repeat (TPR)
READ Repeat and Reset (TPR)
(Display Only)

Read Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write NAK
2. Read Text
3. Write STX EOT (TPR only)

READ Buffer (TB)
READ Buffer and Reset (TBR)
(Display only)

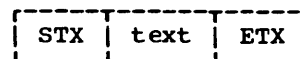
READ Buffer is intended for special applications and for use in diagnosing equipment troubles. It receives the entire contents of the buffer of the specified display station.

1. Write STX and 15 EOT's
2. Write Polling sequence
3. Write Read Buffer Code
4. Read Response
5. Read Text
6. Write STX EOT EOT EOT (TBR only)

Programming Note: At the completion of the Read Buffer operation you must issue a WRITE Erase to erase the screen or else write a message that will overlay the previous buffer.

WRITE MACRO INSTRUCTIONS

The format of a message sent through a WRITE macro instruction is:



BTAM supplies the STX character; it is not provided by the user. The ETX character must be the last character of a message.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
 (Display or Printer)

WRITE Initial addresses a display station or printer, and if the response to addressing is positive, writes an STX and the message text, then reads the response from the addressed unit. If the response to addressing is negative, the operation is posted complete.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Write Code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TIR only)

Programming Notes:

1. If you issue a WRITE Initial to send a message block to the printer and the printer returns a negative response (NAK or EOT) to addressing, the operation is posted complete; the message text is not sent. If the response is positive, the message is sent. If an error occurs during transmission of text, the printer buffer is cleared. You may retry the operation with a WRITE Continue macro.
2. If you issue a WRITE Initial to send a message block to a display station, a positive response is normally received and the text is sent. If during transmission of text, an error occurs, you may retry the operation by issuing a WRITE Continue, but the message containing the error is not cleared. You may also resend the message with a WRITE Erase macro, or with a READ Buffer followed by a WRITE Erase, if several messages were displayed.

WRITE at Line Address (TL)
WRITE at Line Address and Reset (TLR)
 (Display Only)

WRITE at Line Address has the same function as a WRITE Initial, but permits specifying the display line on which the message is to be displayed.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Line Address Code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TLR only)

Display Line	Display Line Addresses	
	2260 or 2265 Code (hex)	EBCDIC
		Code(hex) Character
1	50	F0 0
2	51	F1 1
3	52	F2 2
4	53	F3 3
5	54	F4 4
6	55	F5 5
7	56	F6 6
8	57	F7 7
9	58	F8 8
10	59	F9 9
11	5A	7A :
12	5B	5E ;
13	5C	4C <
14	5D	7E =
15	5E	6E >

Figure 21. IBM 2260 AND 2265 Display Line Addresses

Programming Note: The first byte of the message text must contain a line address character. The message will be displayed starting at the beginning of that line. See Figure 21 for valid display line addresses.

WRITE Erase (TS)
WRITE Erase and Reset (TSR)
 (Display Only)

WRITE Erase has the same function as a WRITE Initial, but also causes the display station screen to be erased before the message is displayed.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Erase code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TSR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)
 (Display or Printer)

WRITE Continue writes a message block and receives a response. This macro is for use following any Read or Write operation that did not include the Reset function.

1. Write STX
2. Write Text
3. Read Response
4. Write STX EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)
(Display Only)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT to indicate that the computer received the message text without error and to stop line activity. This macro is for use following a Read operation when you wish to stop receiving from the display station.

1. Write STX EOT

WRITE Negative Acknowledgment (TN)
(Display Only)

WRITE Negative Acknowledgment writes an EOT, which constitutes a negative response indicating that the computer received the message text with an error and to stop line activity. This macro is for use following a Read operation when you wish to stop receiving from the display station.

1. Write EOT

IBM 2740 COMMUNICATION TERMINAL -- GENERAL INFORMATION

The IBM 2740 Communication Terminal (Model 1 or 2) is available with several features or combinations of features. Of these, BTAM supports the following:

2740 on Nonswitched Lines:

Basic 2740
2740 with Checking
2740 with Station Control
2740 with Checking and Station Control
2740 with Checking and 2760 Optical Image Unit

2740 on Switched Lines:

2740 Dial
2740 Dial, with Checking
2740 Dial, with Transmit Control
2740 Dial, with Checking and Transmit Control
2740 Dial, with Checking and 2760 Optical Image Unit

The channel programs differ for the various feature combinations and are therefore explained separately on the following pages. (See "IBM 2760 Optical Image Unit" for information about the 2760.)

Programming Notes:

1. The 2740 without station control and transmit control does not react to a transmission control unit (TCU) timeout, nor does it time out along with the TCU. Following a TCU timeout, the 2740 is left in transmit text mode is unresponsive to channel commands. The terminal operator must then depress the EOT key or power down and back up to place the 2740 in control receive mode.
2. Only 2740 terminals equipped with the Checking feature provide an automatic response to messages received from the computer. For 2740s not having this feature, responses, if desired, must be entered manually from the 2740 keyboard, and the program must be able to receive such responses. That is, each Write operation for which a response is required must be followed by a Read operation. It is up to the user to establish a convention for responses; for example, to consider the letter Y received from the terminal as a positive response (the terminal received the message correctly) and the letter N received from the terminal as a negative response (the terminal received the message incorrectly and the program should resend the same message).

3. Each message sent to a 2740 Model 2 that is equipped with the Buffered Receive feature must end with an EOT character; the EOT must be supplied by the user program.
4. Multiple-block messages must not be sent to a 2740 Model 2 that is equipped with the Buffered Receive and Checking features, because (a) the contents of the buffer are printed only when an EOT is received from the computer, and (b) all blocks are read into the same buffer. This means that if a multiple-block message is received, only the block received just prior to the EOT will be printed; all previous blocks will have been successively overlaid in the buffer.
5. In sending message text to a 2740 Model 2 with the Buffered Receive feature, be careful to avoid a buffer overflow condition that will occur if the central computer sends a message block exceeding the capacity of the terminal buffer. This can happen even when the length of the message block in main storage is less than the buffer size. For each change in case (upper to lower, or vice versa), the TCU inserts a shift character in the data stream going to the terminal. You should ensure not only that the length of the message block in main storage is shorter than the 2740 buffer, but that it is shorter by an amount sufficient to allow for the inserted shift characters.

6. The Model 2 responds to addressing with a two-character reply. If the response is positive, the first character indicates whether an error occurred during the previous Write operation while transferring data from the buffer to the printer; if an error occurred, its nature is indicated. The second character is the positive response, circle Y.

If the response is negative, the first character indicates the reason for that response; the second character is the negative response, circle N.

The two-character response is received in the DECRESPI field of the DECB for the line.

The operation is posted complete, with or without error, in the event control block for the line, and the appropriate bits are set in the DECFLAGS field of the DECB.

Following each Write operation you should examine the first byte of the

DECRES PN field to determine whether an error occurred, and what kind it is.

The characters (in hexadecimal notation) and their meanings are as follows.

Positive response (second character is Circle Y)

<u>First Character</u>	<u>Meaning</u>
X'01'	(No error; buffer successfully printed)
X'23'	Failure in electronic circuit
X'25'	I/O device failure
X'29'	VRC error in text received on line
X'31'	Parity error in text received on line.

When the first character is other than X'01', BTAM prints message IEA000I (I/O ERR) at the central computer console (and/or teleprocessing or other console, if the operating system includes multiple console support). See Appendix C for the format of this message.

Negative response (second character is Circle N)

<u>First Character</u>	<u>Meaning</u>
X'04'	Terminal is in Bid mode
X'02'	Terminal is in Communicate mode
X'20'	Terminal is in Communicate mode with document device down
X'10'	Terminal is in Local mode
X'13'	Terminal is in Communicate mode but is out of paper
X'08'	Contents of buffer are being printed.

When the first character is X'10, X'13', or X'20', BTAM posts the operation complete-with-error (completion code X'41' in DECSDECB) and prints message IEA000I (I/O ERR) at the central computer console (and/or teleprocessing or other console, if the operating system includes multiple console support). See Appendix C for the format of this message. When the first character is X'02', X'04', or X'08', BTAM posts the operation with normal completion.

IBM 2740 (BASIC)

Read and Write operations for the basic 2740 require no terminal lists.

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows. This is the only macro used to receive text.

1. Write EOT
2. Prepare
3. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state, then writes message text. This is the only macro used to send text.

1. Write EOA and 15 Idle Characters
2. Write Text
3. Write EOT (TIR only)

IBM 2740 WITH CHECKING FEATURE

The macro instructions in this section apply to a 2740 used as an operator's console (under the Multiple Console Support option of OS/VS as well as to a 2740 used as a regular terminal.

Read and Write operations for the 2740 with the checking feature require no terminal lists.

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial and Reset (TIR)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows.

1. Write EOT EOT EOT
2. Prepare
3. Sense
4. Read Text
5. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)

READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)

READ Repeat (TP)

READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Notes

1. Each outgoing message block must end with EOB.
2. Once it is in receive mode, the terminal cannot begin sending message text.

until it receives EOT. Therefore, following one or more Write operations, you must arrange to send EOT to put the terminal in stand-by mode. This may be done by specifying the reset option in the last Write operation (i.e., TIR, TTR, or TVR), or by following the last Write operation by a Write TN macro.

Restriction: If a Read Initial operation immediately follows a Write with Reset operation, the first byte of data may be lost.

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state and turn on the terminal motors, writes message text, and reads the response.

1. Write EOA and 15 idle Characters
2. Write Text
3. Read Response
4. Write EOT (TIR only)

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT (TTR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets it to control mode, monitors the line for an EOA from the terminal and reads the message block that follows.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Prepare
5. Sense
6. Read Text

for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT, or after one or more Write operations, when you wish to begin receiving from the terminal via Read operations (the terminal cannot begin sending text until it receives EOT).

1. Write EOT EOT EOT

Programming Note: WRITE TTV performs exactly the same functions as would be performed by a WRITE Continue (TT) followed by a READ Initial (TI), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is

IBM 2740 WITH DIAL-UP FEATURE

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial and Reset (TIR)

READ Initial answers a call from a terminal and reads a message block.

1. Disable
2. Enable
3. Prepare
4. Read Text
5. Write EOT (TIR only)
6. Disable (TIR only)

READ Conversational (TV)

READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message text that follows. This macro is for reading a message block from a terminal

after a previous READ or WRITE macro has established the line connection.

1. Prepare
2. Read Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, and writes message text.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state and writes message text. This macro is for use following a Read operation to change from receiving text to sending text, when the line connection is already established.

1. Write EOA
2. Write Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP AND CHECKING FEATURES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
-----  
| DIALST,0  
-----
```

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
-----  
| DIALST,dialcount,dialchars  
-----
```

A Write Initial operation that encounters an answering list, which is defined by coding the operand field of a DFTRMLST macro instruction as follows:

```
-----  
| DIALST,0  
-----
```

allows a write to answer a call from a terminal and immediately initiates data transfer to the terminal. In this case, no data (EOA) is expected from the calling terminal.

READ MACRO INSTRUCTIONS

READ Initial (TI) READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, monitors the line for an EOA sent by the terminal, and reads the message block that follows.

1. Disable
2. Enable
3. Prepare
4. Read Text
5. Write EOA EOT (TIR only)
6. Disable (TIR only)

READ Continue (TT) READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP) READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV) READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message block that follows. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT
2. Prepare
3. Read Text
4. Write EOA EOT (TVR only)
5. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI) WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads the response to text.

1. Disable
2. Dial Dial digits (TI and TIR)
3. Enable (TI with Answer List only)
4. Write Pad characters
5. Write EOA
6. Write Text
7. Read Response
8. Write EOT (TIR only)
9. Disable (TIR only)

WRITE Continue (TT) WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT (TVR only)
5. Disable (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets it to control mode, monitors the line for an EOA from the terminal and reads the message block that follows.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Prepare
5. Sense
6. Read Text

Programming Note: WRITE TTV performs the same functions as would be performed by a WRITE Continue (TT) followed by a READ Con-

versational (TV), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (that is, by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP AND TRANSMIT CONTROL FEATURES

CAUTION: A 2740 having the Transmit Control feature is equipped with a Transmit Control switch. This switch must always be in the MTC position when the 2740 is under BTAM control.

DEFINING TERMINAL LISTS

A Read Initial or Write Initial operation that answers a call from a terminal requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
-----  
DIALST,0  
-----
```

A Read Initial or Write Initial operation that calls a terminal requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
-----  
DIALST,dialcount,dialchars  
-----
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Calling List)

READ Initial dials a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOT (TIR only)
8. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering List)

READ Initial answers a call from a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Enable
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOT (TIR only)
8. Disable (TIR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write Selection sequence
2. Read Response
3. Read Text
4. Write EOT (TVR only)
5. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Calling List)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, and writes message text to the terminal.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Answering List)

WRITE Initial answers a call from a terminal, writes an EOA to place it in receive state, and writes message text to the terminal.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes message text to the terminal. This macro is for use fol-

lowing a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP, TRANSMIT CONTROL,
AND CHECKING FEATURES

CAUTION: A 2740 having the Transmit Control feature is equipped with a Transmit Control switch. This switch must always be in the MTC position when the 2740 is under BTAM control.

DEFINING TERMINAL LISTS

A Read Initial or Write Initial operation that answers a call from a terminal requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
-----  
DIALST,0  
-----
```

A Read Initial or Write Initial operation that calls a terminal requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
-----  
DIALST,dialcount,dialchars  
-----
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Calling List)

READ Initial dials a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOA EOT (TIR only)
8. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering List)

READ Initial answers a call from a terminal, writes a selection sequence, and if the response is positive, reads a message block.

1. Disable
2. Enable
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text

7. Write EOA EOT (TIR only)
8. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT EOT EOT
2. Write Selection sequence
3. Read Response
4. Read Text
5. Write EOA EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Calling List)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text to the terminal, and reads a response from the terminal.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT (TIR only)
8. Disable (TIR only)

WRITE Initial (TI)

WRITE Initial and Reset (TIR)
(Using Answering List)

WRITE Initial answers a call from the terminal, writes an EOA to place it in receive state, writes message text to the terminal, and reads a response from the terminal.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT (TIR only)
8. Disable (TIR only)

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT (TVR only)
5. Disable (TVR only)

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write EOT
2. Disable

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{ OPENLST }
{ WRAPLST }, (xx, ...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST ,xx
```

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling character
3. Write Space character
4. Read Response
5. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response is positive, writes an EOA to set the terminal to receive state and writes message text to the terminal. If the terminal sends a negative response to addressing, the operation is posted complete.

1. Write EOT EOT EOT Circle S
2. Write Addressing character
3. Write Space character
4. Read Response
5. Write EOA
6. Write Text
7. Write EOT EOT EOT (TIR only)

IBM 2740 WITH STATION CONTROL AND CHECKING FEATURES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{(OPENLST  
{(WRAPLST), (xx, ...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST,xx
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling character
3. Write Space character
4. Read Response
5. Read Text
6. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)

READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)

READ Repeat (TP)

READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes EOA followed by message text and reads the response from the terminal.

1. Write EOT EOT EOT Circle S
2. Write Addressing character
3. Write Space character
4. Read Response
5. Write EOA
6. Write Text
7. Read Response
8. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal

interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 2760 OPTICAL IMAGE UNIT -- GENERAL INFORMATION

OPERATION AND MESSAGE FORMATS

This section describes the essential functions of the IBM 2760 Optical Image Unit and explains their relationship to the formats of messages to and from the computer. (See the 2760 component description listed in the Preface for detailed information on this terminal device.)

FILMSTRIP POSITIONING

After the operator inserts the filmstrip cartridge into the front of the 2760, the filmstrip drive mechanism positions the filmstrip in accordance with instructions from the user program. These instructions are received in the form of a message of predefined format, called a frame change message. Three characters, designated F, A₁, and A₂, determine how the filmstrip is to be positioned. The F (function) character specifies the direction of filmstrip movement (this character has other functions, discussed below). The A₁ and A₂ characters are codes indicating the amount (that is, number of frames) of filmstrip movement. Figure 22 gives the meanings of the possible values of the F character; Figure 23 gives the codes for the A₁ and A₂ characters.

MODES OF OPERATION

In addition to indicating the direction of filmstrip movement, the F character designates whether the unit is to operate in Manual or Automatic EOM mode and whether or not in Manual Frame Advance mode.

F Char.	Film Movement	Mode of Operation	Manual Frame Advance
Space	Reverse	Auto EOM	Disabled
1	Forward	Auto EOM	Disabled
2	Reverse	Manual EOM	Disabled
3	Forward	Manual EOM	Disabled
4	Reverse	Auto EOM	Enabled
5	Forward	Auto EOM	Enabled
6	Reverse	Manual EOM	Enabled
7	Forward	Manual EOM	Enabled

Figure 22. F (Function) Character Codes

Frames of Film Movement or Number in Image Index Counter	Character Transmitted A ₁ /I ₁	Frames of Film Movement or Number in Image Index Counter	Character Transmitted A ₂ /I ₂
0	Space (C Bit)	0	Space (C Bit)
32	@	1	@
64	—	2	—
96	&	3	&
128	l	4	l
160	/	5	/
192	j	6	j
224	a	7	a
256	2	8	2
288	s	9	s
320	k	10	k
352	b	11	b
384	3	12	3
416	t	13	t
448	ℓ	14	ℓ
480	c	15	c
		16	4
		17	u
		18	m
		19	d
		20	5
		21	v
		22	n
		23	e
		24	6
		25	w
		26	o
		27	f
		28	7
		29	x
		30	p
		31	g

Figure 23. A₁/I₁ and A₂/I₂ Character Codes

Manual vs. Automatic EOM Mode

The 2760 operates in one of two modes when sending a message to the computer. In automatic EOM (end of message) mode the message contains a single set of response point coordinates (explained below), and the message is sent automatically when the terminal operator probes a response point.

In manual EOM mode, more than one set of response point coordinates can be sent in the same message. In this mode, the terminal operator indicates the end of the message by probing the End Entry response point (one of the three Utility response points).

Manual Frame Advance

Positioning of the filmstrip is ordinarily performed upon instruction from the user program. The filmstrip may alternatively be positioned by the terminal operator; this, in conjunction with Manual EOM mode, allows the operator to enter a message containing response points from more than one image (frame). Use of this feature results in fewer program interruptions for repositioning the filmstrip and is therefore more economical of CPU time. Its use also simplifies logical program organization in that the program does not have to provide a frame change message for every possible circumstance.

The terminal operator moves the filmstrip by means of the Film switch on the front of the 2760. He can use the switch only when it has been made operational ("enabled") by the program. The F character sent by the program determines whether the switch is enabled or disabled.

RESPONSE POINTS AND COORDINATES

Each of the two halves of the Optical Image Unit screen, the image screen (right half) and auxiliary screen (left half) has 120 possible response points, in a 10 (horizontal) by 12 (vertical) matrix. Only a few, or perhaps one, of these points will be utilized in any given image (frame) or overlay. Each response point is represented by a set of vertical and horizontal (V and H) coordinates. It is these coordinates that are sent to the computer when the operator probes a response point. The user program must contain a table that associates with each valid response point some value or bit setting representing the response probed by the operator.

Figure 24 gives the V and H coordinates for each of the response points. Each coordinate is represented by a character, which on the communication line is repre-

sented by the corresponding bit pattern in transmission code.

Utility Response Points

Three special response points appear in a vertical row to the right of the image screen. The Load response point, when probed, indicates to the user program that the terminal operator has inserted a filmstrip cartridge into the Optical Image Unit. Upon receiving the V and H coordinates of this response point, the user program should send a frame change message to the 2760 that causes it to advance the filmstrip to the first frame.

The Unload response point, when probed, indicates to the user program that it should send a message that retracts the filmstrip into the cartridge and ejects the cartridge.

The End Entry response point is probed at the end of each message, when the 2760 is operating in Manual EOM Mode.

IMAGE INDEX COUNTER

Within the Optical Image Unit is an electronic counter called an Image Index Counter, which is incremented and decremented in step with filmstrip movement. The counter thus maintains a continuous record of which frame of the filmstrip is currently being projected. The content of this counter is transmitted to the computer at the beginning of each message to the computer or, when manual frame advance is being used, at the beginning of each sequence of response points from a given image. The two characters representing the content of the counter are designated I_1 and I_2 .

Upon receipt by the computer of each message from the 2760, the user program should check the I_1 I_2 characters to ensure that the correct image is being displayed.

Figure 23 shows the characters that represent the numerical content of the image index counter.

If the program sends to the 2760 a message that instructs the mechanism to move the filmstrip to a point that is beyond its last frame, an interlock is activated that prohibits filmstrip movement. This in turn prevents the image index counter from being incremented.

MESSAGE FORMATS

Each message from the computer to the IBM 2760 begins with the sequence EOA PRE o (end-of-address, prefix, lowercase o). The

		Horizontal Positions																						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
L	15	ld	lm	lu	l4	lc	ll	lt	l3	lb	1	lk	ls	l2	la	lj	l/	ll	l&	l-	l@			
T	t5	td	tm	tu	t4	tc	tl	tt	t3	tb	2	tk	ts	t2	ta	tj	t/	tl	t&	t-	t@			
3	35	3d	3m	3u	34	3c	3l	3t	33	3b	3	3k	3s	32	3a	3j	3/	3l	3&	3-	3@			
B	b5	bd	bm	bu	b4	bc	bl	bt	b3	bb	4	bk	bs	b2	ba	bj	b/	bl	b&	b-	b@			
K	k5	kd	km	ku	k4	kc	kl	kt	k3	kb	5	kk	ks	k2	ka	kj	k/	kl	k&	k-	k@	k space	UNLOAD	
S	s5	sd	sm	su	s4	sc	sl	st	s3	sb	6	sk	ss	s2	sa	sj	s/	sl	s&	s-	s@	s space	LOAD	
2	25	2d	2m	2u	24	2c	2l	2t	23	2b	7	2k	2s	22	2a	2j	2/	2l	2&	2-	2@			
A	a5	ad	am	au	a4	ac	al	at	a3	ab	8	ak	as	a2	aa	aj	a/	al	a&	a-	a@	a space	END ENTRY	
J	j5	jd	jm	ju	j4	jc	jl	jt	j3	jb	9	jk	js	j2	ja	jj	j/	jl	j&	j-	j@			
/	/5	/d	/m	/u	/4	/c	/l	/t	/3	/b	10	/k	/s	/2	/a	/j	//	/l	/&	/-	/@			
1	15	ld	lm	lu	l4	lc	ll	lt	l3	lb	11	lk	ls	l2	la	lj	l/	ll	l&	l-	l@			
&	&5	&d	&m	&u	&4	&c	&l	&t	&3	&b	12	&k	&s	&2	&a	&j	&/	&l	&&	&-	&@	(H) Horizontal Coordinates*		
	5	D	M	U	4	C	L	T	3	B		K	S	2	A	J	/	l	&	-	@			

(V) Vertical Coordinates* Auxiliary Screen Vertical Positions Image Screen

* Shown in upper case for ease of reading. The 2760 operates in lower case shift automatically. No case shift characters are required or permitted in messages to or from the unit.

Figure 24. V & H (Vertical & Horizontal) Response Point Coordinate Codes

PRE o characters indicate that the message is intended for the 2760 and not for the IBM 2740 with which it is associated. Similarly, each message from the 2760 to the computer begins with EOA PRE o. The user program should check the input area for the presence of these two characters. (The PRE o sequence is represented by X'3E4C' in transmission code.) Each message in either direction ends with an EOB (end-of-block) character.

Figure 25 presents the formats for messages between the computer and the Optical Image Unit.

Computer to IBM 2760

Output messages, referred to as frame change messages, direct the Optical Image Unit to move the film forward or backward, set the mode for the subsequent response message, and specify the amount of filmstrip travel (that is, number of frames). See Figure 22 for the meanings of the possible values of the F character; see Figure

23 for the coded values representing amount of filmstrip movement (A₁ and A₂ characters).

IBM 2760 to Computer

Input messages, referred to as response messages, indicate to the user program which filmstrip frame is being displayed for the current response, and give the V and H coordinates of the response point or points the operator has probed.

Auto EOM Mode: In this mode, each probe action by the terminal operator causes a complete message, containing the coordinates of one response point, to be sent to the computer.

Manual EOM Mode, Film Switch Disabled: In this mode, a message may contain any number of response point coordinates. The sequence EOA PRE o I₁ I₂ V₁ H₁ is sent to the computer when the first response is probed. Each subsequent set of V and H coordinates is sent individually as each

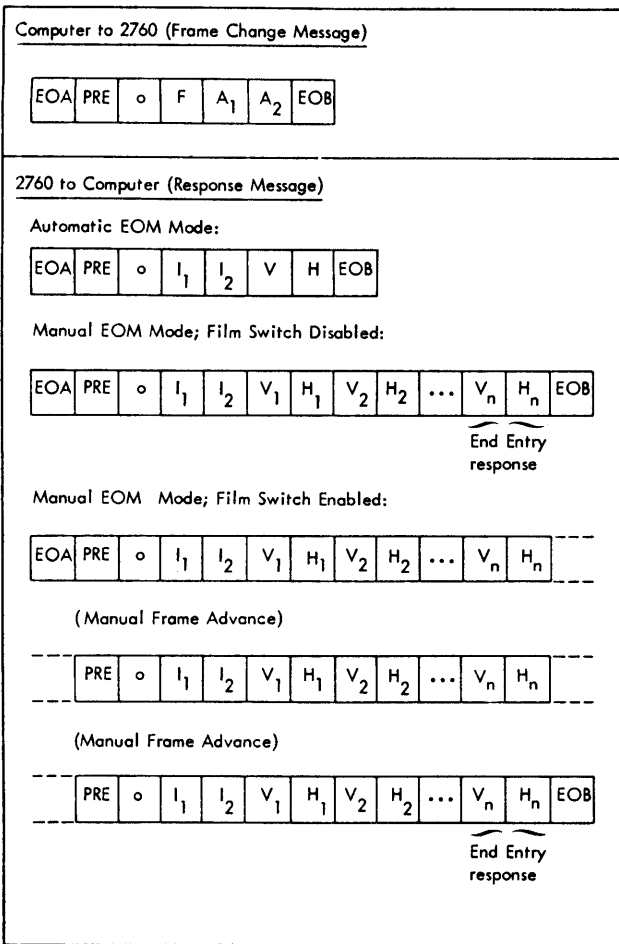


Figure 25. Message Format

response point is probed. The EOB character is automatically sent after the coordinates when the End Entry response point is probed.

The user program should check for the End Entry response point to ensure that the message contains the correct number of responses.

Manual EOM Mode, Film Switch Enabled: In this mode, the terminal operator may reposition the filmstrip to a new frame while entering response points, so that the input message contains responses from more than one image. The image index counter is incremented or decremented each time the filmstrip is moved.

The first probe action by the terminal operator following manual positioning of the filmstrip causes the sequence PRE o I₁ I₂ V₁ H₁ ... to be sent to the computer. Figure 25 illustrates the message resulting from entering response points for three different images.

The only times the operator cannot move the filmstrip, when the Film Switch is enabled, are when the image index counter contains a value of 2 or less, in which case reverse movement is inhibited; when the 2760 detects the hole in the tenth trailer frame, in which case forward movement is inhibited; and when the 2760 is at that moment receiving or executing a message from the computer.

SIGNALS TO OPERATOR

Two kinds of signals inform the terminal operator that his probe actions are correct. One is a visible indicator: the Online light. The second is an audible tone. The use of these signals differs for Automatic EOM, and Manual EOM mode.

Automatic EOM Mode

When the operator enters responses in this mode, the Online light comes on when he probes the response point. The audible tone sounds and the light goes out when the computer has returned a positive answerback, indicating that it received the message without error. The positive answerback may be a circle Y or an EOA (see Error Detection and Recovery).

In describing 2760 operations, the term answerback is used instead of response, to avoid confusion with responses entered by the terminal operator.)

Manual EOM Mode

In this mode, the Online light comes on when the terminal operator probes the first response point and remains on throughout the remainder of the message. This should alert the operator that the terminal is in Manual EOM mode and accordingly is subject to the inter-character time-out imposed by the transmission control unit. That is, each subsequent response should be made within that time limit. It may be well to specifically inform the operator of the time limit by means of a suitable phrase on the image, for each frame for which the limit applies. The Online light goes off when the computer replies with a circle Y, EOA, or EOT.

The audible tone sounds as each response point is probed, to inform the operator that the response has been sent to the computer and that he may probe again. The final probe action (End Entry) results in the tone only after the computer replies with a circle Y or EOA. This signifies to the operator that the complete message was received without error.

If a response point is incorrectly probed, the tone does not sound and the Probe Check light appears. This indicates that the probe touched the screen at too small an angle from the vertical or that it touched outside the designated response point area.

ERROR DETECTION AND RECOVERY

Messages to and from the IBM 2760 are checked for errors by the Record Checking facility of the 2740 to which the 2760 is attached. In addition, the 2760 checks messages from the computer for proper length; all messages from the computer have the same length.

If a text error or record length error is detected in a frame change message, the 2760/2740 sends a circle N (negative answerback) character in response to the EOB that ends the frame change message. If the EROPT operand of the DCB macro for the line group specifies W (retry of write-text errors), BTAM error recovery procedures cause the frame change message (without the EOA character) to be retransmitted up to two more times. If the error condition persists beyond the three attempts, or if EROPT does not specify W, the error condition is posted in the data event control block (DECB) for the line.

If the frame change message is received without error, the 2760/2740 sends a circle Y (positive answerback) character. BTAM responds to the circle Y with an EOT, which causes the 2760 to execute the instructions contained in the frame change message.

If a text error is detected in a response message, and the EROPT operand of the DCB macro for the line group specifies R (retry of read-text errors), BTAM error recovery procedures send a circle N to the 2760/2740, then reread the response message. If the Optical Image Unit is in Automatic EOM mode, it automatically resends the message; if it is in Manual EOM mode, the terminal operator must re-enter the entire response message.

The error recovery procedures respond with circle N and reread the message up to two more times. If the error condition persists beyond the three attempts, or if EROPT does not specify R, the error condition is posted in the DECB for the line.

If the response message is received without error, the operation on which the message was received is posted complete. The EOA character that begins the next frame change message serves as a positive answerback to the 2760/2740.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is activated but there is no Read command in effect to receive the data. Should this occur, the next operation executed for the line to which that 2760 is connected will be posted complete-with-error, indicating that probe data was lost. In order to recover from an error of this kind, you may wish to take one of these suggested actions:

- Issue a WRITE TCO macro that moves the filmstrip to an error-handling frame that will aid the terminal operator in recovering the lost data.
- Issue a WRITE TV macro to write an error message on the printer of the 2740 to which the 2760 is attached.
- Issue a READ TI macro (nonswitched line) or READ TV macro (switched line) to read the next message from the terminal. The operator should be instructed to re-probe the previous response if he does not hear the audible tone within a reasonable interval after probing.

For information on error indicators at Optical Image Unit, see the IBM 2760 Optical Image Unit, Component Description.

ONLINE TESTING

Online tests for the IBM 2760 are initiated at the IBM 2740 terminal keyboard. The test request message can be keyed in whenever the user program issues a READ macro instruction (other than READ Skip) without the Reset option, or it may be keyed in after the filmstrip has been moved by a WRITE TCO macro. In order to use the online test facility, the EROPT field in the DCB for the line group must specify T.

Two tests are available for the 2760: frame change test (type 10) and scan point test (type 11). They are designed to test the filmstrip transport mechanism and the probe response accuracy of the 2760. See Test Type Codes in the section "Online Testing" for descriptions of these tests.

Online tests will generally be run by the IBM Customer Engineer during periods of inactivity or as a startup procedure. Once the testing is completed, the Customer Engineer will unload the filmstrip and the operator can continue operation.

The terminal operator can also interrupt his data entry operation at any time to run a terminal test. However, some provision must be made to reposition the filmstrip to the frame being displayed when the test was

begun, and to reset the modes in effect at the time. A recommended way of doing this is for the user program to save (1) the image index value expected in the response message that would have been received had not the test request message been received instead, and (2) the F-character contained in the last previous frame change message.

Then, upon conclusion of the test, the program would (1) calculate the difference between the values of the image index at the beginning and at the end of the test, and from this difference determine the A_1 and A_2 characters to be sent in the next frame change message; and (2) determine, from the modes at the beginning of the test and the sign of the difference in the image index values, the appropriate F-character for the frame change message. Sending that message to the 2760 would properly position the filmstrip and set the correct modes.

As an example, assume that the filmstrip was positioned at frame 27 at the beginning of the test and that the modes in effect were Manual EOM, Manual Frame Advance disabled. If the previous filmstrip movement had been in the forward direction, the last F-character sent would be 3 (see Figure 14). Then assume that at the end of the test the filmstrip was positioned at frame 44. To reposition it to frame 27 would require a reverse movement of 17, hence the A_1 , A_2 characters in the next frame change message would be Space U (see Figure 15) and the F-character would specify reverse direction. To restore the modes to their original settings (Manual EOM, Manual Frame Advance disabled), the F-character should be 2. In computing the F-character, it is useful to know that the filmstrip direction is determined by bit 6 (0=reverse, 1=forward), the Manual Frame Advance mode is determined by bit 4 (0=disabled, 1=enabled), and the EOM mode is determined by bit 5 (0=automatic, 1>manual).

To request one of the 2760 tests, the following message must be keyed from the 2740:

9 9 9 9 9 x x 4 F A_1 A_2 EOT

xx = 10 for frame change test
 = 11 for scan point test

4 specifies that the test is for the 2740 terminal (of which the 2760 is a component)

F = function control character

A_1 A_2 = amount of film movement

If the operation in effect when a test request message is entered is a Write Invi-

tational Optical (TCO) operation, the on-line test facility performs, at the conclusion of the test, a Read Initial (TI) operation (nonswitched line) or Read Conversational (TV) operation, to receive the next regular message block.

PROGRAMMING CONSIDERATIONS

General Steps for Preparing the User Program

1. Define the information the user program is to obtain from the terminal operator.
2. Divide the required information into questions and statements that are meaningful to the terminal operators and that can be answered by probing response points on the Optical Image Unit screen. Responses may be Yes-No choices, multiple choices, alphabetic or numeric data, etc.
3. Design the sequence of questions to request the information in the most efficient order. Make sure that the questions are coordinated so that each piece of information is obtained at the proper point in the data entry procedure.
4. Make a preliminary design of all the filmstrip frames that request information. Decide on the wording of the questions and the wording and location of the response points.
5. Design on the basis of the frame layouts, a system of tables that will enable the user program to recognize the valid and invalid responses to a question on a frame.
6. Make a final design of all the frames that request information, and modify the tables as necessary to make them more efficient.
7. Arrange the filmstrip layout to minimize film movement (for example, error frames and other frequently displayed frames should be in the middle of the filmstrip).
8. Design a method of initialization so that the program can locate the frame containing the first application image that requests information from the operator even if some of the leader frames have been cut off. The section, Initializing Images, suggests a means for doing this.
9. Design the frames necessary for initialization.

10. Code the user program on the basis of the tables and frames. The program must include the initialization routine as well as the error routine for detecting and notifying the operator of invalid responses.

Initializing Images

Once the filmstrip cartridge is inserted in the Optical Image Unit, five steps of the filmstrip drive mechanism are required to advance the filmstrip out of the cartridge to the point where the first frame can be displayed. It is recommended that three blank frames be left at the beginning of the filmstrip to serve as a leader, to absorb the greater wear that the beginning of the strip receives. The first frame beyond the blank frames is called the initializing image. This image contains a probe response point whose position is unique for that filmstrip, thus serving to identify the filmstrip to the program so that the correct data entry procedure routine can be determined.

A further recommendation is that several initializing images be used, so that a new leading edge can be trimmed on the filmstrip as the original edge becomes worn through use. Each of the initializing images would have response points whose meanings are the same, but whose position differs with each frame. Thus the program, when receiving a response message following the first frame change message, can identify which of the initializing images is being displayed. This information can then be used to modify the Image Index values received in subsequent response messages, thereby compensating for any change in position of the application images (relative to the leading edge of the filmstrip) caused by removing worn frames.

Assume, for example, that a new filmstrip has three blank frames, followed by three initializing frames. The first frame change message would specify a film movement of eight frames. When the filmstrip is new, this message causes the first initializing image to be projected. When the first blank frame is cut off, to provide a new leading edge, the same frame change message will cause the second initializing image to be displayed. Since all subsequent application images are now one frame closer to the leading edge of the filmstrip, it is necessary to subtract one from each image index value received in response messages, in order for the program to correctly identify the frame being displayed for the response points received. By determining which initializing image is being displayed, the program can set the proper decrement value in an index register to modify the received image index values.

Startup Procedure

When the terminal operator is ready to begin a data entry operation, he inserts the appropriate filmstrip cartridge into the Optical Image Unit and probes the Load response point. (If the terminal is connected to the computer by a switched line, he must dial the computer before probing the Load response point.) For either a switched or a nonswitched line, the Standby light on the 2740 must be on and a Read Initial operation must be in progress at the computer. When the operator probes the Load response point, a message containing the coordinates of that point is transmitted to the computer. The program should check the input area for the presence of the Load coordinates, s Space (X'2501'), and upon detecting them, issue a WRITE macro that sends the appropriate frame change message to the 2760. The A₁ and A₂ characters in the frame change message should specify sufficient frame movement to cause the initializing image to be displayed. Assuming that the initializing image is the fourth frame from the beginning of the filmstrip, a forward movement of eight frames is required to position that image for projection.

When the operator probes the response point for the initializing image, the program should issue a frame change message that positions the filmstrip to the first application image to be viewed by the operator. If more than one initializing image is used, as recommended under Initializing Images, the particular frame being viewed by the operator when he enters his response determines the value of the A₁, A₂ characters in the frame change message. For example, assume that the first application image to be viewed is ten frames from the first initializing image. A response entered for the first initializing image should cause the next frame change message to specify eight frames of filmstrip movement. If the response was entered for the second initializing image, however, the frame change message would have to specify a filmstrip movement of seven frames to reach the first application image.

The values of A₁, A₂ in all subsequent frame change messages would be independent of which initializing image was displayed.

Receiving Multiple Message Blocks

Following receipt of a message block from a 2760, the computer must reply with a positive answerback. In the usual conversational operation, in which a sequence of WRITE TCO macros alternates the sending of frame change messages and the receiving of response messages, the macro itself supplies as the positive answerback the EOA

character that begins the next frame change message.

If, however, instead of alternating messages in this manner you wish to receive a message from the 2760 and then receive a message block from the 2740 or the 2760, you should use the READ Continue with Leading Acknowledgment (TTA) macro. This macro sends a positive acknowledgment to the 2760 and then receives another message block from the terminal.

The recommended method for receiving multiple probes from the same filmstrip frame is to specify, in the frame change message that positions that frame, a function character specifying Manual EOM mode. If more than one message block is required to accommodate the response data to be entered from that frame, you should specify Manual EOM mode and receive subsequent response messages by means of a READ TTA macro or a WRITE TCO macro that specifies no filmstrip movement.

You should not issue a READ Continue (TT) macro following receipt of a response message from the 2760, for the following

reason. READ TT sends a circle Y (positive answerback) to the 2760, which causes the 2760 to return an EOT to the computer and to release the probe interlock, which allows the terminal operator to probe response points. When the Read Continue operation receives the EOT, the operation is posted complete; therefore, no Read operation is in effect to receive the next message from the terminal. (Both circle Y and EOA are recognized by the terminal as a positive answerback; however, the Y causes the terminal to reply with an EOT but the EOA does not.)

Sending Message Blocks Alternately to the 2760 and 2740

If it is necessary to send message blocks alternately to the 2760 and the 2740, the message block to the 2760 should be sent first, followed by the message block to the 2740. If the line is not put in control mode after sending to the 2760, the message block to the 2740 must begin with text and must end with an EOB. If the line is put in control mode after sending to the 2760, the message block to the 2740 must begin with an EOA and must end with an EOB.

IBM 2740 WITH CHECKING FEATURE AND 2760
OPTICAL IMAGE UNIT

Read and Write operations for the 2740 with the checking feature and 2760 Optical Image Unit require no terminal lists.

Once the line group data set has been opened, a READ Initial macro may be issued to establish contact with the 2740 to which the 2760 is attached. If the terminal operator wishes to send from the 2740 keyboard, he presses the Bid key and enters the data. If he wishes to begin data entry with the Optical Image Unit, he inserts the appropriate cartridge into the front of the unit and touches the probe to the Load response point. Either action causes the data to be placed in main storage at the location specified by the area operand of the READ macro.

The user program can determine whether the message came from the keyboard or the Optical Image Unit by testing for the presence of the PRE o characters at the beginning of the input area.

All data sent to the Optical Image Unit is in the form of the fixed-length message EOA PRE o F A₁ A₂ EOB, where F represents the function control character and A₁ and A₂ are characters designating the amount of filmstrip movement as explained under "IBM 2760 Optical Image Unit -- General Information."

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows.

1. Write EOT EOT EOT
2. Prepare
3. Sense
4. Read Text
5. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)

READ Continue with Leading Acknowledgment (TTA)

READ Continue with Leading Acknowledgment is for use when you wish to positively acknowledge a message, reset the terminal to standby status, and receive message text from either the 2760 or the 2740.

READ TTA should be used specifically to receive message text from the 2740 or 2760 following receipt of text from the 2760. The EOA is a positive acknowledgment (answerback) to the message block received from the 2760. The EOA sounds the audible tone and activates the 2760 probe by releasing the probe interlock. The EOT sequence resets the terminal to standby status so that either the 2760 or the 2740 may enter message text.

1. Write EOA EOT EOT EOT
2. Prepare
3. Sense
4. Read Text

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state and turn on the terminal motors, writes message text, and reads the response.

1. Write EOA and 15 Idle characters
2. Write Text
3. Read Response
4. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT EOT EOT (TVR only)

WRITE Initial Optical (TIO)

This option is for use when you wish to send a frame change message, but do not require a response from the terminal operator (as when retracting the filmstrip and ejecting the cartridge at the end of a data entry operation), or when you wish to receive the response using a subsequent macro. The macro writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TIO macro the address of the main storage location containing the three-character F A₁ A₂ sequence.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT

Programming Note: If input from the 2760 is expected following execution of the Write Initial Optical operation, you should issue a READ Initial (TI) macro immediately

after completion of the Write TIO operation.

WRITE Invitational Optical (TCO)

This option is for use when you wish to send a frame change message and read a response message from the terminal. The macro functions identically to the WRITE Initial Optical (TIO) macro, but in addition receives message text from the Optical Image Unit or the 2740 keyboard. The Prepare command (see below) monitors the line for an EOA character; when it is received, the Read Text command reads into the input area that follows the EOA.

You must specify in the entry operand of the WRITE TCO macro the address of the main storage location containing the F A₁ A₂ sequence; in the area operand you must specify the address of the input area into which the response message is to be received. If dynamic buffering is used to read the response message, you should specify the length operand as 'S'.

The WRITE TCO macro is the principal macro used in a 2760 application, as it is a convenient means for alternately sending frame change messages and receiving responses from the operator.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT
6. Prepare
7. Sense
8. Read Text

Examples of WRITE TIO and WRITE TCO

Figure 26 illustrates how WRITE TIO and WRITE TCO are coded. The WRITE TIO macro sends a frame change message to move the film forward six frames. The WRITE TCO macro sends a frame change message to move the filmstrip forward 37 frames and then reads a response message from the terminal. (The A₁ character, @, represents a film movement of 32 frames (see Figure 23); its hexadecimal equivalent is X'20'. The A₂ character, /, represents a movement of five frames; its hexadecimal equivalent is X'0B'. Together, the two characters specify a film movement of 37 frames.)

In each case, the F character, 1 (X'02'), specifies (in addition to forward movement) that the response from the 2760 is to be made in Automatic EOM mode with the Film switch (Manual Frame Advance) disabled.


```

WRITE  DECENAME,TIO,DCBNAME,,,FRMSG1,,MF=E
.
.
WRITE  DECENAME,TCO,DCBNAME,INAREA,20,FRMSG2,,MF=E
.
.
FRMSG1  DC  X'020143  (1 SP J (LOWERCASE))
FRMSG1  DC  X'02200B' (1 # /)
INAREA  DS  5F

```

Figure 26. Examples of WRITE TIO and WRITE TCO Macro Instructions (Nonswitched Line)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that

the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

This macro is also used to cause the 2760 to execute the instructions it received in the preceding frame change message.

1. Write EOT EOT EOT

IBM 2740 WITH DIAL-UP AND CHECKING FEATURES
AND 2760 OPTICAL IMAGE UNIT

Once the line group data set has been opened, either the terminal or the computer may establish the line connection. If the terminal is to establish the connection (that is, dial the computer), issue a READ Initial macro instruction that refers to an answering list. When the terminal operator is ready to enter data, he dials the telephone number of the computer.

If he wishes to send from the 2740 keyboard, he presses the Bid key and enters his data. If he wishes to begin data entry with the Optical Image Unit, he inserts the appropriate cartridge into the front of the unit and touches the probe to the Load response point. Either action causes the data to be placed in main storage at the location specified by the area operand of the READ macro.

The user program can determine whether the message came from the keyboard or the Optical Image Unit by testing for the presence of the PRE o characters at the beginning of the input area.

If the computer is to establish the switched line connection, the WRITE TIO macro, explained below, may be used if you wish to send a frame change message immediately following establishment of the line connection. (Alternatively, the connection can be made using a WRITE Initial macro, with the frame change message being sent by a subsequent WRITE TVO or WRITE TCO macro.)

All data sent to the Optical Image Unit is in the form of the fixed-length message EOA PRE o F A₁ A₂ EOB, where F represents the function control character and A₁ and A₂ are characters designating the amount of filmstrip movement, as explained under "IBM 2760 Optical Image Unit -- General Information."

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

DIALST,0

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

DIALST,dialcount,dialchars

(See WRITE Initial Optical macro for calling list required for that macro.)

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, monitors the line for an EOA sent by the terminal, and reads the message block that follows.

1. Disable
2. Enable
3. Prepare
4. Sense
5. Read Text
6. Write EOA EOT EOT EOT (TIR only)
7. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Circle Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)
4. Disable (TTR only)

READ Continue with Leading Acknowledgment (TTA)

READ Continue with Leading Acknowledgment is for use when you wish to positively acknowledge a message, reset the terminal to standby status, and receive message text from either the 2760 or the 2740.

READ TTA should be used specifically to receive message text from the 2740 or 2760 following receipt of text from the 2760. The EOA is a positive acknowledgment (answerback) to the message block received from the 2760, and causes the audible tone to sound and the 2760 probe to become activated (that is, releases the probe interlock). The EOT sequence resets the terminal to standby status so that either the 2760 or the 2740 may enter message text.

1. Write EOA EOT EOT EOT
2. Prepare
3. Sense
4. Read Text

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write Circle N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message block that follows. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT EOT EOT
2. Prepare
3. Sense
4. Read Text
5. Write EOA EOT EOT EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads the response to text.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT EOT EOT (TIR only)
8. Disable (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT EOT EOT (TVR only)
5. Disable (TV only)

WRITE Initial Optical (TIO)

This option is for use when you wish to establish the line connection and send a frame change message to the Optical Image Unit.

The macro issues a Disable command to disable the line in case this was not done previously, dials the terminal, and writes pad characters to provide time fill to allow the terminal motors to reach operating speed. The macro then writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters (F, A₁ and A₂) and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TIO macro the address of a terminal list defined by a DFTRMLST macro as follows:

LIST DFTRMLST DIALST,dialcount,
dialchars,faaseq

DIALST specifies the type of list; dialcount and dialchars specify the number of digits in the telephone number and the digits themselves; and faaseq specifies the three characters constituting the frame change message text. faaseq must be coded as the hexadecimal equivalent of the transmission code bit pattern for the desired characters.

1. Disable frame change charac-
2. Dial Dial digits

3. Write Pad characters
4. Write EOA PRE o
5. Write Frame change characters
6. Write EOB
7. Read Answerback
8. Write EOT EOT EOT

Programming Note: If input from the 2760 is expected following execution of the Write Initial Optical operation, you should issue a READ Conversational (TV) macro immediately after completion of the Write TIO operation.

WRITE Conversational Optical (TVO)

This option is for use when you wish to send a frame change message after the switched line connection has been established, but do not require a response from the terminal operator (as when retracting the filmstrip and ejecting the cartridge at the end of a data entry operation), or when you wish to receive the response using a subsequent macro. The macro writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TVO macro the address of the main storage location containing the three-character F A₁ A₂ sequence.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT

Programming Note: If input from the 2760 is expected following execution of the Write Conversational Optical operation, you should issue a READ Conversational (TV) macro immediately after completion of the Write TVO operation.

WRITE Invitational Optical (TCO)

This option is for use after the line connection has been established, when you wish to send a frame change message and read a response message from the terminal. The macro functions identically to the WRITE

Conversational Optical (TVO) macro, but in addition receives message text from the Optical Image Unit or the 2740 keyboard. The Prepare command (see below) monitors the line for an EOA character; when it is received, the Read Text command reads into the input area the data that follows the EOA.

You must specify in the entry operand of the WRITE TCO macro the address of the main storage location containing the F A₁ A₂ sequence; in the area operand you must specify the address of the input area into which the response message is to be received. If dynamic buffering is used to read the response message, you should specify the length operand as 'S'.

The WRITE TCO macro is the principal macro used in a 2760 application, as it is a convenient means for alternately sending frame change messages and receiving responses from the operator.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT
6. Prepare
7. Sense
8. Read Text

Examples of WRITE TIO, WRITE TVO, and WRITE TCO

Figure 27 illustrates how WRITE TIO, WRITE TVO, and WRITE TCO macros are coded. The WRITE TIO macro dials the telephone number of the 2740 terminal and sends a frame change message; the F A₁ A₂ sequence is coded in the DFTRMLST macro. In this example, the F character is a Space (X'01'), designating reverse movement, and A₁ and A₂ are both "c" (X'67'), representing a filmstrip movement exceeding the length of the filmstrip. This message therefore causes the filmstrip to be retracted and the cartridge ejected. The A₁ and A₂ characters could alternatively be coded as Space Space (X'0101'), representing zero filmstrip movement, then a subsequent WRITE TVO macro could be used to specify the filmstrip movement. This is useful where the amount of film movement may vary from one loading of the filmstrip to another, and so cannot be specified in a terminal list. The same WRITE TIO would be issued regardless of the film movement needed; the subsequent WRITE TVO would use register notation for the entry operand to provide the needed frame change characters.

The WRITE TVO macro sends a frame change message to move the film forward three frames. The F character, 1 (X'02') specifies Automatic EOM Mode with Film switch

```

WRITE    DECBNAME,TIO,DCBNAME,,,LIST,,MF=E
.
.
WRITE    DECBNAME,TVO,DCBNAME,,,FRMSG1,,MF=E
.
.
WRITE    DECBNAME,TCO,DCBNAME,INAREA,20,FRMSG2,,MF=E
.
.
LIST     DFTRMLST  DIALST,4,5003,016767   (SP C C (LOWERCASE))
FRMSG1   DC        X'020161'             (1 SP &)
FRMSG2   DC        X'0D2001'             (6 a SP)
INAREA   DS        5F

```

Figure 27. Examples of WRITE TIO, WRITE TVO, and WRITE TCO Macro Instructions (Switched Line)

(Manual Frame Advance) disabled. The response would be read by a different macro(a READ macro, for example), as the WRITE TVO macro does not read response messages.

The WRITE TCO macro sends a frame change message to move the filmstrip backwards 32 frames, sets the 2760 in Manual EOM mode with the Film switch (Manual Frame Advance) enabled, and reads a response message from the 2760 (or the 2740).

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT EOT EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a Write operation when you wish to break the line connection.

1. Write EOT EOT EOT
2. Disable

IBM 2741 COMMUNICATION TERMINAL

GENERAL INFORMATION

The line control scheme for the IBM 2741 differs from that for some other start-stop terminals (for example, the IBM 1050), in that the terminal and line do not alternate between control mode and text mode, and the polling and addressing functions are absent. Instead the 2741, when in communicate mode (all subsequent discussion presupposes this), alternates between two states: receive and transmit. The 2741 is in a third state, control-receive, between the time it sends an EOT and the time it receives an EOA or EOT from the computer; this state is also entered momentarily when the terminal power switch is turned on or when the mode switch is switched from local to communicate mode. In transmit state, the keyboard is unlocked and the terminal operator can key in data for transmission to the computer. In receive state the keyboard is locked and the terminal can only accept and print data received from the computer. The principal indicator of the state of the terminal is the keyboard. If it is unlocked, allowing the operator to enter data, the terminal is in transmit state; otherwise, it is in receive state.

The terminal alternates between states whenever an end-of-transmission (EOT) character is sent on the line. An EOT sent by the computer always places the terminal in transmit state, and an EOT sent by the terminal always places the terminal in receive state.

The terminal sends an EOT whenever the operator presses the Attention key or the Carrier Return key. These two keys accordingly are the means by which the terminal operator tells the computer that he has finished entering a line of data. The computer sends an EOT whenever it executes a channel program that is to receive data from the terminal, that is, any Read channel program.

Communication between terminal and computer is always initiated by the terminal operator, and can occur anytime after the program sets up the first Read operation, which must be a Read Initial. Conversation begins when the terminal operator sets the mode switch to Communicate (this action sends an end-of-address (EOA) character to the computer). For switched lines, the operator follows this by manually dialing the telephone number of the computer and switching the common-carrier data set to data mode. (See the IBM 2740/2741 Operator's Guide, for detailed dial-up procedures.)

Line control discipline for the 2741 differs from that for other start-stop terminals in the following significant respect. With most terminals, control of the communication line remains vested in the program, except during the relatively small proportion of time that the terminal is in text mode and is actually sending data. If the terminal stops sending data for a period of about 25 seconds, a time-out function in the terminal returns it to control mode. When this happens, the program is again able to initiate activity on the communication line, and the terminal begins monitoring the line for control signals from the computer. This control scheme prevents one terminal on a line from monopolizing use of the line so that the computer is unable to communicate with other terminals on the same line.

The 2741, on the other hand, is intended for conversational use: there is only one terminal per line, and input by the terminal operator and response by the computer alternate, as in an ordinary telephone conversation, until the terminal operator chooses to end the conversation. Each time the terminal is in transmit state, the terminal operator has control; that is, the program can initiate no new activity on the line until the terminal operator returns control to the program by sending an EOT character. The 2741 has no time-out function by which control can be returned to the computer.

Only the terminal operator can end a conversation, either by switching the terminal mode switch to Local or by turning off the terminal power switch.

The sequence of operations between the time the terminal operator begins and ends the conversation with the computer depends upon the logical structure of the program and upon the communications conventions established between terminal operator and the program. These in turn depend on the system application.

Although the differences between line control for the 2741 and for other start-stop terminals result in dissimilar channel programs, you code your READ and WRITE macro instructions in the same way as for other terminals, with the exception of the "entry" operand. Because 2741s use no terminal lists, "entry" is not used, and if coded, it is ignored.

Channel Commands for the IBM 2741

The functions of the commands comprising the channel programs for the 2741 are given below. (The commands are described as they apply to Read and Write operations for the

2741; no inferences should be drawn as to their applicability for other terminals.)

Write EOT Sets the terminal to transmit state.

Write EOA Sets the terminal to receive state.

Inhibit (1) Receives text from terminal into input area.
(2) Receives text from terminal but does not place it in input area. Used for purging the communication line of unneeded text data (used in Read Skip channel program).

Enable For switched lines, conditions the TCU to accept calls from terminals.

Disable When Disable is the first command of Read Initial or Write Disconnect (switched line), it disables the line if, through program logic error, the line is in the enabled condition when the current channel program is started. If the line is already in the disabled condition, which is the normal case, the Disable command has no effect.

Prepare For switched lines, causes the TCU to monitor the line for incoming data.

The function of an Inhibit command, like that of a Read command, is to receive data from a terminal. The difference is that a Read command is terminated by expiration of a timeout interval (if not terminated sooner by receipt of data), while an Inhibit is not ended in this way. Channel programs for the IBM 2741 use Inhibit commands, rather than Read commands, because in 2741 operation under BTAM, an indefinite period may elapse between initiation of a channel program and receipt of data from a terminal.

READ MACRO INSTRUCTIONS

READ Initial (TI) (Nonswitched Line)

Read Initial receives message text (beginning with EOA) from the terminal.

1. Prepare (receives EOA)
2. Inhibit (receives text)

READ Initial (TI) (Switched Line)

READ Initial disables the line (in case this was not done previously), enables it,

then receives message text (beginning with EOA) from the terminal.

1. Disable
2. Enable
3. Prepare (receives EOA)
4. Inhibit (receives text)

READ Continue (TT) READ Conversational (TV)

READ Continue and READ Conversational are identical operations. Each sets the terminal to transmit state, then receives message text (beginning with EOA) from the terminal.

1. Write EOT
2. Prepare (receives EOA)
3. Inhibit (receives text)

READ Skip (TS)

READ Skip receives message text from the terminal but does not place it in main storage.

1. Inhibit (received text is discarded)

WRITE MACRO INSTRUCTIONS

WRITE Continue (TT)

WRITE Continue sends a message segment to the terminal. It is for use after a WRITE Conversational has set the terminal to receive state.

1. Write Text

WRITE Conversational (TV)

WRITE Conversational sets the terminal to receive state and sends it a message segment. It is for use after a Read operation to reverse the direction of transmission.

1. Write EOA
2. Write Text

WRITE Continue Conversational (TTV)

WRITE Continue Conversational sends the terminal a message segment followed by EOT, which sets the terminal to transmit state, then receives message text (beginning with EOA) from the terminal. WRITE TTV is for use following a WRITE Continue or WRITE Conversational, to reverse the direction of transmission.

1. Write Text
2. Write EOT
3. Prepare (receives EOA)
4. Inhibit (receives text)

WRITE Disconnect (TN)
(Switched Line)

WRITE Disconnect disables the line to break off communication with the terminal.

1. Disable

DESIGNING A MESSAGE CONTROL ROUTINE

This section explains how Read and Write operations may be combined to permit conversational communication between terminal and computer.

The first operation, once the line group has been opened, must be a Read Initial.

When the terminal operator establishes communication, the first line of text he types is read into the input area. The Read Initial ends with receipt of the EOT character sent when the terminal operator presses the Return key or the Attention key.

The program should then determine whether it should receive more text from the terminal. This decision might be based on analysis of the data just received. For example, an operating convention might be established by which the terminal operator presses the Return key to signify that he has further input and the Attention key to indicate that he has finished sending. The Return key causes transmission of the new line (NL) character followed by EOT. The Attention key causes transmission of the EOT character only. The program can check the last two characters received from the terminal to determine the action to take: NL EOT indicating that a Read operation should be executed to receive the next text segment from the terminal operator; EOT alone indicating that the program should reply.

If the program is to reply, it should execute a Write Conversational operation. If desired, you can send the reply in several segments by using Write Continue operations. After the last Write, you should execute a Read Initial (nonswitched) or Read Conversational (switched) operation to permit the terminal operator (1) to resume sending input (in which case the sequence just described is repeated) or (2) to signify to the program that he has finished by turning the mode switch to Local or by turning off the terminal power switch. You may substitute a Write Continue Conversational (TTV) for the last of a sequence of Write Continues, to avoid executing a separate Read Initial or Read Conversational; the WRITE TTV performs the functions of the Write Continue and the subsequent Read.

PROGRAMMING CONSIDERATIONS

Operations on Switched Lines

When the terminal operator switches the terminal to Local mode or turns off the power, the operation in progress at that moment is terminated with an indication of Channel End, Device End, and Unit Check in the CSW status byte, and Intervention Required in the sense byte. These indications will be posted in the DECB for the line as DECSDECB=X'41' and DECFLAGS=X'04'. You should check for the presence of these indicators after each Read and Write operation and take appropriate action; ordinarily, the Read Initial should be reissued.

For some applications it will be appropriate for the terminal operator to end the conversation only when the terminal is in transmit state, that is, not to turn off power while receiving text from the computer. It might then be appropriate for the program to record whether the operation in effect when the terminal went off-line was a Read or a Write.

Operations on Nonswitched Lines

- In a conversational environment, it is usually sufficient that the data is made available to the terminal operator, without the necessity that he actually receives it. If it is imperative that the message be received by the operator, he may be required to acknowledge receipt. The acknowledging message is received via the Read operation that follows the sequence of Write operations.

If during transmission from computer to terminal, the terminal operator chooses to break off reception of the message, the data set (modem) that connects the terminal to the line cannot signal this fact to the transmission control unit (TCU). When the operator breaks off reception, the sequence of Write operations then in progress proceeds to conclusion just as though the terminal were still receiving. All Write operations are posted complete without error; that is, completion code is X'7F' and DECFLAGS equals zero.

- For half-duplex nonswitched lines, a Read operation is not posted complete until a message has been received. This may mean that your program must accept "sign-on" messages in any Read operation. If it is desirable to recognize the end of a conversation, the program may wait an appropriate amount of time and, if no message has been received yet, assume that the terminal operator has ended the conversation.

If necessary to purge the Read operation you may issue an IOHALT macro for the line. This will halt the operation and cause posting of the Read. You may then issue another Read Initial operation, with changes to the area and/or length operands. For example, the new Read operation may be intended to receive a "sign-on" message into a different area than regular messages.

Using the Attention Key and 2741 Receive Interrupt Facility

When the terminal is in transmit state, the operator may press the Attention key to signal the computer that he has finished entering data. Pressing this key sends an EOT to the computer and returns the terminal to receive state. The keyboard locks, and the operator can resume entering data only after the program returns the terminal to transmit state by means of a Read operation or Write Continue Conversational operation.

If the 2741 is equipped with the Receive Interrupt feature, the Attention key can be pressed while the terminal is in receive state to interrupt data transmission from the computer, when, for example, the operator has a high-priority message to enter. Pressing the Attention key causes the Write operation then in progress to terminate, and the Channel End, Device End, and Unit Check indications to be set in the CSW status byte and the Intervention Required indication in the sense byte. These indicators will be posted in the DECB for the line as DECSDECB=X'41' and DECFLAGS=X'02'. You should check after each Write operation for the presence of these indicators. When present, it is generally appropriate to issue a Write Continue that sends a NL character to return the terminal's print element carrier to the beginning of the next printing line.

General Considerations

- Dynamic buffering cannot be used for the 2741.
- The usual considerations regarding use of the WAIT and TWAIT macro instructions should be observed. Before issuing any of these macros, you must always check the return code resulting from a Read or Write operation to ensure that the operation was started successfully.

- Messages sent to a terminal must not contain any EOT characters, as these cause the program to lose control.
- IBM 2741 terminals do not perform an automatic carrier return when the print element reaches the end of the print line. To avoid character pile-up at the end of the line, the text sent to the terminal must contain NL (new line) characters at intervals not exceeding the length (in characters) of the line.
- Any printable characters received by the terminal during the time the terminal is executing a carrier return, horizontal tab, or index (line feed) function will be printed erratically. To avoid this occurrence, each New Line (NL), Horizontal Tab (HT), and Line Feed (LF) character must be followed by one or more nonprinting characters, such as the Idle character.

For the line feed function, you should place one Idle character after each LF character in text to be sent to the terminal. For the new line and tab functions, the number of Idle characters needed equals 1.5 plus the number of inches of carrier travel caused by the function, rounded off to the next higher integer. In addition, you may need to place Idle characters at the beginning of each block of text the program sends to the terminal following receipt of an EOT character from the terminal.

The number of Idles required depends on several factors, such as line turnaround time and model of data set used. A recommended practice is to use the same number of characters as are used following a NL character that results in the longest carrier travel.

Example: Assume the length of a print line for a particular application is 7-3/4 inches and tab settings are at 2 and 6 inches. Each HT character should be followed by $1.5 + 4 = 5.5$, or 6 Idles (the 4 derives from the maximum distance of carrier travel [4 inches], caused by an HT character). Each NL character should be followed by $1.5 + 7.75 = 9.25$, or 10 Idles. Also, each block of text sent to a 2741 following receipt of EOT from the terminal should begin with 10 Idles.

AT&T 83B3 SELECTIVE CALLING STATIONS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence (which for the 83B3 is called a Transmitter Start Code).

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{ { OPENLST }, (xxyy, ...)
  { WRAPLST }
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence (which for the 83B3 is termed a Call Directing Code).

To define an addressing list, code the DFTRMLST macro like this:

```
{ OPENLST, (xxyy, ...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

A single V or M character constitutes a negative response; the message text itself signifies a positive response.

1. Write FIGS H LTRS
2. Write TSC
3. Read Response
4. Read Text

READ Skip

READ Skip reads data from the line to clear it. There is no data transferred to storage.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes message text to the terminal.

A single V or M character constitutes a positive response; a negative response is indicated by no response at all. A negative response for any terminal in the list is an abnormal condition. The operation ends and is posted complete-with-error.

1. Write FIGS H LTRS
2. Write CDC
3. Write LTRS
4. Read Response
5. Write Text
6. Write FIGS H LTRS (TIR only)

WRITE Break

This macro instruction causes a series of space characters to be sent, the number is determined by the length operand.

Programming Notes:

1. Each output message must begin with the sequence CR LF LTRS (this serves as the end-of-addressing indicator).
2. You must specify in the WRITE macro the exact length of the message.
3. If you are sending a message with a WRITE TI macro, code FIGS H LTRS at the end of the message (this is the end-of-transmission sequence). If you are sending a message with a WRITE TIR macro, the macro supplies the FIGS H LTRS sequence.

TERMINAL-TO-TERMINAL OPERATION

BTAM does not provide control for terminal-to-terminal traffic on a line on which BTAM provides control of traffic between computer and terminal; however, BTAM does not interfere with terminal-to-terminal traffic. In a system in which such traffic can occur, the operation is as follows.

A READ Initial macro polls the terminal that will become the sending terminal. The sending terminal responds with the addressing code of the terminal with which it wishes to communicate. This code appears to the Read Response command like data, and is therefore received in the input area. The next character is a V or M sent by the receiving terminal as a positive response. It, too, is read into the input area. The sending terminal recognizes the V or M as a positive response and sends a message to the receiving terminal; this message text, too, is read into the input area. Thus, while BTAM does not influence the terminal-to-terminal operation, it does receive into main storage any message sent between the terminals.

WESTERN UNION MODEL 33/35 TWX TERMINALS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list containing a sequence of control and identification characters to be sent to a terminal that calls the computer. The sequence has from 7 to 18 characters. A recommended sequence is:

```
Null CR LF DEL (1 to 12 graphic
characters) CR LF XON
```

To code an answering list, code the operand field of a DFTRMLST macro like this:

```
IDLST,0,numsent,sentchar
```

Example: To define an answering list containing the foregoing character sequence (using RALEIGH as the graphic sequence), you would code:

```
IDLST,0,14,01B150FF4B8233A393E212B15088
```

The characters following the third comma are the hexadecimal representations of the transmission code bit patterns for the recommended sequence:

```
01B150FF --          Null CR LF DEL
4B8233A393E212 --    R A L E I G H
B15088 --            CR LF XON
```

This sequence prints the computer identification, RALEIGH, at the beginning of the next line, and turns on the tape transmitter.

Read Conversational Operation

A Read Conversational operation requires a list containing a sequence of control characters to be sent to the terminal to prepare it to transmit. For this purpose you define an answering list containing the desired characters; the list is not used for the answering function. Define the list by coding the DFTRMLST operand like this:

```
IDLST,0,numcnsent,cntrlseq
```

If the Read Conversational operation is preceded by a Write Initial operation, a recommended sequence is XON (1 to 4 characters of your choice) XOFF; if the preceding operation was a Read Initial, the single character, XON, may be used. These

sequences start the tape transmitter of the terminal. If you wish to read from the keyboard, an appropriate sequence is G Bell A Bell; GA means go ahead, and the bell alerts the terminal operator.

Write Operations

A Write Initial operation requires a calling list containing the same sequence of characters as the called terminal sends when it answers the call from the computer.

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
IDLST,dialcount,dialchars,numrec,tidseq
```

Example: To define a calling list for a terminal whose telephone number is 887-4444 and which will answer with the sequence

```
CR LF I B M 3 5 A S R # 1 CR LF XON
```

you would code:

```
IDLST,7,8874444,17,B1509342B205CCAD82CA
4B05C58DB15088
```

READ MACRO INSTRUCTIONS

Programming Note: Each message sent from the terminal (that is, an incoming message) must end with either the WRU, XON, or XOFF character, or with the EOT sequence. If it ends with the EOT sequence, the next operation must be a Read Initial or Write Initial (EOT resets the terminals to control mode). If the message ends with WRU, XON, or XOFF, the next operation can be a Read Conversational or Write Conversational.

READ Initial (TI)

READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, writes the identification and control sequence, and reads a message block from the terminal.

1. Disable
2. Enable
3. Write Pad characters
4. Write ID-control sequence
5. Read Text
6. Write EOT (TIR only)
7. Disable (TIR only)

READ Conversational (TV)

READ Conversational and Reset (TVR)

READ Conversational writes a control sequence to the terminal and reads message

text from the terminal. This macro is for use following a READ Initial or a WRITE Conversational when the line connection is already established.

1. Write Control sequence
2. Read Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Outgoing messages should not end with a control character or a sequence of control characters (e.g., XON, or XON (user-selected characters) XOFF.

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial calls a terminal and reads the identification sequence of the terminal. If the received ID matches the expected ID that is contained in the terminal list, the macro writes message text to the terminal. If the two ID's do not match, the operation is posted complete-with-error; the message text is not sent.

1. Disable
2. Dial Dial Digits
3. Read Terminal ID sequence
4. Write Text
5. Write EOT (TIR only)
6. Disable (TIR only)

WRITE Conversational (TV)

WRITE Conversational and Reset (TVR)

WRITE Conversational writes message text to the terminal. This macro may be used following a Read operation, to change from receiving text to sending text, and may be issued as many times in succession as necessary to send a message.

1. Write Text
2. Write EOT (TVR only)
3. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

WESTERN UNION PLAN 115A OUTSTATIONS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character is always an X (X'17' is the transmission code bit pattern); the second identifies the terminal.

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{ { OPENLST }, (xxyy, ...)
  { WRAPLST }
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence. The first character is the circuit call code; the second identifies the terminal that is to receive the output message.

To define an addressing list, code the DFTRMLST operand field like this:

```
{ OPENLST, (xxyy, ...) }
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

A single V or M character constitutes a negative response; the message text itself signifies a positive response.

1. Write FIGS H LTRS
2. Write Polling sequence
3. Read Response
4. Read Text

READ Skip

READ Skip reads data from the line to clear it. There is no data transferred to storage.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes message text to the terminal.

A single V or M character constitutes a positive response; a negative response is indicated by no response at all. A negative response for any terminal is an abnormal condition; the operation ends and is posted complete-with-error.

1. Write FIGS H LTRS
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write FIGS H LTRS (TIR only)

WRITE Break

This macro instruction causes a series of space characters to be sent, the number is determined by the length operand.

Programming Notes:

1. Each output message must begin with a Space character (this serves as the end-of-addressing character).
2. You must specify in the WRITE macro the exact length of the message.
3. If you are sending a message with a WRITE TI macro, code FIGS H LTRS at the end of the message (this is the end-of-transmission sequence). If you are sending a message with a WRITE TIR macro, the macro supplies the FIGS H LTRS sequence.

TERMINAL-TO-TERMINAL OPERATION

BTAM does not provide control for terminal-to-terminal traffic on a line on which BTAM provides control of traffic between computer and terminal; however, BTAM does not interfere with terminal-to-terminal traffic. In a system in which such traffic can occur, the operation is as follows.

A READ Initial macro polls the terminal that will become the sending terminal. The sending terminal responds with the addressing code of the terminal with which it wishes to communicate. This code appears to the Read Response command like data, and is therefore received in the input area. The next character is a V or M sent by the receiving terminal as a positive response. It, too, is read into the input area. The sending terminal recognizes the V or M as a positive response and sends a message to the receiving terminal; this message text, too, is read into the input area. Thus, while BTAM does not influence the terminal-to-terminal operation, it does receive into main storage any message sent between the terminals.

WORLD TRADE TELEGRAPH TERMINALS

GENERAL INFORMATION

The name World Trade (WT) telegraph terminals refers to various European teletype-writers using a start-stop 5-level code with two shifts (lettershift and figure-shift) to transfer data over point-to-point telegraph lines.

WT terminals use either the International Telegraph Alphabet No. 2 or the Figure Protected Code ZSC3. World Trade telegraph terminals employ the contention system of line control. When a terminal and the computer each try to send a message, simultaneously, both transmissions are immediately stopped; this is called contention.

A terminal is always ready to receive or to send a message. Normally, the motor of the terminal is off and the first letter-shift character (LTRS) sent or received by the terminal starts the motor, which requires 1.5 seconds to reach operating speed. During this period, the terminal cannot correctly send or receive characters. The motor stops when no character has been transmitted during a period of from 10 to 30 seconds. When the terminal is operating in this manner, it is said to be in Motor-Off mode. Optionally, the terminal can be equipped with a heavy-duty motor which is never switched off; in this case, the terminal is said to be operating in Motor-On mode.

When a terminal is operating in Motor-Off mode, the MONDLY parameter of the DCB macro instruction enables you to specify the number of Mark (Idle) characters corresponding to the 1.5 second period. When you issue a WRITE macro instruction, BTAM recognizes the motor mode of the terminal (motor-off or motor-on) and generates a LTRS character (this can be followed by a user-specified number of Mark characters) that precedes the data to be sent over the line.

Most terminals can be equipped with another optional feature called the Automatic Answerback Unit. This feature enables a sequence of up to 20 identification characters, generated by a mechanical drum, to be sent over the line by either pressing the IAM key or receiving code combination 4 in figures shift.

Telegraph Adapter Description

The World Trade Telegraph Adapter in the TCU recognizes two message end conditions:

FIGS x and FIGS y LTRS. These are established when the IBM 2701, 2702 or 2703 to which the WT terminal is connected is installed: x and y are assigned by the customer on a per-system basis, as follows.

When a terminal is equipped with the Automatic Answerback Unit, FIGS x must be code combination 4 (FIGS D) sent by the terminal WRU key. This character is referred to as the WRU signal. If the terminal is not equipped with the Automatic Answerback Unit, FIGS x may be any other code combination.

The two characters, x and y, cannot be the same. FIGS y immediately followed by a LTRS character causes a Read operation to end. Therefore, FIGS y can be sent by a terminal as data only if it is not followed by LTRS.

The above terminations of a Read operation can be used as end-of-message (EOM) signals. The FIGS y LTRS termination (if not yet used as an EOM signal) or two consecutive EOM signals can represent the end-of-transmission (EOT) signal.

The transmission control unit deletes all incoming LTRS and FIGS characters and updates a shift bit (S) which is added to each character transferred to main storage. Conversely, each change in shift bit setting along a character sequence causes the TCU to send a LTRS or FIGS character ahead of the first message character for which the shift bit was reversed.

Figure 28 shows the relationship of a System/370 byte and a telegraph character configuration.

Contention Resolution

When contention occurs, BTAM sets a completion code of X'7F' in the ECB and turns on bit 3 of DECFLAGS. Contention is resolved by the user program coding and the local operator's action, according to one of the following procedures:

If priority is to be given to the computer, the terminal operator must wait; the program should repeat the Write (or Read TE) operation.

If priority is to be given to the terminal, the program must follow with a READ Continue macro the operation during which contention occurred. The terminal operator continues sending his message.

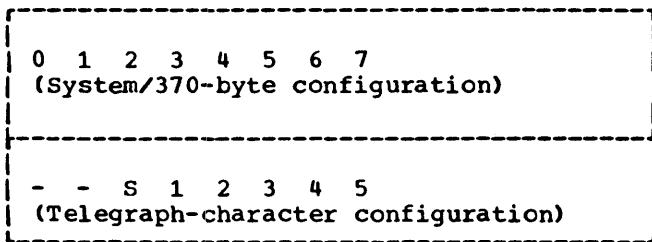


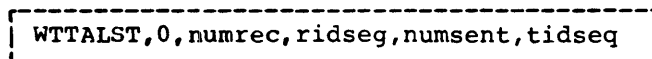
Figure 28. WT Telegraph Code

DEFINING TERMINAL LISTS

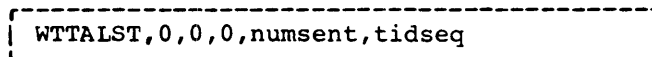
In World Trade telegraph operation terminal lists are used only for the READ Continue with Identification Exchange (TE) macro. (They are not used for READ Initial operations.)

If The Terminal's Identification Sequence is to be Requested

To define a terminal list, code the operand field of a DFTRMLST macro like this:



If The Computer Identification is to be sent to the Terminal



Programming Note: tidseq may specify from 7 to 20 characters (computer identification sequence).

READ MACRO INSTRUCTIONS

Read Initial (TI)

READ Initial monitors the line for a message from a terminal, and reads it into the input area. The Read operation ends when an EOM, EOT, or WRU character is received.

1. Prepare
2. Sense
3. Read Text

Programming Note: A RESETPL macro is effective only if issued when message transmission is not in progress.

READ Continue (TT)

READ Continue reads message text from a terminal following receipt of an EOM character, or when the terminal is given the right to transmit when contention has

occurred. The operation ends when an EOM, EOT, or WRU is received.

1. Read Text

READ Continue with Identification Exchange (TE)

READ Continue with Identification Exchange writes to the remote terminal the computer's identification sequence (defined in the terminal list) and a WRU character. The operation also reads the identification sequence of the terminal (and optionally, message text) into the input area, only if you code WRU=YES in the DCB macro for the line group.

1. Write Mark characters Note 1
2. Write Computer identification seq.
3. Write WRU (or LTRS) Note 2
4. Read Terminal identification Note 3
5. Read Text

Note 1: One LTRS character plus n Mark characters are sent, where n represents the number of Mark characters, as follows:

- a. When the terminal is equipped with the Motor-On optional feature, n is always zero.
- b. When the terminal is not equipped with the Motor-On optional feature, n can take one of the following values:

n=0

if the previous operation was a Write, or if a Read operation ended with EOM or WRU signal.

n=the value given to the MONDLY key-word operand of the DCB macro instruction.

Note 2: The computer sends the WRU signal to ask for the terminal identification, provided that WRU=YES is coded in the DCB macro instruction. Otherwise, the computer sends a LTRS character.

Note 3: The terminal sends its own identification. If the received ID and the expected ID do not match, the operation is posted as complete; no message text is read. Bit 3 of DECFLAGS is set to 1.

Programming Note: The value of the length parameter of the READ macro instruction must equal or exceed the length of the identification sequence generated by the Automatic Answerback Unit. If equal, only the terminal identification sequence is sent to the computer. If the length parameter exceeds the identification sequence length, message text can also be read. This is applicable when WRU=YES is speci-

fied in the DCB macro instruction; otherwise, command (4) is not generated.

Note: (See Note 1 under READ TE above.)

WRITE MACRO INSTRUCTIONS

WRITE Continue (TT)

WRITE Initial (TI)

WRITE Continue sends an output message.

Write Initial sends an output message, preceded by 12 LTRS characters.

1. Write Mark character Note
2. Write Message

1. Write Mark characters Note
2. Write Pad characters
3. Write Message

Note: (See Note 1 under READ TE above.)

PROGRAMMING CONSIDERATIONS FOR USE OF
AUTOPOLL (START-STOP)

Read Initial operations on lines for which the Autopoll feature is used require polling lists different from those used in programmed polling. They are referred to as Start-Stop Autopoll lists. The list may be of the open type (SSALST) or wraparound type (SSAWLST), and may have one or more terminal entries, each containing a single polling character (IBM 1030) or a two-character polling sequence (other terminal types). They are specified in the same way as in lists for programmed polling, with the exception of the 2740. The second polling character in a list for the 2740 must be Space.

To define an Autopoll polling list, code the operand field of a DFTRMLST macro like this:

```
-----  
| {SSALST }  
| {SSAWLST },(xx,...) (for 1030)  
|-----
```

```
-----  
| {SSALST }  
| {SSAWLST },(xxyy,...) (for 1050,1060,2740)  
|-----
```

CHANNEL PROGRAMS

Read Initial operations (and Read Continue and Read Repeat operations, on the 1030 and 1060) using Autopoll require channel programs different from those used in programmed polling. They are explained below by referring to the use of the specific commands that form the program.

READ Initial (TI)
READ Initial and Reset (TIR)
(Open Auto Poll List)

The channel program is:

1. Write EOT EOT EOT
2. Poll
3. NOP
4. Read Index
5. Read Text
6. Write EOA EOT EOT EOT (TIR only)

Command (1) sets the terminals on the line to control mode, as with programmed polling. Command (2) initiates the polling operation. Command (3) is executed only if no response is received from a terminal in the list, or if no terminal in the list returns a positive response to polling, that is, all terminals send negative responses. Execution of command (3) ends the Read operation, which is posted complete in the event control block.

If some terminal in the list returns a positive response, command (3) is skipped; command (4) reads into the first two bytes of the input area the index byte indicating which terminal responded, and the first message character. Command (5) reads the remaining message text into the input area.

Programming Notes:

1. To determine which terminal responded, examine the index byte. You should obtain this index byte not from the input area but from the DECPOLPT field of the DECB for the line. DECPOLPT always contains the index byte, while an I/O error during transmission may prevent the index byte from being placed in the input area.
2. In specifying the length in the READ macro, be sure it is at least one greater than the expected text length, in order to accommodate the index byte.

READ Initial (TI)
READ Initial and Reset (TIR)
(Wraparound Auto Poll List)

The channel program is:

1. Write EOT EOT EOT
2. Poll (Beginning with entry specified in macro)
3. TIC (to command 5)
4. TIC (to command 7)
5. Poll (beginning with first entry in list)
6. TIC (to command 5)
7. Read Index
8. Read Text
9. Write EOA EOT EOT EOT (TIR only)

Command (1) sets the terminals on the line to control mode. Command (2) initiates the polling operation, beginning with the terminal specified by the "entry" operand in the READ macro. If before the end of the list is reached a positive response is returned, the status modifier is set, causing the next command, (3), to be skipped; command (4) transfers to command (7), followed by (8), which functions like commands (4) and (5) in the "open-type" Autopoll operation.

If, however, the end of the list is reached and no positive response has been received, command (3) is executed, giving control to command (5), which restarts the polling operation at the beginning of the polling list. Polling proceeds automatically, and each time the end of the list is reached, command (6) gives control to (5), and the polling starts again. If during a pass through the list, a positive response is received, command (6) is skipped (just

as command (3) is skipped above), and commands (7) and (8) are executed as before.

Programming Notes: The same programming notes given above apply to Autopoll operations with a wraparound list.

Other Types of READ and WRITE

With two exceptions, all other types of READ macro (such as READ Continue) and all types of WRITE macro generate the same channel programs as are shown under the corresponding type in the appropriate sections for the type of terminal concerned: "IBM 1030 Data Collection System," "IBM 1050 Data Communication System," "IBM 2740 with Station Control Feature," and "IBM 2740 with Station Control and Checking Features." The exceptions are READ Continue and READ Repeat for the IBM 1030 and 1060, as shown below. (READ Continue and READ Repeat for the 1050 and for the 2740 with Station Control and Checking are the same as for the non-Autopoll operations for these terminals; READ Continue and READ Repeat are not provided for 2740 with Station Control and without Checking.)

READ Continue (TT) (1030,1060)

The channel program for READ Continue is identical to the program for READ Initial (using either SSALST or SSAWLST), except that the first command is:

1. Write Circle Y EOT EOT EOT

This channel program sends a positive response, then repolls the terminal and receives message text, as in a Read Initial operation.

READ Repeat (TP) (1030,1060)

The channel program for READ Repeat is identical to the program for READ Initial (using either SSALST or SSAWLST), except that the first command is:

1. Write Circle N EOT EOT EOT

This channel program sends a negative response, then repolls the terminal and receives message text, as in a Read Initial operation.

GENERAL INFORMATION

TRANSMISSION CODES

Binary synchronous communications under BTAM control uses one of three transmission codes, as follows:

System/370 to System/370, System/3, 1800, 2770, System/360 (including Model 20), or remote 3270 (not Write Structured Field devices)	} EBCDIC or USASCII
-----------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------

- System/370 to 1130: EBCDIC
- System/370 to 2715: EBCDIC
 (transparent)
- System/370 to 2780: EBCDIC, USASCII
 or Transcode
- System/370 to 2972: EBCDIC

Only EBCDIC may be used between a System/370 and a 2770, 2780 or a remote 3270 with Write Structured Field capability when messages are sent in transparent mode. Only EBCDIC may be used if the central System/370 is communicating with a remote System/370 that is running under BOS (Basic Operating System) or BPS (Basic Programming System). These codes are shown in Appendix E.

You must sometimes enter into message output areas certain line control characters in their USASCII or Transcode form; they must appear in main storage according to the following rules.

- In main storage, bits 1-7 in a System/370 byte correspond to bits 7-1, respectively, of the USASCII character. The zero-bit is always zero (off). When the control unit receives a byte, a parity bit is sent over the line along with bits 1-7 of the byte. Conversely, when seven bits plus a parity bit are received by the transmission control unit from the line, the seven (data) bits are read into main storage right-justified in a byte and the zero-bit is set to zero.
- For Transcode, a similar rule holds. The hexadecimal equivalent is right-justified in a System/370 byte (bits 2-7) and the 0-bit and 1-bit are always set to zero (off). Only bits 2-7 are sent over the line.

REMOTE STATION COMPATIBILITY AND INTERMIXING

Unlike start-stop terminals, BSC stations of different types are compatible in use of line control procedures, so that it is unnecessary to specify at system generation time what specific type or types of remote station are connected to a given communication line. Instead, one of the three types of line supported by BTAM is coded in the UNIT operand of the system generation IODEVICE macro:

- BSC1 indicates that the line is a non-switched point-to-point line.
- BSC2 indicates that the line is a switched point-to-point line.
- BSC3 indicates that the line is a non-switched multipoint line.

USER PROGRAM ANALYSIS

As discussed under "Message Transmission" in the section "Line Control and Message Transmission," the user program must analyze the results of each Read or Write operation to determine whether it completed successfully or unsuccessfully, and what if any exceptional condition occurred. "User Program Analysis Procedure" in the section "Error Recovery Procedures and Error Recording" describes a procedure to follow. In addition, the "Suggested Retry Options for BSC Read and Write Operations," recommends appropriate READ and WRITE macro instructions to issue following various error and exceptional conditions.

LINE AND MESSAGE CONTROL FUNCTIONS

ID Verification

Identification sequences may be exchanged between the central computer and some kinds of remote BSC stations with which communication has been established over a switched line. This facility affords either or both stations (that is, central computer and remote station) the opportunity to verify the identity of the other before message text is transmitted. The terminal list associated with the READ or WRITE macro instruction that established the contact contains the ID sequence to be sent to the remote station, and one or more ID sequences that will be accepted from the remote station.

ID verification is available at either of two levels, which may be termed "regular" and "expanded". In regular ID verification, only one unique ID sequence can be accepted from the remote station, regardless of which of many stations has called (or been called by) the central computer. Further, BTAM makes only one decision regarding continuance of the Read or Write operation. That is, if the received sequence matches the expected sequence (the terminal list contains only one expected sequence), the operation continues, resulting in transfer of text between the stations. If the received sequence does not match the expected sequence, the operation is halted, and text transfer does not occur.

In the expanded ID verification, the user can designate, in the terminal list, many different ID sequences, any of which will be accepted from the remote station; this allows each station to send a unique sequence. Also, contact can be established with stations that do not send ID sequences as well as with those that do. For expanded ID verification, a terminal list having multiple entries is used; this type of list is designated as SWLST. Each entry has a field containing a valid ID sequence that will be accepted from a remote station, and is a control byte. (Each entry may also have a user-data area, at the user's option. This is discussed below.)

After the line connection has been established and an ID sequence (or other data) has been received from the remote station, BTAM scans the terminal list for a matching ID sequence. If one is found, BTAM places the address of the entry containing the sequence in the first fullword of the terminal list, for possible use by the user program. Typically, the program would use this address to determine which remote station called or answered the central computer.

The control byte of an entry contains a user-specified indicator specifying what action BTAM is to take after the ID sequence (or other data) has been received. Examples of actions following a Read Connect operation are: continue with the remainder of the Read Connect operation to read a message block; disconnect the line; or post the operation as complete, without reading a message block.

By setting up the control byte prior to the Read or Write operation, and by checking completion codes and indicators in the DECB following receipt of an ID sequence (or other data) from a remote station, the user program can both determine the status of the operation and influence subsequent BTAM actions.

Each terminal list entry may contain a four-byte user-data field. In this field may be placed a relocatable expression as an address that is to be associated with the ID sequence (or ENQ character) contained in that entry. Typically, the user-data field would contain the address of a subroutine to be called when the remote station represented by the ID establishes contact with the central computer.

For more detailed information on use of expanded ID verification, see the descriptions of the READ Connect, WRITE Connect, DFTRMLST, and CHGNTRY macro instructions.

Error Information Byte (EIB) Mode

BTAM provides the option of specifying, in the DCB macro, whether the TCU is to operate in EIB mode or non-EIB mode. The distinction is as follows: In EIB mode, the TCU, during a receive operation, sends an error information byte into main storage following each IUS (US), ETB, and ETX character received from the communication line. In non-EIB mode, the TCU does not send the EIB into main storage following these characters.

The EIB indicates the presence of either a data check or an overrun error (or no error at all) in the sub-block that immediately preceded the IUS (US), ETB, or ETX character. BTAM does not analyze EIBs. The user program may check them and, where an error is found, take appropriate action, such as issuing a READ Repeat with Leading Graphics macro instruction to request retransmission of that part of the message block that is in error.

Whether or not the TCU is operating in EIB mode, it recognizes the IUS (EBCDIC) or US (USASCII) character as signifying the end of an intermediate block. (IUS is Interchange Unit Separator [an EBCDIC character], and US is Unit Separator [a USASCII character]; the two are equivalent characters.)

Double Addressing (Multipoint Lines)

Transient conditions such as lightning impulses or switching pulses can introduce errors in data transmitted over a communication line. Often, such errors consist of inverted bit settings within the bit pattern representing a character. While errors of this kind occurring in message data are normally detected through checking techniques, they are undetected when they occur in polling and addressing (selection) sequences, which are unchecked. An error wherein one valid polling or addressing character is changed to another can result in polling or addressing the wrong station.

To avoid such an occurrence, double addressing may be employed for certain BSC stations. In this technique, a remote station is represented by two identical characters, rather than one character as in single addressing.

When polled or addressed, the remote station that recognizes the first character compares it with the second. If the two are identical, the station address is presumed to be correct, and the station returns a positive response. If they differ, a transmission error is presumed to have altered one or both of the characters, and the station does not return a response.

The increased polling and addressing reliability this technique affords stems from the improbability that both of the characters would be changed in precisely the same way by a transmission error. For example, the characters BB are far less likely to be converted by an error to CC than they are to be converted to BC, or KB, or FC. (Each of these conversions could result from a single-bit error in each character, where the transmission code is EBCDIC. For example, the letter B, the bit pattern for which is X'C2' (1100 0010), becomes a C (X'C3', 1100 0011) or a K (X'D2', 1101 0010) through a single-bit error.) If a station whose address is K was attached to the line, that station would recognize the first character of the erroneous address KB, but would not respond because the two characters did not match. Thus, a message intended for station B would not be sent to station K instead.

For System/360 Model 20, System/3, 1800, 2715, 2770, 2972, and remote 3270 stations in a multipoint network, double-addressing must be used.

As is always the case in terminal lists, all list entries must have the same length. Therefore, if addresses of different lengths are to be contained in a list (as when single-addressing is used for some stations, double-addressing for others), the shorter addresses must be padded with leading SYN characters so that they are the same length as the longer addresses.

MESSAGE FORMATS

In nontransparent mode, messages appear on the line in the format:

```

-----//-----
| STX | (text) | ETB (or ETX) |
-----//-----

```

The STX (Start of Text) character is required at the beginning of each message

block. (SOH may appear at the beginning of the first message block, however.) ETB denotes the end of a message block and ETX denotes the end of the last block of a message. You must supply in the output area the SOH, STX, ETB and ETX characters. In calculating the length to be specified in a WRITE macro, include the STX and ETX in the number of message characters.

Messages in nontransparent mode may not contain line control characters.

In transparent mode, messages appear on the line in the format:

```

-----//-----
| DLE | STX | (text) | DLE | ETX |
-----//-----

```

Transparent mode allows you to include any bit pattern in the message, regardless of whether the bit pattern represents a line control character.

The DLE STX must appear at the beginning of each message block. DLE ETX denotes the end of the message. You must supply the DLE STX in the beginning of the output area. You do not provide the DLE ETX, as each Write operation of the transparent type automatically sends these characters following your text.

When coding a WRITE macro for sending text in transparent mode, the length must include the DLE STX; the length should not include the ending characters, DLE ETX, as these are sent by a separate command.

When you receive a transparent message from a remote station, it has the format:

```

DLE STX (text) ETB (or)
DLE STX (text) ETX

```

The DLE preceding the ETB or ETX is removed by the TCU before the message enters main storage.

If you issue any WRITE macro that specifies both conversational operation and use of dynamic buffering, the BUFL operand of the DCB macro for the line group must specify at least 24 bytes.

Use of Line Control Characters

Successful transmission of data between central computer and remote station demands thorough familiarity with line control (data link control) procedures. See the general discussion of this subject under "Use of Line Control Characters" in the section "Line Control and Message Transmission."

Use of SOH and STX Characters

Since either an SOH or an STX character appearing at the beginning block of a message resets, but is not included in, the block check character that follows the block, the following practice is recommended. Include as the first character of a heading, following the SOH character, some specific noncontrol character that is never used as the first character following STX in a nontransparent text transmission. You may use any character other than a data link control character or the percent sign (%). Consistent observance of this rule will prevent the processing of text data as a heading or of a heading as text data owing to a transmission error that changes STX to SOH or vice versa. When a message block is received without error, presence of the specific character identifies the block as heading, while absence of that character identifies it as text.

Coordinating BSC Central and Remote Programs

In order to achieve message transmission between two computers using BSC communication, you must be careful to coordinate the central and remote programs so they remain in step. This requires that you be aware of the responses that are valid for message text and for each control character that may be sent over the line. These are as follows.

RESPONSES

Responses to Message Text

ACK-0 or ACK-1 (Pos. response)	The remote station received the text correctly.
NAK (Neg. response)	The remote station wishes to have the text retransmitted.
WACK (Wait-before-transmit)	The remote station wishes to delay transmission. (The only valid response to WACK is ENQ (or EOT); the central computer cannot continue sending message text, but must send ENQ until the remote station responds with the positive acknowledgment for the last message block it received. The central computer may, however, respond to WACK with an EOT, to end the transmission.)

Note: When a remote 3270 printer has been started, WACK is a positive response.

leading graphics	The remote station is transmitting user-supplied, noncontrol characters.
EOT	The remote station is aborting reception of the message because of equipment malfunction or (if the remote station is a computer) program error.
DLE EOT	The remote station is aborting reception of the message and is disconnecting the line because of equipment malfunction or (if the remote station is a computer) program error.
RVI	The remote station wishes temporarily to stop receiving text. The user program may continue sending text, however, or may send an EOT, to end the transmission.

Responses to ENQ

ACK-0	1. The remote station is ready to receive text. 2. Positive response to text.
ACK-1	Positive response to text.
WACK	The remote station wishes to delay transmission.
EOT	The remote station does not wish to receive text.
NAK	The remote station did not acknowledge the last transmission.
Message text	Last receipt was text.

Responses to EOT (Switched Line Only)

EOT	The remote station does not wish to transmit but does not wish to disconnect the line.
ENQ	The remote station wishes to transmit text.

DLE EOT

The remote station is going to disconnect the line.

channel programs should be matched. This example is for System/370-to-System/370 communication on a nonswitched point-to-point line. It shows only the sequence of Read and Write operations; it omits checking of return and completion codes and omits WAIT or TWAIT macros.

You should pay close attention to the commands within channel programs. Figure 29 is an example of how central and remote

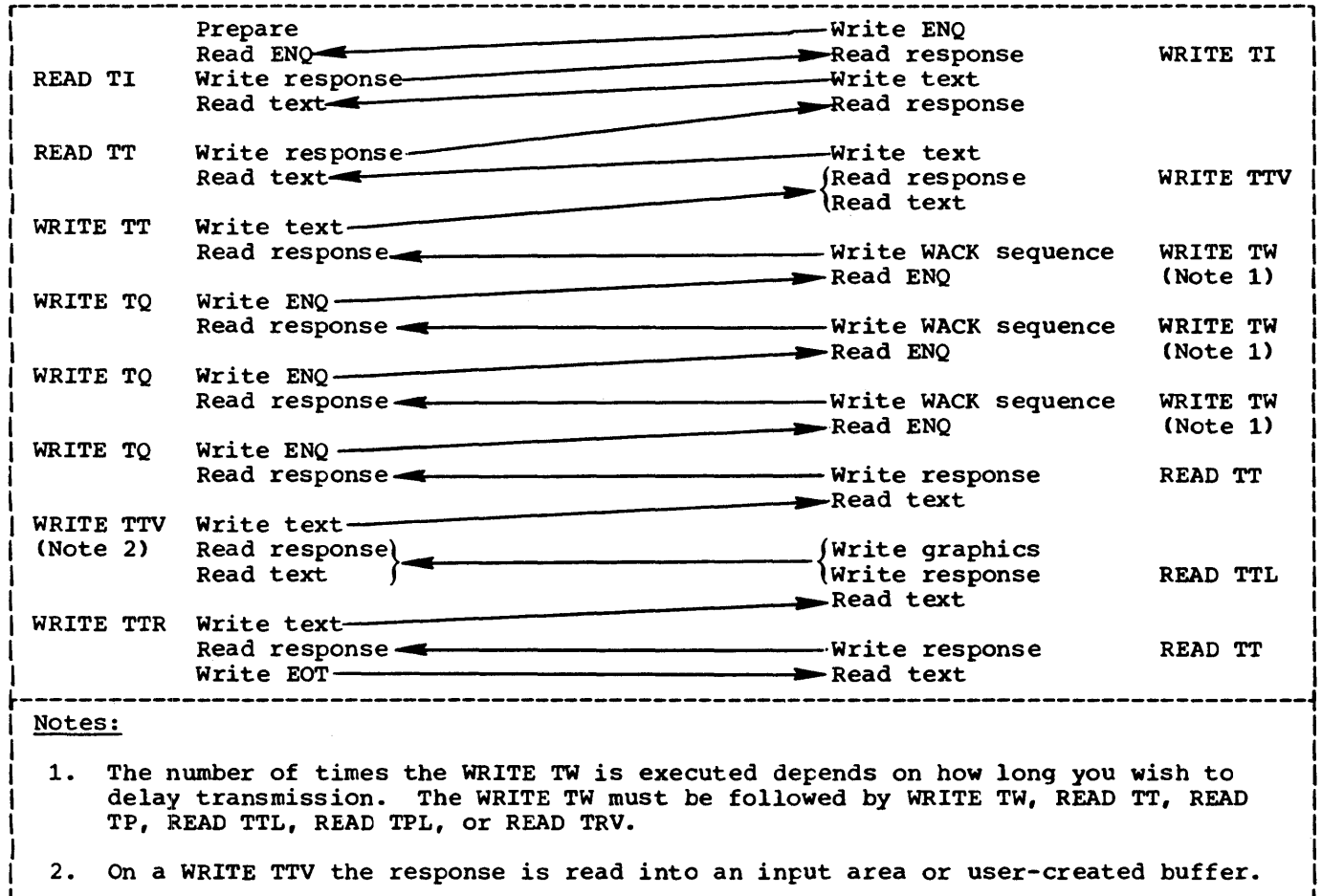


Figure 29. Example of a BSC Message Control Routine

BSC NONSWITCHED POINT-TO-POINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can be connected to a nonswitched point-to-point line, except as noted in individual macro instruction descriptions.

Since BSC operations on nonswitched point-to-point lines use contention-type line control, no terminal lists are used.

The channel programs in this section correspond to an IODEVICE macro UNIT operand of BSC1.

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial monitors the line for an ENQ sent by the remote station, writes a positive response, and reads the message block that follows.

1. Prepare
2. Read ENQ
3. Write Response
4. Read Text

READ Initial Inquiry (TIQ)

READ Initial Inquiry monitors the line for an ENQ sent by the remote station.

1. Prepare
2. Read ENQ

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

Note: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede

the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1).

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote

station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.

2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial writes an ENQ to gain use of the line, and if the response to the ENQ is positive (ACK-0), writes message text and reads the response to text. If the response to ENQ is other than ACK-0, the operation is posted complete, with appropriate indicators set in DECFLAGS.

1. Write ENQ
2. Read Response
3. Write Text
4. Read Response
5. Write EOT (TIR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX)

WRITE Initial Transparent and Reset (TIXR)

WRITE Initial Transparent functions the same as the Write Initial, except that after writing message text it writes the ending characters that must follow the transparent data.

1. Write ENQ
2. Read Response
3. Write Text
4. Write DLE ETX
5. Read Response
6. Write EOT (TIXR only)

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text. (Component selection characters are required unless the Job Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.)

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write TTX (or TTE) or Write TTVX operation to send the transparent text.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block functions the same as WRITE Initial Transparent (TIX) except that it writes DLE ETB instead of DLE ETX following message text.

1. Write ENQ
2. Read Response
3. Write Text
4. Write DLE ETB
5. Read Response

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text. (See programming note above.)

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write Continue (TT) or Write Continue Transparent (TTX) operation to send the transparent text.

WRITE Initial Conversational (TIV)

WRITE Initial Conversational writes an ENQ to gain use of the line, and if the response to ENQ is ACK-0, writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgement, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write ENQ
2. Read ACK-0
3. Write Text
4. Read Response
5. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central comput-

er; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

ETB that must follow transparent data, and reads a response from the remote station.

WRITE Initial Conversational Transparent (TIVX)

WRITE Initial Conversational Transparent writes an ENQ to gain use of the line, and if the response to ENQ is ACK-0, writes message text and the ending characters, DLE ETX, that must follow the transparent data. The macro then reads a response, which may be either the first two characters of a message block or NAK. If the response is message text, the remaining text is read, if not, the operation is posted complete.

1. Write ENQ
2. Read ACK-0
3. Write Text
4. Write DLE ETX
5. Read Response
6. Read Text

Programming Notes:

1. This macro instruction may be used for all types of BSC stations (except as noted in 2, below). However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).
2. This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text.¹

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response
3. Write EOT (TTR only)

WRITE Continue Transparent (TTX)

WRITE Continue Transparent and Reset (TTXR)

WRITE Continue Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Write EOT (TTXR only)

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE

1. Write Text
2. Write DLE ETB
3. Read Response

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending character, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Note: This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last

response (ACK-0, ACK-1, NAK, or a conversational text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-Before-Transmit (TW)

WRITE Wait-before-transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or READ Repeat, or in response to a conversational-type Write operation executed by the remote station (if a computer). The only valid responses to WACK are ENQ and EOT. You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Prepare
3. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to relinquish use of the line. After sending the EOT, the next operation must be an initial-type Read or Write operation, to again gain use of the line.

BSC NONSWITCHED MULTIPOINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can be connected to a nonswitched multipoint line, except as noted in individual macro instruction descriptions.

The channel programs shown in this section correspond to an IODEVICE macro UNIT operand of BSC3.

DEFINING TERMINAL LISTS

In order to achieve Read and Write operations over multipoint lines, you must define appropriate terminal lists (that is, polling or addressing lists) and refer to these lists in your initial-type READ and WRITE macro instructions.

See the explanation of the DFTRMLST macro instruction for general information on defining those lists. Given below are the specific coding requirements for multipoint operations.

Each Read Initial operation requires an open or wraparound polling list, and each initial-type Write operation (Write Initial, Write Initial Transparent, etc.) requires an open addressing list.

Polling List

To define a polling list for any type of BSC station or combination of stations on a multipoint line, code the operand field of a DFTRMLST macro like this:

```
-----  
| {AUTOLST }  
| {AUTOWLST},(tidseq,...)  
|-----
```

tidseq defines an entry in the polling list, and consists of between one and seven polling characters, followed by an ENQ character, all of which must be coded as the hexadecimal equivalents of their transmission code bit patterns.

All polling list entries must be the same length. Therefore, if polling sequences of different lengths are to be contained in a list, the shorter sequences must be padded with leading SYN characters so that they are the same length as the longer sequences.

In defining a polling list of either the open (AUTOLST) or wraparound (AUTOWLST) kind, you must code, following the entries for the stations to be polled, an entry of length equal to the others, and containing EOT characters (in hexadecimal equivalent

of the transmission code bit patterns). For example, if the entries for the stations each contain five polling characters plus ENQ, the last entry must be coded as six EOT characters.

Addressing List

To define an addressing list for any type of BSC station or combination of stations on a multipoint line, code the DFTRMLST operand field like this:

```
-----  
| OPENLST,(tidseq,...)  
|-----
```

tidseq consists of between one and seven addressing characters, followed by an ENQ character, all of which must be coded as the hexadecimal equivalents of their transmission code bit patterns.

All addressing list entries must be the same length. Therefore, if addresses of different lengths are to be contained in a list, the shorter addresses must be padded with leading SYN characters so that they are the same length as the longer addresses.

Note: See the discussion of double addressing in the General Information section at the beginning of this chapter.

READ MACRO INSTRUCTIONS

READ Initial (TI) (Using Open Polling List [AUTOLST])

READ Initial initiates an Autopoll operation to cause the TCU automatically to poll each of the stations in the polling list. If a positive response to polling is received from any station, the macro reads into the input area the index byte indicating which station is sending the message, followed by the message block. The first byte of the input area contains the index.

1. Write EOT
2. Poll (at list entry specified in READ macro)
3. NOP (terminates instruction sequence)
4. Read Index
5. Read Text

Command (1) sets the stations on the line to control mode. Command (2) initiates the polling operation. Command (3) is executed if no station in the Auto Poll list returns a positive response to polling, that is, all stations send negative responses. Command (3) is skipped if no response is received from a station in the Auto Poll list. Execution of command (3) ends the Read operation, which is posted complete in the event control block.

If some station in the list returns a positive response, command (3) is skipped; command (4) reads into the first two bytes of the input area the index byte indicating which station responded, and the first message character. Command (5) reads the remaining message text into the input area.

Programming Notes:

1. To determine which station responded, examine the index byte. You should obtain this index byte not from the input area but from the DECPOLPT field of the DECB for the line. DECPOLPT always contains the index byte, while an I/O error during transmission may prevent the index byte from being placed in the input area.
2. In specifying the length in the READ macro, be sure it is at least one greater than the expected text length, in order to accommodate the index byte.

READ Initial (TI)

(Using Wraparound Polling List [AUTOWLIST])

READ Initial initiates an Auto Poll operation to cause the TCU automatically to poll each of the stations in the polling list. If a positive response is received from any station, the macro reads into the input area the index byte indicating which station is sending the message, followed by the message block. The first byte of the input area contains the index.

1. Write EOT
2. Poll (at list entry specified in READ macro)
3. TIC (to command (5))
4. TIC (to command (7))
5. Poll (at beginning of list)
6. TIC (to command (5))
7. Read Index
8. Read Text

Command (1) sets the stations on the line to control mode. Command (2) initiates the polling operation, beginning with the station specified by the "entry" operand in the READ macro. If a positive response is returned before the end of the list is reached, the status modifier is set, causing the next command, (3), to be skipped; command (4) transfers to command (7), followed by (8), which functions like commands (4) and (5) in the "open-type" Auto Poll operation.

If, however, the end of the list is reached and no positive response has been received, command (3) is executed, giving control to command (5), which restarts the polling operation at the beginning of the polling list. Polling proceeds automati-

cally, and each time the end of the list is reached, command (6) gives control to (5), and the polling starts again. If a positive response is received during a pass through the line, command (6) is skipped (just as command (3) is skipped above), and commands (7) and (8) are executed as before.

Programming Notes: The same programming notes given above apply to Auto Poll operations with a wraparound list.

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

Note: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode or a remote IBM 3270; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an

IBM 2780 using Transcode or a remote IBM 3270; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1.)

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.
2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.
3. The remote 3270 always responds to the READ Interrupt macro with an EOT. The problem program must determine whether all data was received by checking for an ETX at the end of the previous message block. If an ETB is present instead, all data was not received. If the rest of the data is wanted, the problem program can reread the message.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial and Reset (TIR)

WRITE Initial addresses a remote station and if the response to addressing is positive, writes message text, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Read Response
6. Write EOT (TIR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX)

WRITE Initial Transparent and Reset (TIXR)

WRITE Initial Transparent addresses a remote station, and if the response to addressing is positive, writes message text and ending characters DLE ETX, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETX
6. Read Response
7. Write EOT (TIXR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text to a device with Write Structured Field capability. For all other remote 3270 devices, text transmission must always be in nontransparent mode.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block addresses a remote station, and if the response to addressing is positive, writes message text and ending characters DLE ETB, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETB
6. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text to a device with Write Structured Field capability. For all other remote 3270 devices, text transmission must always be in nontransparent mode.

This Page Intentionally Left Blank

WRITE Initial Conversational (TIV)

WRITE Initial Conversational addresses a remote station and if the response to addressing is positive, writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write EOT
2. Write Addressing Sequence
3. Read Response
4. Write Text
5. Read Response
6. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2770, and 2972 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Initial Conversational Transparent (TIVX)

WRITE Initial Conversational Transparent addresses a remote station, and if the response to addressing is positive, writes message text and the ending characters, DLE ETX, that must follow the transparent data. The macro then reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETX
6. Read Response
7. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text, to a device with Write Structured Field capability. For all other remote 3270 device, text transmission must always be in nontransparent mode.

2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue (TT)

WRITE Continue and Reset (TTR)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response
3. Write EOT (TTR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text to a device with Write Structured Field capability. For all other remote 3270 devices, text transmission must always be in nontransparent mode.

WRITE Continue Transparent (TTX)

WRITE Continue Transparent and Reset (TTXR)

WRITE Continue Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Write EOT (TTXR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because transmission to these types of stations must always be in nontransparent mode.

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE ETB, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETB
3. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text to a device with Write Structured Field capability. For all other remote 3270 devices, text transmission must always be in nontransparent mode.

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or

NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2770, and 2972 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending character, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2972, because text transmission to this station must always be in nontransparent mode.

For a remote IBM 3270, this macro instruction can only be used when sending message text to a device with Write Structured Field capability. For all other remote 3270 devices, text transmission must always be in nontransparent mode.

2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last response (ACK-0, ACK1, NAK or a conversational text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-Before-Transmit (TW)

WRITE Wait-before-transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or Read Repeat, or in response to a conversational-type Write operation executed by the remote computer. The only valid responses to WACK are ENQ and EOT. You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to relinquish use of the line. After sending the EOT, the next operation must be an initial-type Read or Write operation, to again gain use of the line.

1. Write EOT

BSC SWITCHED POINT-TO-POINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can communicate with the central computer over a switched line, except as noted in individual macro instruction descriptions.

Contact between central computer and a remote station over a switched line can be established in numerous ways, representing the various combinations of these alternatives:

- Is the central computer to call a remote station or answer a call from a remote station?
- Is the calling or answering function to be automatic or manual? That is, are the TCU and the common carrier equipment at the central computer equipped to perform the calling (dialing) or answering function under program control, or must the operator at the central computer perform these functions?
- Is the data set (modem) at the central computer capable of generating and transmitting a "data tone" to signify to a calling station that data transfer can proceed, or must the user program supply the data tone?
- Once contact has been established, is the central computer to send an ID sequence to the remote station, is the central computer to receive an ID sequence from the remote station, or both (or neither)? If ID sequences are to be received from remote stations, do all stations with which contact may be established have to send the same ID sequence? Or can each send a unique sequence?
- Once contact has been established, is the direction of the first message transmission to be toward the remote station (that is, a Write Text operation), or toward the central computer (that is, a Read Text operation)?

Each of the various available combinations of the foregoing alternatives is represented by a combination of a specific READ or WRITE macro instruction option and a terminal list having a specific format and content. These are shown in Figure 30.

DEFINING TERMINAL LISTS

See the explanation of the DFTRMLST macro instruction for general information on defining these lists. Given below are the

specific coding requirements for switched point-to-point operations.

Automatic Calling and Answering - With Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro instruction like this:

```
[SWLST,AD,dialcount,dialchars,entrylength,
[userlength],idcount,idsent
|{,(authsequence[,controlvalue]
|[,userdata])}...]
```

This type of list is for use with a WRITE Connect (TC) macro instruction.

To define an answering list, code the operand field of a DFTRMLST macro like this:

```
[SWLST,AN,entrylength,
[userlength],idcount,idsent
|{,(authsequence[,controlvalue]
|[,userdata])}...]
```

This type of terminal list is for use with a READ Connect (TC) or Read Connect with Tone (TCW) macro instruction.

See the section DFTRMLST Macro -- SWLST Form, for detailed information on coding DFTRMLST macros of the SWLST form.

Automatic Calling and Answering - Without Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
[BSCLST,dialcount,dialchars,numrec,ridseq,
[numsent,tidseq]
```

The ridseq operand must end with ACK-0; the tidseq operand must end with ENQ. This type of list is for use with a READ Initial or WRITE Initial macro instruction.

To define an answering list, code the DFTRMLST operand field like this:

```
[BSCLST,0,numrec,ridseq,numsent,tidseq]
```

The ridseq operand must end with ENQ; the tidseq operand must end with ACK-0. This type of list is for use with a READ Initial macro instruction.

To:	and to...	and if Expanded ID Verification...	to be used, issue a...	macro that refers to a terminal list defined like this:
CALL a remote station, using...	Read text	is not ³	READ TI	BSCNST,dialcount, dialchars,numrec, ridseq,numsent,tidseq (See Note 1)
Automatic Calling	Write text	is	WRITE TC (followed by WRITE TT)	SWLST,AD,dialcount, dialchars,entrylength, [userlength],idcount idsent{[(authsequence[,controlvalue][,userdata])]}...
		is not	WRITE TI ⁶	BSCNST,dialcount, dialchars,numrec, ridseq,numsent,tidseq
Manual Calling	Write text ⁴	is	WRITE TC (followed by WRITE TT)	SWLST,MD,entrylength,[userlength], idcount,idsent {[(authsequence [,controlvalue] [,userdata])]}...
		is not	WRITE TC (followed by WRITE TT)	WTLIST,0,numrec, ridseq,numsent, tidseq
ANSWER a remote station,using...	Read text	is	READ TC or TCW	SWLST,AN,entrylength, [userlength], idcount,idsent {[(authsequence [,controlvalue] [,userdata])]}...
		is not	READ TI	BSCNST,0,numrec, ridseq,numsent, tidseq (see Note 2)
Automatic Answering				
Manual Answering	Read text ⁵	is not ³	READ TI	(if data set [modem] automatically generates tone) BSCNST,0,numrec, ridseq,numsent,tidseq (if data set [modem] does not automatically generate tone) WTLIST, 0,numrec,ridseq, numsent,tidseq, length,area

¹Alternatively, if no ID sequences are required, the list may be coded as:

DIALST,dialcount, dialchars

²Alternatively, if no ID sequences are required, the list may be coded as: DIALST,0

³Expanded ID verification not available.

⁴Text cannot be read from the remote station.

⁵Text cannot be written to the remote station.

⁶Or WRITE TIX, TIV, or TIVX.

Figure 30. Summary of BSC Switched Line READ and WRITE Macro and Terminal List Options

Omitting ID Sequence: If no ID sequences are desired, omit, in the ridseq and tidseq operands, all but the ENQ and ACK-0 characters. When no ID characters are specified, the numrec, ridseq, numsent, and tidseq operands are as follows:

	<u>Calling List</u>	<u>Answering List</u>
numrec	2	1
ridseq ¹	ACK-0	ENQ
numsent	1	2
tidseq ¹	ENO	ACK-0

(Alternatively, if ID sequences are not desired, you may define a calling list using a DFTRMLST in which the operand field is coded DIALST,dialcount,dialchars; an answering list using a DFTRMLST in which the operand field is coded DIALST,0.)

Manual Calling² - with Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
SWLST,MD,entrylength,
[userlength],idcount,idsent
[,{(authsequence[,controlvalue]
[,userdata])}]...
```

This type of list is for use with a WRITE Connect (TC) macro instruction.

Manual Calling and Answering - Without Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
WTLIST,0,numrec,ridseq,numsent,tidseq
```

The ridseq operand must end with ACK-0; the tidseq operand must end with ENQ. The sequence specified by tidseq may contain up to 15 characters (excluding the ENQ). This type of list is for use with a WRITE Connect macro instruction.

¹The ridseq and tidseq operands must be coded in hexadecimal representation of the appropriate transmission code bit patterns of the ENQ and ACK-0 characters.
²Manual answering with expanded ID verification is not available.

An answering list may be coded in one of two ways, depending on whether the data set (modem) at the answering station (the central computer) is designed to automatically generate a data tone upon receiving a call.

Data Sets Without Tone: If the data set does not generate a tone, BTAM sends a user-specified character sequence that the operator at the calling station hears as an audible tone.

To define an answering list for a line equipped with a data set that does not generate a tone, code the operand field of a DFTRMLST macro like this:

```
WTLIST,0,numrec,ridseq,numsent,tidseq,
length,area
```

The ridseq operand must end with an ENQ; the tidseq operand must end with ACK-0. The sequence specified by tidseq may contain up to 15 characters (excluding the two-character sequence, ACK-0).

The length and area operands specify the length of the character sequence used as a data tone and the address of that sequence.

The data tone should be about three seconds long. To obtain a tone of this duration requires a length of about 255 characters, for a 600 bits-per-second communication line, or about 450 characters, for a 1200 bps line. A sequence of X'FF' is recommended for the data tone. This type of list is for use with a READ Initial macro.

Data Sets With Tone: To define an answering list for a list for a line equipped with a data set that generates a tone, code the DFTRMLST operand field like this:

```
BSCLIST,0,numrec,ridseq,numsent,tidseq
```

The ridseq and tidseq operands are as explained above (under Data Sets Without Tone). This type of list is for use with a READ Initial macro.

If no ID sequences are desired, omit in the ridseq and tidseq operands all but the ENQ and ACK-0 characters. When no ID characters are specified, the numrec, ridseq, numsent, and tidseq operands are as shown above under Automatic Calling and Answering (Without Expanded ID Verification).

DEFINING TERMINAL LIST (SWLST) FOR EXPANDED ID VERIFICATION

Answering List

A READ Connect macro for Automatic Answering, with Expanded ID Verification, requires an answering list defined as follows:

Name	Operation	Operands
symbol	DFTRMLST	SWLST,AN,entrylength, [userlength],idcount, idsent{[, (authsequence [, controlvalue] [, userdata])}...}

SWLST

specifies a list structure for expanded BSC ID verification.

AN

specifies that an answering list (to be used by the READ Connect or Read Connect with Tone macro) is to be defined.

entrylength

specifies the number of bytes to be allocated for each list entry containing a user-defined authorized ID ENQ sequence. The integer specified should equal the number of bytes required to accommodate the authorized ID ENQ sequence of maximum length, plus the userdata field, if present (4 or 0), plus one (for the entry's control byte). Authorized ID ENQ sequences of less than the maximum length are assembled left-justified within the fixed-length ID field allocated for each entry. Each userdata field (if any) and control byte have the same offset within all entries. (The value specified may be zero if no other sequence than ENQ, alone, is expected and ENQ is not put in the list.)

userlength

specifies whether a four-byte userdata field is to be allocated for each list entry containing an authorized ID ENQ sequence. A code of 4 means to allocate; 0 means not to allocate. The default option is 0.

idcount

specifies the length (in bytes) of the field required to accommodate the ID characters (if any) and ACK-0 defined by the idsent operand. The range permitted is 2 (ACK-0 alone) through 17; up to 15 ID characters may be specified.

idsent

specifies the hexadecimal representation of the ID ACK-0 sequence to be sent to the remote station. While the ID characters (if any) are of your choosing, the ACK-0 sequence is required. Upon receiving an ID ENQ sequence during execution of a READ Connect, BTAM checks the control byte value of the corresponding list entry, and transmits the ID ACK-0 sequence if the checked value is 0. (See the discussion of the controlvalue suboperand, or the discussion of the Read Connect channel program, for the explanation of the BTAM actions performed for the various control byte values.)

authsequence

specifies the hexadecimal representation of an authorized ID ENQ sequence. Each ID ENQ sequence is defined in a separate sublist along with its corresponding control byte value and user data (if any). You should code a separate ID ENQ sequence for each authorized sequence that can be received on a Read Connect operation using the particular answering list being defined. ID ENQ sequences of varying lengths can be defined within the same DFTRMLST macro. Each sequence specified must include the ENQ character at the end. You may define ENQ alone as an authsequence operand to service remote stations not employing ID verification.

controlvalue

each list entry assembled for an authsequence sequence has an associated control byte, the value of which determines the automatic BTAM action to be performed when the sequence is received on a Read Connect operation using the list. The values and corresponding BTAM actions are:

0

specifies that BTAM is to send the idsent sequence and read a message block (if any) from the calling station. If you omit the controlvalue suboperand within a sublist, this value is assumed. (If the controlvalue operand is omitted, two commas must precede a coded userdata operand in the same sublist, because they are positional operands within the sublist.) You may specify this value for a list entry containing an ID ENQ sequence or the single ENQ character.

1

specifies that BTAM is to break the line connection and restart the channel program at the Enable command (to

await a new call). You may specify this action if BTAM is not to service a particular calling station at the time of the call. Typically, this action would be specified for reasons of priority (time-of-day scheduling). You may specify this value for a list entry containing an ID ENQ sequence or the single ENQ character.

2

specifies that BTAM is to post normal completion of the Read Connect immediately, with the address of the received ENQ character in the first word of the answering list. This permits control to be returned to the user program so that it can specify the subsequent actions to be performed. You may specify this value only for a list entry containing a single ENQ character (i.e., not containing an ID sequence). This permits the user program to issue a subsequent READ Continue or READ Repeat macro to send ACK-0 or NAK to a calling station that is not prepared to receive an ID sequence.

userdata

specifies the relocatable expression to be assembled right-justified in the userdata field of the associated list entry. If you omit this suboperand and userlength specifies 4, four non-initialized bytes are allocated for the corresponding list entry. (No boundary alignment can be assumed for the user data field.)

Notes:

1. A maximum of 194 sublists can be coded for an answering list of the SWLST form.
2. The CHGENTRY macro can be used to change the control byte of an answering list of the SWLST form during program execution.

The first fullword of the list is the area in which BTAM stores the address of the entry containing the ID ENQ sequence corresponding to the received sequence. See Appendix A for the format of the assembled answering list.

Calling List

For Automatic or Manual Calling, with Expanded ID Verification, A WRITE Connect macro requires a calling list defined as follows:

Name	Operation	Operands
symbol	DFTRMLST	SWLST, {AD}, {MD} [dialcount, dialchars,] entrylength, [user- length], idcount, idsent [,{(authsequence [, controlvalue] [, userdata])}]...]

SWLST

specifies a list structure for expanded BSC ID verification.

AD

specifies that an auto-dial calling list is to be defined. In this case, the dialcount and dialchars operands are required so that program-initiated dialing can occur. The corresponding Write Connect channel program begins with a Dial command.

MD

specifies that a manual-dial calling list is to be defined. In this case, omit the dialcount and dialchars operands, because the dialing operation is initiated by the central computer operator. The Write Connect channel program with which a manual-dial calling list is used begins with an Enable command.

dialcount

specifies the number of dial characters (bytes) used in the dialing operation. Code this operand only if you code AD as the preceding operand.

dialchars

specifies the decimal digits of the telephone number to be dialed. Code this operand only if you also code AD.

entrylength

specifies the number of bytes to be allocated for each list entry containing a user-defined authorized ID ACK-0 sequence. The integer specified should equal the number of bytes required to accommodate the authorized ID ACK-0 sequence of maximum length, plus the userdata field, if present (4 or 0), plus one (for the entry's control byte). Authorized ID ACK-0 sequences of less than the maximum length are assembled left-justified within the fixed-length ID field allocated for each entry. Each userdata field (if any) and control byte have the same offset within all entries.

userlength

specifies whether a four-byte userdata

field is to be allocated for each list entry containing an authorized ID ACK-0 sequence. A code of 4 means to allocate; 0 means not to allocate. The default option is 0.

idcount

specifies the length (in bytes) of the field required to accommodate the ID characters (if any) and ENQ defined by the idsent operand. The range permitted is 1 (ENQ alone) through 16; up to 15 ID characters may be specified.

idsent

specifies the hexadecimal representation of the ID characters (if any) and ENQ to be sent to the remote station. Typically, the ID characters to be sent will convey station identification. The ID characters, if any, are of your choosing; the ENQ character is required.

authsequence

specifies the hexadecimal representation of an authorized ID ACK-0 sequence. Each ID ACK-0 sequence is defined in a separate sublist along with its corresponding control byte value and user data (if any). You should code a separate ID ACK-0 sequence for each authorized sequence that can be received from remote (answering) stations. ACK-0 must be coded following each ID sequence; it must not be coded where no ID sequence is used. BTAM checks for reception of ACK-0 or NAK, alone, on a Write Connect operation without requiring that they appear in the list.

controlvalue

each list entry assembled for an authsequence sequence has an associated control byte. For any received ID sequence terminated by ACK-0, BTAM ignores the control byte. When a valid ID sequence terminated by NAK is received during a Write Connect operation, BTAM examines the control byte of the entry whose ID matches the received ID. The control byte value determines the BTAM action to be performed. The values and BTAM actions are:

0

specifies that upon receipt of the sequence, BTAM is to post completion of the operation immediately.

1

specifies that upon receipt of the sequence, BTAM is to resend the ID ENQ sequence. This option has meaning only when the ID NAK sequence has been sent, indicating that the remote sta-

tion is not ready to receive, and you wish to retry, expecting that the remote station will shortly become ready to receive. The maximum number of retries performed for this control byte value is seven. If more retries than this are desired, you can reissue the WRITE Connect macro; BTAM bypasses the initial Enable or Dial command if the line connection is already established.

userdata

specifies the relocatable expression to be assembled right-justified in the userdata field of the associated list entry. If you omit this suboperand and userlength specifies 4, four non-initialized bytes are allocated for the corresponding list entry. (No boundary alignment can be assumed for the user data field.)

Notes:

1. A maximum of 192 sublists can be coded for a calling list of the SWLST form.
2. The control byte values for a calling list cannot be changed by use of the CHGNTRY macro.

The first fullword of the list is the area in which BTAM stores (prior to completion posting) the address of the list entry associated with the received ID sequence. See Appendix A for the format of the assembled calling list.

READ MACRO INSTRUCTIONS

READ Initial (TI)
(Using Automatic Calling List -- BSCLST DIALST)

READ Initial calls a remote station, writes the central computer's identification sequence and ENQ to the station, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation continues by writing EOT (indicating that the central computer does not wish to send), reading ENQ and responding with ACK-0, then reading a message block from the remote station. If the identifications do not match, the Read operation ends with command (3) and is posted as complete.

1. Dial Dial Digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write EOT
5. Read ENQ
6. Write ACK-0
7. Read Text

READ Initial (TI)
(Using Automatic Answering List -- BSCLST, DIALST)

READ Initial answers a call from a remote station, reads the identification sequence of the remote station and an ENQ, writes ACK-0 to indicate that the central computer is ready to receive, and reads a message block from the remote station.

1. Enable
2. Read ID ENQ
3. Write ID ACK-0
4. Read Text

READ Initial (TI)
(Using Manual Answering List -- BSCLST, WTLIST)

A READ Initial macro using a manual answering list is for use where the central computer is not capable of automatically answering calls from remote stations; the operator at the central computer must answer them manually. Operation is as follows.

The channel program first enables the line so that calls can be received. When the telephone rings, the computer operator answers it, and may verify the identity of the calling station (if that call was initiated by the remote station operator rather than automatically). The operator then places the data set (modem) in data mode. (This terminates the Enable command.)

If the manual answering list is of the WTLIST format (used where the data set (modem) does not automatically generate a data tone), the channel program then sends a user-specified character sequence that the operator at the remote station hears as a tone. If the list is of the BSCLST format (used where the data set does generate a tone), the channel program does not send the character sequence.

The channel program then reads an identification sequence, ending in ENQ, from the remote station. If the sequence does not match the expected sequence, the Read Initial operation ends at this point, and is posted complete-with-error in the event control block. If the two sequences do match, the channel program sends the identification sequence of the central computer, then reads a message block from the remote station.

1. Enable
2. Write Data Tone Characters (for WTLIST only)
3. Read ID ENQ
4. Write ID ACK-0
5. Read Text

Programming Note: It may be desirable, after issuing the READ Initial, to send a message to the console operator (using the WTO macro), instructing him to answer calls received by the computer.

READ Connect (TC) (Expanded ID Verification) (Using Automatic Answering List - SWLST)

READ Connect is used to allow initial contact to be established with a remote BSC station and to perform a specific action based on the ID sequence, if any, received from the remote station. The possible actions include reading message blocks, disconnecting the line, and immediately returning control to the user program.

After the sequence is received, BTAM analyzes it. If the sequence matches one of the authorized sequences in the answering list, BTAM places the address of the entry containing the matching ID-ENQ sequence (or ENQ alone) in the first fullword of the list. BTAM then examines the control byte of that list entry to determine which action to take.

If the control byte value is 0, BTAM restarts the channel program to send the ID ACK-0 sequence (or ACK-0 alone) given in the list, and then reads a message block, if any. If the control byte value is 1, BTAM restarts the channel program to break the line connection, and then restarts the channel program from the beginning Enable command. If the control byte value is 2, BTAM immediately posts normal completion (X'7F'). (A control byte value of 2 is for use when no ID sequence is employed, and you wish to follow normal completion (X'7F') of the Read Connect operation with a READ Continue macro.)

If the received sequence does not match any of the authorized ID-ENQ sequences (or ENQ alone), BTAM determines whether ENQ alone, an invalid sequence, or DLE EOT was received. If ENQ alone was received, BTAM posts normal completion (X'7F').

If an invalid sequence was received, BTAM retries the Read ID ENQ command up to seven times. If all retries are unsuccessful, BTAM disconnects the line, turns on bit 3 of DECFLAGS, and posts a completion code of X'7F' (normal completion). If DLE EOT was received, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F').

If a timeout occurs on the Read ID ENQ command, BTAM disconnects the line and restarts the channel program at the Enable command.

This macro is used only when the expanded ID verification facility is to be employed. The entry operand of the READ Connect macro must specify the name of an answering list of the SWLST format, as defined by a DFTRMLST macro. The channel program generated for the READ Connect macro is:

1. Enable
2. Read ID ENQ (or ENQ alone)
3. Write ID ACK-0 (or ACK-0 alone)
4. Read Text
5. Write DLE EOT
6. Disable
7. TIC to Enable command

READ Connect with Tone (TCW) (Expanded ID Verification) (Using Automatic Answering List -- SWLST)

READ Connect with Tone functions the same as READ Connect (TC), as described above, except that the channel program contains an added command, Write Data Tone Characters. This macro is for use on a line equipped with an automatic answering unit that does not automatically send a data tone upon receiving a call. Upon completion of the Enable command, which occurs when a call is received, the channel program sends a user-specified character sequence that the operator at the calling station hears as an audible tone.

The character sequence that constitutes the tone must be coded in the user program. The address and the length of the tone character sequence must be specified in the outarea and outlength operands of the READ TCW macro instruction.

The data tone should be about three seconds long. To obtain a tone of this duration requires a length of about 255 characters, for a 600 bits-per-second communications line, or about 450 characters for a 1200 bps line. A sequence of X'FF' is recommended for the data tone. (Notice that the address and length of the tone sequence are specified in the READ macro, not in the DFTRMLST macro, as is the case for manual answering, without expanded ID verification.)

The channel program generated for the READ Connect with Tone macro is:

1. Enable
2. Write Data Tone Characters
3. Read ID ENQ (or ENQ alone)
4. Write ID ACK-0 (or ACK-0 alone)
5. Read Text
6. Write DLE EOT
7. Disable
8. TIC to Enable command

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

Note: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Inquiry Monitor (TQM)

Read Inquiry Monitor reads an ENQ (or other response) from a 3275 display station equipped with the Dial feature and does not time out. It is an alternative to READ Inquiry and prevents a timeout when the 3275 has nothing to send within the allotted time.

1. Prepare
2. Read ENQ

This macro instruction does not prevent disconnect if the CPU calls the 3275 and the 20-second interval timer feature is installed and enabled, and the operator does not respond within 20 seconds.

Programming Note: If the application program has a write operation to perform or does not wish to wait indefinitely for the 3275 to send in a response, the application program can issue a RESETPL macro to terminate the Prepare.

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1).

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.
2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI) (Using Automatic Calling List -- BSCLST)

WRITE Initial calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text to the remote station and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX) (Using Automatic Calling List -- BSCLST)

WRITE Initial Transparent calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text and the ending characters DLE ETX to the remote station, and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Write DLE ETX
6. Read Response

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text. (Component selection characters are required unless the Job Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.) The reason is that when operating in transparent mode, the 2770 and 2780 do not recognize com-

ponent selection characters within message text.

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write TTVX or Write TTX (or TTE) operation to send the transparent text.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text, and the ending characters DLE ETB to the remote station, and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Write DLE ETB
6. Read Response

Programming Note: See programming note under Write TIX macro instruction.

WRITE Initial Conversational (TIV) (Using Automatic Calling List -- BSCLST)

WRITE Initial Conversation calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text to the remote station and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Read Response
6. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Initial Conversational Transparent (TIVX) (Using Automatic Calling List -- BSCLST)

WRITE Initial Conversational Transparent calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text and the ending characters DLE ETX to the remote station and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Write DLE ETX
6. Read Response
7. Read Text

Programming Notes:

1. This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).
2. This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text. (Component selection characters are required unless the JOB Select switch (2770) or Mode switch (2780) is set for printing or punching (or some

other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.)

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write Continue (TT) or Write Continue Transparent (TTX) operation to send the transparent text.

WRITE Connect (TC)
(Using Manual Calling List -- WTLIST)

A WRITE Connect macro is for use where calls to remote stations must be initiated manually by the console operator rather than by program control. Operation is as follows.

The channel program first enables the line so that calls may be initiated. After issuing the WRITE Connect macro, the program must inform the console operator (as by a WTO macro) to dial the remote station. The operator dials the call, and upon hearing a data tone from the remote station, places the data set (modem) in data mode. (This terminates the Enable command.)

The channel program then writes to the remote station the identification sequence of the central computer, then reads the identification sequence of the remote station.

If the received sequence matches the expected sequence, the operation is posted complete (without error) in the event control block. If the sequences do not match, the operation is posted complete-with-error

This macro does not write message text to the remote station; one or more WRITE Continue macros should be issued for this purpose following the WRITE Connect macro.

1. Enable
2. Write ID ENQ
3. Read ID ACK-0

WRITE Connect (TC) (Expanded ID Verification)
(Using Automatic or Manual Calling List -- SWLST)

WRITE Connect is used to originate a call to a remote BSC station, either through program-initiated (automatic) dialing or through manual dialing, and to cause exchange of identification sequences (or ENQ and ACK-0) between the central computer and the remote station.

The entry operand of the WRITE Connect macro must specify the name of a calling

list of the SWLST format, as defined by a DFTRMLST macro. If the DFTRMLST macro specifies the AD operand, the automatic-dialing channel program is generated; if DFTRMLST specifies the MD operand, the manual dialing channel program is generated.

If the response from the called remote station is an ID ACK-0 sequence that matches one of the authorized ID ACK-0 sequences in the calling list, BTAM places the address of the entry containing the matching ID in the first fullword of the list and posts normal completion (X'7F').

If the response from the remote station is an ID NAK sequence, the ID portion of which matches the ID portion of one of the authorized ID ACK-0 sequences, BTAM places the address of the entry containing the matching ID in the first fullword of the list, then examines the control byte of that entry. If the control byte is 0, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If the control byte is 1, BTAM retries the Write ID ENQ (or ENQ alone) command.

If the response from the remote station is an invalid ID sequence (that is, one that does not match any of the authorized ID sequences in the calling list), BTAM retries the Write ID ENQ (or ENQ alone) command.

In the two foregoing situations in which BTAM retries the Write ID ENQ (or ENQ alone) command, the maximum number of retries is seven.

If all retries are unsuccessful, and a valid ID NAK sequence was received on the last retry, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If all retries are unsuccessful, and an invalid ID sequence was received on the last retry, BTAM breaks the line connection, turns on bit 3 of DECFLAGS, and posts normal completion (X'7F').

If the response from the remote station is ACK-0 (with no preceding ID), BTAM posts normal completion (X'7F'). If the response is NAK (with no preceding ID), BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If the response is WACK, BTAM turns on bits 0 and 1 of DECFLAGS and posts normal completion (X'7F').

If no response at all is received from the remote station, BTAM retries the Write ID ENQ (or ENQ alone) command up to seven times; if all retries are unsuccessful, BTAM breaks the line connection, sets X'01' in DECSSEN0, and posts a completion code of X'41'.

The channel program for automatic dialing is:

1. Dial
2. Write ID ENQ (or ENQ alone)
3. Read ID ACK-0 or ID NAK response

The channel program for manual dialing is:

1. Enable
2. Write ID ENQ (or ENQ alone)
3. Read ID ACK-0 or ID NAK response:

Programming Note: If the Write Connect operation ends with ID NAK, NAK, or WACK and you reissue the WRITE Connect macro, BTAM starts the channel program at the second command (Write ID ENQ) if the line connection is still established at the time the macro is issued. Otherwise, BTAM starts the channel program at the first command (Enable or Dial).

WRITE Continue (TT)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response

Programming Notes: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Continue Transparent (TTX)

WRITE Continue Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE ETB, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETB
3. Read Response

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending characters, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Note: This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last response (ACK-0, ACK-1, NAK, or a conversational text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-before-Transmit (TW)

WRITE Wait-before-Transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or READ Repeat, or in response to a conversational-type Write operation executed by the remote computer. The only valid responses to WACK are ENQ and EOT.

You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to indicate to the remote station that the central computer has no more message text to send, and reads a response. This macro is for giving the remote station the opportunity to transmit.

1. Write EOT
2. Read Response

WRITE Reset Monitor (TRM)

WRITE Reset Monitor writes an EOT to a 3275 equipped with the Dial feature to indicate to the 3275 that the central computer has no more message text to send, and awaits a response without timing out. It is an alternative to WRITE Reset and prevents a timeout occurring when the 3275 has nothing to send within the allotted time.

1. Write EOT
2. Prepare
3. Read ENQ

This macro instruction does not prevent disconnect if the CPU calls the 3275 and the 20-second interval timer feature is installed and enabled, and the operator does not respond within 20 seconds.

Programming Note: If the application program has a write operation to perform or does not wish to wait indefinitely for the 3275 to send in a response, the application program can issue a RESETPL macro to terminate the Prepare.

WRITE Break (TB)

WRITE Break sends a Disable command to the TCU, causing the TCU to break the switched line connection. This macro does not inform the remote station that the connection is to be broken.

1. Disable

WRITE Disconnect (TD)

WRITE Disconnect writes DLE EOT, indicating to the remote station that the line connection is to be broken, then sends a Disable command to the TCU, causing the TCU to break the switched line connection.

1. Write DLE EOT
2. Disable

LOCAL READ AND WRITE OPERATIONS

LOCAL IBM 3270 DISPLAY SYSTEM

For information about using READ and WRITE macro instructions for the local 3270 display system, see the section "IBM 3270 Information Display System -- Programming Considerations."

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial reads modified fields from a local 3270 display station after an attention interruption has been generated by the display station operator.

1. Select command
2. Read modified command

READ Modified (TM)

READ Modified reads modified fields from a local 3270 device independently of action by the display station operator.

1. Select command
2. Read modified command

READ Modified from Position (TMP)

READ Modified from Position reads modified fields from a local 3270 device beginning at a specified location in the buffer.

1. Select command*
2. Write command (to set buffer address)
3. Read modified command

READ Buffer (TB)

READ Buffer reads the entire buffer of a local 3270 device.

1. Select command*
2. Read buffer command

READ Buffer from Position (TBP)

READ Buffer from Position reads the entire buffer of a local 3270 device beginning at a specified location.

1. Select command*
2. Write command (to set buffer address)
3. Read buffer command

*For more information concerning the Prepare to Read Operations see, IBM 3270 (Local) Information Display System -- Programming Considerations, Prepare to Read Operations, in this book.-

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

WRITE Initial writes a message to a local 3270 device.

1. Select command*
2. Write command

WRITE Erase (TS)

WRITE Erase clears the buffer of a local 3270 device to nulls (binary zeros) and then writes a message to the device.

1. Erase/write command

WRITE Erase Alternate (TSA)

WRITE Erase Alternate places the specified display or printer into a mode of using the alternate buffer size for a device, clears the buffer to nulls (binary zeros) and then writes a message to the device. This macro instruction does not apply to the 3272 Control Unit.

1. Erase/write alternate command

WRITE Structured Field (TSF)

WRITE Structured Field transfers a data stream containing a structured field (s) to a specified device. The Write Structured Field macro instruction applies only to those devices with structured field capability. Devices unable to support structured fields, will reject this macro with a command reject error. This instruction will not cause any specific device action; however, the structured field contains a type field which will cause a device action. For more information on structured fields see, IBM 3270 Information Display System, Library Users Guide, GA23-0058 for the manuals applicable to the configuration.

1. Write structured field command

WRITE Unprotected Erase (TUS)

WRITE Unprotected Erase clears all unprotected fields in the buffer of a local 3270 device to nulls (binary zeros).

1. Erase all unprotected command
2. NOP

This Page Intentionally Left Blank

This chapter contains miscellaneous programming considerations for communicating between a central computer and any of the remote computers supported by OS/VS BTAM as remote stations: IBM System/370, IBM System/360 (including Model 20), IBM System/3, IBM System/7, IBM 1130, and IBM 1800. These considerations are in addition to those shown under "General Information" in the section "BSC Read and Write Operations" and under the major sections within that section covering the three types of line configuration (nonswitched point-to-point, nonswitched multipoint, and switched point-to-point).

Except where noted, these considerations apply equally to all of the foregoing types of remote computers.

Transmission over Nonswitched Point-to-Point Line

Initial contact between the central computer and the remote computer over a nonswitched point-to-point line is on a contention basis. That is, the line remains idle until either of the computers sends an ENQ character to the other computer, signifying its intent to begin a transmission. Sending the ENQ character is called bidding for the line.

Ordinarily, both computers will not simultaneously bid for the line. On rare occasions, however, bidding will be simultaneous. When this happens, one computer must defer to the other. The control programs in the two computers must be coordinated so that this deferral takes place.

In BTAM, you accomplish this by coding MODE=CNTRL in the DCB macro for the line, if you wish the central computer (or the remote computer, if it also is running under BTAM) to retain control. Conversely, you omit MODE=CNTRL if you wish to defer to the other computer. The opposite choice must then be made for the other computer. Assume that both computers are running under BTAM. In one BTAM program, MODE=CNTRL would be specified; in the other it would be omitted. If the remote computer is running under a control program other than BTAM, the equivalent action must be taken to assure coordination between the two computers.

When this coordination is done, the computer that is to retain control automatically resends the ENQ character, and the computer that is to defer executes a Read

command (or equivalent) in order to listen for the ENQ sent by the computer retaining control. Transmission begins when one computer successfully sends the ENQ character to the other; that is, receives an ACK-0 (positive acknowledgment). Thus, the direction of transmission is established by which computer sends the ENQ.

When the deferring computer is running under BTAM, and contention occurs, BTAM immediately ends the operation that sent the ENQ character and posts the operation complete-with-error (X'41' in the DECSDECB field of the DECB). When the computer retaining control is running under BTAM, and contention occurs, BTAM automatically restarts the operation to resend the ENQ character up to seven times.

Transmission over Nonswitched Multipoint Line

(Applicable only to System/360 Model 20 as a remote station.)

Communication between the central computer and a remote computer over a nonswitched multipoint line begins when the central computer places the line in control mode by sending an EOT character, then initiates contact with the desired computer by sending that computer's polling or addressing sequence.

Transmission from remote computer to central computer is initiated when BTAM, in the central computer, executes an initial-type READ macro instruction. This causes the polling sequences for each of the remote computers to be sent automatically, in turn, until either the last computer represented in the polling list is polled (for an open-type list) or until the last computer represented in the polling list is polled following execution of a RESETPL macro for the line (for a wraparound-type list).

Transmission from central computer to remote computer is initiated when BTAM, in the central computer, executes an initial-type WRITE macro instruction. This causes the addressing sequence for the desired remote computer to be sent; the WRITE macro specifies the addressing list entry representing the desired remote computer.

The details of the polling and addressing functions are discussed under BSC Nonswitched Multipoint Operation in the BSC Read and Write Operations chapter.

The polling and addressing sequences are specified identically for all types of remote computers. That is, up to seven polling or addressing characters may be sent. There is, however, a distinction in the handling of these characters by the 1800 as opposed to the System/3 and the 1130. The 1800 hardware itself responds to the first two characters of the polling sequence. Any further characters, if any, in the sequence are merely received by the user program; the hardware does not recognize them as polling characters. For the System/3 and the 1130, however, there is no hardware-generated response to the polling or addressing sequence; all of the characters are received by the program, which decides what response to return to the central computer.

As explained under the General Information section of the BSC Read and Write Operations chapter, double addressing must be used for the System/360 Model 20, System/3, and 1800; that is, the first two polling or addressing characters must be identical. Double addressing, though not required for the 1130, is advisable for the reasons mentioned in the General Information section. For the Model 20 and the 1800, the first two polling characters are set in the hardware at installation time; for the System/3 and the 1130, polling characters are specified in the program.

Transmission over Switched Point-to-Point Line

Communication between the central computer and a remote computer via the switched telephone network begins when either computer calls the other. The call may be made manually by the computer operator or it may be made automatically, where the transmission control unit at the computer is equipped with an automatic calling unit. Similarly, the operator at the called computer may answer manually or the TCU may answer automatically if it is equipped with an automatic answering unit.

BTAM provides the choice of calling a distant computer automatically or manually, and of answering calls from a distant computer automatically or manually.

Once the line connection is established, the calling computer sends an ENQ character to bid for use of the line, as is done in a nonswitched point-to-point (contention) system. The called computer, upon answering and successfully receiving the ENQ, returns an ACK-0 sequence (positive acknowledgment). Unlike a contention system, however, you may arrange for either the ENQ or the ACK-0, or both, to be preceded by from one to 15 identification characters. This allows the control program at the com-

puter receiving the ID characters to verify that the computer sending those characters is authorized to communicate with it. In BTAM, these ID characters are user-defined in the terminal list referred to by the macro instruction that initiates the transmission. The action BTAM is to take if an invalid ID sequence or a negative acknowledgment is received may also be user-specified.

The various alternatives to be used are determined by which type of READ or WRITE macro instruction you issue to initiate the transmission and by which type of terminal list you provide for use by that macro. See the explanations in the BSC Switched Point-to-Point Operation section of the BSC Read and Write Operations chapter. Figure 30 summarizes the choices.

System/3 Notes

Data Formats: IBM System/3 RPG II support uses the following formats for transmission of data. These formats must be followed when sending data to System/3 from a CPU.

- Non-transparent, non-ITB (End of Intermediate Transmission Character):
STX-data-ETX or ETB
- Non-transparent, ITB:
STX-data-ITB-data-ITB-data-ETX or ETB
- Transparent, non-ITB:
DLE-STX-data-ETX or ETB

Data can be either blocked or unblocked but must be of fixed length. Fixed record length and unblocked implies non-ITB mode and requires that all data between ITBs be of the same length.

Conversational Mode: Only one response to conversational data is allowed by System/3. To maintain proper line discipline, System/3 will send or accept a NULL message (STX-ETX sequence), in lieu of a data transfer following a conversational response.

Examples of situations in which this is necessary follow:

- I. System/370 Point-to-Point Contention BTAM Program (Initiate a Read)
 - READ TI
 - 1. Prepare
 - 2. Read ENQ
 - 3. Write ACK-0
 - 4. Read text

WRITE TT

5. Write text
6. Read response

WRITE TV

7. Write NULL message (STX-ETX sequence)
8. Read text

After the conversational transfer of data (items 4 and 5), the BTAM programmer must write a null message before continuing. If item 7 were a Write text, the System/3 would reply with an EOT.

II. System/370 Point-to-Point Contention BTAM Program (Initiate a Write)

WRITE TIV

1. Write ENQ
2. Read response
3. Write text
4. Read text

READ TT

5. Write ACK-1
6. Read NULL message (STX-ETX)

WRITE TV

7. Write text
8. Read text

After the conversational transfer of data (items 3 and 4), the BTAM programmer must realize that the System/3 will be sending a null message (item 6) instead of normal data.

WACK and TTD Responses: System/3 will transmit WACK or TTD at two second intervals during a wait time specified by an RPG programmer (default of 180 seconds). An EOT (Disc) sequence is sent after the elapsed time.

System/7 Notes

Programming Support: A System/7 with the BSCA feature (feature #2074) is supported by BTAM as a standard BSC processor station, similar to a System/3. The application program in the System/7 can communicate with the central computer using any of the following BSC programming support:

Assembler Language READ/WRITE macro instructions
Assembler Language GET/PUT macro instructions
FORTRAN IV READ/WRITE statements

With the BSCA feature, the System/7 is loaded from the central computer over a switched or multipoint BSC line. For a switched line, initial program load (IPL) is signalled in the ID exchange of the BSC station. For a multipoint line, IPL is transmitted in the selection sequence of the tributary station.

Switched-line IPL: DC1 [EBCDIC]
DC1 [EBCDIC]
ENQ

Multipoint IPL: Station selection sequence
DC1 [EBCDIC]
DC1 [EBCDIC]
ENQ

BSC Service Programs: Service programs can be loaded into a System/7 over a BSC line. These programs can be used to:

- Bootstrap-load an application program from a central computer (\$UBIPL)
- Transmit the contents of System/7 index registers to a central computer (\$UBREG and \$UBRGC)
- Transmit the contents of System/7 storage to a central computer (\$UBDMP)

If a program in the central computer tries to communicate with a System/7 program immediately after loading (using \$UBIPL on a switched line), the central computer should be prepared to reissue the Write Reset macro instruction. The Write Reset is necessary because the Read ENQ operation may time-out before the System/7 is fully prepared to respond.

Supported Functions: The following is a summary of the BSCA functions supported for System/7.

Line Types

Leased point-to-point
Switched point-to-point (manual answer)
Switched point-to-point (manual call)
Switched point-to-point (auto answer)
Multipoint tributary

Character Modes

ASCII (nontransparent only)
EBCDIC (transparent, nontransparent, and remote IPL)

BSC Line Functions

TTD (for specific duration)
WACK (for specific duration)
Data check retry
Switched line ID exchange (optional)
Limited conversational facilities (optional)

Multipoint tributary automatic negative response (optional)

- EOT poll response
- NAK selection response

Null message reception
Leading graphics reception
SOH (optional)
ITB nontransparency

Line Speeds/Clocking

1200 to 50,000 b.p.s.
Internal clocking (1200, 2000, and 2400 b.p.s.)
External clocking (modem)

The following BSC functions are not supported:

Transmission of ITB transparency
Remote workstation (such as HASP and RJE)
Central multipoint
Auto-Call

INITIATING TRANSMISSION TO AN IBM 2780

Transmission over Nonswitched Point-to-Point Line

Initial contact between the central computer and an IBM 2780 over a nonswitched point-to-point line is on a contention basis. That is, the line remains idle until either the computer or the 2780 sends an ENQ character to the other, signifying its intent to begin a transmission. Sending the ENQ character is called bidding for the line. Ordinarily, the central computer and the 2780 will not simultaneously bid for the line. On rare occasions, however, bidding will be simultaneous. When this happens, the central computer must defer to the 2780. To cause this to happen, do not code MODE=CNTRL in the DCB macro for the line, as doing so would cause BTAM to retain control.

When contention occurs, that is, both computer and 2780 send ENQ simultaneously, BTAM turns on bit 3 in the DECFLAGS field of the DECB for the line, posts the Write operation that sent the ENQ complete-with-error (X'41' in DECSDECB), and returns control to the user program. The user program should immediately issue an initial-type READ macro; this will cause BTAM to detect the next ENQ character sent by the 2780, respond by sending ACK-0, and then read message text from the 2780.

Transmission over Nonswitched Multipoint Line

Communication between the central computer and an IBM 2780 over a nonswitched multipoint line begins when the central computer places the line in control mode by sending an EOT character, then initiates contact with the 2780 by sending the addressing sequence for that 2780 or by polling the line to which the 2780 is connected.

Transmission from 2780 to central computer is initiated when BTAM executes an initial-type READ macro instruction. This causes the polling sequences for each of the remote 2780s to be sent automatically, in turn, until either the last 2780 represented in the list is polled (for an open-type list) or until the last 2780 represented in the list is polled following execution of a RESETPL macro for the line (for a wraparound list).

Transmission from central computer to 2780 is initiated when BTAM executes an

initial-type WRITE macro instruction. This causes the addressing sequence for the desired 2780 to be sent; the WRITE macro specifies the addressing list entry representing the 2780.

The details of the polling and addressing functions are discussed under BSC Non-switched Multipoint Operation in the BSC Read and Write Operations chapter.

The polling sequence for an IBM 2780 is always

x 6 ENQ

where the x may be any uppercase or lowercase character used as the station address of the 2780. This address is established at the time the 2780 is installed. The second character is always a 6 to indicate the card reader.

The addressing sequence for an IBM 2780 is always

x 3 ENQ (to address the printer) or

x 4 ENQ (to address the punch)

where the x is the station address, as described for the polling sequence.

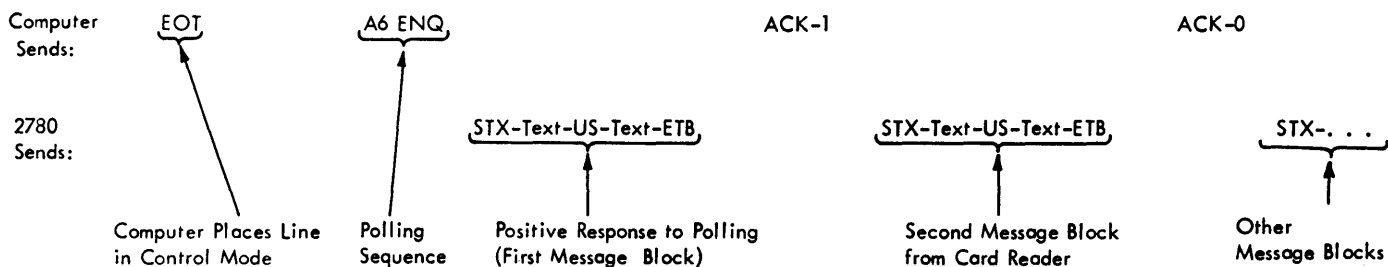
Polling and addressing examples are shown in Figure 31.

Transmission over Switched Point-to-Point Line

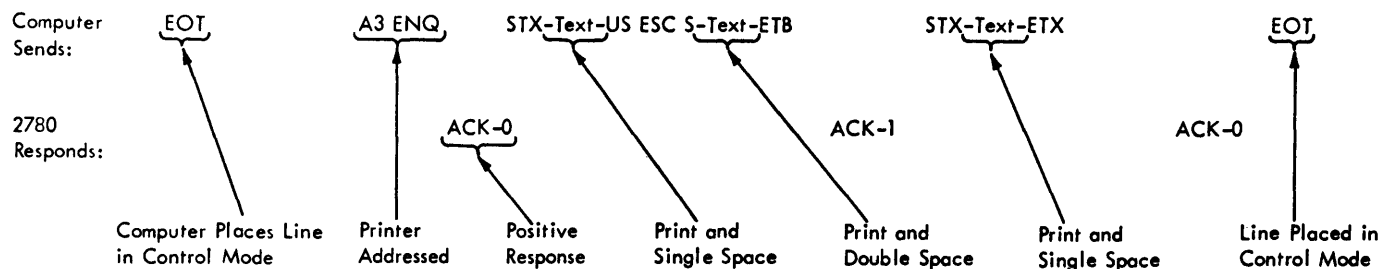
Communication between the central computer and an IBM 2780 via the switched telephone network begins when either the computer or the 2780 calls the other. The call may be made manually by the computer or terminal operator or it may be made automatically, where the 2780 or the transmission control unit at the computer is equipped with an automatic calling unit. Similarly, the operator at the computer or the 2780 may answer manually or the computer or 2780 may answer automatically if it is equipped with an automatic answering unit.

Once the line connection is established, the calling station (computer or 2780) sends an ENQ character to bid for use of the line, as is done in a nonswitched point-to-point (contention) system. The called station, upon answering and successfully receiving the ENQ, returns an ACK-0 sequence (positive acknowledgment). Unlike a contention system, however, you may

Example 1: Card reader polled



Example 2: Printer addressed



Example 3: Card Punch addressed

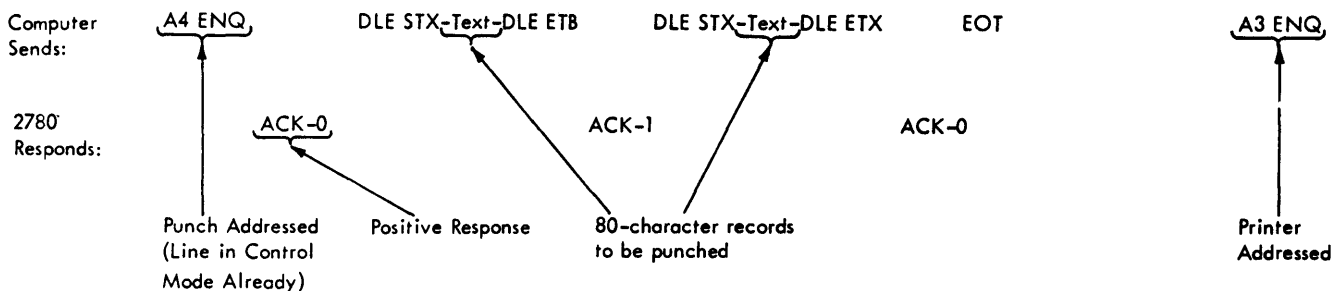


Figure 31. Multipoint Polling/Addressing Operations for 2780

arrange for either the ENQ or the ACK-0, or both, to be preceded by from one to 15 identification characters. This allows the 2780 operator or BTAM to verify that the station sending those characters is authorized to communicate with it. In the 2780 these characters, like the station address, are established at the time the 2780 is installed. In BTAM, these characters are user-defined in the terminal list referred to by the macro instruction that initiates the transmission. The action BTAM is to take if an invalid ID sequence or a negative acknowledgment is received may also be user-specified.

The various alternatives to be used are determined by which type of READ or WRITE macro instruction you issue to initiate the transmission and by which type of terminal list you provide for use by that macro. See the explanations in the BSC Switched Point-to-Point Operation section of the BSC Read and Write Operations chapter. Figure 30 summarizes the choices.

END-TO-END CONTROL CHARACTERS

In the character set of the IBM 2780 are four characters that provide secondary end-to-end control functions required by the printer, card reader, and card punch of the 2780.

One of these characters, BEL, is not used when the 2780 communicates with the central computer. (Its function in terminal-to-terminal operation is to cause the audible alarm to sound.)

Another end-to-end control character is EM (end-of-medium). When the card reader detects this character punched in a card it is currently reading, the reader ejects the card. When the EM character is sent to the card punch, it causes the punch to eject the card. These functions occur, however, only when the EM appears in nontransparent text. If it appears in transparent text, it is ignored. Regardless of whether it is in nontransparent or transparent text,

however, it is punched in the card, when sent to the card punch.

The remaining two end-to-end characters are ESC (Escape) and HT (Horizontal Tab). The ESC character is used in component selection (except multipoint lines), in vertical forms control, and in horizontal formatting. The HT character is used in horizontal formatting, when the 2780 is equipped with the Printer Horizontal Format Control feature.

The use of the ESC and HT characters is discussed under the functions with which they are associated.

COMPONENT SELECTION

In communicating with an IBM 2780 over a switched or nonswitched point-to-point line (but not a multipoint line), you must perform component selection when transmitting messages to the 2780, if the Mode switch at the 2780 is set at Transmit or Receive. If the Mode switch is set to Print or Punch, the 2780 ignores component selection messages; all messages it receives are automatically printed or punched. The Mode switch is manually set by the 2780 operator.

Component selection is accomplished by sending a nontransparent message beginning

with a two-character escape sequence. To select the punch, begin the message (following the STX character) with ESC 4. To select the printer, begin the message (after STX) with any one of the vertical forms control escape sequences (for example, ESC /, ESC A, ESC B). Thus, the vertical forms control escape sequences (discussed below) perform the dual function of selecting the printer and controlling the forms motion for the records in which they appear.

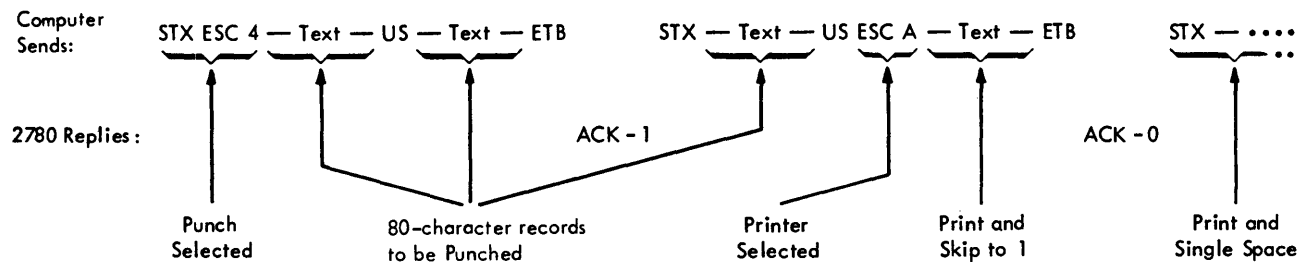
Once a component is selected, you need not reselect it in successive records to be sent to that component.

Because the 2780 recognizes component selection escape sequences only in nontransparent messages it receives, the first message, and any subsequent messages containing component selection sequences, must always be in nontransparent mode. Intervening messages can be in transparent mode. Figure 32 illustrates the use of component selection sequences.

Nonswitched Point-to-Point Line

When the 2780 finishes sending messages to the central computer, it transmits ETX (or DLE ETX). The computer responds to this with a positive acknowledgment (ACK-0 or ACK-1) if it detected no errors. Upon

Example 1: Nontransparent mode



Example 2: Transparent mode

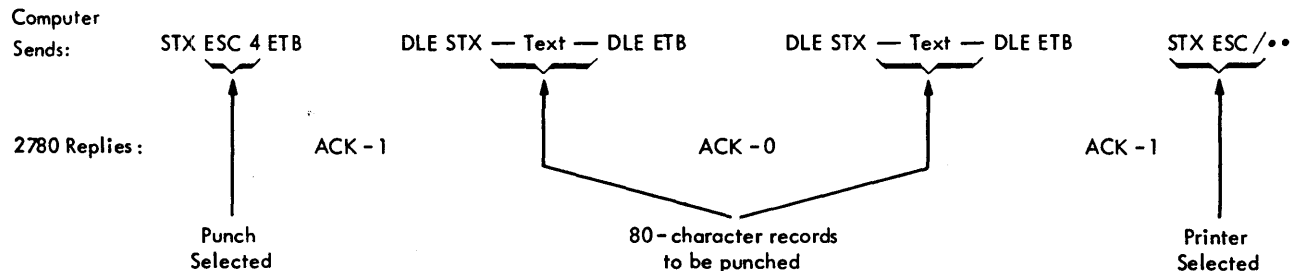


Figure 32. Examples of Component Selection for 2780

receiving the positive acknowledgment, the 2780 sends an EOT to the computer.

When the 2780 is equipped with the Automatic Turnaround feature, the card punch automatically enters ready status after the card reader reads a blank card. This permits the computer, upon receiving EOT, to immediately bid for the line (by sending ENQ) and select the punch. Without this feature, selection of the punch following a card reader operation requires operator intervention. Automatic selection of the printer, however, does not require the presence of the Automatic Turnaround feature.

Switched Point-to-Point Line

The 2780 normally does not transmit the disconnect signal (DLE EOT) to the central computer; it thus lets the user program at the central computer decide when to break the switched line connection after a transmission. As mentioned above, when the 2780 receives a positive acknowledgment to the last message it sends, it returns an EOT to the computer. The user program may then send the disconnect signal (DLE EOT) or, via a WRITE Inquiry macro, bid for use of the line.

When the user program sends an EOT to the 2780, the 2780 reacts in one of two ways.

If the card reader is in ready status, the 2780 bids for use of the line by sending ENQ. Upon receiving the ENQ, BTAM posts normal completion (X'7F' in the DECSDECB field of the DECB). The user program may, if it is ready to receive, then issue a READ Continue (TT or TTL) macro.

If the card reader is not in ready status, the 2780 does not respond to the EOT. Instead, it continues to monitor the line for an ENQ from the computer. If it receives no ENQ within about 20 seconds, the 2780, if equipped with the Automatic Answering feature, will then break the line connection (go "on-hook").

Vertical Forms Control

Vertical positioning of forms at the 2780 printer is controlled by predefined escape sequences. These are shown in Figure 33.

You must place the sequence in the first two character positions of each message block (print line record) you send to the printer. Exception: If you wish single spacing and, for a switched or nonswitched point-to-point line, you require no component selection characters, you may omit the escape sequence (ESC /, for EBCDIC, or ESC Q, for USASCII). For double or triple spacing, or skipping to a channel of the

<u>Escape Sequence</u>		
<u>USASCII</u>	<u>EBCDIC, Transcode</u>	<u>Forms Motion After Printing</u>
ESC Q	ESC /	Single space
ESC R	ESC S	Double space
ESC S	ESC T	Triple space
ESC A	ESC A	Skip to ch. 1
ESC B	ESC B	Skip to ch. 2
ESC C	ESC C	Skip to ch. 3
ESC D	ESC D	Skip to ch. 4
ESC E	ESC E	Skip to ch. 5
ESC F	ESC F	Skip to ch. 6
ESC G	ESC G	Skip to ch. 7
ESC H	ESC H	Skip to ch. 8

(In EBCDIC, the ESC character is identical to the PRE (Prefix) character.)

Figure 33. Vertical Forms Control Escape Sequences

printer carriage control tape, you must begin the print line record with the appropriate escape sequence. See Example 2 of Figure 31 for examples of escape sequences for vertical forms control.

HORIZONTAL FORMAT CONTROL

The IBM 2780 may be equipped with the Printer Horizontal Formatting Control feature. This feature allows the printer of the 2780 to receive from the line instructions on arranging across the page the text contained in subsequent print line records it receives. To tell the 2780 the format desired, the user program at the central computer must send a format record each time the format is to be changed. The format record most recently received by the printer governs the horizontal format of all subsequent print line records. (Sending a format record is equivalent to setting the tab stops on a typewriter.)

Each format record must begin with the ESC HT sequence. The record contains other HT characters, and intervening SP (space) characters such that a HT appears at each position of the format record corresponding to the print line position where a stop is to be made (that is, a tab stop is to be set), with SP characters occupying all other format record positions. No characters other than HT and SP may appear in the format record. It is not necessary to make the format record as long as the print line; it may end at the last HT character. The final character of the format record is an ETB. See Figure 34 for an example.

Format records must always be sent in nontransparent mode.

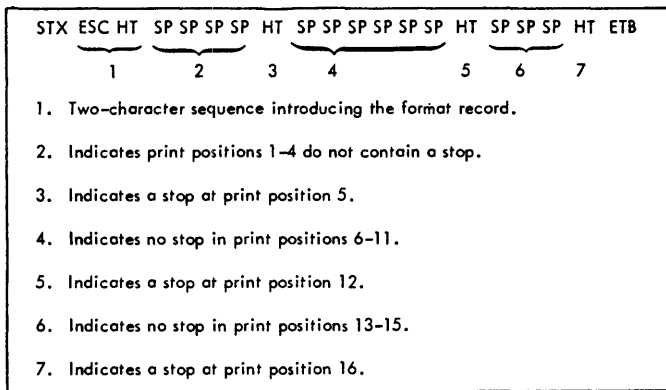


Figure 34. Example of a 2780 Format Record

The 2780 printer is governed by the last-received format record until:

- It receives a new format record.
- Power is removed from the terminal.
- A card is read by the card reader (EBCDIC and USASCII codes only)
- A record is received by the card punch (EBCDIC and USASCII codes only)

For terminals using Transcode, the printer retains the format record even if card reading or punching occurs following use of the printer.

Once a format record has been sent to the printer, HT characters are used in subsequent print line records the user program sends to the printer to cause skipping to the next stop position on the print line. This is equivalent to pressing the Tab key of a typewriter.

Example: Assuming that the printer is governed by the format record shown in Figure 34, if you wished to print the two characters A,B at print positions 1 and 2, and the three characters C,D,E at positions 12, 13, and 14, you would send this print line record:

```
STX A B HT HT C D E ETB
```

If HT characters appear in nontransparent text when no format record is in effect, or if it appears in the text beyond the format record position corresponding to the last stop on the print line, an error occurs--overrun of the print line.

If HT characters appear in transparent text, the 2780 does not recognize them as end-to-end control characters but treats them as text characters.

If the 2780 is not equipped with the Printer Horizontal Formatting Control feature, the 2780 does not recognize HT characters as end-to-end control characters regardless of whether they appear in transparent or nontransparent text.

MULTIPLE RECORD TRANSMISSION

The 2780 contains a 400-character buffer for receiving data from the communications line. This buffer can accommodate one, two, or more records depending on whether the terminal is equipped with the Multiple Record Transmission feature and whether transparent or nontransparent mode is used.

2780 without Multiple Record Transmission Feature

Without the feature, the 2780 can receive or send two nontransparent records per transmission. The first record is ended by the unit separator (US) character (also called interchange unit separator -- IUS -- in EBCDIC) which provides the ITB function, while the second record ends with ETB or ETX. Thus, in nontransparent mode, a single transmission appears on the line as:

```
STX ...text... US ... text... ETB (or ETX)
```

When received by the 2780, only the text is printed or punched. The US and ETB (or ETX) cause the punch to eject the card, but are not themselves punched into the card (unlike EM (end-of-medium), which is punched into the card).

The STX character is mandatory at the beginning of the first record, and optional at the beginning of the second (after US).

Upon receiving a two-record transmission, the 2780 checks each record individually. If the first record is invalid and the second is invalid, it prints or punches only the first record. If the central computer subsequently resends the two-record transmission, the 2780 prints or punches only the second record, as the first has already been successfully printed or punched. If in the initial transmission the first record is valid, the 2780 prints or punches neither record, even if the second record is valid. This avoids processing records out of sequence.

The 2780 can send two transparent records per transmission, in this format:

```
DLE STX ...text... DLE US DLE STX ... text... DLE ETB (or DLE ETX)
```

Mode	Maximum number of records per transmission	Maximum number of data characters ¹ per record
Nontransparent	2 without MRT 7 with MRT	(variable length record) 80 (to punch) 80 (from reader) print line ² (to printer)
Transparent	1 (to 2780) 2 (from 2780 w/o MRT) 4 (from 2780 with MRT)	(fixed length record) 80 (to punch) 80 (from reader) print line ² (to printer)

¹Not including control characters (STX, US, etc.) or escape sequences.
²The print line may be 80, 120, or 144 positions.

Figure 35. IBM 2780 Records: Number and Length

(The distinction between ETB and ETX is that ETX is used to end the last block in a transmission.)

Although the 2780 can send two-record transmissions in transparent mode, BTAM can send the 2780 only one record per transmission. The format is:

DLE STX ...text... DLE ETB (or DLE ETX)

2780 with Multiple Record Transmission Feature

With the feature, the 2780 can send or receive up to seven records per transmission, in nontransparent mode, and can send up to four records, in transparent mode. BTAM can send the 2780 only one record per transmission in transparent mode.

The maximum number of records per transmission, and the maximum number of data characters per record, are shown in Figure 35.

Although escape sequences cannot perform component selection in transparent mode, they may be used in transparent-mode transmissions to the printer to perform vertical forms control. They are optional in this case, however, and do not count as part of the fixed record length shown in Figure 35. The only case in which the escape sequence is mandatory in transparent mode is if the first data character in the record coincides with the ESC character.

PROGRAMMING CONSIDERATIONS

Maximum Transmission Length

The maximum number of characters in one transmission to the 2780 is 400. All characters in the transmission, including escape sequences, end-to-end control characters, US, ETB, and ETX, occupy positions

in the buffer, with one exception. STX characters are deleted by the 2780 control unit as they arrive and do not appear in the buffer. The limitation of 400 is in addition to other restrictions already mentioned, such as print or punch record length and number of records per transmission.

Automatic Turnaround Feature

If the 2780 is equipped with the Automatic Turnaround feature, the first blank card placed in the card reader hopper behind cards being read and transmitted will cause the 2780 to stop card reading and send an ETX to the central computer. (The blank card is not sent over the line). In this case the last message block from the 2780 will be:

STX ...text... US ETX

Nonmixing of Modes within a Transmission

Within one transmission you cannot send records in both transparent and nontransparent modes. That is, a transmission of the form

STX ...text... US DLE STX ...text... DLE ETB

is not permitted.

Responses by 2780 to Abnormal Conditions

There are two possible responses by the 2780 to abnormal conditions occurring when the 2780 is receiving a transmission from the central computer: NAK and EOT.

The 2780 sends a response of NAK for line errors. For example, the received message block may contain a parity error. The 2780 returns a NAK to request the central computer to retransmit. Another kind of line error may result in a

received message block containing too many characters. (A user-program error could produce the same result.) Again, the 2780 returns a NAK to request retransmission. Clearly, if the user program has actually sent too many records or characters in one transmission, the user program must be corrected; no amount of retransmission could clear the error condition.

The 2780 sends EOT in response to certain user-program errors that it can detect, such as failure to send a format record to the printer before sending print line records containing horizontal format control, or sending too many characters in a record.

The 2780 also sends EOT when it detects certain internal errors, such as a punch jam, printer forms check, buffer parity check, etc. When these conditions occur, the 2780 immediately responds to the central computer with EOT, signifying that the 2780 operator must intervene before operation can resume.

Recovery from such a disruption depends on the user-program restart and recovery procedures and on the 2780 operator procedures. The operator's instructions may, for example, tell him to prepare the 2780 to expect retransmission of the message block to which the 2780 responded with EOT. Or he may be instructed to prepare the 2780 to receive a new message block, unrelated to the previous one. In any event, the operator's response and the user program's response to these conditions should be consistent.

In certain instances the 2780 will not immediately respond with EOT when an internal error occurs. This happens only if the error condition occurs during the printing or punching of the last record in a received message block. Upon verifying that the last record in a block is free of error, the 2780 responds with the appropriate acknowledgment to the central computer

and overlaps processing of the last-received record with receipt of the next block. If an internal error occurs during processing of the last record, the 2780 responds to the next block with an EOT. The 2780 operator can intervene to process the record on which the error actually occurred without retransmission of that record. The user program would then retransmit the next message block.

A description of internal errors and operator procedures may be found in IBM 2780 Data Transmission Terminal Component Description.

In sending to the central computer, the 2780 makes special use of the ENQ character to indicate the occurrence of an internal error. If a buffer parity or overrun error occurred in the message block it is sending, the 2780 sends a message block of the form:

```
STX ...text...  US ....text...  ENQ
(or)
STX ...text...  ENQ
```

A message ending with ENQ is invalid, and BTAM error recovery procedures automatically respond to the 2780 with a NAK character. The 2780 then sends EOT to indicate that it is aborting transmission until the operator corrects the error condition.

If an internal error occurs during processing of a record being readied for transmission to the central computer, it sends STX ENQ (with no text). The central computer and 2780 then exchange NAK and EOT, as above.

When it receives the EOT, BTAM turns on bit 7 of the DECFLAGS field of the DECB and posts abnormal completion, with a code of X'41'. Subsequent transmission by the 2780, after communication with the central computer is resumed, normally begins with the record on which the error occurred.

OS/VS BTAM supports Binary Synchronous Communication between a System/370 and an IBM 2790 Data Communications System (via an IBM 2715 Transmission Control Unit) over point-to-point (switched and nonswitched) and multipoint nonswitched line configurations.

The 2715 is available in two models: local and remote. The local 2715 (2715 Model 1) is attached directly to the multiplexer channel of a System/370. A Binary Synchronous Module is provided in the 2715 Model 1 to simulate a Binary Synchronous Communications Adapter, operating on a point-to-point nonswitched line. The remote 2715 (2715 Model 2) can be connected to an IBM 2701 Data Adapter Unit attached to a System/370 multiplexer or selector channel, or to an IBM 2703 Transmission Control attached to a System/370 multiplexer channel. Communication between the System/370 and the 2715 (local and remote) follows the line control conventions of Binary Synchronous Communications. To the System/370 programmer, a local 2715 is indistinguishable from a remote 2715 (operating on a point-to-point nonswitched line).

For detailed descriptions of the 2790 system, see IBM 2790 Data Communication System Component Description.

The transmission code supported is transparent EBCDIC. When communicating with a System/370, the 2715 sends error information and diagnostic messages to the system. The type of message (error information or diagnostic) is indicated in a special header that precedes the message text and is transparent to the user.

- Diagnostic messages are operator awareness messages that are printed on the System/370 console or the 2740, if available. These messages are the result of the 2715 completing diagnostics.
- Error information messages are recorded on disk. These messages are sent when the 2715 has filled a sector of its integral disk with error information data or by user request.

Note: If one of these diagnostic or error information messages is received by BTAM on a READ Initial operation, BTAM changes the optype to READ Continue and restarts the operation. The initial flag bit in the DECB is left on in this case. If one of these messages is received on a multipoint

READ Initial operation, the input area may contain the index byte and EOT.

The READ, WRITE, and CONTROL options available for the 2715 are listed in Figure 19.

BTAM provides a set of macro instructions to enable the user to describe the processing that must be performed on his input. The 2715 cannot be programmed by the user. IBM-supplied microcode in the 2715 interprets processing requirements through a set of tables generated by the user-coded macros.

There are 18 user macros that can be used with the 2790 System. Seven of the macros are for the basic system, four are for the Pulse Count feature, and seven are for the 2798 Guidance Display Unit. The 18 macros, when assembled by the System/370 Assembler, generate 21 types of tables. The tables are transmitted in object form to the 2715 by the user's BTAM program. These tables contain pointers and index values, as well as parameters used by the microcoded routines. (See Appendix M for a sample 2790 program.)

2715 PULSE COUNT FEATURE

The 2790 Data Communications System is a data collection and data communication system. The 2715 Pulse Count feature is provided to allow the 2790 system user to dynamically control and monitor production work flow. This feature adds another major capability to the 2790 system.

Some of the highlights of the 2715 Pulse Count feature are:

- The 2793 Area Station is the only area station in the 2790 system on which pulse counters can be attached.
- Up to 63 counters are allowed on a 2793 area station.
- Up to 1008 counters are allowed on a 2790 system.
- These counters can have a decimal count from 0 to 29,999.
- The READ/WRITE capability is available for all counters.
- The overflow interrupt capability is available on all counters.

- Implicit/explicit counter addressing at the DEU level is permitted.
- Count testing can be performed on up to 504 counters in the system.
- Schedule readout capability is available for up to 504 counters in the system.

The 2715 Pulse Count feature has many possible uses. Some of the functions that can be performed with it are:

- Appending counts automatically to transactions entered on a 2795/2796/2797 Data Entry Unit.
- Monitoring the current progress of counters by requesting readouts of counters for printing at the 2740, the area station 1053 printer, or the System/370.
- Setting counters to predetermined counts, and when these counts are reached, automatic printouts of the counters are routed to the 2740, the area station 1053, or the System/370.
- Monitoring for unassigned production with printout notification at the 2740, the area station 1053, or the System/370.
- Monitoring counters on a scheduled basis and informing the user when a counter is not advancing. Printouts can be directed to the 2740, area station 1053, or the System/370 to alert the user of such "no-count" production conditions.
- Readout of counters on a scheduled basis for analysis by users.
- Scheduling up to 15 user-selected time schedules for flexibility in implementing the "count test" and "scheduled readout" functions described above.

Counter Testing

Two types of counter testing can be automatically performed by the 2715 if the user so desires. The user specifies one of 15 possible test schedules for each of 504 counters by means of the user table. He can also specify the type of count testing that is to be automatically performed on each of up to 504 counters. The 2715 will scan the counter table and perform one of two count tests, "no count" or "unassigned production," depending on what the user specifies in the tables and also whether or not count testing is enabled.

The user has the ability to enable or disable count testing from either the System/370 or the 2790 DEUs via user-specified transaction lists. When initiated from a DEU, the desired action must be specified in the transaction list (CTRLIST macro). If explicit counter addressing is specified, this address must be within the data entry. Implicit counter addressing at a DEU implies counters 1 through 32 only. All count test entries must be the last data entry from the DEU.

The user can enable or disable count testing from the System/370 or the 2715 operator's console for all counters on an area station, or all counters on the system. If the user disables count testing for all counters on the system, he can enable all counters and either have the 2715 continue from the previous stop point in the timing of the schedules, or have the 2715 reinitialize all schedules and start again.

Three possible count testing actions can be initiated on an individual counter basis:

1. No-count test can be started. The 2715 automatically stops unassigned production testing in this case.
2. Unassigned production test can be started. The 2715 automatically stops no-count test in this case.
3. All count testing can be stopped.

The no-count test informs the user that a counter is not advancing. The no-count test is executed on a user-defined schedule on an individual counter basis. When the 2715 detects that a counter is not advancing, a message signifying a no-count condition is generated and routed to the destination defined by the user in the ROUTE operand of the ASCTR macro instruction. The message indicates which counter has not advanced when it should have. The 2715 then disables further no-count testing for that counter until the user has corrected the situation and has enabled further testing. Count testing for all other counters remains in progress.

The unassigned production test informs the user that a counter is advancing when it should not. The unassigned production test is executed on a user-defined schedule on an individual counter basis. When the 2715 detects a counter advancing, a message signifying an unassigned production test condition is generated and routed to the destination designated by the user, in the ROUTE operand of the ASCTR macro instruction. This message indicates which counter has advanced when it should not. The 2715

then disables further unassigned production testing for that counter until the user has corrected the situation and has enabled further testing. Count testing for all other counters remains in progress.

Scheduled Counter Readout

An individual counter readout function is provided so that the 2715 will automatically read up to 504 counters on user-defined schedules if the user so desires. The user specifies one of 15 possible schedules for each of up to 504 counters in the table macros (CTRGROUP and CTRSCHED). As the 2715 reads a counter, the transaction header is attached and the transaction is stored automatically on disk as deferred data. The transaction header contains the counter address and associated time stamp.

The user can enable or disable scheduled readout from the System/370 or the 2715 operator's console for all counters on an area station or all counters on the system. He can also enable or disable scheduled readout for a single counter from the System/370. If the user disables scheduled readout for all counters on the system, he can enable scheduled readout on all counters and either have the 2715 continue from the previous stop point in the timing of the schedules or have the 2715 reinitialize all schedules and start again.

DEU Set Counter Capability

The initiating transaction from a DEU consists of one or more data entries. For multiple data entries, the set counter function is contained within the last data entry. There is no set counter capability on the 2795 DEU. Only the 2796 and 2797 DEUs have this capability. The user selects the proper transaction list by setting the top left knob on the 2796 or the left knob on the 2797. (Transaction expansion may not be used.) The last step in the transaction list must be the counter appendage. The counter control byte in the counter appendage step of the transaction list specifies the set counter operation and also specifies whether or not implicit or explicit counter addressing is to be used. For implicit addressing, the user does not have to specify the counter address. The DEU address is automatically translated to a counter address. The user specifies the high-order byte of the 5-digit value to which the counter is to be set by turning the top right knob on the 2796 or the right knob on the 2797 to the high-order digit of the value. This digit must be 0, 1, or 2, because the value cannot exceed 29,999. The user specifies the low-order four digits of the counter value in the four digit-rocker switches on the 2796

or by keying them in on the manual entry digit keys of the 2797.

Note: There are 6 possible manual entry positions on the 2797. Only the 4 low-order positions are used for the low-order value of the set counter operation.

Explicit addressing requires that the counter address be contained in the data entry. Explicit addressing may be accomplished by DEU manual entry or by manual entry and card or badge entry. If manual entry is used, the lower left and lower right knobs on the 2796 or the two high order positions of the 2797 manual entry field specify the counter address. If manual and card or badge entry are used, columns 77 and 78 of the card or columns 19 and 20 of the badge must contain the counter address.

The status of a DEU-initiated set counter operation is indicated via normal status reporting. Unsuccessful set counter transactions initiated by a DEU are signified by raising the red error flag on the DEU Set counter functions may not be routed to ASLOG printer.

DEU Read Counter Capability

The initiating transaction from the DEU consists of one or more data entries. For a transaction that contains multiple data entries, the last data entry must contain the information necessary to initiate a read counter. The 2795, 2796, and 2797 DEUs have the read counter capability. The user selects the proper transaction list by setting the left knob on the 2795 or 2797 or the top left knob on the 2796. The last step in the transaction list must be the counter appendage. The counter control byte in the counter appendage step of the transaction list specifies the read counter operation to be performed, and also specifies whether implicit or explicit counter addressing is to be used. Read counter values are appended to the normal transaction and routed according to the user-defined routing designation in the transaction list.

The read counter capability includes both implicit and explicit counter addressing for all read operations except Read Group and Read Group Residual, for which explicit counter addressing must be used. The following read operations may be defined in the last step of the transaction list:

- Read (single or group)
- Read Residual (single or group)

- Read and Reset (single counter)
 - Read and Set* (single counter)
- *For 2796 and 2797 only.

For implicit counter addressing, the user does not have to specify the counter address. The DEU address is automatically translated to the counter address. For explicit counter addressing, the user may manually set the lower-left and lower-right knobs on the 2796 to the counter address, or he may put the counter address in columns 77 and 78 of the card or columns 19 and 20 of the badge. For the 2797, the user may manually enter the two digits for the counter address in the manual entry digit keys (these two digits must be left justified), or he may put the counter address in columns 77 and 78 of the card or columns 19 and 20 of the badge. Explicit counter addressing for the 2795 may only be specified from columns 77 and 78 of the card or columns 19 and 20 of the badge. Manual entry is not possible on the 2795.

For explicit counter addressing on the Read Group or Read Group Residual, the user may manually set the lower-left and lower-right knobs on the 2796 to the starting counter address, and the first two digit-rocker switches to the ending counter address, or he may put the starting and ending counter addresses in columns 77-80 of a card or 19-22 of a badge. For the 2797, the user may manually enter the two digits for the starting counter address followed by the two digits for the ending counter address in the manual entry digit keys (these four digits must be left justified), or he may put the starting and ending counter addresses in columns 77-80 of a card or columns 19-22 of a badge. For the 2795, the user must specify the starting and ending counter addresses in columns 77-80 of a card or columns 19-22 of a badge. For a Read and Set operation, the user specifies the set counter value in the same way as for the set counter operation (transaction expansion may not be used), with the top right knob and the four digit-rocker switches on the 2796, or the right knob and the four low-order digits from the manual entry digit keys on the 2797.

The counter transaction and count value may be logged at any area station for 1053 display by message routing, but the counter transaction must consist of only one step in addition to the counter appendage. Any of the read operations (except Read Group and Read Group Residual) may be routed to an area station 1053 printer by implicitly or explicitly specifying the output destination. Read and Set operations (READSET) may not be routed to the ASLOG printer. The implicit routing address is

in the transaction list, while the explicit routing address is from the card or badge entry. Implicit message routing is done by using both the left and right knobs on the 2795 and 2797 or the upper left and right knobs on the 2796 to address the transaction list. For explicit message routing to an area station 1053, the user must specify the 1053 address in columns 71 and 72 of the card or columns 13 and 14 of the badge. This applies to any DEU.

Note: Transaction expansion is a prerequisite to message routing. Storage expansion (32K) is a prerequisite to transaction expansion.

Setting a counter to a certain value implies that the user wants to know when the counter reaches that value. The set counter function sets a counter to a value of 29,999 minus the value specified, so that when the user-specified count is reached, the user is alerted to an overflow interrupt for that counter. The read counter function allows the actual value of the counter to be read. The read residual counter function allows the value of the counter to be subtracted from 29,999, so that the residual difference is read.

For example, if a user sets a counter to a value of 10 and immediately performs a read counter function, the value read will be 29,989. If he immediately performs a read residual function on the counter, the value read will be 10.

2715 TABLES

There are 18 user macros that, when assembled, generate 21 types of tables to be transmitted in object form to the 2715. The types of tables are:

- Table Definition Block.
- Area Station Table.
- Data Entry Unit Table.
- Transaction Group Tables.
- Transaction List Tables.
- Area Station Sequence Table.
- Area Station Counter Table.
- Counter Table.
- Schedule Table.
- Data Entry Unit Sequence Table.
- Data Entry Unit Index Table.

- System Parameter Table.
- Transaction Table.
- GDU List Table.
- Parameter List Number Table.
- Parameter List Table.
- Display Guidance Table.
- GDU Area Station Table.
- GDU Sequence Table.
- Identification Table.
- Translate Table.

Descriptions of these tables and the macros that generate them follow.

Note: The user must assemble all his macros at the same time since the relationship among the tables is established by labels.

Table Definition Block: The Table Definition Block contains a pointer to each of the other tables. It is defined by the CONFIGUR macro instruction.

Area Station Table (AS Table): The Area Station Table contains one entry per area station. Each entry is one byte and contains a numeric pointer that relates the specified area station to a particular transaction group within the Transaction Group Table. The maximum size of the AS Table is 100 bytes for a 2715 having 32K bytes of storage and 64 bytes for a 2715 having 16K bytes of storage. The AS Table is defined by the AS macro instruction.

Data Entry Unit Table (DEU Table): The Data Entry Unit Table contains one entry for each area station defined in the system. This entry (0-99) is used for all data entry units attached to the designated area station. If there are no attached data entry units, the entry contains a value to indicate this condition. The DEU table is defined by the AS macro instruction.

The position of the entry in the table is relative to the position of the area station address within the valid range of addresses. For example, the first entry in the DEU Table is for the data entry units attached to the area station with ID=0; the second for those attached to the area station with ID=1; etc.

Each entry in the DEU Table is one byte and is used to gain access to the transaction group associated with all the data

entry units attached to the area station. This indicates that all data entry units attached to an area station must use a common transaction group.

Transaction Group Table (TGROUP Table):

Each transaction group consists of nine halfword (two-byte) entries that contain pointers to a transaction list or to another transaction group. Each entry corresponds to a transaction code (a transaction key on an area station or the value of the left rotary knob on a 2795 or 2797 or the top left rotary knob on a 2796 Data Entry Unit). Each entry contains a pointer to a transaction list that defines the operating procedure associated with the specified transaction code. If nine transaction lists are not sufficient, an indication can be set in one or more of the transaction group entries to permit a transaction expansion function in which a secondary value (the first digit of input from an area station or the value of the right hand rotary knob on a data entry unit) is used to index another transaction group. Therefore, it is possible for an area station or data entry unit to refer to nine TGROUP entries, any or all of which may indicate secondary indexing. This allows a data entry unit to perform a maximum of 81 distinct transaction functions, while allowing 81 functions for area stations (see TGROUP in the Macro Descriptions section).

All area stations that have the same operating characteristics must refer to the same transaction group, using the area station address and the corresponding entry in the AS Table. The same is true for data entry units, using the area station address and the corresponding entry in the DEU Table. There can be up to 63 transaction groups, each of which uniquely specifies an area station or data entry unit capability. The transaction groups are defined by TGROUP macro instructions.

Transaction List Tables (TRLIST Tables):

Each Transaction List Table consists of a three-byte identification and routing header field and either an internal message or from one to sixteen data entry steps. The header field determines the destination of the completed transaction. Each data entry step is generated by an ASLIST or DEULIST macro and determines whether checking is to be performed on the input. If an ASLIST macro generated a data entry step, the step contains the number of the next guidance light to be turned on (more than one guidance light number is included if the user chooses to include error checking in his transaction step, for example, via LENGTH and DIGIT operands of the ASLIST macro).

A TRLIST Table is defined in any one of three ways:

- A TRLIST macro followed by one or more ASLIST macro instructions,
- A TRLIST macro followed by one or more DEULIST macro instructions,
- A TRLIST macro followed by one or more ASLIST (DEULIST) macro instructions with specification for message routing with an internal message.

The first Transaction List Table always refers to all IBM 1035 Badge Readers, if there are any on the system.

Area Station Sequence Table (AS-SEQ Table):

A transaction from an area station may comprise a discrete number of processing steps (for example, badge, card, card). The AS-SEQ Table keeps track of the last step of the transaction entered from each area station. The AS-SEQ Table has one entry per area station. Each entry is one byte and contains the step number (0-15). The maximum size of the AS-SEQ Table is 100 bytes, one byte for each of the 100 possible area stations. The AS-SEQ Table is defined by the AS macro instruction.

Area Station Counter Table (ASCTR Table):

The Area Station Counter Table contains one entry per area station. Each entry is two bytes and contains a displacement to the group of counters in the Counter Table for that particular area station. Each entry also contains routing information for counter overflow and count test response messages. All counters attached to an area station have counter overflow and count test response messages routed to the same destination.

Each entry in the Area Station Counter Table is used to gain access to the counters in the Counter Table associated with this area station. The displacement in each entry, plus the counter address, allows the 2715 to index to individual counters. Scheduled readout and count testing are performed at the individual counter level.

The maximum size of the Area Station Counter Table is 202 bytes: 2 bytes for each of 100 area stations, plus 2 additional control bytes. This table is truncated at the highest assigned area station. Unassigned area stations below the highest assigned area station require 2 bytes of unused main storage. However, the highest assigned area station with counters can be any assigned area station less than or equal to the highest assigned area station defined by the AS macro operand ID=n. The Area Station Counter Table is defined by the ASCTR macro instruction.

Counter Table (CTR Table): The Counter Table consists of 2 bytes of control information for each counter in the system that requires testing. The table is organized on a group basis. Each area station that has one or more counters (to be tested) constitutes a group. Each group is truncated at the highest counter address. Counters for which scheduled readout and count testing are not to be done require 2 bytes each in the counter table if their addresses are less than the highest counter address. Only 504 counters can be defined in the Counter Table. The Counter Table is defined by CTGROUP macro instructions. Each CTGROUP macro defines a single counter.

Schedule Table: The Schedule Table consists of 2 bytes for each user-defined count test schedule or readout schedule. The maximum number of schedules that can be defined is 15, so the maximum size of the Schedule Table is 30 bytes. These schedules determine the frequency (in minutes) with which the Counter Table entry will be tested. Scheduled readout and count test operations can refer to any of the 15 schedules that are defined. The maximum frequency that can be specified is 2047.

Data Entry Unit Sequence Table (DEU-SEQ Table): A transaction from a data entry unit may comprise a discrete number of steps. The DEU-SEQ Table keeps track of the last step entered from each data entry unit on the system. Each one-byte entry contains the step number (0-15) of the transaction initiated by each data entry unit. The maximum size of the DEU-SEQ Table is 1,024 bytes (one byte for each of the 1,024 data entry units available on the 2790 system). The DEU-SEQ Table is defined by the AS macro instruction.

Data Entry Unit Index Table (DEU-INDEX Table): When a data entry is received from a data entry unit, the 2715 receives the area station address and the data entry unit address. The area station address is used as an index to the desired entry in the DEU-INDEX Table. (There is one entry per area station.) The one-byte entry in the DEU-INDEX Table contains a pointer to the desired entry in the DEU-SEQ Table. However, since the DEU-SEQ Table can be up to 1,024 bytes long, two additional bits are necessary. Using bits 6 and 7 of the corresponding entry in the DEU Table extends the addressing capability of the DEU-INDEX Table entry to the full 1,024 possible entries of the DEU-SEQ Table. Use of the data entry unit address allows indexing to the specific counter associated with the data entry unit sending the transaction. The DEU-INDEX Table is defined by the AS macro instruction.

System Parameter Table: The System Parameter Table has a fixed core location and maintains an index of other 2798 GDU table locations. The System Parameter Table contains the following information:

System error guidance:

- Invalid function error guidance
- Premature termination error guidance
- Monitor key error guidance

Table pointers:

- GDU Sequence Table pointer
- Transaction Table pointer
- Parameter List Number Table pointer
- Identification Table pointer
- Translate Table pointer
- GDU AS Table pointer

Identification characters:

- "Get" identification character
- "Store" identification character

Counts:

- GDU list count
- Identifier count

The System Parameter Table is generated by the CONFIGUR macro. The information in the table is determined from the CONFIGUR macro operands. There is only one System Parameter Table per 2790 System with 2798 GDUs.

Transaction Table: The Transaction Table is used to index the desired GDU list from a GDU operator entered transaction code. Each entry in the Transaction Table may contain a pointer to a GDU list associated with a transaction code. There are 100 possible transaction codes so there are 100 possible entries in the Transaction Table. Each entry in the table is 2 bytes long for a maximum table size of 200 bytes. The table is truncated at the highest assigned transaction code. But for each entry skipped between zero and the highest assigned value two bytes of core are reserved, just as if this value had been assigned. Each entry in the Transaction Table is generated by a GDUTRANS macro.

GDU List Table: The GDU List Table contains up to 100 GDU lists. Each GDU list entry will contain a transaction header, a variable number of GDU steps (up to 16), an all zero byte, and implicit text (if specified). Each GDU list entry is generated by a combination of the TRLIST macro and from 1 to 16 GDULIST macros. The TRLIST macro generates the transaction header. Each GDULIST macro generates a 5-byte GDU step with the following information:

- A one-byte parameter list number.
- Two bytes of normal guidance to be sent to the operator guidance panel on the

GDU. This guidance is used to light a combination of 16 lights.

- A two-byte display guidance pointer used to:
 - a. Point to a display guidance message in the Display Guidance Table, or
 - b. Point to an identifier in the Identifier Table.

Implicit text is defined in the last GDU-LIST macro in the GDU list entry. Each GDU list entry can contain a minimum of 10 bytes and a maximum of 85 bytes plus implicit text.

Parameter List Number Table: The Parameter List Number Table contains up to 127 addresses of the parameter lists. Each entry in the table is two bytes for a maximum table size of 254 bytes. This table is generated by PARAMNUM macros, each of which generates a two-byte entry. The table is truncated at the highest defined parameter list number defined by the PLN operand of the PARAMNUM macro.

Parameter List Table: The Parameter List Table contains up to 127 entries. The data in each parameter list entry defines the types of checks that are performed on a data entry. The first two bytes of a parameter list contain a check field and a function field in which the checks and/or functions associated with this list are denoted. Following these two bytes are the check lists if any are required. These check lists provide the test information and error guidance for the checks performed on a data entry. The check lists are variable lengths, depending on the tests to be done. The parameter lists are packed decimal. Each parameter list is generated by a PARMLIST macro.

Display Guidance Table: The Display Guidance Table contains the various messages used as display guidance for the GDUs on the loop. Each entry in the Display Guidance Table contains a length byte and from 1 to 16 data bytes. Each entry in the table is generated by a DISPGUID macro.

Guidance Display Unit Area Station Table (GDUAS Table): The Guidance Display Unit Area Station Table, in conjunction with the GDU device address, contains pointers which provide entries into the GDU Sequence Table. Each entry is two bytes and is generated by the GDUAS macro. The maximum size of the GDUAS table is 200 bytes. This table is truncated at the highest assigned area station. Unassigned area stations below the highest assigned area station require two bytes of unused main storage.

Guidance Display Unit Sequence Table: The Guidance Display Unit Sequence Table consists of one byte of zeros for each GDU on the 2790 system. Each byte is used for inquiry display and GDU sequence. This table is generated by the GDUTRANS macro.

Identification Table: The Identification Table contains the areas to maintain the GDU identifiers. Fourteen bytes are generated for each identifier. The total number of identifiers is specified in the System Parameter Table. This table is generated by the STEND macro.

Translate Table: The Translate Table is used in conjunction with the translate function and contains a maximum of eight entries. Each entry contains a translate character, the length of the text into which it is to be translated, and from 1 to 14 bytes of text. One byte of zeros follows the last entry in the table. The maximum table size is 129 bytes. Each entry in this table is generated by a TRANSLAT macro.

Loading the Tables

The assembly output of the user macros is a set of tables, in object form, needed for the 2715 internal operation. The user must write a BTAM routine to load these tables into the 2715. During transmission of the object text from the System/370 to the 2715, the user must include two headers before each card's data: a message header and a transaction header (Figure 39). The user must provide DLE STX in front of the message (see format below). BTAM provides the ending characters. All cards must be transmitted, one at a time, to the 2715.

D	S	Message	Transaction	Object Card
L	T	Header	Header	
E	X	2 bytes	8 bytes	80 bytes

When loading the tables, the user must first be sure that the 2790 system is inactive. This is accomplished by using the defined control transactions to "stop 2790 input" and to purge the disk of all deferred data ("Read deferred data"). When activity at the 2715 has ceased, the user must initially send a "table load start" control transaction. This is followed by the transmission of the table object cards as "table load data" control transactions (see the Message Format section).

Columns 73-80 of the object deck contain a program identification and a sequence number, which are checked by the 2715. The program identification (columns 73-76) is determined from a named TITLE card

generated by the CONFIGUR macro instruction. Both the identification and the sequence are checked by microcoded routines in the 2715. The completion of the data load is signaled by the END card (END in columns 2-4). After transmitting the END card, the user must send a "table load end" control transaction. If an error is found in either the program identification or the sequence field of any card, the table load is rejected.

When transmitting the tables, the maximum message length is 128 bytes. When the last message of the table load has been sent, the user should transmit an EOT. The 2715 bids for the line and then transmits a message indicating the status of the table load (see the Message Format section). The table load is rejected if any of the following conditions occurs:

1. Invalid program identification sequence field.
2. Improper control field in transaction header.
3. 2715 table size exceeds storage available.
4. Data has not been purged from the 2715 integral disk.
5. The system is active.

MACRO INSTRUCTIONS

The macros coded for the 2790 System must be in the following order:

- CONFIGUR
- AS
- GDUAS (optional)
- TGROUP
- ASCTR (optional)
- CTGROUP (optional)
- CTRSCHEd (optional)
- GDUTRANS (optional)
- PARAMNUM (optional)
- PARMLIST (optional)
- DISPGUID (optional)
- TRANSLAT (optional)
- TRLIST

- ASLIST
- DEULIST (optional)
- GDULIST (optional)
- CTRLIST (optional)
- STEND

Configuration Macro (CONFIGUR): The Configuration macro generates the table definition block that contains pointers to the other user tables.

Area Station Definition Macro (AS): The Area Station Definition macro permits building an exhaustive list of all area stations present in the system. In addition, each macro logically attaches area stations and data entry units to their associated transaction groups.

Guidance Display Unit Area Station Macro (GDUAS): The GDUAS macro is used to build an entry in the GDU Area Station Table.

Transaction Group Macro (TGROUP): By coding the Transaction Group macro, the user establishes a pointer to a set of transaction lists that can be associated with the transaction keys of a group of area stations, with the left-hand knob positions of a group of 2795 or 2797 Data Entry Units, or with the top left-hand knob positions of a group of 2796 Data Entry Units.

With each of the nine transaction keys on an area station, the operator can select up to nine transaction lists. With each position of the left-hand knob on a 2795 or 2797 or of the top left-hand knob on the 2796 Data Entry Unit, the operator can also select up to nine transaction lists (position 0 is reserved). Normally, each transaction key or position of the left-hand knob is associated with only one transaction list. Since groups of area stations and groups of data entry units usually have the same operating procedure, a given set of area stations must have corresponding transaction keys associated with identical transaction lists.

Area Station Counter Macro (ASCTR): By coding the ASCTR macro, the user defines each area station that has pulse counters, and establishes a displacement to the counter group in the Counter Table associated with each particular area station. In addition, the user defines routing information for counter overflow and count test response messages for all the counters on each area station.

Counter Group Macro (CTRGROUP): By coding the CTRGROUP macro, the user can define two bytes of control information for each coun-

ter on an area station for which scheduled readout or count testing is to be done.

Counter Schedule Macro (CTRSCHED): The CTRSCHEd macro defines the count test schedules and the readout schedules that can be used by all the counters on the system.

Guidance Display Unit Transaction Macro (GDUTRANS): The GDUTRANS macro is used to build an entry in the Transaction Table. A GDUTRANS macro must be coded for every transaction code that will be used in communication with the 2798 GDUs.

Parameter List Number Macro (PARAMNUM): The PARAMNUM macro is used to define an entry in the Parameter List Number Table for use with 2798 GDUs.

Parameter List Macro (PARMLIST): The PARMLIST macro is used to generate an entry in the Parameter List Table. The data in this entry defines the types of checks that are performed on a data entry from a 2798 GDU.

Display Guidance Macro (DISPGUID): The DISPGUID macro is used to define a display guidance message in the Display Guidance Table.

Translate Table Macro (TRANSLAT): The TRANSLAT macro is used to build an entry in the Translate Table.

Transaction List Macro (TRLIST): The Transaction List macro is used with the Area Station List macro or the Data Entry Unit List macro to define a transaction, or it is used to define a user-specified message. The TRLIST macro enables the user to control the destination of the completed transaction or a predefined message. Each transaction list has a DEU or AS transaction code associated with it.

Area Station List Macro (ASLIST): When the transaction is to be initiated by an area station, the Transaction List macro is followed by one or more Area Station List macros. The ASLIST macro is used to define one step of a transaction for a 2791 Area Station; the transaction code is the value of the transaction key pressed by the operator.

Data Entry Unit List Macro (DEULIST): When the transaction is to be initiated by a data entry unit, the Transaction List macro is followed by one or more Data Entry Unit List macros. The DEULIST macro is used to define one step of a transaction for a data entry unit; the transaction code is the value of the left-hand rotary knob of a 2795 or 2797 and of the top left-hand rotary knob of a 2796.

Guidance Display Unit List Macro (GDULIST):
The GDULIST macro is used to define one step of a GDU transaction list for a 2791 or 2793 Area Station with 2798 GDUs attached.

Counter List Macro (CTRLIST): When the transaction is to be initiated by a data entry unit, the TRLIST macro is followed by one or more DEULIST macros, and, optionally, by a CTRLIST macro instruction. The CTRLIST macro defines the last step of a transaction for a data entry unit that is attached to a 2793 Area Station using pulse counters.

Statement End Macro (STEND): The Statement End macro indicates the end of all user macros.

Macro Descriptions

The macros are arranged in the following section in the same order as they must appear in the assembly.

The macros must be assembled together. The first assembly statement must be the CONFIGUR macro (there must not be a TITLE, CSECT, or START card). The last assembly statement must be the END card. There must not be any other macros or code inserted into the assembly of the user tables.

CONFIGUR (Configuration) Macro Instruction

The CONFIGUR macro is used to generate the table definition block, which contains pointers to the 2715 tables. The format of the CONFIGUR macro is:

Name	Operation	Operand
[symbol]	CONFIGUR	{16 } {CORE={32 } } [, PC={NO } {YES }] {NO } {GDU={YES }] {, FUNCERR=(absexp, ...) } {, ENDERR=(absexp, ...) } {, MONERR=(absexp, ...) } {, GETID=absexp } {, STORID=absexp } {, IDCOUNT=absexp } {NO } {, INQDISP={YES }]

symbol
the name of the macro is optional.

CORE
the CORE parameter specifies the 2715 storage size. The only valid values

are 16 and 32. If an invalid value is specified, an MNOTE is issued and no code is generated. If the CORE parameter is omitted, 16 is assumed.

If CORE=16 is coded, then the size of all tables built must not exceed 1,280 bytes. If CORE=32 is coded, then the size of all tables built must not exceed 4,096 bytes. The size of all tables built is calculated by using Figure 36. The size will be the total of all macros used.

PC

the PC operand indicates whether pulse count macros are coded in this assembly. If PC=YES is specified, pointers to the Area Station Counter Table, the Counter Table, and the Schedule Table are included in the Table Definition Block. Coding PC=YES adds six bytes to the Table Definition Block. If PC=NO is coded, the pointers to the Area Station Counter Table, the Counter Table, and the Schedule Table are omitted, but two bytes of zeros are added. If the PC operand is omitted, PC=NO is assumed.

The PC operand is valid only if CORE=32 is specified. If PC=YES is coded and CORE=16 (or the CORE operand is omitted), an MNOTE is issued and no code is generated.

GDU

indicates whether 2798 Guidance Display Units are used on the 2715. If this operand is omitted, GDU=NO is assumed. If GDU=YES is coded, CORE=32 must also be specified.

FUNCERR

this operand indicates the error guidance that will be returned to the 2798 GDU when the following invalid functions are recognized:

- An invalid length is specified on a Get Identifier function. Normally, the GDU operator keys the 'Get ID' character, the two digit identifier address, and the ENTER key upon entering a GDU step having a get condition in its function field. An invalid length occurs when more than two characters are entered for the identifier address following the 'Get ID' character.
- An invalid length on a Translate function. This error occurs when a GDU step is entered that has a translate function associated with it and more than one keyed character is entered.

Macro	Min. Bytes Used	Additional Considerations
CONFIGUR	22	Add 4 if PC=YES. Add 22 if GDU=YES. Add 14 times the number coded if IDCOUNT=n.
AS	4	Add 4 for each skipped ID in sequence. Add 1 for each DEU attached.
GDUAS	1	Add 1 for each skipped ID in sequence.
TGROUP	18	
ASCTR	2	Add 2 for each skipped ID in sequence. Add 2 after last ASCTR macro.
CTRGROUP	2	Add 2 for each skipped counter in sequence.
CTRSCHED	2	For each schedule.
GDUTRANS	2	Add 2 for each skipped TRCODE operand.
PARMLIST	2	Add 3 for CKLNTH= Add 3 for CKMOD11= Add 3 for CKMOD10= Add 8 for CKOR= Add 3 for CKNONUM= Add 3 for CKNUM= Add 5 for CKRANGE= plus 1 for each position in CKRANGE field. Add 3 for CKAND= plus 1 for each position in CKAND field.
DISPGUID	1	Add 1 for each text character.
TRANSLAT	16	
TRLIST	5	
ASLIST	5	Add 1 for each implicit text character.
DEULIST	5	Add 1 for each implicit text character.
GDULIST	5	Add 1 for each implicit text character.
CTRLIST	5	Add 1 for each implicit text character.
STEND	0	Add 1 for each area station in system if INQDISP=YES in CONFIGUR macro.

Figure 36. 2715 Macro Storage Site Estimates

- An invalid address on a Get Identifier or Store Identifier function. This error occurs when the two digit identifier address is not in the Identifier Table, or the two digit identifier address has not been specified.
- A non-translatable character is specified on a Translate function. This error occurs when the character to be translated is not found in the Translate Table.
- A non-numeric character is recognized during a range check.

The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lights on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 37.)

ENDERR

this operand indicates the error guidance that will be returned to the 2798 GDU when a premature termination occurs. A premature termination occurs when the number of characters received in a data entry from a 2798 is not sufficient to complete all of the checks specified by this GDULIST macro (with exception of the CKLENGTH macro). The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lights on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 37.)

MONERR

this operand indicates the error guidance that will be returned to the 2798 GDU when a Monitor key check error occurs. The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lights on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 37.)

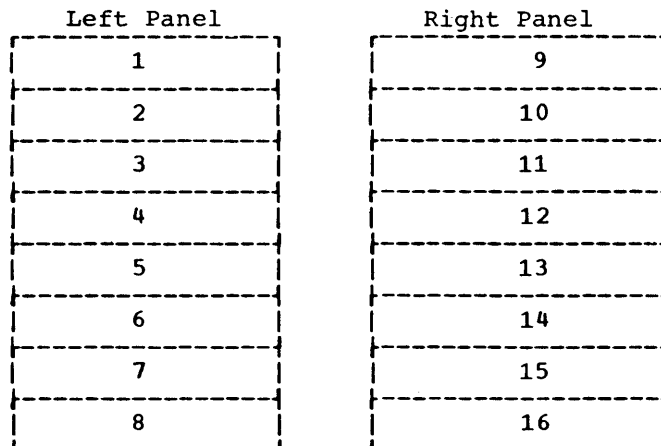


Figure 37. 2798 GDU Guidance Panels

Keyboard Character	Hexa-decimal Equivalent	Keyboard Character	Hexa-decimal Equivalent
A	C1	2	F2
B	C2	3	F3
C	C3	4	F4
D	C4	5	F5
E	C5	6	F6
F	C6	7	F7
G	C7	8	F8
H	C8	9	F9
I	C9	TAB	05
J	D1	NEWLINE	15
K	D2	LINEFEED	25
L	D3	SPACE	40
M	D4	.	4B
N	D5	+	4E
O	D6	&	50
P	D7	!	5A
Q	D8	\$	5B
R	D9	*	5C
S	E2	;	5E
T	E3	-	60
U	E4	/	61
V	E5	,	6B
W	E6	?	6F
X	E7	:	7A
Y	E8	#	7B
Z	E9	@	7C
0	F0	=	7E
1	F1	"	7F

Figure 38. 2798 GDU Keyboard Character Conversion

GETID

this operand specifies the GET Identification character that is used for the GET Identifier function. The value of this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: \$ & - / , # @ " : ? ! ; * + TAB NEWLINE LINEFEED SPACE. (See Figure 38.)

STORID

this operand specifies the Store Identification character that is used for the Store Identifier function. The value of this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: . \$ & - / , # @ " = : ? ! ; * + TAB NEWLINE SPACE LINEFEED. (See Figure 38.)

IDCOUNT

this operand specifies the number of identifiers that will be used. The value of this operand may be from 0 to 100. This operand must be coded if GDU=YES. Every time the 2715 is ICPLed, the predefined text 'NOT USED' will be defined in every identifier in the identifier table. The user should

use the Store Identifier function prior to using the Get Identifier function after an ICPL if he expects useful information to be in the identifier table.

INQDISP

this operand indicates whether Inquiry Display will be used on the 2715. If this operand is omitted, INQDISP=NO is assumed. If INQDISP=YES is coded, CORE=32 must also be coded.

AS (Area Station) Macro Instruction

The AS macro is used to build an entry in the Area Station Table and the Data Entry Unit Table, and a corresponding entry in the Data Entry Unit Index Table. In each table, the position of the entry to be built relative to the beginning of the table is determined from the ID parameter. Each AS macro requires (4 + absexp) bytes of 2715 storage (absexp is the value of the DEGROUOP operand). The format of the AS macro is:

Name	Operation	Operand
[symbol]	AS	ID=absexp [,ASGROUP=symbol] [,DEGROUOP=(symbol, absexp)]

symbol

the name field is optional.

ID

each area station is assigned a specific address (X'80' - X'E3') at system installation time, and the value of "absexp" is the decimal representation of that address (see Figure 39). ID is used to determine the position of an entry in the AS Table, DEU Table, and DEU-INDEX table. The value of the ID parameter must be from 0 to 99 when CORE=32 in the CONFIGUR macro; however, all values in this range need not be specified. The value of the ID parameter must be from 0 to 63 when CORE=16 or when the CORE operand is omitted in the CONFIGUR macro; however, all values in this range need not be specified. If one is omitted, a warning message is generated. The values 100 through 128, which would generate hexadecimal values E4 through FF, are not valid. The AS macros must be in ascending sequence by ID. An AS macro found to be out of sequence or in error terminates the assembly of this macro instruction. The ID of a macro in error is subsequently handled the same as an omitted ID.

Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)	Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)	Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)
0	80	33	A1	66	C2
1	81	34	A2	67	C3
2	82	35	A3	68	C4
3	83	36	A4	69	C5
4	84	37	A5	70	C6
5	85	38	A6	71	C7
8	86	39	A7	72	C8
7	87	40	A8	73	C9
8	88	41	A9	74	CA
9	89	42	AA	75	CB
10	8A	43	AB	76	CC
11	8B	44	AC	77	CD
12	8C	45	AD	78	CE
13	8D	46	AE	79	CF
14	8E	47	AF	80	D0
15	8F	48	B0	81	D1
16	90	49	B1	82	D2
17	91	50	B2	83	D3
18	92	51	B3	84	D4
19	93	52	B4	85	D5
20	94	53	B5	86	D6
21	95	54	B6	87	D7
22	96	55	B7	88	D8
23	97	56	B8	89	D9
24	98	57	B9	90	DA
25	99	58	BA	91	DB
26	9A	59	BB	92	DC
27	9B	60	BC	93	DD
28	9C	61	BD	94	DE
29	9D	62	BE	95	DF
30	9E	63	BF	96	E0
31	9F	64	C0	97	E1
32	A0	65	C1	98	E2
				99	E3

Figure 39. AS or GDUAS Macro Id Parameter Decimal and Hexadecimal Equivalents

ASGROUP

the ASGROUP parameter is valid for the 2791 only. The value of "symbol" is the name of the transaction group for this area station and must appear in the name field of a TGROUP macro. If the name does not appear, an assembly error occurs. The ASGROUP parameter builds an entry in the AS table.

data entry units. That is, "symbol" must be the same as the name of the transaction group for the 2795s connected to this area station, or of the dummy transaction group that precedes the transaction group for the 2796s connected to this area station (see Figure 40 for examples).

DEGROUP

symbol

the name of the transaction group with which the data entry units on this area station are associated is specified by "symbol". It must appear in the name field of a TGROUP macro; if not, an assembly error occurs. "symbol" builds an entry in the DEU Table. "symbol" must be identical to the name of the first (or only) TGROUP macro that defines a transaction group for

absexp

the value of the absolute expression is the number of data entry units attached to this area station. This number must not exceed 32, since this is the maximum number of data entry units that can be attached to any one area station. The value is used to build an entry in the DEU-INDEX table.

The DEGROUP operand may be omitted if no data entry units are attached to this area station. The entries in the DEU table and

Name	Operation	Operands
* DEFINING TRANSACTION GROUPS FOR BOTH 2795 AND 2796 DATA ENTRY UNITS		
	AS	ID=59,DEGROUP=(GROUP1,5) (TOTAL OF 5 DEU'S)
GROUP1	TGROUP	TC1=ATTENDNC,TC2=SETUP,TC3=PRODN,... (2795'S)
GROUP2	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2796'S)
* DEFINING TRANSACTION GROUP TABLE FOR 2796 AND DUMMY TRANSACTION GROUP TABLE FOR 2795		
	AS	ID=59,DEGROUP=(GROUP1,2) (TOTAL OF 2 2796'S)
GROUP1	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2795'S)
GROUP3	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2796'S)
* DEFINING TRANSACTION GROUP TABLE FOR 2795 ONLY		
	AS	ID=59,DEGROUP=(GROUP1,3) (TOTAL OF 3 2795'S)
GROUP1	TGROUP	TC=1ATTENDNC,TC2=SETUP,TC3=PRODN,... (2795'S)
* DEFINING TRANSACTION GROUP FOR 2797 AND DUMMY TRANSACTION GROUP FOR 2795 AND 2796		
	AS	ID=59,DEGROUP=(GROUP1,3) (TOTAL OF 3 2797'S)
GROUP1	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2795'S)
GROUP2	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2796'S)
GROUP3	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2797'S)

Figure 40. Examples of Defining Transaction Group Tables for Data Entry Units

the DEU-INDEX table corresponding to this area station are then defined with a value indicating there are no DEUs. The DEGROUP parameter must be coded for the 2793.

GDUAS (Guidance Display Unit Area Station) Macro Instruction

The GDU Area Station macro instruction is used to build an entry in the GDU Area Station Table. The position of the entry to be built relative to the beginning of the table is determined from the ID operand. Each GDUAS macro requires 2 bytes of 2715 storage. The format of the GDUAS macro is:

Name	Operation	Operand
{symbol}	GDUAS	ID=absexp, GDUNUMB=absexp

symbol

the name field of this macro is optional.

ID

each area station with attached 2798 GDUs is assigned a specific address (X'80' - X'E3') at system installation time, and the value of 'absexp' is the decimal representation of that address (see Figure 38). ID is used to determine the position of an entry in the GDUAS Table. The value of the ID operand must be from 0 to 99. The GDUAS macros must be in ascending

sequence by ID. A GDUAS macro found to be out of sequence or in error terminates the assembly of this macro instruction.

GDUNUMB

this operand specifies the number of GDUs on this area station. The value of this operand can be from 1 to 16 for a 2793 and from 1 to 12 for a 2791 Model 3.

Note: The GDUAS macro should only be coded for an Area Station with 2798 GDUs attached. The GDUAS macro must be coded once for every area station with 2798 GDUs attached.

TGROUP (Transaction Group) Macro Instruction

The TGROUP (Transaction Group) macro defines entries in a TGROUP Table. Each keyword operand associates a transaction list with a transaction code. The maximum number of TGROUP macros allowed is 63. The TGROUP macro instructions must follow the last AS macro instruction coded. A macro sequence error occurs if they do not. Each TGROUP macro requires 18 bytes of 2715 storage.

The three types of data entry units (2795, 2796, and 2797) require separate transaction groups, one for all 2795s connected to an area station, followed by one for all 2796s connected to the same area station, and immediately followed by one

for all 2797s connected to the same area station.

If all three types of DEUs are connected to the same area station, the user must code three TGROUP macro instructions. The first defines the transaction group for the 2795s, the second defines the transaction group for the 2796s, and the third defines the transaction group for the 2797s. If only 2797s are connected to the area station, three TGROUP macros must still be coded. The first two define dummy transaction groups and the third defines the transaction group for the 2797s. If only 2796s are connected to the area station, two TGROUP macros must be coded. The first will be a dummy and the second will define the transaction group for the 2796s. If only 2795s are connected to the area station, one TGROUP macro must be coded. The dummy groups are required because the 2715 microcoded routines expect to find the groups for the three types of DEUs in the same relative main storage position from the beginning of the user tables.

Where two or more TGROUP macros are coded for the same area station, the name of the first macro must appear in the DEGROUP operand of the AS macro for that area station. The first operand of any dummy TGROUP macro must be identical to the first operand of the following non-dummy TGROUP macro for the same area station. Only one operand need be coded for dummy TGROUP macros.

Figure 40 shows examples of how TGROUP macros may be coded. The format of the TGROUP macro is:

Name	Operation	Operand
symbol	TGROUP	[[TCn=(symboln[,E])]]

symbol

the name field is required for this macro instruction.

TCn

if TCn=symboln is coded, the transaction code "n" is associated with the transaction list referred to by "symboln". At least one TCn operand must be present. An MNOTE is issued if all operands are omitted. The value of "n" must be between 1 and 9 inclusive. If TCn=(symboln,E) is coded, it indicates that this is a transaction expansion entry and that the transaction is associated with the transaction group referred to by "symboln". The transaction group referred to by "symboln" must have transaction code 1 defined. When using the transaction

expansion function, the format of the TGROUP referred to varies for area stations and data entry units. The first character of data received indicates the desired entry in the TGROUP Table.

Each entry in the transaction group referred to points to a transaction list. If the transaction expansion is for an area station, the first step of each of these transaction lists must be identical. Transaction expansion must be used when generating the transaction list in which message routing is to be specified. A transaction expansion entry must not refer to another transaction expansion entry.

ASCTR (Area Station Counter) Macro Instruction

The ASCTR macro is used to generate the Area Station Counter Table. The Area Station Counter Table requires two bytes of control information for each of up to 100 area stations, plus two additional bytes, for a maximum of 202 bytes. This table is truncated at the highest assigned area station, that is, the area station with the highest ID. Unassigned area stations below the highest assigned area station will each have two bytes defined in the Area Station Counter Table by BTAM at assembly time. The highest assigned area station with counters can be any assigned area station less than or equal to the highest assigned area station that is defined by the AS macro operand ID=n.

Associated with each area station with counters is a displacement that provides for scanning of the counter table. This displacement is used with the counter address to provide an index to individual counter level control. Schedule readout and count testing are performed at the individual counter level.

This macro also allows routing specification to be specified for counter overflow and count test response messages. All counters attached to an area station must have these messages routed to the same destination.

The format of the ASCTR macro instruction is:

Name	Operation	Operand
[symbol]	ASCTR	ID=absexp, HIGHCTR=absexp, ROUTE=(CPU DISK [, LOG] [, ASLOG] [, EXTALRM] [, NEXTAS=absexp]

The macro is coded once for each area station with pulse counters. The maximum number of ASCTR macros that can be coded is 100.

symbol
the name field is optional.

ID
specifies the decimal representation of the address of the area station on which pulse counters are available. The ID operand is not required for all area stations, but the ID operands must be in ascending sequence. An ID operand out of sequence causes an invalid table assembly. The ID operand may have values from 0 to 99.

HIGHCTR
specifies the number of the highest counter on this area station that scheduled readout or count testing may be performed on. Values for the HIGHCTR operand may range from 0 to 63, since only 63 counters are allowed on any given area station. A value of 0 indicates that no area station counters use scheduled readout or count testing.

ROUTE
specifies the destination of counter overflow and count test response messages. At least one destination must be specified, and if only one is specified, the parenthesis are not coded. The CPU and DISK suboperands are mutually exclusive.

CPU
specifies that counter overflow and count test response messages should be routed directly to the CPU.

DISK
specifies that counter overflow and count test response messages should be routed to the 2715 integral disk.

LOG
specifies that counter overflow and count test response messages should be routed to the 2740 attached to the 2715.

ASLOG
specifies that counter overflow and count test response messages should be routed to the area station 1053 printer from which the overflow was initiated.

EXTALRM
specifies that counter overflow and count test response messages should be routed to the 1053 printer on the area station from which the overflow was initiated, and that the external alarm contact closure at the area station should be activated.

The above suboperands specifying routing information for counter overflow and count test response messages need not be coded in any given order.

NEXTAS
specifies the decimal representation of the address of the next higher area station with pulse counters on which count testing or readout functions may be scheduled. The NEXTAS operand may have values from 0 to 99. This operand must be coded when HIGHCTR=0 is coded, but is not necessary for any other HIGHCTR value. NEXTAS=0 must be coded if there is no higher area station that has pulse counters on which count testing or readout functions may be scheduled. Unless NEXTAS=0 is coded, the NEXTAS operand must be greater than the ID operand for this ASCTR macro. If the NEXTAS operand is greater than 0, the ASCTR macro referred to by the value of the NEXTAS operand must have a HIGHCTR operand value greater than 0.

CTRGROUP (Counter Group) Macro Instruction

The CTRGROUP macro is used to generate the Counter Table. The macro must be coded once for each counter in the system on which schedule readout or count test functions are to be performed. A CTRGROUP macro must be coded for the counter whose value was specified in the HIGHCTR operand of the ASCTR macro instruction for this area station. The counter Table is organized on a group basis. Each group consists of the highest counter with scheduled readout or testing and all counters (whether scheduled or unscheduled) below it on the same area station. Each group is truncated at the highest counter scheduled for readout or testing, with a maximum of 63 counters allowed per area station. The Counter Table consists of two bytes of control information for each of these counters (scheduled and unscheduled) plus two additional bytes at the end of the table. Up to 504 counters may be scheduled for the entire system. Each CTRGROUP macro defines two bytes of control information for a particular counter.

The format of the CTRGROUP macro instruction is:

Name	Operation	Operand
[symbol]	CTRGROUP	ctrno, [sro], [cttest], ID=absexp [,SROENAB={NO }] {YES } [,CTINIT={ NULL NCT UNASP }]

symbol

the name field is optional.

ctrno

identifies the counter on which schedule readout or count testing is to be done. The value of the ctrno operand must be from 1 to 63; however, all values in this range need not be specified. All counters must be specified in ascending sequence.

sro

indicates which readout schedule is to be used for this particular counter. The value of the sro operand must be from 0 to 15. A 0 value indicates that schedule readout is not to be performed for this counter. If the sro operand is omitted, no schedule readout will be performed. This operand must not specify a test schedule greater than the highest test schedule defined by the CTRSCHED macro instruction.

cttest

indicates which count test schedule is to be used for this particular counter. The value of the cttest operand must be from 0 to 15. A 0 value indicates that count testing is not to be done. If the cttest operand is omitted, count testing is not performed. This operand must not specify a test schedule greater than the highest test schedule defined by the CTRSCHED macro instruction.

ID

specifies the decimal representation of the address of the area station on which this particular counter is defined. This operand is required.

SROENAB

specifies whether or not schedule readout is to be automatically started by the 2715 at ICPL time. If SROENAB=YES is coded, this indicates that schedule readout is to be automatically started by the 2715. Coding SROENAB=NO indicates that schedule readout is not to be automatically

started by the 2715; the user can initiate schedule readout with a control request at a later time. SROENAB=YES must not be coded if the sro operand is 0 or is omitted. If the SROENAB operand is omitted, SROENAB=NO is assumed.

CTINIT

specifies an initial count test condition that is to be started by the 2715 after an ICPL for this counter. Coding CTINIT=UNASP indicates that unassigned production testing is to be started by the 2715. Coding CTINIT=NCT indicates that no-count testing is to be started by the 2715. Coding CTINIT=NULL indicates that neither unassigned nor no-count testing is to be started by the 2715 for this counter. CTINIT=UNASP or CTINIT=NCT must not be coded if the cttest operand is 0 or is omitted. If the CTINIT operand is omitted, CTINIT=RESET is assumed. Count testing can be initiated later by a control request if it is not automatically started at ICPL time.

CTRSCHED (Counter Schedule) Macro Instruction

The CTRSCHED macro defines the count test schedules and the readout schedules to be used by all the pulse counters in the 2790 System. The count test and readout schedules may be any of 15 possible schedules in the Schedule Table. The format of the CTRSCHED macro instruction is:

Name	Operation	Operand
[symbol]	CTRSCHED	sched,...

symbol

the name field is optional.

sched

specifies a count test schedule or readout schedule in minutes. This operand must be coded once for each schedule interval to be defined, but the maximum number of schedules that can be coded is 15. The value of this operand must be between 1 and 2047.

GDUTRANS (Guidance Display Unit Transaction) Macro Instruction

The GDU Transaction macro is used to build an entry in the Transaction Table. The position of the entry to be built relative to the beginning of the table is determined from the TRCODE operand. A GDUTRANS macro must be coded for every transaction code that will be used in communication with the 2798 GDUs. Each GDUTRANS macro requires

two bytes of 2715 storage. The format of the GDUTRANS macro is:

Name	Operation	Operand
[symbol]	GDUTRANS	TRCODE=absexp, TRLIST=symboln

symbol
the name field of this macro is optional.

TRCODE
this operand is the value of a transaction code. Each transaction code is associated with a particular GDU List. The TRCODE operand is used to determine the position of an entry in the Transaction Table. The value of the TRCODE operand must be from 00 to 99. However, all values in this range need not be specified. The GDUTRANS macros must be in ascending sequence by TRCODE operands. A GDUTRANS macro found to be out of sequence or in error terminates the assembly of this macro instruction.

TRLIST
this operand associates a GDU transaction list with the transaction code indicated in the TRCODE operand. The transaction list (TRLIST) referred to by 'symboln' is associated with the TRCODE operand.

PARAMNUM (Parameter List Number) Macro Instruction

The PARAMNUM macro is used to define an entry in the Parameter List Number Table for use with 2798 GDUs. The Parameter List Number macro is coded once for every parameter list defined by the user with the PARMLIST macro. Up to 127 PARAMNUM macros can be coded. This macro requires two bytes of 2715 storage. The format of the PARAMNUM macro is:

Name	Operation	Operand
[symbol]	PARAMNUM	PLN=absexp, PARMLST=symboln

symbol
the name field of this macro is optional.

PLN
this operand specifies the parameter list number that is to be associated with the parameter list referred to by the PARMLST operand. The value of this operand must be between 1 and

127. The PARAMNUM macros must be in ascending sequence by PLN. A PARAMNUM macro found out of sequence or in error terminates the assembly of this macro instruction. The value of the PLN operand must be 1 greater than the PLN operand of the previous PARAMNUM macro.

PARMLIST
this operand specifies the name of a parameter list defined by a PARMLIST macro.

PARMLIST (Parameter List) Macro Instruction

The PARMLIST macro is used to generate an entry in the Parameter List Table. The data in the parameter list defines the types of checks that are to be performed on a data entry from a 2798 GDU. The entries in the Parameter List Table to be used by the 2715 are selected by a pointer from the Parameter List Number Table. Every PARMLIST macro must be referred to by a PARMLIST operand in the PARAMNUM macro. The size of the Parameter List entry is variable depending on the type of tests requested. The format of the PARMLIST macro is:

Name	Operation	Operand
symbol	PARMLIST	[CKLENGTH=(length-absexp, errguidance-absexp,...)] NO [,CKMONKY= YES] [,CKMOD11=(length-absexp, position-absexp, errguidance-absexp,...)] [,CKRANGE=(position1-absexp, position2-absexp, hilowchars-absexp,...)] [,LOWGUID=(absexp,...)] [,HIGUID=(absexp,...)] ERROR [,RNGTST= DATA] [,CKMOD10=(length-absexp, position-absexp, errguidance-absexp,...)] [,CKOR=(position-absexp, checkchar1-hexchar,... checkcharn-hexchar)] [,ORGUID=(absexp,...)] [,CKAND=(position1-absexp, position2-absexp, checkchar1-hexchar,... checkcharn-hexchar)] [,ANDGUID=(absexp,...)] [,CKNONUM=(position1-absexp, position2-absexp, errguidance-absexp,...)] [,CKNUM=(position1-absexp, position2-absexp, errguidance-absexp,...)] NO [,TRANSL= YES] NO [,IDENT= YES]

symbol

the name field must be specified and must be the same name as defined by the PARMLST operand in the PARAMNUM macro.

Component Description: IBM 2790 Data Communication System, GA27-3015.

length

this suboperand specifies the length of the modulus 11 check field. The value of this suboperand may be from 1 to 15. The length does not include the self-check character.

CKLNGTH

this operand causes the 2715 to check the data entry to determine if it is the length specified.

length

this suboperand specifies the length of the data entry and its value may be from 1 to 17.

position

this suboperand specifies the starting position of the modulus 11 check field. The value of this suboperand may be between 2 and 16.

Note: The first byte in the data entry is the Operational Status byte that is generated by the 2715. Therefore, the value specified by the length suboperand will always be one more than the number of characters entered by the GDU operator. For example, if the user doesn't expect any characters to be entered from the GDU keyboard he must assign a value of 1 to the length suboperand.

errguidance

this suboperand specifies the error guidance that is returned to the 2798 GDU if the modulus 11 check is not satisfied. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when a CKMOD11 error condition exists. (See Figure 37.)

errguidance

this suboperand specifies the error guidance that is returned to the 2798 GDU if the data length is incorrect. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when a CKLNGTH error condition exists. (See Figure 37.)

Example: CKMOD11=(6,2,2,10,14) will result in a modulus 11 check of a 7 digit field (the seventh digit is the self-check digit) starting at GDU data entry position 2. If the self-check digit does not satisfy the modulus 11 check, the error guidance lights for line 2 on the left panel and lines 10 and 14 on the right panel will be turned on.

Example: If the user coded CKLNGTH=(8,2,4,16) and the GDU operator entered any number of characters other than 7, then the error guidance lights for lines 2 and 4 on the left panel and line 16 on the right panel will be turned on.

CKRANGE

this operand causes the 2715 to check the specified field to ensure that it is neither less than the specified low test value nor higher than the specified high test value. This check also tests the data to ensure that it is numeric.

CKMONKY

this operand indicates whether the 2715 will check to determine if the 2798 Monitor key is on. If CKMONKY=YES is coded, the 2715 checks that the Monitor key is on. If a Monitor key error is encountered when CKMONKY=YES, the error guidance, as specified by the MONERR operand in the CONFIGUR macro, is returned to the 2798 GDU.

position1

this suboperand specifies the starting position of the field for which the range check is performed. The value may be from 2 to 17.

position2

this suboperand specifies the last position of the field for which the range check is performed. The value may be from 2 to 17.

CKMOD11

this operand causes the 2715 to perform a modulus 11 check on the field specified and tests the data in the field to ensure that it is numeric. The last position in the specified field must contain the self-check character. A detailed description of modulus 11 checking can be found in

hilowchars

this suboperand specifies the high and low test digits (0-9) for each character in the field. Up to 16 'hilowchars' may be coded. If the high and low test digits coded for a position of a field are the same digit, the

2715 will check that the test position is indeed that digit.

Example: CKRANGE=(8,10,91,80,63) will cause the 2715 to check for a 3 digit number starting in data entry position 8 and ending in position 10. The 3 digit number in positions 8 through 10 must be greater than or equal to 103 and less than or equal to 986.

LOWGUID

this operand specifies the error guidance that is returned to the 2798 GDU if the specified field in the CKRANGE check is lower than the low test value. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the low test condition exits. (See Figure 37.)

Example: From the example associated with the CKRANGE operand, the user may code LOWGUID=(8,9) to inform the GDU operator if the number he entered at the 2798 GDU is less than 103. If the number is less, the error guidance lights for line 8 on the left panel and line 9 on the right panel will be turned on.

HIGUID

this operand specifies the error guidance that is returned to the 2798 GDU if the specified field in the CKRANGE check is higher than the high test value. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the high test condition exists. (See Figure 37.)

Example: From the example associated with the CKRANGE operand, the user may code HIGUID=(10,12) to inform the GDU operator if the number he entered at the 2798 GDU is greater than 986. If the number is greater, the error guidance lights for lines 10 and 12 on the right panel will be turned on.

RNGETST

this operand indicates the action to be taken when the data entry fails to comply with a CKRANGE check. If RNGETST=ERROR is coded and the CKRANGE fails, the data entry is not accepted and the desired error guidance is returned to the 2798 GDU. If RNGETST=DATA is coded and the CKRANGE fails, the data entry is accepted and the desired error guidance is returned with the normal guidance for the next

step. RNGETST is the only error condition for which data can be accepted.

CKMOD10

this operand causes the 2715 to perform a modulus 10 check on the field specified and tests the data in the field to ensure that it is numeric. The last position in the specified field must contain the self-check character. A detailed description of modulus 10 checking can be found in Component Description: IBM 2790 Data Communication System, GA27-3015.

length

this suboperand specifies the length of the modulus 10 check field. The value of this suboperand may be from 1 to 15. The length does not include the self-check character.

position

this suboperand specifies the starting position of the modulus 10 check field. The value of this suboperand may be from 2 to 16.

errguidance

this suboperand specifies the error guidance that is returned to the GDU if the modulus 10 check is not satisfied. Each value of this suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when CKMOD10 error condition exists. (See Figure 37.)

Example: CKMOD10=(4,1,16) will result in a modulus 10 check of a 5 digit field (the fifth digit is the self-check digit) starting at GDU data entry position 1. If the self-check digit does not satisfy the modulus 10 check, the error guidance light for line 16 on the right panel will be turned on.

CKOR

this operand causes a check by the 2715 to ensure that the character received in the position specified in the data entry is one of the check characters specified by the user. There may be one to five unique check characters associated with this test and only one must compare.

position

this suboperand specifies the position in the data entry that is checked for the character comparison. The value of this suboperand may be from 2 to 17.

checkchar1,checkcharn

each suboperand defines a check character. From 1 to 5 of these suboperands may be coded. The value of the suboperand may be the hexadecimal equivalent of any of the characters: A-Z, 0-9, or any of the special characters: . \$ % - / , # " @ = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Figure 38.)

Example: CKOR=(3,D3,F5,7C,61) will check the fourth data entry position (the third character entered by the operator) to ensure that it contains one of the characters: L, 5, @, or /. If the character is not one of the four specified, the user may code the following operand.

ORGUID

this operand specifies the error guidance that is returned to the 2798 GDU if the CKOR check indicates an error. The error occurs when the character in the specified data entry position does not equal any of the check characters. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when a CKOR error condition exists. (See Figure 37.)

Example: From the example associated with the CKOR operand, the user may code ORGUID=(2,3,4) to inform the GDU operator if the third character he entered at the 2798 GDU is not equal to one of the specified characters. If the character is not equal, the error guidance lights for lines 2, 3, and 4 on the left panel will be turned on.

CKAND

this operand causes the 2715 to check the characters received in the consecutive positions specified to ensure that they match all of the specified check characters.

position1

this suboperand specifies the starting position of the field for which the CKAND compare is started. The value of this suboperand may be from 2 to 17.

position2

this suboperand specifies the last position of the field for which the CKAND compare occurs. The value of this suboperand may be from 2 to 17.

checkchar1,checkcharn

each suboperand defines a check char-

acter and from 1 to 16 characters may be coded. The value of the suboperand may be the hexadecimal equivalent of any of the characters: A-Z, 0-9, or any of the special characters: . \$ % - / , # " @ = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Figure 38.)

Example: CKAND=(7,10,C2,E3,C1,D4) will check the consecutive data entry positions 7 through 10 to ensure they contain the characters B, T, A, and M in that order. If an error occurs, the user may code the following operand. If the hexadecimal value X'00' is used as a check character, the character in the corresponding position of the data field will not be checked.

Example: CKAND=(2,5,C1,C2,00,C3) will check data entry positions 2, 3, and 5 to ensure they contain the characters A, B, and C in that order. The character in data entry position 4 will not be checked.

ANDGUID

this operand specifies the error guidance that is returned to the 2798 GDU if the CKAND check indicates an error. This error occurs when the characters received in the consecutive positions specified do not match all of the specified check characters. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when a CKAND error condition exists. (See Figure 37.)

Example: From the example associated with the CKAND operand, the user may code ANDGUID=9 to inform the GDU operator if the specified characters do not match. If the characters do not match, the error guidance light for line 9 on the right panel will be turned on.

CKNONUM

this operand causes the 2715 to check a specified field to ensure that no numeric characters are received.

position1

this suboperand specifies the starting position of the field to be checked. The value may be from 2 to 17.

position2

this suboperand specifies the last position of the field to be checked. The value may be from 2 to 17, but must be greater than or equal to the 'position1' suboperand.

errguidance

this suboperand specifies the error guidance that is returned to the GDU if a numeric character is received and a CKNONUM check is performed. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the CKNONUM error condition exists. (See Figure 37.)

Example: CKNONUM=(2,17,3,6) causes the 2715 to check positions 2 through 17 of the GDU data entry to ensure that all the characters are non-numeric. If any of the characters in the specified positions are numeric, the error guidance lights for lines 3 and 6 on the left panel will be turned on.

CKNUM

this operand causes the 2715 to check a specified field to insure that all numeric characters are received.

position1

this suboperand specifies the starting position of the field to be checked. The value may be from 2 to 17.

position2

this suboperand specifies the last position of the field to be checked. The value may be from 2 to 17, but must be greater than or equal to the 'position1' suboperand.

errguidance

this suboperand specifies the error guidance that is returned to the GDU if a non-numeric character is received and a CKNUM check is performed. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the CKNUM error condition exists. (See Figure 37.)

Example: CKNUM=(8,8,15) causes the 2715 to check position 8 of the GDU data entry to ensure that the character in this position is numeric. If the character in the specified position is non-numeric, the error guidance light for line 15 on the right panel will be turned on.

TRANSL

this operand indicates if the translate function will be performed on the transaction step that uses this parameter list. If TRANSL=YES is coded, the translate function will be used. If TRANSL=NO is coded, or the operand is omitted, the translate function

will be used. This operand can not be used if any other operand in the PARMLIST macro except CKMONKY is used. The TRANSL and IDENT operands are mutually exclusive.

IDENT

this operand indicates if a store or get identifier function may be performed on the transaction step that uses this parameter list. If IDENT=YES is coded, the store or get identifier function may be used. If IDENT=NO is coded or the operand is omitted, the store or get identifier function may not be used. Other checks may be specified. The IDENT and TRANSL operands are mutually exclusive.

Note: Only three of the following seven check operands can be coded on a PARMLIST macro:

- CKMOD11
- CKRANGE
- CKMOD10
- CKOR
- CKAND
- CKNONUM
- CKNUM

The CKLNGLTH and CKMONKY operands may be coded on any PARMLIST macro, regardless of how many other check operands are coded.

Only one check may be performed on a given data position in the 2798 GDU data entry. The seven check operands listed above must not overlap. A particular position in the data entry cannot be covered by more than one check. This does not apply to either the CKLNGLTH or CKMONKY operands.

DISPGUID (Display Guidance) Macro Instruction

The DISPGUID macro is used to define a display guidance message in the Display Guidance Table. A DISPGUID macro must be defined for every display guidance message the user defines. The display guidance address in the GDU step of the GDU list is used by the 2715 to address a particular display guidance message in the Display Guidance Table. The DISPGUID macro requires from 2 to 17 bytes of 2715 storage. The format of the macro is:

Name	Operation	Operand
{symbol}	DISPGUID	DISPMSG='text' {YES} [, SUPPRES={NO}]

symbol

the name field of this macro is optional.

DISPMSG

this operand defines a user specified display guidance message. The text must not exceed 16 characters.

SUPPRES

this operand indicates whether the display guidance message is returned to the 2715 after it is displayed at the GDU display guidance and the operator presses the GDU Enter Key. Coding SUPPRES=YES or omitting the operand indicates that the defined data that was written to the GDU by the 2715 is not to be returned with the operator added data to the 2715. Only that data inserted by the GDU operator will be returned. Coding SUPPRES=NO will cause the defined data and operator inserted data to be returned, up to a maximum of 16 characters. If the operator inserted data plus the defined data exceed 16 characters, the defined data will be moved to the left and the right most characters lost.

The maximum number of DISPGUID macros that can be issued depends only on the user table size limitation.

TRANSLAT (Translate Table) Macro Instruction

The TRANSLAT macro instruction builds an entry in the Translate Table. A maximum of eight TRANSLAT macros may be coded and 3 to 16 bytes of 2715 storage are required for each. This macro is coded once for each character that is translated. The format of the TRANSLAT macro is:

Name	Operation	Operand
[symbol]	TRANSLAT	TRANSCH=hexchar, TRANTXT='text'

symbol

the name field of this macro is optional.

TRANSCH

this operand defines the character that is translated. The value for this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: . \$ % - / , # @ " = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Figure 38.)

TRANTXT

this operand defines the user specified translate text.

Example: An assembly line worker is required to enter the character C from a 2798 GDU each time he builds and tests a specific clutch. The programmer coded the following in the TRANSLAT macro: TRANSCH=C3,TRANTXT='4 SPEED CLUTCH'. The 2715 checks the character entered for this Translate transaction and replaces the C with the text '4 SPEED CLUTCH'. The text is now displayed at the 2798.

Note: Each character assigned to a text must be unique, that is, assign a different character to each text.

TRLIST (Transaction List) Macro Instruction

The Transaction List macro is used with the Area Station List macro and the Data Entry Unit List macro to define a transaction. When the transaction is initiated by an area station, the Transaction List macro is followed by one or more ASLIST macros. When the transaction is initiated by a data entry unit, the TRLIST macro is followed by one or more DEULIST macros. The DEULIST macros may be followed by a CTRLIST macro. The first transaction list must be for all of the IBM 1035 Badge Readers. This consists of a TRLIST macro instruction followed by one DEULIST macro instruction.

The Transaction Lists created by the TRLIST, ASLIST, DEULIST and CTRLIST macro instructions are composed of two elements: a header, and either an internal message or from one to sixteen data-entry steps. The header information is provided in the TRLIST macro instruction. The TRLIST macro must follow the last TGROUP macro.

The TRLIST macro is used to generate the transaction headers for GDU lists and can be referred to by the GDUTRANS macro.

Message routing can be specified (NULL or absexpl coded) only when the TRLIST is part of a transaction expansion. Message routing means that explicit or explicit/implicit text is to be routed to 1053 printer on an area station. The only ways the user can specify message routing are to code ROUTE=NULL or ROUTE=absexpl in the TRLIST macro. Coding ROUTE=DISK, CPU, or IOG does not imply message routing, but that the transaction is to be routed to the specified ROUTE parameter. Transaction routing does not mean routing to an area station 1053 printer.

Each TRLIST macro requires [5 + (text length + 1)] bytes of 2715 storage. The format of the TRLIST macro is:

Name	Operation	Operand
symbol	TRLIST	TRID=absexp1 [,ROUTE= [{ DISK }]] [{ CPU }]] [,LOG] [{ ,NULL }] [{ ,absexp2 }]] [,TEXT= [{ NO }]] [{ YES }]] [,INQDISP= [{ NO }]] [{ YES }]] [,DEM0D10= [{ NO }]] [{ YES }]] [,DEM0D11= [{ NO }]] [{ YES }]] [,GDU= [{ NO }]] [{ YES }]]

symbol
the name field is required for this macro instruction.

TRID=absexp1
specifies a transaction identifier. The user assigns a value from 0 to 159 to "absexp1," and the 2715 places this value in the transaction control byte of the transaction header for priority and deferred data. The value of "absexp1" must be in ascending order with the other TRID parameters coded in the program; however, values may be omitted (a warning message is generated at assembly time). Since the user receives the transaction header with a message, the transaction identifier allows him to determine which TRLIST macro processed the data in the 2715.

ROUTE
the ROUTE operand specifies the destination of the data records (transactions) that originate on one of the devices attached to the 2715. At least one destination must be specified, and if only one is specified the parentheses are not coded.

DISK
specifies that the transaction should be routed to the 2715 integral disk; that is, the message is a deferred message.

CPU
specifies that the transaction should be routed directly to the CPU; that is, the message is an inquiry or a priority message.

LOG
specifies that the transaction is to be routed to the 2740 attached to the 2715.

NULL
specifies that the first data entry of the transaction is the destination address of the message, that is, the hexadecimal address of an area station. The message is to be routed to the printer attached to that area station.

absexp2
specifies the decimal representation of the address of an area station (see Figure 39). The message is to be routed to the printer attached to that area station.

Note: The suboperands of the ROUTE parameter may be coded in any order. If one is omitted, commas need not be coded to indicate the omission.

TEXT= NO
 YES
specifies that a message defined in a subsequent ASLIST, DEULIST, GDULIST or CTRLIST macro is to be routed.

INQDISP
the INQDISP operand indicates whether this transaction is an Inquiry Display transaction. Coding INQDISP=YES specifies that inquiry display will be used in this transaction. Coding INQDISP=YES requires that INQDISP=YES be coded in the CONFIGUR macro. Coding INQDISP=YES requires one extra GDULIST macro to end this transaction list. See GDULIST macro description for details.

DEM0D10
the DEM0D10 operand indicates whether the 2715 will perform a Modulus 10 self check on all or part of a data entry from an area station or data entry unit. Coding DEM0D10=YES specifies that Modulus 10 self checking will be performed on a data entry in this transaction. Coding DEM0D10 requires CORE=32 to be coded in the CONFIGUR macro. This operand is mutually exclusive with the DEM0D11 operand. This operand does not apply to the 2798 GDU.

DEM0D11
the DEM0D11 operand indicates whether the 2715 will perform a Modulus 11 self check on all or part of a data entry from an area station or data

entry unit. Coding DEMOD11=YES specifies that Modulus 11 self checking will be performed on a data entry in this transaction. Coding DEMOD11=YES requires CORE=32 to be coded in the CONFIGUR macro. This operand is mutually exclusive with the DEMOD10 operand. This operand does not apply to the 2798 GDU.

GDU

this operand allows for 100 additional transaction identifiers (TRID) to be specified by the user. The normal range of identifiers is from 0 to 159 and the additional identifiers range from 0 to 99. Coding GDU=YES resets the TRID operand checking and allows for a maximum of 100 more transaction identifiers to be specified. The checking resumes with the new identifiers which may or may not be unique identifiers. If non-unique identifiers exist, the user must also check the device address in the transaction header to determine if the transaction is for a 2798 GDU. All TRLIST macros for GDU transactions and the associated GDULIST macros should be the last macros coded before STEND. (See Appendix M.)

Example: This example represents a series of 260 TRLIST macros with all other macros omitted:

```

TRLIST0    TRLIST  TRID=0,ROUTE=DISK
TRLIST1    TRLIST  TRID=1,ROUTE=CPU
TRLIST2    TRLIST  TRID=2,ROUTE=DISK
.          .
.          .
.          .
TRLIST159  TRLIST  TRID=159,ROUTE=CPU
GDUTR0     TRLIST  TRID=0,ROUTE=DISK,
              GDU=YES
.          .
.          .
GDUTR99    TRLIST  TRID=99,ROUTE=CPU,
              GDU=YES

```

ASLIST (Area Station List) Macro Instruction

The Area Station List macro instruction is used to define one step of a transaction list for a 2791 Area Station. One to sixteen ASLIST macros may follow a TRLIST macro. If more than sixteen are used, the excess macros are flagged as errors in the assembly. Each ASLIST macro requires 5 bytes of 2715 storage. If the message operand is coded, the ASLIST macro requires additional storage of length-of-text-plus-one bytes. The format of the ASLIST macro instruction is:

Name	Operation	Operand
{symbol}	ASLIST	device-code,NORM=absexp [,LENGTH=(absexp1, absexp2)] [,DIGIT=(absexp1, absexp2,absexp3)] [,ENTRY={ $\frac{1}{M}$ }] [,MSG='text'] [,INQDISP=absexp [,MODULUS=(absexp1, absexp2,absexp3)] [,SELTRAN={ $\frac{NO}{YES}$ }]

symbol

the name field of this macro instruction is optional.

device-code

this operand indicates the device to be activated at the 2791 Area Station. The accepted values are:

- B - Badge
- C - Card
- M - Manual entry
- O - OEM input

NORM

this operand indicates which guidance light on the area station should be switched on if no error is recognized in the previous step of the transaction (see Figure 38). (The first step is considered to be the acceptance of the transaction code.) This value must be from 1 to 31.

LENGTH

absexp1

specifies the significant length of the data entry (the number of data characters excluding blanks). This may be any value from 0 to 81; the maximum length depends on the input device -- card reader, badge reader, manual entry, OEM entry.

absexp2

specifies which guidance light should be switched on if the number of characters received is different from the value specified by "absexp1". The value of "absexp2" must be from 1 to 31. (See Figure 41.)

No length error checking takes place if the LENGTH parameter is not coded. If the LENGTH parameter is omitted, or if zero is specified, no significant length checking is done.

31	30	29	28
27	26	25	24
23	22	21	20
19	18	17	16
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	SELECT TRANS- ACTION
ON LINE	REPEAT CLEAR	IN PROCESS	CARD IN

Figure 41. ASLIST Operand Values for Guidance Lights by Position on the Area Station

DIGIT

absexp1

specifies the position or column of the value in the data entry that is to be compared with the value specified in "absexp2". The value of "absexp1" must be from 1 to 15.

Note: The actual position of the first byte of data received from the input devices depends on the device. For the local badge reader, card reader, keyboard, and OEM devices on the 2791, the first byte of data is in position 2. (Position 1 is the Monitor key.)

absexp2

specifies a value, from 0 to 9, to be compared with a specified value in the data entry.

absexp3

indicates which guidance light should be switched on if the specified values do not match. This value must be from 1 to 31. (See Figure 38.)

If this operand is omitted, no error checking takes place.

The DIGIT operand can not be coded if DEMOD10=YES or DEMOD11=YES in the TRLIST macro.

ENTRY

this operand allows a processing step to accept multiple input data entries until the operator calls for the next step to be activated. If the ENTRY

operand is omitted, or if ENTRY=1 is coded, there will be only one data entry for this step. If ENTRY=M is coded, this step may be repeated until ended by the operator. User-documented instructions to the operator must reflect the fact that the total number of bytes of data entered must not exceed the maximum transaction length of 247. This operand must not be coded on the first or only ASLIST macro following a TRLIST macro.

INQDISP

this operand specifies which guidance light on the area station is turned on when an Inquiry Display transaction is received by the 2715 and routed to the CPU as priority data. This is a user specified guidance such as "Inquiry in Process." The value of the operand must be from 2 to 31 (0 is reserved for Select Transaction and 1 is reserved for an aborted inquiry). Coding this operand requires INQDISP=YES to be coded in the TRLIST macro for this transaction.

MODULUS

this operand indicates the field in this data entry for which the 2715 performs either a Modulus 10 or Modulus 11 self check algorithm. Error guidance is also specified when the self check fails.

absexp1

specifies the starting position of the field for the modulus check.

absexp2

specifies the length of the field on which the modulus check is performed. The value can be from 1 to 15. This length does not include the self check character.

atsexp3

specifies which guidance light is turned on if the modulus check fails. This value must be from 1 to 31.

Note: The MODULUS operand can not be coded unless DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro. This operand is mutually exclusive with the DIGIT operand.

SELTRAN

this operand allows the Select Transaction light on the 2791 Area Station to be turned on at the completion of a transaction, instead of the first guidance light. Coding SELTRAN=YES on any ASLIST macro after the first ASLIST macro in any transaction causes the Select Transaction light to be turned on at the completion of a tran-

saction. If the operand is omitted or if SELTRAN=NO is coded, the first guidance light is turned on at the completion of the transaction. SELTRAN=YES can not be coded on the first ASLIST macro in a transaction.

MSG='text'
 defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may only be specified for the last ASLIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro.

DEULIST (Data Entry Unit List) Macro Instruction

The Data Entry Unit List macro is used to define one step of a transaction list for a data entry unit or to define a transaction for the 1035 Badge Reader. For a 2796 and 2797 DEU one to thirteen DEULIST macros and for a 2795 DEU one to sixteen DEULIST macros may follow a TRLIST macro. If more than sixteen are used, the excess macros are flagged as errors in the assembly. Each DEULIST macro requires 5 bytes of 2715 storage. If the MSG operand appears, the DEULIST macro requires additional storage of length-of-text-plus-one bytes. The format of the DEULIST macro instruction is:

Name	Operation	Operand
[symbol]	DEULIST	[DIGIT=(absexp1, absexp2)] [,LENGTH=absexp1] [,MSG='text'] [,MODULUS=(absexp1, absexp2)] [,DIGIT2=(absexp1, absexp2)]

symbol
 the name of the DEULIST macro is optional.

DIGIT

absexp1
 specifies the position or column of the value in the data entry that is to be compared with the value specified in "absexp2". The value of "absexp1" must be from 1 to 15. If the specified values do not match, the red error button on the data entry unit pops up, and the operator must reenter correct data.

Note: The actual position of the first byte of data received varies depending on the device. For a data entry unit (2795, 2796, 2797), the first byte of data is in position 3. Positions 1 and 2 are the Monitor key and setting of the right-hand knob. For a 1035 badge reader, the first byte of data is in position 1.

absexp2
 specifies a value, from 0 to 9, to be compared with a specified value in the data entry.

The DIGIT operand can not be coded if DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro for this transaction.

LENGTH

absexp1
 specifies the significant length of the data entry (the number of data characters excluding blanks).

To determine the required data entry length, use the following formulas (see Programming Notes for data entry format):

- 1035: Reads a badge = value from 1 to 10 or 0
 - 2795: ID+RK+CDBD = value from 2 to 12 or 0
 - 2796: MON+TRK+CDBD+BLK+BRK+ROCK = value from 8 to 18 or 0
 - 2797: MON+RK+CDBD+MAN = value from 8 to 18 or 0
- BLK = bottom left knob (1 byte)
 BRK = bottom right knob (1 byte)
 CDBD = card or badge (0 to 10 bytes)
 ID = ID code (1 byte)
 MAN = manual entry (6 bytes)
 MON = Monitor key (1 byte)
 RK = right knob (1 byte)
 ROCK = digit-rocker switches (4 bytes)
 TRK = top right knob (1 byte)

If zero is specified or if the LENGTH parameter is omitted, no significant length check is performed. If an invalid length is detected, the red error button pops up.

MODULUS

the MODULUS operand indicates the field in this data entry for which the 2715 performs either a Modulus 10 or Modulus 11 self check algorithm. The MODULUS operand can not be coded unless either DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro for this transaction. This operand is

mutually exclusive with the DIGIT operand, but not the DIGIT2 operand.

absexp1
specifies the starting position of the field for which the modulus check is performed. The value can be from 2 through 16 corresponding to the last data positions in the data entry.

absexp2
specifies the length of the field for which the modulus check is performed. This value can be from 1 to 15. This length does not include the self check character.

DIGIT2
the DIGIT2 operand specifies a position in the data entry that is checked by the 2715 for a specified value. This operand can be coded when either the DIGIT operand or the MODULUS operand is coded or when neither is coded.

absexp1
specifies the position of the value in the data entry that is compared with the value specified in "absexp2." The value of "absexp1" must be from 1 to 15. If the specified values do not match, the red error button on the data entry unit pops up and the operator must reenter the correct data.

absexp2
specified a value from 0 to 9 that is compared with a specified value in the data entry.

MSG
defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may be specified only for the last DEULIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro. If a CTRLIST macro is coded, the MSG operand may be specified only in the CTRLIST macro.

GDULIST (Guidance Display Unit List) Macro Instruction

The GDULIST macro instruction is used to define one step of a GDU transaction list for a 2791 or 2793 Area Station with 2798 GDUs attached. One to sixteen GDULIST macros may follow a TRLIST macro. If more than sixteen are coded, the excess macros are flagged as errors in the assembly. Each GDULIST macro requires 5 bytes of 2715 storage. If the MSG operand is coded, the GDULIST macro requires additional storage

equal to the length of the MSG text. If an inquiry display transaction (INQDISP=YES in the TRLIST macro) is coded, one extra GDULIST macro must be coded as the last entry of the transaction list. This macro supplies normal guidance light number and display message number only and initiates no checking or parameter list references.

The format of the GDULIST macro is:

Name	Operation	Operand
[symbol]	GDULIST	PARAMNO=absexp [, (NORGUID=absexp, ...)] [, {DISPMSG=symbol}] [, {IDENT=absexp}] [, MSG='text'] [, ENTRY={ $\frac{1}{M}$ }]

symbol
the name field in this operand is optional.

PARAMNO
this operand indicates the parameter list number to be used by the 2715 to get to a parameter list that defines the type of checks to be performed on the data entry for this GDU step. The value of this operand must be defined in a PLN operand of the PARAMNUM macro. The value of the PARAMNO operand must be from 1 to 127.

NORGUID
this operand indicates the normal guidance that will be sent to the operator guidance panel on the GDU when this step is entered. The value of this operand can be from 1 to 16 and up to 16 suboperands can be coded. Each suboperand represents a light on the guidance panel that will be turned on when this particular step is entered.

DISPMSG
this operand specifies the name of the DISPGUID macro that defines the message to be displayed on the 2798 Display Guidance Panel when this step in the GDU transaction is entered.

IDENT
this operand specifies an identifier in the Identifier Table to be displayed on the 2798 Display Guidance Panel when this step in the GDU transaction is entered. The value of the operand must be between 0 and 99 and must be less than the value of the IDCOUNT operand of the CONFIGUR macro (except when the IDCOUNT=0). The DIS-

PMSG and IDENT operands are mutually exclusive. Every time the 2715 is ICPLed, the predefined text 'NOT USED' will be defined in every identifier in the identifier table. The user should use the Store Identifier function prior to using the Get Identifier function after an ICPL if he expects useful information to be in the identifier table.

Example: If the IDCOUNT operand of the CONFIGUR macro indicates there are 6 identifiers in the Identifier Table (IDCOUNT=6), and the user wants to display the fifth identifier when the step associated with a GDULIST macro is entered, then he must code IDENT=4 in this macro (IDENT=0 is the first identifier available).

Note: The user is made aware of the fact that he has not stored any text in a particular Identifier since he performed his table load by having the text 'NOT USED' defined in every Identifier in the Identifier Table at assembly time. When the user displays a particular Identifier as specified by the IDENT operand of a GDULIST macro and sees the text 'NOT USED,' he should realize that he has never stored any text in the Identifier.

CTRLIST (Counter List) Macro Instruction

The CTRLIST macro is used to define the last step of a transaction for a data entry unit that is attached to a 2793 Area Station with pulse counters. This macro generates a five-byte data entry step for pulse count. The counter appendage step must be the last step in a transaction.

The format of the CTRLIST macro instruction is:

Name	Operation	Operand
[symbol]	CTRLIST	DEVCOD= { B } { C } { M } , CTRADR= { IMP } { EXP } , CTRRD= { SINGLE } { GROUP } , CTTEST= { NULL } { SETNCT } { SETUNAS } { RESET } , CTROP= { READ } { SET } { READSET } { READRST } { RDRESID } { NULL } [,MSG='text']

symbol

the name field is optional.

DEVCOD

indicates the way the data entry is entered at the DEU. If DEVCOD=B is coded, a badge will be used; if DEVCOD=C, a card will be used; and if DEVCOD=M, manual entry will be used. M may not be specified for a 2795 DEU.

CTRADR

indicates whether implicit or explicit counter addressing is to be used. Coding CTRADR=EXP indicates that explicit counter addressing is to be used. Explicit counter addressing is entered within the last data entry. This entry is retained as data in normal transaction assembly. Addressing is specified as decimal digits with values from 1 to 63.

Coding CTRADR=IMP indicates that implicit counter addressing is to be used. Implicit counter addressing is valid only from a DEU and implies that only the first 32 counters can be used. For implicit counter addressing, the device address of the DEU initiating the request (from X'C0' to X'DF') will be converted to a counter

MSG

defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may be specified only for the last GDULIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro. If a CTRLIST macro is coded, the MSG operand may be specified only in the CTRLIST macro.

ENTRY

this operand allows a processing step to accept multiple input data entries until the operator calls for the next step to be activated. If the ENTRY operand is omitted, or if ENTRY=1, there will be only one data entry for this step. If ENTRY=M, this step may be repeated until ended by the operator. User documented instructions to the operator must reflect the fact that the total number of bytes of data entered must not exceed the maximum transaction length of 247. This operand must not be coded on the first or only GDULIST macro following a TRLIST macro.

device address (from X'1' to X'20') and used as the implied address.

CTRRD

indicates how counters are to be read. If CTRRD=SINGLE is coded, the counters are to be interrogated individually. Coding CTRRD=GROUP indicates that counters are to be interrogated on a group basis. Group reads are done on a from/to basis with a 16-counter maximum.

CTTEST

specifies the count test options. Coding CTTEST=NULL indicates that there is no change in the present count test condition. Coding CTTEST=SETNCT indicates that no-count test will be enabled and the unassigned production test will be disabled. Coding CTTEST=SETUNAS indicates that the no-count test will be disabled and the unassigned production test will be enabled. Coding CTTEST=RESET disables all testing conditions.

CTROP

indicates the type of counter request to be performed. Coding CTROP=READ indicates that the counters are not to be reset after a single or group read. Coding CTROP=SET indicates that the counters are to be set to the value specified by the user at the DEU. Coding CTROP=READSET indicates that the counters are set to the value specified by the user at the DEU after a single or group read. SET and READSET are valid only for 2796 and 2797 DEUs and may not be routed to the ASLOG printer. Coding CTROP=READRST indicates that the counters are to be reset to zero after a single or group read. Coding CTROP=RDRESID indicates a read residual function, after which the counters are not reset. Coding CTROP=NULL indicates that no read or set counter functions will be performed in this transaction.

MSG

defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This TRLIST macro must also have specified TEXT=YES.

STEND (Statement End) Macro Instruction

The Statement End macro instruction is used with or without a name and must have no operands. It is used to indicate the end of all user macros. This must be the last card processed before the assembler END card. The STEND macro instruction compares

the total number of bytes generated for the 2715 tables with the maximum allowable size for the user's particular 2715 (see CONFIGUR). If the size of the tables exceeds the maximum, an MNOTE is issued indicating the assembly is invalid.

Name	Operation	Operand
[symbol]	STEND	

symbol

the name field of this macro is optional.

Note: A warning MNOTE is generated by this macro.

PROGRAMMING NOTES

The following general operational characteristics should be remembered when communicating with a 2715:

- When priority data has been read to exhaustion (EOT received), the user should write a control message to the 2715 requesting deferred data and then read that data until an EOT is received.
- When a data message has been accepted by the 2715 but cannot be routed to its ultimate destination, the 2715 sends the message back to the System/360 with the transaction control byte unchanged, and an error code in the zone field of the second byte of the time field of the transaction header. The error codes are:

2740 not attached	1110
2740 not operational	1101
Incomplete transaction	1100
1053 not attached	1011
1053 not operational	1010
2740 overload	1001
MSG routine overload	1000
Invalid request from CPU	0111
Counter not attached (Pulse Count feature)	0101
Device not operational (Pulse Count feature)	0011

- The devices attached to a 2790 system may vary in their ability to transmit blanks. This may affect the length of data entered, that is, data from local card reader, badge reader, etc.
- A 2715 data entry consists of the following:

2795 Data Entry Unit

LEFT KNOB	ID CODE	RIGHT KNOB	10 DATA BYTES
-----12 BYTES-----			

2796 Data Entry Unit

TOP LEFT KNOB	MON- ITOR KEY *	TOP RIGHT KNOB	10 DATA BYTES	BOT- TOM LEFT KNOB	BOTTOM RIGHT KNOB	4 DIGIT ROCKER SWITCH- ES
-----18 BYTES-----						

* Has a value of 1, 2, or 3.

2797 Data Entry Unit

LEFT KNOB	MON- ITOR KEY **	RIGHT KNOB	10 DATA BYTES	6-DIGIT MANUAL ENTRY BUFFER
-----18 BYTES-----				

** Has a value of 4, 5, or 6.

2798 Guidance Display Unit

TRANS- ACTION CODE BYTE	OPERA- TIONAL STATUS BYTE	MAXIMUM OF 16 DATA BYTES
-----17 BYTES-----		

2791 Area Station

TRANSACTION CODE	MONITOR KEY*	1 TO 80 DATA BYTES**
-----1 TO 81 BYTES-----		

*Not included with data entries from 1035 Badge Readers
 X'F0' = Key off
 X'F1' = Key on
 **Card reader-80 bytes
 Badge reader-10 bytes
 Manual entry-6 bytes
 OEM entry-10 bytes

The Monitor key on an area station or a data entry unit allows the operator to add an approval to a given transaction. Approval is accomplished through the transmission of a unique character that is activated by placing a key in a two-position lock switch for the 2791 Area Station and a

three position lock switch for a 2796 or 2797 Data Entry Unit.

Note: The 2715 removes the first character, which is the transaction code (from a transaction key on an area station or the value of the left-hand knob on a 2795 or 2797 or of the top left-hand knob on a 2796 Data Entry Unit).

- The user must provide input/output areas or buffers of at least 640 contiguous bytes to allow for the maximum message length that can be received from the 2715.
- A separate assembly of the following macros is required for table generation:

```
CONFIGUR
AS
TGROU
ASCTR (optional)
CTRGRU (optional)
CTRSCH (optional)
TRLIST
ASLIST (DEULIST)
CTRLIST (optional)
STEND
```

- The first assembly statement must be the CONFIGUR macro (there must not be a TITLE, CSECT, or START card). The last assembly statement must be the END card.

Notes:

1. When transaction expansion is specified, all TRLISTS referred to by this group must be such that the device selection and normal guidance in the first data entry of each of these transaction lists are identical.
2. Storage expansion (32K core) is a prerequisite to transaction expansion.
3. Transaction expansion is a prerequisite to message routing.
4. A transaction expansion entry must not refer to another transaction expansion entry.
5. The first transaction must be for all of the IBM 1035 Badge Readers.
6. Chaining data entries is not allowed for the IBM 1035 Badge Reader.
7. The value coded in the LENGTH parameter must be equal to the number of data characters (nonblank) plus 1.
8. MSG operand may only be specified for the last ASLIST, DEULIST or CTRLIST

macro associated with any TRLIST macro.

9. The last entry of a transaction cannot be a multiple entry.
10. The maximum transaction length on a multiple entry is 247 bytes.
11. All DEUs attached to an area station must use a common transaction group (TGROUP). If 2795, 2796, and 2797 DEUs are attached to the same area station, three TGROUP macro instructions must be coded, but only one DEGROU operand is coded in the AS macro for this area station. See Figure 37 for examples and the discussion of the TGROUP macro instruction for details.

MESSAGE FORMAT

The user communicates with the 2715 using BTAM READ and WRITE macro instructions and BSC line control procedures. When reading from the 2715, the length of the message is text length plus 3 (DLE STX is received at the beginning of the message and ETX is received at the end). The maximum length for text received is 640 characters.

When writing to the 2715, the number of bytes coded in the length operand of the WRITE macro instruction is text length plus 2 (the user must insert DLE STX in front of the text). The total number of bytes written is text length plus 4 (BTAM inserts DLE ETX at the end of the text). The maximum length for text written is 128 characters.

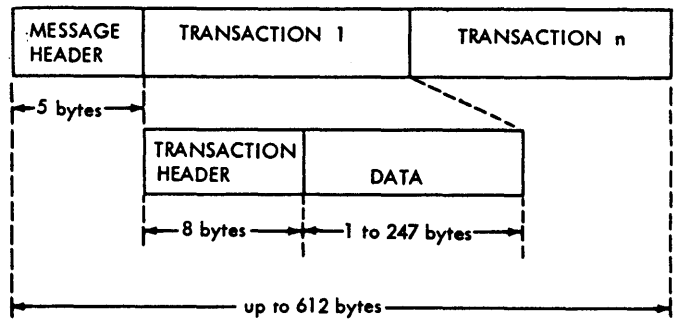
Each message transmitted or received is composed of one or more transactions, preceded by a message header. Each of the transactions is composed of a transaction header and data. When transmitting to the 2715, these headers must be provided by the user in correct format. Message formats are shown in Figure 42.

Message Header -- System/370 to 2715

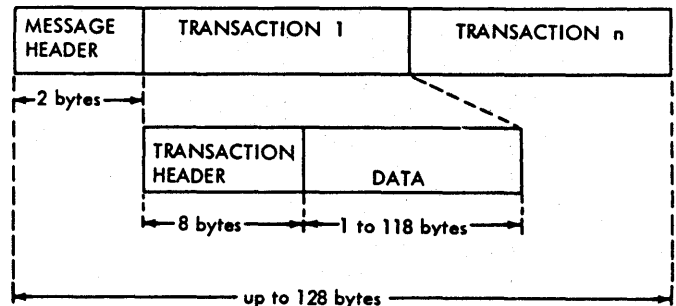
The message header is two bytes and has the following format:

Byte 0	Message length
Byte 1	Message control byte

Message Length: The message length is a one-byte count, in hexadecimal, of the number of characters in the message, including headers and data. The BSC framing control characters are not included in this count. For transmission from the System/370 to the 2715, the message length should not exceed 128 bytes.



2715 to System/370



System/370 to 2715

Figure 42. Message Formats

Message Control Byte: The message control byte is used to indicate one of three possible destinations for output data, as shown below.

MESSAGE DESTINATION	CONTROL BYTE
1053 Printer or Pulse Count	X'01'
2715 Control	X'02'
2740 Terminal	X'04'

Message Header -- 2715 to System/370

The message header is five bytes and has the following format:

Bytes 0 - 2	Work day number
Bytes 3 - 4	Restart number

Work Day Number: The work day number is a three-byte EBCDIC field used as a date field. The date may be omitted, in which case the field is undefined.

Restart number: The restart number is a two-byte field that defines the type of data and associated restart information. The format of this field is:

Byte 3: bits 0-4 Low-order bits of cylinder address.

bit 5 Track.

bits 6-7 Sector.

Byte 4: bit 0 Reserved.

bit 1 If on, indicates deferred data.

bit 2 If on, indicates priority data.

bits 3-4 Reserved.

bits 5-7 High-order bits of cylinder address.

Note: When a data message has been accepted by the 2715 but cannot be routed to its ultimate destination, the 2715 sends it back to the System/370 with an error code inserted in the zone field of the second byte of the time stamp. See the Time Stamp description for definition of the error codes. The transaction control byte still contains the "System/370 to 2715" indication.

Area Station Address: For transmission from the 2715 to the System/370 the Area Station Address field usually contains the area station address. The field contains a hexadecimal value (see Figure 39).

- For priority data and deferred data, the field contains the area station address.
- For responses to control transactions, the field is undefined.

For transmission from the System/370 to the 2715, the field normally contains the area station address. The user must specify the address in hexadecimal (see Figure 39). When the field contains an invalid area station address, the transaction is returned intact with an error code (see Programming Notes). The area station address field should be zero for control or 2740 operations.

Device Address: For transmission from the 2715 to the System/370 the Device Address field identifies the sending data entry unit, 1053 Printer, 1035 Badge Reader, OEM device, or 2791 resident card, badge, or manual entry, or the actual counter address. This field is zero if it is control information. Addresses in this field are represented in hexadecimal form.

For transmission from the System/370 to the 2715, the field usually contains the address of the printer on the area station. The field is zero for the 2740 or control transactions.

Time Stamp: The time stamp is a four-byte field that contains the value of the clock when the data was received. It is carried in conventional form, in hours and minutes, as EBCDIC characters. The field may be omitted on output to the 2715. If the field is omitted, four zero EBCDIC characters (X'F0') must be inserted.

An error condition will be encoded into the zone bits of the second byte to preserve the original time stamp. Note that the zone bits of the first time byte may also be changed.

Transaction Header

The transaction header is eight bytes and has the following format:

Byte 0	Transaction length
Byte 1	Transaction control byte
Byte 2	Area station address
Byte 3	Device address (counter address)
Bytes 4 - 7	Time stamp

Transaction Length: For transmission from the 2715 to the System/370, the transaction length is a hexadecimal count of the number of bytes in a transaction, including the header. The count may not exceed 255; therefore, the maximum number of bytes of data is 247.

For transmission from the System/370 to the 2715, the transaction length is a user-provided hexadecimal count of the number of bytes in a transaction, including the transaction header. The count must not exceed 126; therefore, the maximum number of bytes of data is 118. The 2715 checks the summation of all transaction lengths against the message length. If they do not agree, the 2715 transmits an EOT, aborting the transmission.

Transaction Control Byte: The transaction control byte is a binary code that specifies the type of transaction. Values for the transaction control byte are shown in Figure 43. If the value in a control transaction is not recognized by the 2715, a message is returned to the System/370.

Message Type and Function	Control Byte	Data Entry Bytes (d_1, d_2, \dots, d_n)
System/370 to 2715		
- Data transactions		
- 1053 printer data	X'FA'	d_1-d_n (max=118 bytes)
- 2740 terminal data	X'FB'	d_1-d_n (max=118 bytes)
- Control transactions		
- Bypass area station; causes the specified area station to go offline.	X'C1'	d_1 =area station address in hex
- Restore area station; causes the specified area station to go online.	X'C2'	d_1 =area station address in hex
- Bypass segment; causes the specified segment of the transmission line to be bypassed.	X'C3'	d_1 =segment to be bypassed in EBCDIC
- Restore segment; causes the specified segment of the transmission line to be restored to operation.	X'C4'	d_1 =segment to be restored in EBCDIC
- Read deferred data; causes deferred data to be sent from the 2715 disk to the System/370.	X'C5'	none
- Stop 2790 input.	X'C6'	none
- Start 2790 input.	X'C7'	none
- Alarm messages:		
• Text; sends user error message to the area station 1053.	X'CD'	d_1-d_n =user error message in EBCDIC
• Alarm; causes alarm bell to ring at the area station 1053.	X'CE'	none
• Alarm and text; sends alarm and user error message to area station.	X'CF'	d_1-d_n =user error message in EBCDIC
- User table load start.	X'D1'	none
- User table load data; defines the following data as 2715 tables.	X'D2'	d_1-d =one object card from the user's assembly of 2715 macro instructions
- User table load end.	X'D3'	none
- CPU restart; recovers deferred data that was received subsequent to the specified restart number (used with the checkpoint/restart capability).	X'D4'	d_1-d_2 =restart number that was checkpointed (in hexadecimal)
- 2715 restart; attempts to recover data that has been buffered at the 2715 and not yet transmitted to the System/370 after a 2790 or 2715 irrecoverable error or stop.	X'D5'	d_1-d_2 =deferred restart number log (in hexadecimal) d_1-d =priority restart number log (in hexadecimal)
- Sort area station errors; causes the 2715 to scan the error logout file and extract error statistics for the address specified in the fourth byte of the transaction header.	X'D6'	none
- Read partial error log; causes error data in the 2715 error logout file to be transmitted to the System/370.	X'D7'	none
- Reinitialize 2715 disk (will not be initiated unless all 2790 input is stopped and all deferred data is transmitted to the System/370).	X'D8'	none
- Set day stamp.	X'E2'	none
- Monitor day number; causes the previously set day stamp to be monitored.	X'E3'	none
- Monitor time; causes Real-time clock to be monitored.	X'E4'	none

Figure 43 (Part 1 of 2). Transaction Control Byte Usage

Message Type and Function	Control Byte	Data Byte Entries (d ₁ , d ₂ , ... d _n)
- Reset deferred data mode; causes the 2715 to stop queuing deferred data from the disk to be transmitted to the System/370 (the 2715 will continue to transmit the deferred data already queued).	X'E6'	none
- All Pulse Count transactions:	X'FC'	d ₁ =counter control byte in EBCDIC d ₂ -d ₃ =EBCDIC value of last counter in the group d ₂ -d =EBCDIC value to which the counter is to be sent
2715 to System/370		
- Data Transactions	X'00' X'7F' ¹	
- Control Transactions		
- Positive response to CPU request	X'CA'	
- Negative response to CPU request	X'CB'	
- Invalid response to CPU request	X'CC'	
- Positive response to 2715 request	X'DA'	
- Negative response to 2715 request	X'DB'	
- Response to invalid 2715 request	X'DC'	
- User defined	X'FO' X'F9' X'FF' X'FD'	
- Automatically initiated response		
- Unsolicited 2715 response		
- Special pulse counter transactions		
- Positive response to CPU request	X'CA'	d ₁ =counter control request d ₂ =control definition same as X'CA'
- Invalid response to CPU request	X'CC'	
- Positive response to operator initiated pulse counter control request	X'DA'	d ₁ =counter control request d ₂ =control definition
- Pulse count transaction for CPU	X'ED'	
- Response for pulse counters	X'EF' ²	

¹This is the value of the transaction identifier.

²Diagnostics can be run on a counter or counters for a specific area station while the system is still active with normal customer transactions. If the user wants to save the counter values, a diagnostic code can be issued to route all the counter values to the system. After the counters have been returned to the system, they may be restored to the original values by the user program.

Figure 43 (Part 2 of 2). Transaction Control Byte Usage

The following error codes are assigned:

X'E' 2740 not attached -- The 2740 is not attached to the system, and the 2740 was specified in a user table entry. The transaction-list number in the header identified the incorrect user-table entry.

X'D' 2740 intervention required -- The 2740 requires intervention because it

has power off, is out of paper, or is in improper mode.

X'C' Incomplete transaction -- This transaction is incomplete due to one of several causes:

- Operator aborted the transaction.
- Byte count was exceeded on a repeat transaction.

- Stop loop was executed and transaction was not completed in the time allowed.
- Incomplete communication with a counter for any request.

X'B' 1053 not attached -- The transaction was addressed to an area station that did not have a 1053 attached. This can be due to CPU program problems if the CPU originated the transaction, due to user table problems in the case of message routing with implicit addressing, or due to operator errors in message routing with explicit addressing.

X'A' 1053 not operational -- The addressed station has a 1053 attached, but for some reason it is not operational.

X'9' 2740 overload -- The 2740 was specified in so many transactions that a significant part of 2715 buffering was queued for the 2740 and system operation was affected. In this case, the 2715 will flag transactions with this error code, bypass the 2740, and send them to the processor as priority data. Transactions already on the 2740 queue are not affected and print out at the 2740. When the 2740 queue clears, the system will revert to its normal operation.

X'8' Message-routing overload -- The output queue contained so much of the 2715 buffering that system operation was affected. The 2715 will flag transactions with this code and route them to the processor as priority data. Transactions already on the output queue are handled normally. When the output queue clears, the system will return to normal operation.

X'7' Invalid request from CPU -- The original transaction from the CPU is returned to the CPU due to one of the following causes:

- An improper command.
- Wrong transaction length.
- An invalid value specified from a DEU to set a counter (Pulse Count feature).

X'5' Counter not attached -- The counter specified is not present on the area station (Pulse Count feature).

X'3' Device not operational -- The specified counter is present, but is not operational due to a busy condition in the counter adapter (Pulse Count feature).

Note: In the X'8' and X'9' cases, the user program still has access to the 2740 or 1053 output within normal output limitations. Thus the user may reroute this traffic under control of his program as he wishes.

It is the user's responsibility to restore the zone bits in the first and second bytes of the time-stamp field whenever he detects an error if he wishes to restore the time field to true EBCDIC representation (for example, if he were to reroute the transaction).

Data with 2798 Transactions

The first byte of data of every step in a transaction from a 2798 indicates whether the monitor key was on or off at the 2798 from which the transaction was entered. This monitor key byte is either X'FA' indicating monitor key off, or X'FB' indicating monitor key on. Following the monitor key byte can be a data field containing from 0 to 16 data characters received from the 2798 GDU. When the user is analysing a 2798 transaction, he can separate each step by comparing for a X'FA' or X'FB' (or both) in the transaction depending upon whether or not he expects the monitor key to be on or off.

Data with Counter Control

The first byte of data is the counter control byte indicating the type of pulse counter operation this transaction results from, as shown in Figure 44. The counter control byte can be the response to any of the read functions (Read, Read Residual, Read and Reset, Read and Set, Read Group), or one of the following operations:

<u>Counter Control Operation</u>	<u>Counter Control Byte (in Hex)</u>
Scheduled Readout	F4
No-Count Test Failure	F2
Unassigned Production Test Failure	F1
Overflow Interrupt	F6
Power Interrupt	F7
Invalid Transaction from AS	F0

The second byte of data always contains a blank character (X'40'). The third and fourth bytes contain the address of the counter that the operation resulted from. In the case of the Read Group operation, the third and fourth bytes contain the address of the first counter in the group. The fifth byte of data is another blank character. The next five bytes contain the counter value. Except for the following operations, there is no more data in the data area.

Counter Control Operation	Counter Control Byte (in hex)
Set no-count testing (NCT), reset unassigned production testing (UNASP)	01
Reset NCT, set UNASP	02
Reset all count testing functions	03
Set counter	20
Set counter, set NCT, reset UNASP	21
Set counter, reset NCT, set UNASP	22
Set counter, reset all count testing functions	23
Read counter	80
Read counter, set NCT, reset UNASP	81
Read counter, reset NCT, set UNASP	82
Read counter, reset all count testing functions	83
Read residual	A0
Read residual, set NCT, reset UNASP	A1
Read residual, reset NCT, set UNASP	A2
Read residual, reset all count testing functions	A3
Read and reset counter	C0
Read and reset counter, set NCT, reset UNASP	C1
Read and reset counter, reset NCT, set UNASP	C2
Read and reset counter, reset all count testing functions	C3
Read and set counter	E0
Read and set counter, set NCT, reset UNASP	E1
Read and set counter, reset NCT, set UNASP	E2
Read and set counter, reset all count testing functions	E3
Read group	88
Read group residual	C8
Disable schedule readout (single counter)	B0
Enable schedule readout (single counter)	B1
Disable schedule readout and count testing (all counters on an AS)	B2
Enable schedule readout and count testing (all counters on an AS)	B3
Disable all schedule readouts (on 2790 System)	B4
Enable all schedule readouts (on 2790 System)	B5
Disable all count testing functions (on 2790 System)	B6
Enable all count testing functions (on 2790 System)	B7
Disable all schedule readouts and count test functions (on 2790 System)	BB
Enable all schedule readouts and count test functions by continuing from stop point	BC
Enable all schedule readouts and count test functions by reinitializing all schedules	BD

Figure 44. Counter Control Operation

	<u>Operation</u>	<u>Message</u>
• Read Group.		
• Read Group Residual.	No-Count Test Failure	NCTF
• No-Count Test Failure	Unassigned Production Test Failure	UPTF
• Unassigned Production Test Failure.	Overflow Interrupt	OVFL
• Overflow Interrupt.	Power Interrupt	POWER

- Power Interrupt.

CONTROL TRANSACTIONS

For the Read Group and Read Group Residual operations, all the remaining counter values are contained in the data area, and each is separated by a blank character (X'40'). For other operations, the following EBCDIC messages are in the data area, preceded by a blank character:

Control transactions are formatted the same as other types of transactions, using the transaction control byte of the transaction header to indicate the action to be performed. The data field of the message is used to identify the specific object of the action, for example, the identification number of the area station to be restored (the data field may or may not be present,

depending on the nature of the transaction control type).

The control transaction types, as presently defined, are:

- System/370 to 2715 (sent by the user program):
 - Bypass area station.
 - Restore area station.
 - Bypass segment.
 - Restore segment.
 - Stop 2790 input.
 - Start 2790 input.
 - User table load start.
 - User table load data.
 - User table load end.
 - CPU restart.
 - 2715 restart.
 - Sort area station errors.
 - Read partial error log.
 - Reinitialize disk.
 - Set day stamp.
 - Monitor day number.
 - Monitor time.
 - Reset deferred data mode.
 - Read deferred data.
 - All pulse count transactions.
 - Alarm.
 - Text.
 - Alarm and Text.
- 2715 to System/370 (sent to user program):
 - Positive response to CPU request.
 - Negative response to CPU request.
 - Response to invalid CPU request.
 - Positive response to 2715 request.
 - Negative response to 2715 request.
 - Response to invalid 2715 request.
 - User defined.
 - Automatically initiated response.
 - CE-initiated response.
 - Unsolicited 2715 response.
 - Pulse count responses to 2715 operator-initiated requests.
 - Pulse count transactions destined for CPU
 - Response for pulse counters.

In addition to the above transactions, there are two types of messages that are transparent to the user (that is, non-user data).

1. Error records are recorded by BTAM on a disk file; and
2. Diagnostic information (automatic or resulting from intervention at the 2715 local) is printed by BTAM on the System/370 Console or the 2740, if available. Diagnostic information from the 2715 remote goes to the 2740 Data Communications Terminal.

Pulse Count Transactions

All pulse count transactions initiated from a System/370 have a control byte of X'FC' in byte 1 of the 8-byte transaction header. Byte 2 contains the area station address and byte 3 the counter address (in hexadecimal). Particular kinds of pulse counter operations are specified in the transaction text or data. The first byte of the transaction text is the counter control byte. This byte specifies the counter operation requested. Only one data byte (the counter control byte) is required for all counter operations except the Set functions, the Read and Set functions, and the Read Group functions.

For the Read Group and Read Group Residual operations, two additional data bytes must follow the counter control byte. These two bytes are the EBCDIC value of the last counter in the group. The upper limit of the last counter is 63, since there can be only 63 counters on a single area station.

For the Set functions and the Read and Set functions, five additional data bytes must follow the counter control byte. These five bytes contain the EBCDIC value to which the counter is to be set. The value must be between 0 and 29,999 in EBCDIC. These five additional data bytes are required for the following operations:

- Set Counter
- Set Counter and Set No-Count Testing and Reset Unassigned Production Testing
- Set Counter and Reset No-Count Testing and Set Unassigned Production Testing
- Set Counter and Reset all count testing functions
- Read and Set
- Read and Set and Set No-Count Testing and Reset Unassigned Production Testing
- Read and Set and Reset No-Count Testing and Set Unassigned Production Testing
- Read and Set and Reset all count testing functions

The counter control operations and the hexadecimal representation of the counter control bytes are shown in Figure 44.

Overflow Interrupt

An overflow interrupt message is transmitted to the user-defined routing indication (specified in the ASCTR macro instruc-

tion) whenever any counter reaches a value of 30,000.

Power Interrupt

The reporting of initial power-up or power failure at an area station results in a power interrupt message being transmitted to the user-defined routing indication (specified in the ASCTR macro instruction). Until the power interrupt is reported from the area station, all counter transactions will be incomplete transactions.

EXTERNAL ALARM CONTACT FEATURE

The Area Station External Alarm Contact feature is provided as a method of alerting the operator at the area station level that an alarm condition exists in his area. This feature on a 2791-1 or 2793-1 Area Station allows the attachment of an external device at the area station 1053 printer, which can make use of a contact closure to operate some kind of external alarm whenever the EBCDIC character for BELL (X'2F') is received at the area station 1053 printer.

Three types of alarm messages can originate from either the System/370, the 2740 attached to the 2715, or an area station or data entry unit. The three types of messages are:

1. Alarm
2. Text
3. Alarm and text

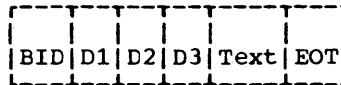
The alarm message causes the 2791/2793 alarm hardware to be activated. The text message consists of data that is printed on the 1053 printer. The alarm and text message consists of data that causes the 2791/2793 alarm hardware to be activated and that causes the data to be sent to the 1053 printer. If the 1053 is not available, alarm or alarm and text messages are routed to the CPU. The 2791/2793 alarm hardware is activated for the alarm or alarm and text messages whether or not the 1053 printer is available. Text messages initiated at the System/370 or 2740 must be supplied by the user with the transaction request. Area station and data entry unit requests may have text supplied as explicit or implicit text.

The alarm messages initiated from the System/370 are handled as normal System/370 to area station 1053 printer output messages. The transaction control byte defines the type of alarm message, as follows:

<u>Control Byte Value</u>	<u>Alarm Message Type</u>
X'CD'	Text
X'CE'	Alarm
X'CF'	Alarm and text

The data can be any normal user data. For alarm or alarm and text messages, the 2715 generates the alarm character to send to the 1053 (the user does not have to do this).

The alarm message initiated from the 2740 is handled as a special control request. This request must be coded as follows:



where:

BID is the 2740 BID key.

D1 is the type of request and can have the following values:

- Y for alarm message.
- Z for text message.
- X for alarm and text message.

D2 and D3 represent the area station address to which the message is to be sent (decimal 00 to 99).

Text is any user text up to 127 characters.

EOT is the 2740 EOT key.

Messages originating at an area station or data entry unit are initiated by an input transaction in conjunction with the 2715 user tables. The implicit or explicit area station address, if other than the area station address of the originating station, indicates that two messages will be created by the 2715. One message will be the alarm message that will be sent to the area station that initiated the transaction. The second message will contain the data to be routed to some other area station 1053 printer. This second message will not be an alarm message. If, however, the user desires to send alarm and text to another area station, the first two characters of the text must be the alarm, text, or alarm and text characters. If the implicit or explicit area station address is the address of the transaction initiator, then only an alarm message will be generated by the 2715. Message routing and implicit text will be specified in the transaction list header for messages originating at an area station or data entry unit. The first two bytes of implicit text must define the type of alarm message desired. For alarm messages, these bytes will be deleted from text sent to the 1053. The following table shows the format of the first two bytes of the implicit text for alarm messages. Implicit text is specified in the 2715 user tables in the MSG operand of the last ASLIST or DEULIST macro or the

CTRLIST macro in a defined transaction. The user must multipunch a 0-7-8-9 sequence for the EBCDIC BELL character specified in the table. The SPACE character indicates that nothing is punched in this column of the card.

<u>Message Type</u>	<u>First Byte</u>	<u>Second Byte</u>
Alarm	BELL	SPACE
Text	SPACE	BELL
Alarm and Text	BELL	BELL

2740 TRANSACTIONS

The 2740 Communications Terminal is a standard feature on the 2715 remote and an optional feature on the 2715 local. It is used with the 2715 as both an input (inquiry, control) and output (response, error logging) device.

A message originated by the System/370 and destined for the 2740 terminal has a maximum length of 128 bytes (two-byte message header, eight-byte transaction header, 118 bytes of text). These messages have the following format:

- Bytes 1 and 2: Message header (these bytes are not printed on the 2740).
- Bytes 3 through 10: Transaction header (Bytes 4, 5, and 6 are printed in hex.

Following these six printed characters [two for each byte] is a space. Bytes 7 through 10 are printed as they appear in main storage. There will be a total of 11 characters printed, including the space.)

- Data: The maximum length of the actual data text is 118 bytes if the margins of the 2740 are set to maximum printing space.

A message entered from the 2740 may be formatted by the 2715 as a normal 2740-initiated request and routed directly to the System/370 user or will be treated as a control request. The 2715 will format a standard eight-byte transaction header, inserting the first character entered from the keyboard in the control byte of the header. If this byte is numeric, a four-byte time stamp will be added, subsequent characters from the 2740 will be inserted as text, and the message will be routed to the System/370.

If the first character entered from the 2740 is not numeric, the 2715 will not add a time stamp and will treat the message as a control request.

SYSTEM CONFIGURATION

An IBM 2770 can communicate with a System/370 over a nonswitched line (point-to-point or multipoint) or a switched line. The 2772 Control Unit must be equipped with the Multipoint Data Link Control feature for use on a multipoint line. A control unit for use on a switched line can be equipped with an Automatic Answering feature, if desired.

TRANSMISSION CODES

The IBM 2770 communicates with the System/370 using either of two transmission codes, EBCDIC or USASCII, as selected when the 2770 is ordered. If the 2770 is equipped with the EBCDIC Transparency feature, text data can contain any of the 256 EBCDIC bit patterns. That is, when text data is sent in transparent mode, the EBCDIC bit patterns representing data link control and terminal control characters are treated simply as data, and do not cause the control functions usually effected by these bit patterns to occur. This feature allows transmission of various kinds of raw data, such as packed decimal numbers, floating-point numbers, and machine-language programs. When transmission is in nontransparent mode, however, the data link and format control characters are recognized as such, and thus cannot appear as normal text.

TERMINAL POLLING AND SELECTION

In order to activate a terminal so that data transmission can occur, the central computer transmits on the communications line a specific character sequence that identifies the input or output component (and in the case of multipoint lines, the terminal as well) from which data is to be received, or to which data is to be sent. This procedure is called polling when an input device is involved, and selection when an output device is involved; the character sequences are called polling sequences and selection sequences. Specific polling and selection sequences are assigned to 2772 Control Unit input and output adapters, rather than to specific device types, as is the case with some other terminal types. The specific adapter-to-device correspondence is established by the customer engineer when the 2770 system is installed. You must be aware of the correspondence in order to

select correctly the character sequences you need. These sequences are given below.

Point-to-Point

In point-to-point communication, you may perform component selection by one of two methods. The output device may be selected by the Job Select Switch on the 2770 operator control panel, or it may be selected by the transmission of a device control character. If more than one device is assigned by the Job Select Switch, the device control character is mandatory. DC1, DC2, and DC3 are the device control characters for output devices attached to output adapters 1, 2, and 3.

Device control characters for point-to-point lines may be sent as separate message blocks, or accompanied by text data, in the format STX DCx [text] ETB, or STX DCx [text] ETX. You code this message block in the output area referenced by a WRITE macro instruction. A device control character can be sent only as a nontransparent block, and it must be the first block of a message, that is, following EOT or the first block of conversational reply.

The polling function is not used for a 2770 on a point-to-point line, since message transmission from terminal to computer is initiated only by the terminal.

Multipoint

Message transmission between computer and 2770 via a multipoint line is initiated only by the computer, using a Read Initial or Write Initial operation. You code polling and selection sequences in terminal lists (called polling lists for polling sequences, and addressing lists for selection sequences). (The terms selection and addressing are used here synonymously.) The DFTRMLST macro instruction is used to create the terminal lists. The READ or WRITE macro instruction that initiates message transmission sends the polling or selection sequences contained in the list.

Polling and selection sequences consist of four characters. The first is the terminal address, which may be any alphabetic character; it identifies an individual terminal and is set by the customer engineer when the terminal is installed. In a polling sequence, this character must appear in uppercase, for example, A. When in a

selection sequence, it must be lowercase, for example, a.

The second character is always identical to the first. The third character in the sequence is a component polling or selection character. The characters DC1, DC2, and DC3 select the output devices attached to output adapter 1, 2, and 3, respectively. The characters 5, 6, and 7 poll the input devices attached to input adapters 1 (keyboard), 2, and 3; 0 causes a general poll, resulting in receipt of data from any ready input device.

The fourth character in the sequence is always ENQ (inquiry), which elicits a response from the terminal control unit that indicates whether the polled or selected component is ready.

TEMPORARY TRANSMISSION DELAYS

In communication between the IBM 2770 and the central computer, message transmission may need to be delayed because of conditions at the 2770. The 2770 signals the central computer that delay is necessary by sending one of several data link control sequences, the specific one depending on the reason for the delay. These sequences, and the automatic BTAM response or appropriate user program response to each, are as follows.

2770 Unable to Send (STX ENQ)

When the 2770 is unable to send (STX ENQ) during transmission of text from the 2770 to the central computer, the 2770 becomes temporarily unable to transmit. It sends a TTD sequence (STX ENQ) in lieu of text. Upon receiving this sequence, BTAM automatically transmits a NAK character. Transmission of STX ENQ and NAK alternates until (1) the 2770 once again is able to transmit, in which case it resumes transmitting text, or until (2) the BTAM retry count including errors as well as TTD's reaches seven, or (3) until EOT is transmitted by the 2770 (STX ENQ followed by EOT is a forward abort sequence). In cases 2 and 3 BTAM turns on bit 7 of DECFLAGS and posts a completion code of X'41'. When a X'41' is posted with bit seven set in DECFLAGS, the application should transmit EOT and reestablish contact at a later time. (If the source of the incoming data is an IBM 50 Magnetic Data Inscrubber cartridge, it may be desirable for the user program to send NAK characters until a total of about 60 seconds has elapsed from the time the STX ENQ sequence was received. The MDI cartridge requires 45 seconds to rewind, and several seconds are required for the terminal operator to mount a new cartridge so that transmission can resume.)

2770 Unable to Receive Text (WACK)

At the time the 2770 receives an ENQ or selection characters from the central computer, or after it has already received one or more message blocks, it may be unable temporarily to receive text into the buffer. This condition occurs when the current contents of the buffer are being transferred to an output device. When this happens, the 2770 sends a WACK sequence instead of the usual alternating acknowledgment.

Upon receiving the WACK, BTAM turns on bits 0 and 1 of DECFLAGS and posts a completion code of X'7F'. The user program should check DECFLAGS for this response, and if WACK was received, should send an ENQ character (as by a WRITE Inquiry macro). The ENQ should be sent regardless of whether the WACK was received in response (1) to text or (2) to the initial ENQ character (nonswitched point-to-point line) or ID ENQ sequence (switched point-to-point line). For a multipoint line, however, the ENQ should be sent only if the WACK was received in response to text. If it was received in response to initial selection, the user program should resend the selection characters, that is, reissue the WRITE macro. To determine whether the WACK was received in response to text or selection, examine the DECTPCOD field of the DECB. If it contains X'06', the WACK was received in response to selection. If it does not contain X'06', the WACK was received in response to text.

The user program should be arranged to keep responding to WACK sequences in this manner until the 2770 responds normally or until the user program wishes to abandon communication with the 2770 for the time being. In the latter case, the user program should issue the appropriate macro to break off transmission.

2770 Wishes to Transmit (RVI)

At the time the 2770 receives a selection sequence from the central computer (multipoint line only), it may signal the central computer that it wishes to transmit instead of receive. To do this, it sends an RVI sequence instead of an ACK-0 (the normal response to selection) BTAM accepts the RVI in lieu of the ACK-0, turns on bits 1 and 6 of DECFLAGS, and posts a completion code of X'7F'. The user program should check DECFLAGS for this response and proceed as follows.

When the RVI is received in response to selection, the program should issue a READ Initial macro if it wishes to allow the 2770 to transmit.

TERMINAL FUNCTION CONTROL

There are six characters in each code (EBCDIC and USASCII) that control terminal functions. These are sometimes referred to as end-to-end control characters (as distinguished from data link control characters).

- **EM (end-of-medium)**
This character is used to indicate the end of data on paper or magnetic tape. It is transmitted as data and reproduced in paper tape at the 1018 paper tape punch. (It is not sent to the IBM 50 Magnetic Data Inscrber, as this is an input device only.)
- **IRS (Interchange Record Separator) (EBCDIC)**
RS (Record Separator) (USASCII)
This character is used to indicate the end of data in a punched card. When the contents of a card are read into the buffer, the control unit inserts an IRS (RS) character into the buffer following the last data character read from the card. If the contents of a buffer are sent to the paper tape punch, the IRS (RS) characters are also punched in the tape, so that cards can be punched from the tape. When sending data from the buffer to the card punch or printer, each IRS (RS) character encountered in data causes the control unit to command the card punch to eject a card, or the printer to perform the new line function.
- **NL (New Line)**
The NL character defines a print line when data is to be printed. If data containing NL characters is sent to a card punch or paper tape punch, the NL characters are punched.
- **DC1**
DC2 (Device Control)
DC3
These characters are used to activate specific devices attached to the 2772 control unit. Their use is explained under Terminal Polling and Selection.
- **ESC (Escape)**
This character and a defined graphic character that follows it are called an escape sequence. Escape sequences are used to control formatting of data on output devices, as explained under 2213 Printer and 2265 Model 2 Display Station.
- **VT (Vertical Tab)**
FF (Forms Feed)
These two characters are used to control formatting on the 2213 printer, as explained under 2213 Printer.

Placing terminal function control characters in message text is not a BTAM function; they must be placed there by the terminal operator, programmer, or preparer of input media (for example, cards, tape).

2213 Printer

Vertical forms control for the printer may be regulated by a carriage control tape contained within the printer, or by control commands consisting of escape sequences (ESC followed by a defined character). An escape sequence specifies the number of line spaces to be skipped following printing of the line in which the escape sequence appears, or specifies the channel number of the carriage control tape that is to govern forms motion. The escape sequence must be contained in the first two positions of a record sent to the printer. These two characters are not printed on the forms. The escape sequences and their corresponding functions are given in Figure 45.

EBCDIC Sequence	USASCII Sequence	Forms Motion After Printing	Skip to Carriage Control Tape Channel
ESC /	ESC Q	Single space	
ESC S	ESC R	Double space	
ESC T	ESC S	Triple space	
ESC A	ESC A		1
ESC B	ESC B		2
ESC C	ESC C		3
ESC D	ESC D		4
ESC E	ESC E		5
ESC F	ESC F		6
ESC G	ESC G		7
ESC H	ESC H		8
ESC I	ESC I		9
ESC J	ESC J		10
ESC K	ESC K		11
ESC L	ESC L		12
ESC M	ESC M	Space suppress	

Figure 45. IBM 2213 Vertical Forms Control Escape Sequences

Vertical forms control may also be actuated by the Vertical Tab (VT) character, which causes skip-to-channel-2 of the carriage control tape and the Forms Feed (FF) character, which causes skip-to-channel-1. These two characters differ in effect from the escape sequences in that forms motion takes place immediately upon detection of the VT or FF character, whereas forms motion caused by detection of an escape sequence does not occur until the entire line containing the escape sequence has been printed.

2265 Model 2 Display Station

Two 2265 Model 2 Display Station control functions are activated by two-character escape sequences contained within the message data sent to the display station via a Write operation. These functions and their associated escape sequences are:

Erase/Write (ESC U)

Write at Line Address (ESC ').

The escape sequence must be the first two characters following the STX character that begins a message or message block. Both the ESC U and ESC ' sequences may be contained in a single message (though not in the same block).

Erase/Write: To erase the screen of a display station the station must be selected and the program must send:

STX ESC U (text) ETX or

STX ESC U (text) ETB.

The screen is erased, the cursor is positioned at the first available display position of the screen (upper left corner), and the data represented by (text) is displayed.

Write at Line Address: This control function allows the program to select a specific line where the data containing the escape sequence is to be displayed. The program must send:

STX ESC ' x (text) ETX or

STX ESC ' x (text) ETB

where x represents the line address. The line address is a hexadecimal code specifying the display line where the message data is to begin. Display line numbers and corresponding line addresses are shown in Figure 46.

Function	Escape Sequence	
Erase Screen	ESC U	
Erase screen and display message	ESC U (text)	
Write at Line Address	ESC ' x (text)	
Display Line Number	<u>Address Code</u>	
	15 lines	12 lines
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	A	A
11	B	B
12	C	C
13	D	
14	E	
15	F	

Figure 46. IBM 2265 Erase and Write-at-Line Address Control Characters

Conversational Mode

This special feature enables the 2772 to accept a text response to an inquiry without having to be selected before receiving the response. With this feature, the user may include in his BTAM program the coding required to initiate a Write Continue operation to the 2770 immediately following the last block of data received from the same 2770 on a Read operation. This Write Continue operation may be followed by other Write Continue operations to the same 2770. To read more data from the 2770, the user must issue another READ Initial macro to poll the input unit again.

FIELD-CONTROL OPERATION

This special feature permits operator or program entry of three field modifiers that can be entered individually in any given character location in the display buffer.

Protected Data

Two of the three field modifiers provided by the Field-Control Operation special feature serve to identify the beginning and

end of a field of data that is to be protected. They are:

- Protected-Data-Field Modifier (ESC Z): The presence of this modifier in the display buffer identifies the start of a field of protected data and prevents manual erase or over-write of the data.
- End-Field Modifier (ESC 9): The presence of this modifier in the display buffer identifies the end of a protected-data field and automatically terminates the field-control operation.

These field-control modifiers are entered into the system via escape (ESC) sequences. The characters Z and 9 are stored in the display buffer and they are reproduced on the display screen as:

Protected Data Field Modifier --
(End-Field Modifier --)

When the (and) symbols are not preceded by ESC, they may be used as normal data. During a Read or Write operation, if the display cursor encounters a Protected-Data-Field Modifier it moves over the field until it reaches the End-Field Modifier. The cursor then locates in the next display position beyond the End-Field Modifier and normal Read or Write operation continues.

Tab Set

The presence of the Tab Set Character Field Modifier in the display buffer identifies the position as a Tab Set character location. When a Horizontal Tab (HT) is received from the central computer or from the keyboard, the HT character is stored in the first unprotected character space, and then the cursor automatically advances to one character space beyond the next Tab Set character.

The Tab Set Character Field Modifiers are entered into the system via a format message. The first data in this message should be ESC HT, to set up a tab-set sequence. Each HT following this represents a Tab Set character. A vertical bar is displayed in each line from and including the line containing the cursor to the bottom of the screen for each Tab Set character. This character cannot be written within a protected field. Caution must be exercised to see that the cursor is not positioned in a protected field when a Tab Set character is to be written. The tab-set sequence is not terminated until the New Line (NL) character is entered.

Once this format message has been stored, messages may be transmitted without spaces, as they need contain only the Hori-

zontal Tab (HT) character to provide formatting. The HT character is stored and will be read back to provide printer formatting and/or better communications line efficiency. If an HT is sent and there are no Tab Field Modifiers, the cursor is positioned at the beginning of the next line.

RECORD FORMATS

The basic 2772 control unit has two 128-byte buffers. As a special feature the 2772 can have buffers of 256 bytes each. The basic 2772 can send or receive one message block per transmission. Thus, in non-transparent mode, messages appear on the line in the format:

```
-----  
|STX|...text...| ETB (or ETX) |  
-----
```

In transmission to the 2772, only the text portion of the message is transferred to the output device. The definition of text depends on the type of device.

For the paper tape punch, text consists of the data characters and any end-to-end control characters present in the data. For the card punch, any escape sequences, NL and EM characters are considered text and do not cause ejecting of a card from the punch; the IRS (RS), ETB, and ETX do cause card ejection. For the printer, the NL and IRS (RS) characters and escape sequences are not considered part of text.

In nontransparent mode, the maximum record length is 128 characters for the basic 2772 and 256 characters for the 2772 with the Expanded Buffer feature. The STX, ETB, ETX, and the device control characters (DCx) do not go into the buffer. Records exceeding the buffer size cause an I/O buffer overrun error, which causes the 2772 to send a NAK in response to the received block. BTAM will retry the Write operation seven times, then post the operation complete with error.

Each IRS (RS) character in data transferred from the buffer to the card punch causes the card currently being punched to be ejected and a new card fed. This action also occurs if no IRS (RS) character has been detected by the time 80 consecutive data characters have been sent to the punch.

Printing: Data to be sent to the printer may be formatted into print lines of 132 characters or less by the use of IRS (RS)

or NL characters. If neither of these characters is detected by the time 132 characters have been sent to the printer, successive data is printed on the next line. The new line function also occurs if the printer reaches a tab stop.

Display: Records exceeding the length of the display line are not truncated, but are continued on the next display line. In nontransparent mode, variable length records may be sent to the 2772. The number of records per transmission is not restricted except by buffer size. The STX, ETB, ETX, and DCx characters do not enter the buffer. All other characters, including escape sequences and end-to-end control characters, occupy positions in the buffer. In transparent mode, variable length blocks may be sent to the 2772. A block consists of one record, since end-to-end controls are not recognized in transparent mode. The length of the block may not exceed the buffer size.

Transmission of Blank Cards

Basic 2772: In either transparent or nontransparent mode, blank cards are read into the buffer and transmitted just as are cards containing data.

2772 with Expanded Buffer Feature: In nontransparent mode, data from the card reader is packed. That is, each card is read into the buffer, then scanned from column 80 backward until a data character is reached. The control unit then inserts an IRS (RS) character in the buffer at the next position. The next card is read into the buffer beginning at the next following position. Thus, card definition is maintained while unnecessary blanks at the end of the card are deleted. For this reason, the 2772, when equipped with the Expanded Buffer feature, does not transmit blank cards, in nontransparent mode. In transparent mode, data is not packed in the manner indicated above, and blank cards are transmitted.

Communication between the central computer and the station control unit of an IBM 2972 General Banking Terminal system employs an eight-bit transmission code and BSC multipoint data link control procedures. The makeup of the character set is as follows.

BSC Data Link Control Characters

The transmission code bit patterns for the data link control characters are identical to the EBCDIC bit patterns for the same characters. The 2972 station control unit can send and can receive and respond functionally to, these data link control characters and sequences:

<u>Character</u>	<u>Bit Pattern (Hex)</u>
STX	02
ETX	03
DLE	10
ETB	26
ENQ	2D
SYN	32
EOT	37
NAK	3D
ACK-0	1070
ACK-1	1061
WACK	106B
RVI	107C

The 2972 does not send the SOH (X'01') and ITB (X'1F') characters. However, it can receive them, but does not respond functionally to them. (This provides compatibility with other types of remote BSC stations that may be attached to the same multipoint line.)

Graphic and Terminal Function Control Characters

Graphic characters are the alphabetic and numeric characters and the special symbols that can be printed on the 2980 teller and administrative stations, or that these stations can send to the central computer. HT (horizontal tab), NL (new line), and Pass-book Index are examples of terminal function control characters. The 2972 station control unit passes graphic and control characters between the communications line and the 2980s connected to the station control unit.

The character sets for the different models of the 2980 vary in the specific characters they include and in the individual transmission code bit patterns that represent the characters. For example, the bit pattern X'D3' represents a 6, in numeric shift, and I, in alphabetic shift, for the 2980 Model 1. For the Model 4, however, the same bit pattern, X'D3', represents I, in numeric shift, and Q, in alphabetic shift.

BTAM does not provide translation tables for user-program translation between EBCDIC and transmission codes. Appendix K shows the correspondence between each transmission code bit pattern and the characters that bit pattern represents, for each of the models (1, 2, and 4) of the 2980 stations.

IBM 3270 INFORMATION DISPLAY SYSTEM -- PROGRAMMING CONSIDERATIONS

The control units, display stations, and printers that make up the IBM 3270 Information Display System are supported by BTAM under control of OS/VS. Support for local or remote 3270 display systems or both can be included in BTAM. For helpful information on programming the 3270, see Introduction to Programming the IBM 3270, GC27-6999.

3270 DEVICES SUPPORTED

BTAM supports the following remote 3270 control units and their attachable devices:

- . 3271 Control Unit, Models 1 or 2
- . 3274 Control Unit Model 1C (BSC)
- . 3275 Display Station
- . 3276 Control Unit Display Station

The remote 3270 control unit must be attached to either a 2701 Data Adapter Unit or 2703 Transmission Control Unit or 3704 or 3705 in emulation mode.

BTAM supports the following local 3270 Control Units and attachable devices:

- . 3272 Control Unit, Model 1 or 2
- . 3274 Control Unit, Model 1B or 1D

The local 3270 control unit must be attached to a selector, multiplexer, or block multiplexer channel.

Throughout this publication, local 3270 refers to a 3272 Model 1 or 2 Control Unit or to a 3274 Model 1B or 1D Control Unit with attachable devices. A remote 3270 refers to a 3271 Model 1 or 2 Control Unit, 3274 Model 1C Control Unit, 3275 Display Station, or 3276 Model 1, 2, 3, 4 Control Unit with attachable devices. Except where noted, references to the 3272 also apply to the 3274 Model 1B and 1D. References to the 3271 also apply to the 3274 Model 1C and the 3276. Where functions apply to some 3270 devices and not others, these exceptions are noted. The following summarizes these exceptions:

FUNCTION	APPLIES TO	
	Local	Remote
Alternate Buffer Size	3274 Models 1B and 1D	3274 Model 1C
Write Structured Field	3274 Model 1D	3274 Model 1C (with Write Structured Field capability)
Prepare to Read	3274 Model 1D	None.

The following 3270 control units are supported by BTAM as 3272 and 3271 Control Units:

<u>CONTROL UNIT</u>	<u>ATTACHMENT</u>	<u>COMPATIBLE WITH</u>
3274 1B, 1D	Local	3272
3274 1C	BSC	3271
3276 1,2,3,4	BSC	3271

For more information about remote and local 3270 configurations, see IBM 3270 Information Display System, Library Users Guide.

3270 CAPABILITIES SUPPORTED

BTAM supports the following remote and local 3270 capabilities:

- Read modified fields from device buffer
- Write to device buffer
- Erase and write to device buffer
- Transmit structured field data to a device
- Erase and write alternate size device buffer
- Erase all unprotected fields in device buffer
- Read modified fields from device buffer from position
- Read from device buffer
- Read from device buffer from position

In addition, BTAM supports the remote 3270 capability of copying from the buffer of one remote device into the buffer of another remote device on the same control unit.

A remote 3270 display system can consist of the remote 3270 control unit and devices attached to nonswitched multipoint BSC lines or the 3275 station equipped with the Dial feature attached to switched point-to-point lines or both. Nonswitched and switched systems are discussed separately in this chapter.

For remote 3270 display systems, capabilities are used through a combination of BTAM READ and WRITE macro instructions for

nonswitched multipoint BSC stations (or switched point-to-point BSC stations for the 3275 with dial feature) and data link and end-to-end control characters in output messages. For local 3270 display systems, capabilities are used through local types of BTAM READ and WRITE macro instructions (specified by means of the otype operand).

BTAM provides additional capabilities for application programs to use the large (alternate) display or printer size of the devices attached to the 3274 1B, 1C and 1D and the 3276 1,2,3,4 Control Units.

BTAM supports the Prepare to Read operation in OS/VS1 and OS/VS2 MVS for the 3274 Model 1D Control Unit. Prepare to Read permits overlap of host and control unit activity and also frees both the channel and control unit resources. For more information refer to IBM 3270 (Local) Information Display System -- Programming Considerations, Prepare to Read Operations in this book.

BTAM supports the Write Structured Field operation in OS/VS1 and OS/VS2 MVS for the 3274 Model 1D Control Unit and the 3274 Model 1C Control Unit. Write Structured Field allows application programs to transmit structured fields to those devices.

For more information about remote and local 3270 capabilities, see IBM 3270 Information Display System, Library Users Guide.

NONSWITCHED REMOTE 3270 DISPLAY SYSTEM

Nonswitched remote 3270 display stations and printers are supported by BTAM as BSC stations connected to nonswitched multipoint lines using either EBCDIC or ASCII transmission code.

Control Unit Number	Address Character	EBCDIC Hexadecimal Representation	ASCII Hexadecimal Representation
0	SP	40	20
1	A	C1	41
2	B	C2	42
3	C	C3	43
4	D	C4	44
5	E	C5	45
6	F	C6	46
7	G	C7	47
8	H	C8	48
9	I	C9	49
10	¢ ([)	4A	5B
11	.	4B	2E
12	<	4C	3C
13	(4D	28
14	+	4E	2B
15		4F	21
16	&	50	26
17	J	D1	4A
18	K	D2	4B
19	L	D3	4C
20	M	D4	4D
21	N	D5	4E
22	O	D6	4F
23	P	D7	50
24	Q	D8	51
25	R	D9	52
26	! (])	5A	5D
27	S	5B	24
28	*	5C	2A
29)	5D	29
30	;	5E	3B
31	—]	5F	5E

Figure 47. Control Unit Addresses for Polling List Entries and for Identification in Input Messages (with Hexadecimal Representations in EBCDIC and ASCII)

DEFINING COMMUNICATIONS LINE GROUPS

See "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System," and see Appendix D.

The UNIT operand of the IODEVICE system generation macro instruction must specify BSC3 for the nonswitched 3270 display system. The DCB macro instruction operands that apply to the nonswitched 3270 display system are: DSORG, MACRF, DDNAME, BUFNO, BUFL, BUFCB, EXLST, BFTEK, LERB, EROPT, DEVD, MODE, and CODE.

DEFINING AND MODIFYING TERMINAL LISTS

See "Defining and Modifying Terminal Lists" in the general section "Defining the Teleprocessing System," see "Defining Terminal Lists" under the heading "BSC Nonswitched Multipoint Operation" in the section "BSC Read and Write Operations," and see Appendix A.

Control Unit Number	Address Character	EBCDIC Hexadecimal Representation	ASCII Hexadecimal Representation
0	-	60	2D
1	/	61	2F
2	S	E2	53
3	T	E3	54
4	U	E4	55
5	V	E5	56
6	W	E6	57
7	X	E7	58
8	Y	E8	59
9	Z	E9	5C
10	(\)	6A	7C
11	,	6B	2C
12	%	6C	25
13		6D	5F
14	Σ	6E	3E
15	?	6F	3F
16	0	F0	30
17	1	F1	31
18	2	F2	32
19	3	F3	33
20	4	F4	34
21	5	F5	35
22	6	F6	36
23	7	F7	37
24	8	F8	38
25	9	F9	39
26	:	7A	3A
27	#	7B	23
28	@	7C	40
29	'	7D	27
30	=	7E	3D
31	"	7F	22

Figure 48. Control Unit Addresses for Selection List Entries (with Hexadecimal Representations in EBCDIC and ASCII)

Defining Terminal Lists

The DFTRMLST macro instruction is used to define terminal lists for the remote 3270 display system.

Each control unit has a one-character polling address (see Figure 47) and a one-character selection address (see Figure 48). Each display station or printer has its own one-character address for specific polling and selection (see Figure 49), and all devices share a one-character address for general polling (see Figure 49). Double addressing is used for both control unit and device. Each five-character polling or selection sequence has the format:

```
[xx|xx|yy|yy|ENQ]
```

where xx is the hexadecimal representation in EBCDIC or ASCII of the control unit address for polling or selection, yy is the hexadecimal representation of the device address, and ENQ is X'2D' for EBCDIC or X'05' for ASCII. In the polling list, the last entry must be five EOT characters (X'37' for EBCDIC or X'04' for ASCII).

This Page Intentionally Left Blank

Device Number	Address Character	EBCDIC	ASCII
		Hexadecimal Representation	Hexadecimal Representation
0	SP	40	20
1	A	C1	41
2	B	C2	42
3	C	C3	43
4	D	C4	44
5	E	C5	45
6	F	C6	46
7	G	C7	47
8	H	C8	48
9	I	C9	49
10	† ([)	4A	5B
11	.	4B	2E
12	<	4C	3C
13	(4D	28
14	+	4E	2B
15		4F	21
16	&	50	26
17	J	D1	4A
18	K	D2	4B
19	L	D3	4C
20	M	D4	4D
21	N	D5	4E
22	O	D6	4F
23	P	D7	50
24	Q	D8	51
25	R	D9	52
26	! (])	5A	5D
27	\$	5B	24
28	*	5C	2A
29)	5D	29
30	;	5E	3B
31	—	5F	5E
General Poll	"	7F	22

Figure 49. Device Addresses for Polling and Selection List Entries and for Identification in Input Messages (with Hexadecimal Representations in EBCDIC and ASCII)

Examples: To define an open polling list for devices 1, 2, and 3 on control unit 7 using EBCDIC transmission code:

```
OPLIST DFTRMLST AUTOLST,(C7C7C1C12D,
C7C7C2C22D,C7C7C3C32D,
3737373737)
```

To define a wraparound polling list for devices 1, 2, and 3 on control unit 7 and a general poll on control unit 8 using ASCII

```
WPLIST DFTRMLST AUTOWLST,(4747414105,
4747424205,4747434305,
4848222205,0404040404)
```

To define a selection list for devices 1, 2, and 3 on control unit 7 using EBCDIC

```
SLIST DFTRMLST OPENLST,(E7E7C1C12D,
E7E7C2C21D,E7E7C3C32D)
```

Modifying Terminal Lists

The CHGNTRY macro instruction is used to modify terminal lists for the nonswitched 3270 display system. (If wraparound polling is being done, the RESETPL macro instruction is used first to terminate polling.)

Example: To suspend the poll on device 3 (from the first example under "Defining Terminal Lists"):

```
SPOLL CHGNTRY OPLIST,AUTOLST,3,5,,SKIP
```

BUFFER MANAGEMENT

See the general section "Buffer Management." Programmer buffering or dynamic buffering can be used for the remote 3270 display system.

CODE TRANSLATION

See the general section "Code Translation," and see Appendix E.

All remote 3270 messages can be translated between EBCDIC and ASCII, except for those transmitted to devices with Write Structured Field capability. The messages sent to Write Structured Field devices are transmitted in EBCDIC because the Write Structured Field operation does not permit a one-to-one translation of the data stream to ASCII code. To allow all other remote 3270 messages to be translated between EBCDIC and ASCII using the BTAM RASA and SASA translation tables, in the I/O interface code for six-bit structured data in all 3270 messages the setting of the two high-order bits is determined by the setting of the six low-order bits in the byte (see Figure 50). Six-bit structured data includes the WCC and CCC, attribute character, cursor and buffer addresses, remote control unit address, remote device address, and sense and status bytes; for more information, see IBM 3270 Information Display System, Library Users Guide.

Examples: Line 1 of Figure 52 represents a message received from a display station using ASCII transmission code. (If the control unit and device addresses are to be checked against the entry in the polling list, this should be done before translation, since the terminal list entries are

00 0000 40 SP	01 0000 50 &	10 0000 60 -	11 0000 F0 0
00 0001 C1 A	01 0001 D1 J	10 0001 61 /	11 0001 F1 1
00 0010 C2 B	01 0010 D2 K	10 0010 E2 S	11 0010 F2 2
00 0011 C3 C	01 0011 D3 L	10 0011 E3 T	11 0011 F3 3
00 0100 C4 D	01 0100 D4 M	10 0100 E4 U	11 0100 F4 4
00 0101 C5 E	01 0101 D5 N	10 0101 E5 V	11 0101 F5 5
00 0110 C6 F	01 0110 D6 O	10 0110 E6 W	11 0110 F6 6
00 0111 C7 G	01 0111 D7 P	10 0111 E7 X	11 0111 F7 7
00 1000 C8 H	01 1000 D8 Q	10 1000 E8 Y	11 1000 F8 8
00 1001 C9 I	01 1001 D9 R	10 1001 E9 Z	11 1001 F9 9
00 1010 4A c	01 1010 5A !	10 1010 6A ;	11 1010 7A :
00 1011 4B .	01 1011 5B S	10 1011 6B ,	11 1011 7B #
00 1100 4C <	01 1100 5C *	10 1100 6C %	11 1100 7C @
00 1101 4D (01 1101 5D)	10 1101 6D _	11 1101 7D '
00 1110 4E +	01 1110 5E ;	10 1110 6E >	11 1110 7E =
00 1111 4F I	01 1111 5F ~	10 1111 6F ?	11 1111 7F "

EBCDIC
Bits
23 4567

EBCDIC
Hex

Graphic
Character

Note: The I/O interface code is obtained by overlaying columns 4,5,6, and 7 of standard EBCDIC code on columns C,D,E, and F.

in transmission code.) Line 2 of Figure 52 indicates the characters contained in the message. Line 3 shows the message after the TRNSLATE macro instruction has been used to translate from ASCII to EBCDIC. Line 4 indicates the content of the message for the problem program.

Line 1 of Figure 53 indicates the content of a message from the problem program. Line 2 shows the message before the TRNSLATE macro instruction has been used to translate from EBCDIC to ASCII. Line 3 indicates the characters contained in the message. Line 4 represents the message to be sent to a display station using ASCII transmission code.

ACTIVATING AND DEACTIVATING THE TELEPROCESSING SYSTEM

See the general section "Activating and Deactivating the Teleprocessing System."

LINE CONTROL AND MESSAGE TRANSMISSION

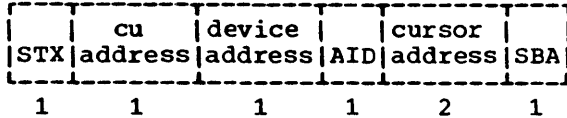
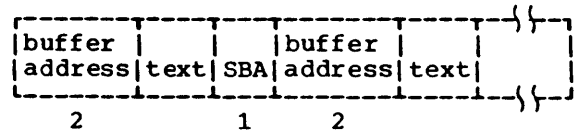
See the general section "Line Control and Message Transmission," see "READ Macro Instructions" and "WRITE Macro Instructions" under the heading "BSC Nonswitched Multipoint Operation" in the section "BSC Read and Write Operations," and see Appendixes B and G.

Read Operations

Read Modified Fields for Operator Input:
To poll a device or control unit and, after some action by the display station operator, to read a message block, use the READ TI macro instruction. The polling list entry either specified by or in the polling list specified by the entry operand determines whether a single device or all devices on a control unit are polled.

Figure 50. I/O Interface Code for Six-bit Structured Data

- If the operator pressed the ENTER key or pressed a PF key, the READ TI macro instruction causes a normal read. If the device buffer was formatted, the buffer specified by the inoutarea operand contains an index byte and a message block with the format:



If the operator selected detectable fields with the selector pen, the message block has the same format except that it contains no text.

TO DO THIS...	USE THESE MACRO INSTRUCTIONS ¹ ...	WITH THESE DEVICE CONTROL CHARACTERS IN THE OUTPUT DATA STREAM...	
		CHAR	HEX (EBCDIC)
Read Modified Fields for Operator Input	READ Initial (TI) Then READ Continue(s) (TT)	Not used	—
Read Modified Fields Independent of Operator Action	WRITE Initial Conversational (TIV)	ESC 6	X'27F6'
	Then READ Continue(s) (TT)	Not used	—
Read Modified Fields from Position ²	WRITE Initial (TI)	ESC 1	X'27F1'
	Then WRITE Continue Conversational (TTV)	ESC 6	X'27F2'
	Then READ Continue(s) (TT)	Not used	—
Read Buffer	WRITE Initial Conversational (TIV)	ESC 2	X'27F2'
	Then READ Continue(s) (TT)	Not Used	—
Read Buffer From Position ²	WRITE Initial (TI)	ESC 1	X'27F1'
	Then WRITE Continue Conversational (TVT)	ESC 2	X'27F2'
	Then READ Continue(s) (TT)	Not Used	—
Write	WRITE Initial and Reset (TIR) or	ESC 1	X'27F1'
	WRITE Initial (TI) and/or WRITE Continue(s) (TT)	ESC 1	X'27F1'
	Then WRITE EOT (TR) or WRITE Continue and Reset (TTR)	Not used	—
Erase and Write Alternate ³	Same as for Write	ESC =	X'277E'
Erase and Write	Same as for Write	ESC 5	X'27F5'

Figure 51 (Part 1 of 2). Macro Instructions and Device Control Sequences for Nonswitched 3270 Function

TO DO THIS...	USE THESE MACRO INSTRUCTIONS ¹ ...	WITH THESE DEVICE CONTROL CHARACTERS IN THE OUTPUT DATA STREAM...	
		CHAR	HEX (EBCDIC)
Write Structured Fields ⁴	WRITE Initial Transparent Block (TIE) and/or WRITE Continue Transparent Block (TTE)	ESC 3	X'27F3'
	Then WRITE Continue Transparent Text (TTX)	ESC 3	X'27F3'
	Then WRITE EOT (TR)	Not Used	---
	or		
	WRITE Initial Transparent Text (TIX)	ESC 3	X'27F3'
	Then WRITE EOT (TR)	Not Used	---
Erase Unprotected Fields	WRITE Initial and Reset (TTR)	ESC ?	X'276F'
	or		
	WRITE Initial (TI)	ESC ?	X'276F'
	Then WRITE EOT (TR)	Not Used	---
Copy	WRITE Initial and Reset (TIR)	ESC 7	X'27F7'
	or		
	WRITE Initial (TI)	ESC 7	X'27F7'
	Then WRITE EOT (TR)	Not Used	---

1 Typically, a WAIT or TWAIT macro instruction is issued to determine I/O completion of each READ or WRITE macro instruction.

2 In order to effect the Read Modified Fields from Position and Read Buffer from Position functions, a WRITE Initial must be issued first to establish the screen position by specifying an SBA address, and then WRITE Conversational must be issued to send the escape command (it will also read up to 256 bytes of data).

3 Not applicable to devices attached to 3271 or 3275 Control Units.

4 Applicable only to remote 3274 Model 1C devices with Structured Field capability.

Figure 51 (Part 2 of 2). Macro Instructions and Device Control Sequences for Nonswitched 3270 Function

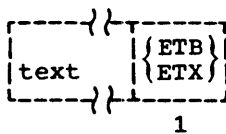
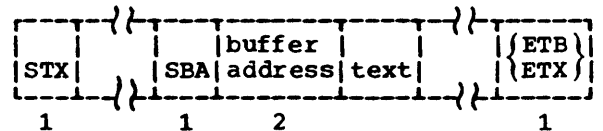
① ASCII message (in hexadecimal)	02	47	41	27	44	48	11	44	20	4A	2E	20	53	4D	49	54	48	03
② Message characters	STX	G	A	'	D	H	DC1	D	Space	J	.	Space	S	M	I	T	H	ETX
③ EBCDIC translation (in hexadecimal)	02	C7	C1	7D	C4	C8	11	C4	40	D1	4B	40	E2	D4	C9	E3	C8	03
④ Message content	start of text	control unit: 7	device: 1	AID: ENTER key	cursor address: 0264	SBA order	buffer address: 0256	message text: J. SMITH										end of text

Figure 52. Sample Input Message (Showing Translation from EBCDIC to ASCII)

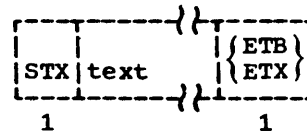
① Message content	start of text	escape command: 1	WCC	SBA order	buffer address: 0064	SF order	attribute byte	message text: ENTER										end of text
② EBCDIC message (in hexadecimal)	02	27	F1	C3	11	C1	40	1D	60	C5	D5	E3	C5	D9	03			
③ Message characters	STX	ESC	1	C	DC1	A	Space	{GS}	-	E	N	T	E	R	ETX			
④ ASCII translation (in hexadecimal)	02	1B	31	43	11	41	20	1D	2D	45	4E	54	45	52	03			

Figure 53. Sample Output Message (Showing Translation from EBCDIC to ASCII)

If the device buffer was unformatted, the buffer specified by the inoutarea operand contains an index byte and a message block with the format:



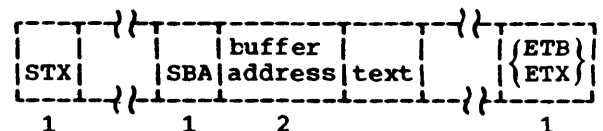
If a specific poll was used and the device buffer was unformatted, subsequent message blocks have the format:



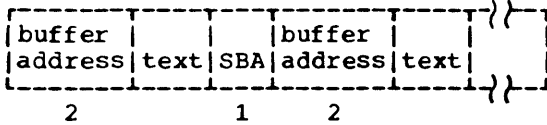
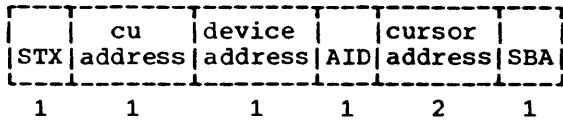
Data from remote 3270 devices is blocked with a nominal length of 256 bytes (including data link control characters). The actual length of a block can be calculated from the DECCOUNT field in the DECB.

Since a response may contain more than one block, use READ TT macro instructions to read blocks until an EOT is received. If a specific poll was used and the device buffer was formatted, subsequent message blocks have the format:

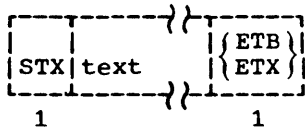
A general poll may result in messages from more than one device attached to the control unit polled. The last block of a message from one device ends with an ETX. If a general poll was used and the device buffer was formatted, a message block following a block ending with an ETB has the format:



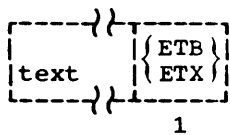
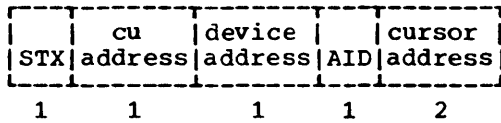
The message block following a block ending with an ETX has the format:



If a general poll was used and the device buffer was unformatted, a message block following a block ending with an ETB has the format:

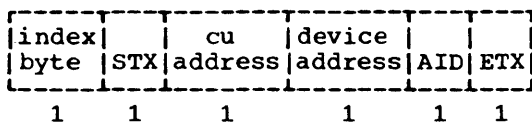


The message block following a block ending with an ETX has the format:



Note: A message block received in response to a READ TT macro instruction is not preceded by an index byte.

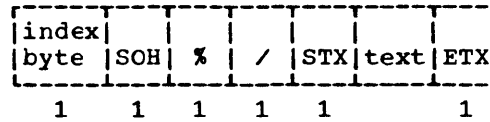
- If the operator pressed the CLEAR key or pressed a PA key, the READ TI macro instruction causes a short read. The buffer specified by the inoutarea operand contains:



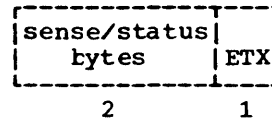
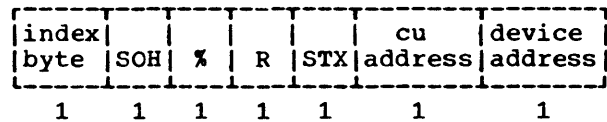
- If a card or cards were read by means of the operator identification card reader, the buffer specified by the inoutarea operand contains a message in the same format as the message result-

ing from pressing the ENTER key. The cursor address contains the address of the last character read from the identification card (the LRC character).

- If a test request message was entered and EROPT=T was not specified in the DCB, the buffer specified by the inoutarea operand contains:



- If an error status message was read, the buffer specified by the inoutarea operand contains:

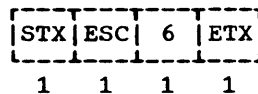


To terminate a read operation, issue a READ TRV macro instruction after receiving a message block ending with an ETX. If a permanent I/O error occurs, either issue a READ TP macro instruction to reread the block or issue a WRITE TR macro instruction to terminate the operation.

Examples of READ macro instructions for reading modified fields for operator input are:

```
READ  DECB1,TI,DCB1,INBUF1,256,PNTRY1,2
READ  DECB2,TT,DCB1,INBUF2,256,,2
```

Read Modified Fields: To select a device and, independently of action by the display station operator, to read from the device, use a WRITE TIV macro instruction with the entry operand specifying the selection entry of the device and the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Follow the WRITE macro instruction with READ TT macro instructions.

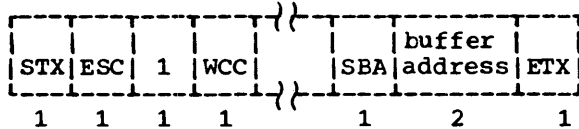
In the first message block, if the AID byte contains neither C'-' nor C'Y' or if the first byte is an SOH, the buffer specified by the inoutarea operand contains a

message block with one of the formats described under "Read Modified Fields for Operator Input." Otherwise, the contents of the buffer are unpredictable; the message is probably nonexistent or incomplete.

Examples of WRITE and READ macro instructions for reading modified fields are:

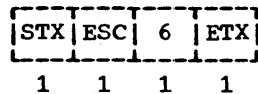
```
WRITE  DECB1,TIV,DCB1,(INBUF1,OUTBUF),
      (256,4),SNTRY2,3
READ   DECB2,TT,DCB1,INBUF1,256,,3
```

Read Modified Fields from Position: To select a device and read from the device, use a WRITE TI macro instruction with the entry operand specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags and the last buffer address should indicate where the read modified operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



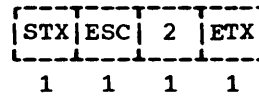
Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Modified Fields."

Examples of WRITE and READ macro instructions for reading modified fields from position are:

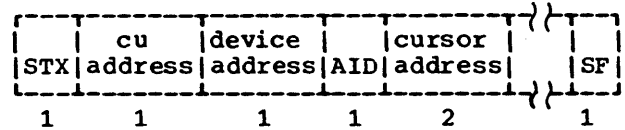
```
WRITE  DECB1,TI,DCB1,OUTBF1,8,SNTRY2,1
WRITE  DECB2,TTV,DCB1,(INBUF1,OUTBUF2),
      (256,4),,1
READ   DECB3,TT,DCB1,INBUF1,256,,1
```

Read Buffer: To select a device and read from the device, use a WRITE TIV macro instruction with the entry operand specifying the selection entry of the device and the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):

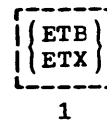
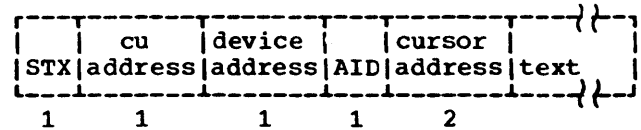


Follow the WRITE TIV macro instruction with READ TT macro instructions.

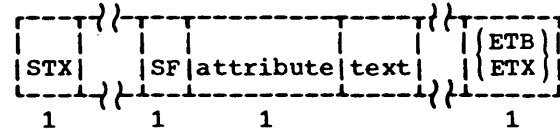
If the device buffer was formatted, the first message block in the buffer specified by the inoutarea operand has the format:



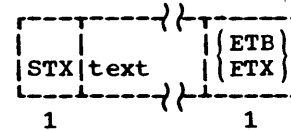
If the device buffer was unformatted, the first message block in the buffer specified by the inoutarea operand has the format:



If the device buffer was formatted, subsequent message blocks have the format:



If the device buffer was unformatted, subsequent message blocks have the format:

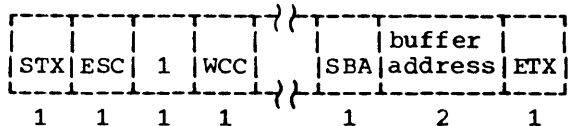


Examples of WRITE and READ macro instructions for reading a buffer are:

```
WRITE  DECB1,TIV,DCB1,(INBUF1,OUTBUF),
      (256,4),SNTRY2,2
READ   DECB2,TT,DCB1,INBUF1,256,,2
```

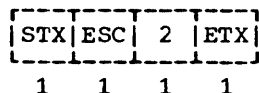
Read Buffer from Position: To select a device and read from the device, use a WRITE TI macro instruction with the entry

operand specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags (if their setting is wanted in the input message), and the last buffer address should indicate where the read buffer operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Buffer."

Examples of WRITE and READ macro instructions for reading a buffer from position are:

```
WRITE  DECB1, TI, DCB1, OUTBF1, 8, SNTRY3, 3
WRITE  DECB2, TTV, DCB1, (INBUF1, OUTBF2),
      (256, 4), , 3
READ   DECB3, TT, DCB1, INBUF1, 256, , 3
```

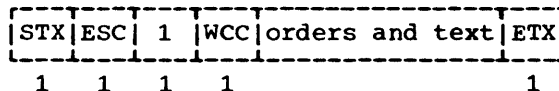
Write Operations

Write Buffer: To select a device and write a message block or blocks, use one of the following sequences of WRITE macro instructions:

- WRITE TIR
- WRITE TI
WRITE TR
- WRITE TI
WRITE TTR
- WRITE TI
WRITE TT
.
.
.
WRITE TR

- WRITE TI
WRITE TT
.
.
.
WRITE TTR

with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand of the WRITE TIR, WRITE TI, WRITE TT, or WRITE TTR macro instruction specifying a buffer that contains (in EBCDIC or ASCII):



For information about the WCC and the orders and text that may follow it, see IBM 3270 Information Display System, Library Users Guide. An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs.

Examples of WRITE macro instructions for writing a buffer are:

```
WRITE  DECB1, TIR, DCB1, OUTBUF, 128,
      SNTRY1, 2
WRITE  DECB1, TI, DCB1, OUTBUF, 128,
      SNTRY1, 2
WRITE  DECB2, TR, DCB1, , , 2
WRITE  DECB1, TI, DCB1, OUTBF1, 256,
      SNTRY2, 3
WRITE  DECB2, TTR, DCB1, OUTBF2, 128, , 3
WRITE  DECB1, TI, DCB1, OUTBF1, 256,
      SNTRY3, 2
WRITE  DECB2, TT, DCB1, OUTBF2, 256, , 2
WRITE  DECB3, TR, DCB1, , , 2
WRITE  DECB1, TI, DCB1, OUTBF1, 256,
      SNTRY2, 1
WRITE  DECB2, TT, DCB1, OUTBF2, 256, , 1
WRITE  DECB3, TTR, DCB1, OUTBF3, 128, , 1
```

Erase and Write Buffer: To select a device, clear its buffer to nulls (binary zeros), and write a message block or blocks, use one of the sequences of WRITE macro instructions listed under "Write Buffer" with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand of the WRITE TIR, WRITE TI, WRITE TT, or WRITE TTR macro instruction specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	5	WCC	orders and text	ETX
1	1	1	1		1

Examples of WRITE macro instructions for erasing and writing a buffer are the same as those given under "Write Buffer."

Erase Write Alternate: To select a device, invoke its alternate buffer size, clear its buffer to nulls (binary zeros), and write one or more message blocks, use one of the sequences of WRITE macro instructions listed under "Write Buffer" with the entry operand of the WRITE TIR or WRITE TI macro instructions specifying the selection entry of the device and the inoutarea operand of the WRITE TIR, WRITE TI, WRITE TT, or WRITE TTR macro instruction specifying a buffer that contains:

STX	ESC	=*	WCC	ORDERS AND TEXT	ETX
1	1	1	1		1

* X'7E' in EBCDIC

Examples of WRITE macro instructions for invoking alternate buffer sizes, erasing and writing a buffer are the same as those given under "Write Buffer." This command does not apply to the 3271 Control unit or the 3275 Display Station.

Programming Notes:

1. The alternate buffer mode will remain in effect until an Erase/Write command (ESC 5) is sent from the host (via the application or Test Request processing). Operation of the CLEAR key for displays and Power-On for both displays and printers will cause the device to enter default buffer mode.
2. Default and Alternate buffer sizes for the 3278 Display Station and 3287 and 3289 Printers are:

	Model			
	1	2	3	4
Default	480	1920	1920	1920
Alternate	960	1920	2560	3440

3. All commands except Erase/Write and Erase/Write Alternate are based on the current (last invoked) buffer size. Any application that invokes alternate

buffer mode should return the device to its default buffer size before the application terminates. This will reduce the possibility of starting a second application (which may not be aware of alternate buffer sizes) with the device in a buffer size other than the one expected.

Write Structured Field: This operation causes a remote 3274 Model 1C with Write Structured Field capability to enter Write Structured Field mode. This operation transfers a data stream containing structured fields to a specified device, but will not cause any specific device action. The structured field contains a type field which will cause a device action.

Unlike other remote 3270 operations, Write Structured Field requires data transparency because full eight bit bytes describe many of the device actions. Data transparency means that data, including normally restricted data-link line-control characters are treated as specific bit patterns. Thus any bit pattern can be transmitted as information data and not as line-control data. To enter transparent mode, use Write transparent macros. If contact has not previously been established with the selected device, a WRITE TIE or WRITE TIX is issued with the entry operand specifying the address of the device's entry in an addressing terminal list. If contact is already established a WRITE TTE or WRITE TTX is issued, and the entry operand is not applicable and should be omitted. In both cases, the area operand specifies an output area containing:

DLE	STX	ESC	3*	STRUCTURED FIELD (S)	**
1	1	1	1		

* X'F3' in EBCDIC

** The DLE ETX or DLE ETB characters are supplied by BTAM when the application program uses BTAM transparency macro instructions.

To terminate Write Structured Field transmission, a WRITE TR macro is issued.

An example of the Write Structure Field operation is:

```
SENDMSG WRITE DECBl,TIX,DCBl,OUTBUF,
          300,SNTRY1,2
ENDWRT WRITE DECBl,TR,DCBl,,,,2
```

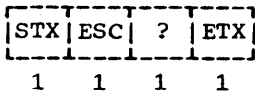
PROGRAMMING NOTES:

- Any application program using Write Structured Field for a remote 3270 should, before terminating, return the device to its 3270 (default) mode. This will avoid starting a second application program, which may not use Write Structured Field, with the device in Write Structured Field mode. To reset Write Structured Field mode, issue an Erase Write Alternate or Erase/Write command with a WCC of .1..

BTAM will automatically reset Write Structured Field mode during Request-for Test (RFT) processing, because BTAM issues an Erase/Write command with a WCC of .1..

- Write Structured Field data can not be translated to ASCII code.

Erase Unprotected Fields: To select a device and set all unprotected fields in its buffer to nulls (binary zeros), use the WRITE TIR macro instruction or the WRITE TI and WRITE TR macro instructions with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):

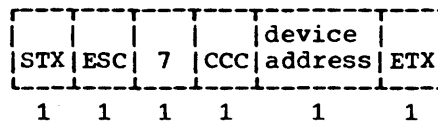


For more information about the results of this operation, see IBM 3270 Information Display System, Library Users Guide.

Examples of WRITE macro instructions for erasing all unprotected fields are:

```
WRITE DECBI,TIR,DCBI,OUTBUF,4,SNTRY1,2
WRITE DECBI,TI,DCBI,OUTBUF,4,SNTRY1,2
WRITE DECBI,TR,DCBI,,,,2
```

Copy: To select a device and copy into its buffer the contents of the buffer of another device on the same control unit, use the WRITE TIR macro instruction or the WRITE TI and WRITE TR macro instructions with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):



For more information about the CCC and the device address that follows it, see IBM 3270 Information Display System, Library Users Guide.

Examples of WRITE macro instructions for copying into a buffer are:

```
WRITE DECBI,TIR,DCBI,OUTBUF,6,SNTRY1,2
WRITE DECBI,TI,DCBI,OUTBUF,6,SNTRY1,2
WRITE DECBI,TR,DCBI,,,,2
```

Programming Notes:

If the response to a READ TI, READ TT, WRITE TIV, or WRITE TTV macro instruction is a message block ending with an ETB, follow with a READ TT macro instruction, not a WRITE TT or WRITE TTV macro instruction.

If a WRITE TI macro instruction is used to erase unprotected fields, follow with a WRITE TR macro instruction, not a WRITE TT, WRITE TIV, or WRITE TTV macro instruction.

If a WRITE TI or WRITE TT macro instruction is used to start a printer, follow with a WRITE TR macro instruction, not a WRITE TT, WRITE TIV, or WRITE TTV macro instruction.

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

See the general section "Error Recovery Procedures and Error Recording," and see Appendixes B and C.

Error Conditions

An error status message should be read from the remote 3270 device if:

- A WRITE TI, WRITE TIR, or WRITE TIV macro instruction receives an RVI sequence in response to selection (completion code is X'7F'; bits 1 and 6 are on in the DECFLAGS field of the DECB).
- A WRITE TI, WRITE TIR, or WRITE TIV macro instruction receives an EOT in response to text (completion code is X'41'; bit 1 is on in the DECFLAGS field; an EOT is in byte 1 of the DECRSPN field of the DECB).

- A WRITE TIV or READ TI macro instruction receives a text block ending with an ENQ (completion code is X'41'; bit 1 is on in the DECFLAGS field).

To receive the error status message, the problem program should issue a READ TI macro instruction using the polling entry of the device for which completion was posted. See Appendix C for a description of the error status message and suggested actions based on its contents.

Exceptional Conditions

If the completion code is X'7F' and bit 6 is on in the DECFLAGS field of the DECB, an error status message was received in

response to a READ TI macro instruction (or in response to a READ TT macro instruction if a general poll was used).

If the completion code is X'7F' and bits 0 and 1 are on in the DECFLAGS fields of the DECB, a WACK was received in response to a WRITE TI or WRITE TT macro instruction. If the write operation started a printer, this is a normal completion. A WRITE TR macro instruction must follow to reset the line.

RETRY OPTIONS

See the BSC3 retry options in the general section "Suggested Retry Options for BSC Read and Write Operations."

ONLINE TESTING

See "Online Testing for Binary Synchronous Communications Lines" in the general section "Online Testing."

To receive standard IBM maintenance for a remote 3270 display system, the online testing facility must be available.

SWITCHED REMOTE 3270 DISPLAY SYSTEM

IBM 3275 Display Stations equipped with the dial feature are supported by BTAM as BSC stations connected to switched point-to-point lines using either EBCDIC or ASCII transmission code.

DEFINING COMMUNICATIONS LINE GROUPS

See "Defining Communications Line Groups" in the chapter "Defining the Teleprocessing System," and see Appendix D.

The UNIT operand of the IODEVICE system generation macro instruction must specify BSC2 for the switched 3275 display station.

The DCB macro instruction operands that apply to the switched 3275 display station are: DSORG, MACRF, DDNAME, BUFNO, BUFL, BUFCB, EXLST, BFTEK, LERB, EROPT= C, T, and E, DEVD=BS, MODE, CODE.

DEFINING AND MODIFYING TERMINAL LISTS

See "Defining and Modifying Terminal Lists" in the chapter "Defining the Teleprocessing System," see "Defining Terminal Lists" and "Defining Terminal List (SWLST) Expanded ID Verification" under the heading "BSC

Switched Point-to-Point Operation" in the chapter "BSC Read and Write Operations," and see Appendix A.

DFTRMLST Macro

The DFTRMLST macro instruction is used to define terminal lists for the switched 3275 display station. In a switched 3275 system the terminal list contains numbers to be dialed and the unique identification (ID) number to be received from each 3275. The terminal list consists of one or more entries, one entry for each 3275 being defined. Entries in the terminal list are referred to when a connection-initiating READ or WRITE macro instruction is coded.

Three list types may be used for the switched 3275: BSCLST, SWLST, and WTLIST. The calling and answering list options and the associated READ or WRITE macros are described under "BSC Switched Point-to-Point Operation" in the "BSC Read and Write Operations" section.

ID Verification

ID verification must be requested in programming input/output with a switched 3275. BTAM verifies that an identification (ID) number received from the remote device is the same as a number furnished in a terminal list defined in the application program. This assures that communication between CPU and the switched 3275 will take place only with previously defined 3275s.

The ID for a particular 3275 with the dial feature is determined by IBM and the user and is hardwired into the 3275; it is always four characters in length. The first character (lower case "f" for EBCDIC; upper case "F" for ASCII) provides terminal type identification and is assigned by IBM. The other three characters are assigned by the customer or by IBM. Graphic characters only can be assigned. The assignment is transparent to the code (EBCDIC or ASCII). The graphic assignments that can be used appear in the IBM 3270 Information Display System Library Users Guide.

The switched 3275 always sends its ID on both a CPU calling operation and a CPU answering operation. The CPU may or may not, at the user's option, transmit an ID during a calling operation or an answering operation. Since a 3275 cannot verify an ID from the CPU, this facility is not useful for switched 3275 applications. If a CPU ID is used, it cannot exceed 15 characters and must not be used except when establishing initial contact. The possible ID and control character sequences are summarized in Figure 53.

Type of Terminal List	Transmission Sequence
Calling from CPU	CPU sends ID ENQ or ENQ 3275 Dial sends ID ACK-0
Answering from CPU	3275 Dial sends ID ENQ CPU sends ID ACK-0 or ACK-0

Figure 54. ID and Control Character Sequences for Calling or Answering Terminal Lists

Modifying Terminal Lists

The CHGNTRY macro instruction is used only to change the value of a control byte in an answering list of the SWLST format for a switched 3275. See "CHGNTRY Macro Instruction" in the chapter "Defining the Teleprocessing System."

BUFFER MANAGEMENT

See the chapter "Buffer Management." Programmer buffering or dynamic buffering can be used for any remote 3270 system, including a switched 3275.

CODE TRANSLATION

See the chapter "Code Translation" and Appendix E.

The TRNSLATE macro can be used to translate switched 3275 messages between EBCDIC and ASCII codes when ASCII transmission is used. To use the TRNSLATE macro, the BTAM translation tables must be assembled in the application program by use of the ASMTRTAB macro instruction.

Bytes containing binary information (the WCC, attribute character, cursor and buffer address, and sense/status bytes) must be encoded in EBCDIC, translated to ASCII on output, and translated to EBCDIC on input with standard BTAM translation tables. The standard BTAM translation tables for BSC are RASA for incoming messages and SASA for outgoing messages.

ACTIVATING AND DEACTIVATING THE TELEPROCESSING SYSTEM

See the chapter "Activating and Deactivating the Teleprocessing System."

LINE CONTROL AND MESSAGE TRANSMISSION

In switched point-to-point operations, several types of READ and WRITE macro instructions are used to make initial connection between the CPU and the remote station. Some of these macro instructions do no more than make the connection; others make the connection and then, as part of the same operation, read or write a message block. Following the READ or WRITE in which connection is established, other types of READ and WRITE macro instructions are used to carry on the input/output operations.

The types of READ and WRITE macros used for making a connection are:

- READ Connect (TC)
- READ Connect with Tone (TCW)
- READ Initial (TI)
- WRITE Connect (TC)
- WRITE Initial (TI)
- WRITE Initial Conversational (TIV)

All READ and WRITE macros applicable to the switched 3275 are shown in Figure 55. A discussion of the function and the channel program for each macro instruction type is given under the heading "BSC Switched Point-to-Point Operation" in the chapter "BSC Read and Write Operations."

To perform operations at the switched 3275, refer to:

1. Figure 30, which specifies the READ or WRITE macro instruction sequence and terminal list required to make a connection and read or write text for any calling or answering situation (automatic or manual, expanded ID verification if desired).
2. Figure 55, which gives the macro instruction sequence and the two-character ESC (command) sequence required to control the 3275. If the 3275 is already connected to the central computer, Figure 55 also lists the READ or WRITE Continue macro required to transmit the ESC sequence to the 3275.

To do this...	Use these macro instruction sequences...	With this two-character 3275 command sequence in the output data stream...	
		Sequence	Hex
Read Modified Fields for Operator Input	READ Connect, READ Initial, or READ Continues to read the first or only block of a message.	Not used	-
	Then one or more READ Continues to read successive blocks until EOT is received.		
Read Modified Fields Independent of Operator Action	WRITE Initial Conversational or WRITE Continue Conversational to write ESC 6 and read the first message block.	ESC 6	X'27F6'
	Then READ Continues until the buffer has been read and EOT is received.	Not used	-
Read Modified Fields from Position	Write Initial or WRITE Continue to set the position from which to read.	ESC 1	X'27F1'
	WRITE Continue Conversational to write ESC 6 and read the first or only message block.	ESC 6	X'27F6'
	Then one or more READ continues to read successive blocks until EOT is received.	Not used	-
Read Buffer	WRITE Initial Conversational or WRITE Continue Conversational to write ESC 2 and read the first message block.	ESC 2	X'27F2'
	Then READ Continues until the buffer has been read and EOT is received.	Not used	-
Read Buffer from Position	WRITE Initial or WRITE Continue to set the position from which to read.	ESC 1	X'27F1'
	WRITE Continue Conversational to write ESC 2 and read the first message block.	ESC 2	X'27F2'
	Then READ Continues to read successive blocks until EOT is received.	Not used	-
Write	WRITE Initial or WRITE Continue or WRITE Connect* followed by WRITE Continue to write the first or only message block.	ESC 1	X'27F1'
	WRITE Continues as desired to write subsequent message blocks.	ESC 1	X'27F1'
	WRITE Reset or WRITE Reset Monitor to return the line to control mode.	Not used	-

Figure 55 (Part 1 of 2). Macro Instruction Sequences to Perform Read and Write Functions at the Switched 3275

To do this...	Use these macro instruction sequences...	With this two-character 3275 command sequence in the output data stream...	
		Sequence	Hex
Erase and Write	WRITE Initial or WRITE Continue or WRITE Connect* followed by WRITE Continue to write the first or only message block.	ESC 5	X'27F5'
	WRITE Continues as desired to write subsequent message blocks.	ESC 1	X'27F1'
	WRITE Reset or WRITE Reset Monitor to return the line to control mode.	Not used	-
Erase Unprotected Fields	WRITE Initial or WRITE Continue or WRITE Connect* followed by WRITE Continue to erase all unprotected fields in the buffer.	ESC ?	X'27F6'
	WRITE Reset or WRITE Reset Monitor to return the line to control mode.	Not used	-

*If WRITE TC, WRITE TT is required (for calling the 3275), place the ESC sequence in the output data stream of the WRITE TT. (WRITE TC connects but does not write a text message.)

Figure 55 (Part 2 of 2). Macro Instruction Sequences to Perform Read and Write Functions at the Switched 3275

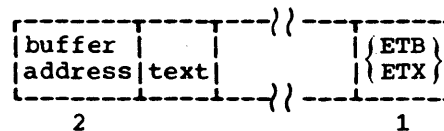
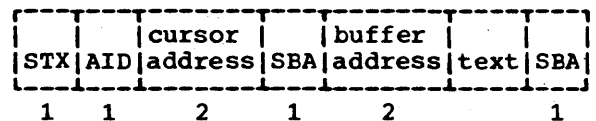
Read Operations

Read Modified Fields for Operator Input:
To connect a 3275 and, after some action by the display station operator, to read a message block, use the READ TI, TC, or TCW macro instruction for the particular switched system as described under "BSC Switched Point-to-Point Operation" in the chapter "BSC Read and Write Operations." If the connection was previously made, use the READ TT macro instruction to read a message block. The entry operand on connection operations specifies the name of a DFRMLST macro instruction that defines the terminal list. The entry operand is not coded in READ or WRITE macro instructions issued after connection is made.

No 3275 command (ESC code) is required to read modified fields; the normal operation of the 3275 reads only modified fields. A maximum of 256 bytes of data are read as the result of one READ macro instruction. READ Continues are required to read additional 256-byte blocks until an ETX is detected indicating the end of the message. An additional READ TT is required to read an EOT, which restores the data link to control mode.

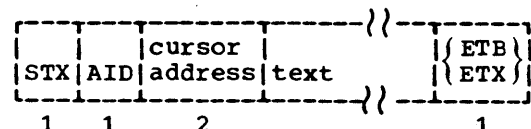
- If the operator pressed the ENTER key or pressed a PF key, the READ TI, TC, or TCW macro instruction causes a norm-

al read. If the 3275 buffer was formatted, the buffer specified by the inoutarea operand contains a message block with the format:

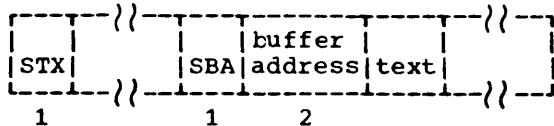


If the operator selected detectable fields with the selector pen, the message block has the same format except that it contains no text.

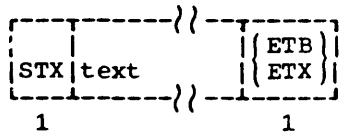
If the 3275 buffer was unformatted, the buffer specified by the inoutarea operand contains a message block with the format:



Data from remote 3270 devices is blocked with a nominal length of 256 bytes (including data link control characters). The actual length of a block can be calculated from the DECCOUNT field in the DECB. Since a response may contain more than one block, use READ TT macro instructions to read blocks until an EOT is received. If the 3275 buffer was formatted, subsequent message blocks have the format:

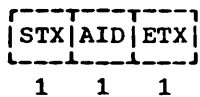


If the 3275 buffer was unformatted, subsequent message blocks have the format:

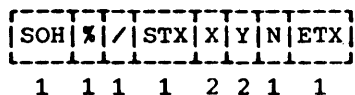


The last block of a message from the 3275 ends with an ETX.

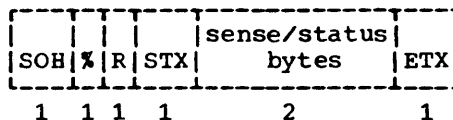
- If the operator pressed the CLEAR key or pressed a PA key, the READ TI, TC, TCW, or TT macro instruction causes a short read. The buffer specified by the inoutarea operand contains:



- If a card or cards were read by means of the operator identification card reader, the buffer specified by the inoutarea operand contains a message in the same format as a message resulting from pressing the ENTER key. The cursor address contains the address of the last character read from the identification card (the LRC character).
- If a test request message was entered and EROPT=T was not specified in the DCB, the buffer specified by the inoutarea operand contains:



- If an error status message was read, the buffer specified by the inoutarea operand contains:



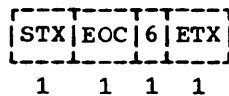
To terminate a read operation, issue a READ TRV macro instruction after receiving a message block ending with an ETX. If a permanent I/O error occurs, either issue a READ TP macro instruction to reread the block or issue a WRITE TR macro instruction to terminate the operation.

Examples of READ macro instructions for reading modified fields for operator input are:

```

READ  DECB1,TI,DCB1,INBUF1,256,PNTY1,2
READ  DECB2,TT,DCB1,INBUF2,256,,2
  
```

Read Modified Fields: To connect a 3275 and, independently of action by the display station operator, to read from it, use a WRITE TIV macro instruction with the entry operand specifying the name of the terminal list and the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Follow the WRITE macro instruction with READ TT macro instructions.

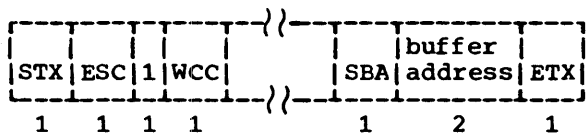
In the first message block, if the AID byte contains neither C-' nor C'Y' or if the first byte is an SOH, the buffer specified by the inoutarea operand contains a message block with one of the formats described under "Read Modified Fields for Operator Input." Otherwise, the contents of the buffer are unpredictable; the message is probably nonexistent or incomplete.

Examples of WRITE and READ macro instructions for reading modified fields are:

```

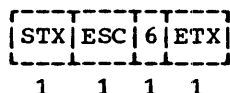
WRITE  DECB1,TIV,DCB1,(INBUF1,OUTBUF),
      (256,4),SYNTRY2,3
READ   DECB2,TT,DCB1,INBUF1,256,,3
  
```

Read Modified Fields from Position: To connect a 3275 and read from it, use a WRITE TI macro instruction with the entry operand specifying the name of the terminal list and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags and the last buffer address should indicate where the read modified operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



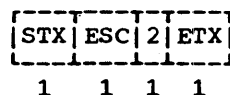
Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Modified Fields."

Examples of WRITE and READ macro instructions for reading modified fields from position are:

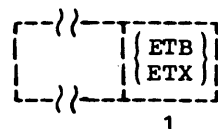
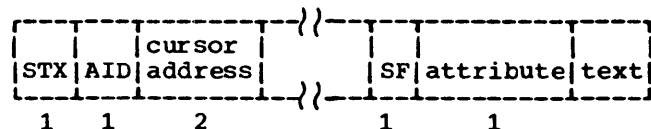
```
WRITE  DECB1, TI, DCB1, OUTBF1, 8, SNTRY2, 1
WRITE  DECB2, TTV, DCB1, (INBUF1, OUTBUF2),
      (256, 4), , 1
READ   DECB3, TT, DCB1, INBUF1, 256, , 1
```

Read Buffer: To connect a 3275 and read from it, use a WRITE TIV macro instruction with the entry operand specifying the name of the terminal list and the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):

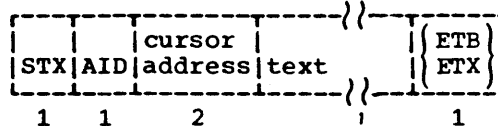


Follow the WRITE TIV macro instruction with READ TT macro instructions.

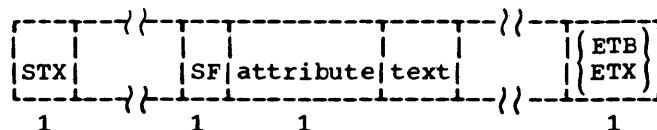
If the 3275 buffer was formatted, the first message block in the buffer specified by the inoutarea operand has the format:



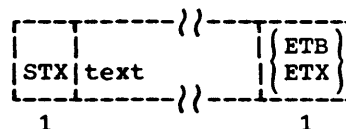
If the 3275 buffer was unformatted, the first message block in the buffer specified by the inoutarea operand has the format:



If the 3275 buffer was formatted, subsequent message blocks have the format:



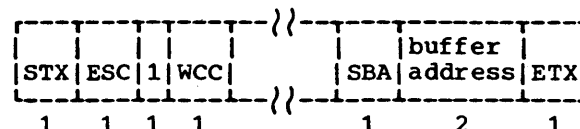
If the 3275 buffer was unformatted, subsequent message blocks have the format:



Examples of WRITE and READ macro instructions for reading a buffer are:

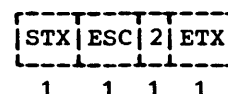
```
WRITE  DECB1, TIV, DCB1, (INBUF1, OUTBUF),
      (256, 4), SNTRY2, 2
READ   DECB2, TT, DCB1, INBUF1, 256, , 2
```

Read Buffer from Position: To connect a 3275 and read from it, use a WRITE TI macro instruction with the entry operand specifying the name of the terminal list and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags (if their setting is wanted in the input message), and the last buffer address should indicate where the read buffer operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):



Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Buffer."

Examples of WRITE and READ macro instructions for reading a buffer from position are:

```
WRITE  DECB1, TI, DCB1, OUTBF1, 8, SNTRY3, 3
WRITE  DECB2, TTV, DCB1, (INBUF1, OUTBF2),
      (256, 4), , 3
READ   DECB3, TT, DCB1, INBUF1, 256, , 3
```

Write Operations

Write Buffer: To connect a switched 3275 and write a message block or blocks, use the appropriate sequence of WRITE macro instructions described under "BSC Switched Point-to-Point Operation" in the chapter "BSC Read and Write Operations" and in Figure 55.

The entry operand of the WRITE TC or WRITE TI macro instruction specifies the name of the terminal list and the inoutarea operand of the WRITE TI or WRITE TT macro instruction specifies a buffer that contains (in EBCDIC or ASCII):

```
-----
|STX|ESC|1|WCC|orders and text|ETX|
-----
1  1  1  1                               1
```

For information about the WCC and the orders and text that may follow it, see IBM 3270 Information Display System, Library Users Guide. An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs.

The line must be reset to control mode to allow the terminal operator to request control of the line when he is ready. Since WRITE TR will allow the operator only nine seconds to respond before timeout occurs, a WRITE TRM can be used to wait for terminal response. If WRITE TRM is used, the programmer should use system time facilities to determine when terminal operator response time has become too long.

Examples of WRITE macro instructions for writing a buffer are:

```
WRITE  DECB1, TI, DCB1, OUTBUF, 128, SNTRY1, 2
WRITE  DECB2, TR, DCB1, , , , 2

WRITE  DECB1, TI, DCB1, OUTBF1, 256, SNTRY3.2
WRITE  DECB2, TT, DCB1, OUTBF2, 256, , 2
WRITE  DECB3, TR, DCB1, , , , 2
```

Otype TRM can be used in place of TR in these examples.

Erase and Write Buffer: To connect a 3275, clear its buffer to nulls (binary zeros),

and write a message block using one of the sequences of WRITE macro instructions listed in Figure 55 with the entry operand of the WRITE TC or WRITE TI macro instruction specifying the name of the terminal list and the inoutarea operand of the WRITE TI or WRITE TT macro instruction specifying a buffer that contains (in EBCDIC or ASCII)

```
-----
|STX|ESC|5|WCC|orders and text|ETX|
-----
1  1  1  1                               1
```

Examples of WRITE macro instructions for erasing and writing a buffer are the same as those given under "Write Buffer."

Erase Unprotected Fields: To connect a 3275 and set all unprotected fields in its buffer to nulls (binary zeros), use one of the sequences of WRITE macro instructions listed in Figure 55 with the entry operand of the WRITE TC or WRITE TI macro instruction specifying the name of the terminal list and the inoutarea operand of the WRITE TI or WRITE TT macro instruction specifying a buffer that contains (in EBCDIC or ASCII):

```
-----
|STX|ESC|?|ETX|
-----
1  1  1  1
```

For more information about the results of this operation, see IBM 3270 Information Display System, Library Users Guide.

An example of WRITE macro instructions for erasing all unprotected fields is:

```
WRITE  DECB1, TI, DCB1, OUTBUF, 4, SNTRY1, 2
WRITE  DECB1, TR, DCB1, , , , 2
```

Otype TRM can be used in place of TR in this example.

Operator Considerations

The terminal operator can not make a successful connection if the 3275 STATUS indicator is on at the time he plans to dial the CPU. The operator should be instructed to turn off the STATUS indicator by turning the power off and then on again. If the error is recurrent, it will recur after a successful connection is made.

It is preferable to control disconnecting from the application program rather than depending on the terminal operator to complete the input/output session by operating the DISCONNECT switch. (Refer to "Programming Notes" in this section for the case in which the 3275 has no more data to send to the CPU.) However, if the applica-

tion depends on the terminal operator to end a session, the DISCONNECT switch sends a DLE EOT sequence to the CPU. If the data set associated with the TCU attached to the CPU has an automatic disconnect feature, disconnection occurs, and the line is disabled. If the data set at the CPU does not have this feature, the program must issue a WRITE Disconnect upon recognizing the DLE EOT in the inout data stream or DECRESFN field of the DECB. The WRITE Disconnect sends a DLE EOT to the device and the line is disabled.

Programming Notes: If a message block ending with an ETB is received in response to a READ TI, READ TT, READ TC, READ TP, READ TRV, READ TCW, WRITE TIV, WRITE TTV, or WRITE TQ, do not follow with WRITE TT or WRITE TTV. If the text is received without error, the next operation should always be READ TT macro instruction.

If a WRITE TI is used to erase unprotected fields, do not follow with WRITE TT or WRITE TTV. If the operation completes without error, the next operation should always be WRITE TR or WRITE TRM.

If a WRITE TI or WRITE TT is used to start a printer (WCC specifies "start print"), do not follow with WRITE TT or WRITE TIV. If the operation completes without error (WACK response to text received), the next operation should always be WRITE TR or WRITE TRM.

If a switched 3275 has no more data to be read (completion code of X'41' and timeout), the next operation should be a WRITE TD, WRITE TQ, or WRITE TRM. (Refer to the chapter "Suggested Retry Options for BSC Read and Write Operations.")

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

See the chapter "Error Recovery Procedures and Error Recording," and see Appendixes B and C.

Error Conditions

An error status message should be read from the switched 3275 if:

- A READ TI (Auto Call) or WRITE TI, TIV, or TC macro instruction receives an ID NAK response to a CPU call to a switched 3275 (completion code is X'7F'; bit 1 is on in the DECFLAGS field of the DECB)
- A WRITE TQ macro instruction receives a NAK response to ENQ (completion code is X'41'; NAK is in byte 1 of the DECRESFN field of the DECB)

- A WRITE TI, TIV, TT, or TTV macro instruction receives an EOT response to write text (completion code is X'41'; bit 1 is on in the DECFLAGS field of the DECB; EOT is in byte 1 of the DECRESFN field of the DECB)

To receive the error status message, the problem program should take the following steps:

1. Issue a WRITE TRM to write EOT and read ENQ (the 3275 must bid for the line).
2. Issue READ TT to read the error status message.
3. Issue another READ TT to acknowledge receipt of the error status message and receive EOT.
4. Examine the sense/status bits and proceed accordingly. See Appendix C for a description of the error status message and suggested actions based on its contents.

Exceptional Conditions

If the completion code is X'7F' and bit 1 is off and bit 6 is on in the DECFLAGS field of the DECB, an error status message was received in response to a READ TI, READ TT, READ TC, or READ TCW macro instruction.

If the Completion code is X'7F' and bits 0 and 1 are on in the DECFLAGS field of the DECB, a WACK was received in response to a WRITE TI or WRITE TT macro instruction. If the write operation started a printer, this is a normal completion. A WRITE TR or WRITE TRM macro instruction must follow to reset the line.

RETRY OPTIONS

See the switched 3275 BSC2 retry options in the chapter "Suggested Retry Options for BSC Read and Write Operations."

ONLINE TESTING

See "Online Testing for Binary Synchronous Communications Lines" in the chapter "Online Testing."

To obtain standard IBM maintenance for a switched 3275 display system, the online testing facility must be available.

LOCAL 3270 DISPLAY SYSTEM

The functions provided by BTAM for remote stations have been extended to support

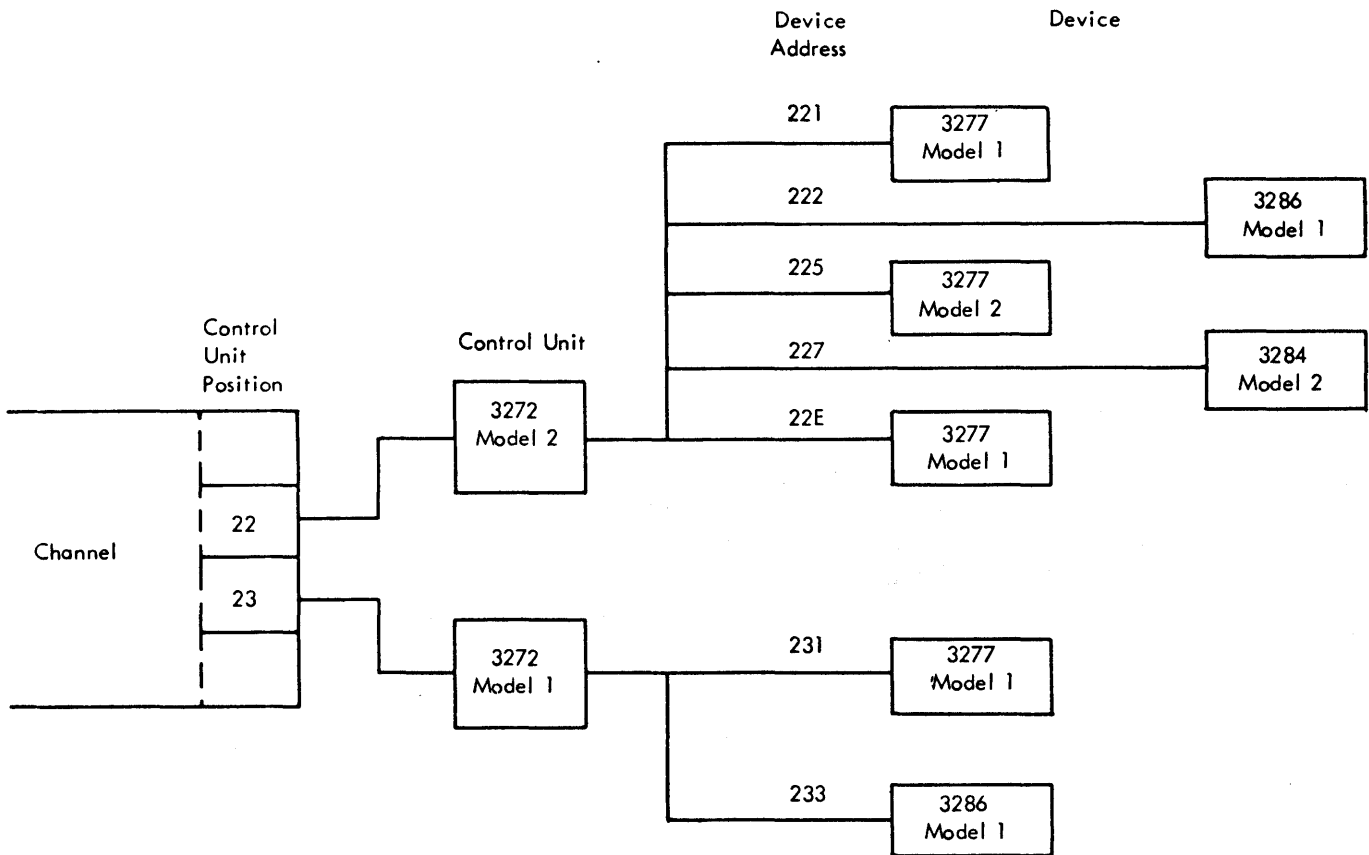


Figure 56. Sample Local 3270 Display Systems Showing Device Addresses

local 3270 display stations and printers as local devices using EBCDIC code.

```
IODEVICE UNIT=3277,ADDRESS=221, . . .
IODEVICE UNIT=3286,ADDRESS=222, . . .
IODEVICE UNIT=3277,ADDRESS=225, . . .
IODEVICE UNIT=3284,ADDRESS=227, . . .
IODEVICE UNIT=3277,ADDRESS=22E, . . .
IODEVICE UNIT=3277,ADDRESS=231, . . .
IODEVICE UNIT=3286,ADDRESS=233, . . .
```

DEFINING THE LOCAL 3270 DISPLAY SYSTEM

Identifying Local 3270 Devices

The control unit and devices in a local 3270 display system are identified as local during system generation. The IODEVICE system generation macro instruction operands that apply to the local 3270 display system are: UNIT, ADDRESS, MODEL, and FEATURE. For more information about the IODEVICE system generation macro instruction, see Appendix D. For more information about models and features of the local 3270 display system, see IBM 3270 Information Display System, Library Users Guide.

Example: The following system generation macro instructions are used to identify the local 3270 display systems shown in Figure 55:

```
IODEVICE UNIT=3277,ADDRESS=221, . . .
IODEVICE UNIT=3286,ADDRESS=222, . . .
IODEVICE UNIT=3277,ADDRESS=225, . . .
IODEVICE UNIT=3284,ADDRESS=227, . . .
IODEVICE UNIT=3277,ADDRESS=22E, . . .
```

Grouping Local 3270 Devices

Local 3270 devices are grouped together in the same way as remote terminals. Each local 3270 device is equivalent to a communication line. Each group of local 3270 devices is equivalent to a line group.

Local 3270 devices may be grouped during system generation by means of the UNITNAME macro instruction. Or line groups may be specified during program execution by means of the UNIT parameter of the DD statement.

Local 3270 devices making up a line group are associated with one DCB. The DDNAME in the DCB must be the same as the name of the DD statement for the line group. A line group can contain up to 60 devices. A line group need not include all the devices on a control unit; it can include devices from several control units. Each local 3270 device is identified by its relative line number (RLN). The relative line numbers are determined by the order in which devices are grouped during system generation or program execution.

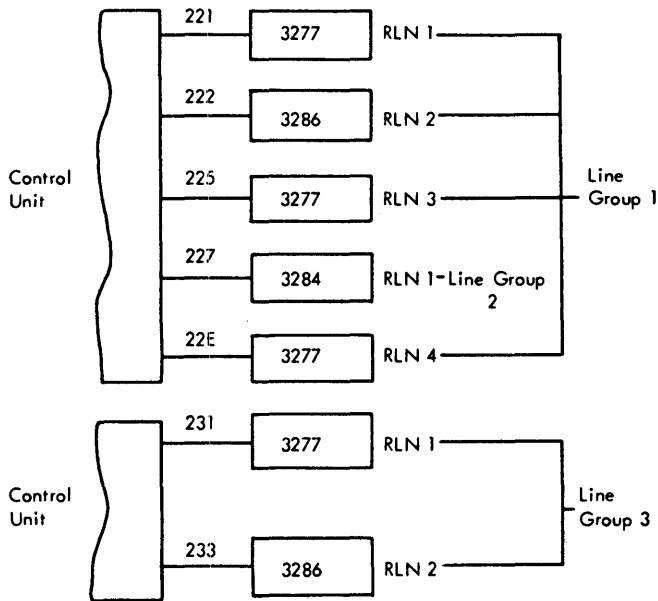


Figure 57. Line Group and Relative Line Numbers for Example 1

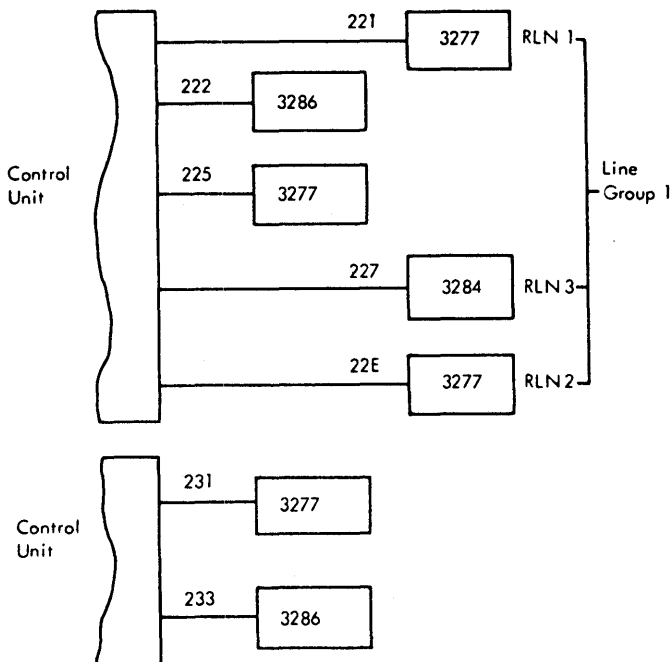


Figure 58. Line Group and Relative Line Numbers for Example 2

Example 1: To define (during system generation) the line groups and relative line numbers shown in Figure 57 for the local 3270 display systems shown in Figure 56, see "Example 1" under "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." (Address 221 would be coded in place of 021, 222 in place of 022, etc.)

Example 2: To define (during program execution) the line groups and relative line numbers shown in Figure 58, see "Example 2" under "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." (Address 221 would be coded in place of 021, 227 in place of 027, etc.)

For a description of the DCB macro instruction, see "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." The DCB macro instruction operands that apply to the local 3270 display system are: DSORG, MACRF, DDNAME, BUFNO, BUFL, BUFCB, EXLST, EROPT, and READYQ.

The local 3270 device can be identified in a READ or WRITE macro instruction by means of the DCB and the relative line number; no terminal list is needed. Therefore, the DFTRMLST macro instruction is not used for the local 3270 display system.

BUFFER MANAGEMENT

See the general section "Buffer Management." Dynamic buffering cannot be used for the local 3270 display system. The REQBUF and RELBUF macro instructions can be used for manipulating buffer pools.

CODE TRANSLATION

Since only EBCDIC code is used, code translation does not apply to the local 3270 display system.

In the I/O interface code for six-bit structured data in all 3270 messages, the setting of the two high-order bits is determined by the setting of the six low-order bits in the byte (see Figure 47). Six-bit structured data includes the WCC, attribute character, and cursor and buffer addresses; for more information, see IBM 3270 Information Display System, Library Users Guide.

ACTIVATING AND DEACTIVATING THE LOCAL 3270 DISPLAY SYSTEM

See the general section "Activating and Deactivating the Teleprocessing System." The OPEN and CLOSE macro instructions are used to activate and deactivate line groups of local 3270 devices. The LOPEN macro instruction is used to activate a specific local 3270 device when the OPEN macro instruction has been unsuccessful.

ATTENTION INTERRUPTIONS AND READ INITIAL OPERATIONS

When the operator of a local 3270 display station carries out certain actions, such as pressing the ENTER or CLEAR key, pressing a PF or PA key, or selecting a detectable field with the selector pen, an I/O interruption, called an attention interruption, occurs. If a DCB has been opened for a line group that includes the display station, the attention interruption is recorded for the display station. If a DCB has not been opened, the attention interruption is ignored.

The READ TI macro instruction for the local 3270 display system includes the dcb operand, which specifies a DCB, and the rln operand, which specifies a local 3270 display station in the line group associated with the DCB. When a READ TI macro instruction is issued, it causes a check (beginning with the display station specified by the rln operand) of whether an attention interruption has been recorded for any of the local 3270 display stations in the specified line group.

If an attention interruption has occurred, a read initial operation is started to read a message from the display station from which the attention interruption came. When the message has been received, the READ macro instruction is posted complete, and the relative line number of the display station is placed into the DECPOLPT field of the DECB. Only one attention interruption is serviced for each READ TI macro instruction, and only one message is read. After a READ TI macro instruction has been issued, the problem program cannot issue another READ or WRITE macro instruction specifying the same DCB until either the read initial operation has been posted complete or the read request has been canceled by means of the RESETPL macro instruction. Attention interruptions that occur between read initial operations are recorded for the display stations and serviced by later READ TI macro instructions.

If no attention interruption has occurred for the display stations associated with the DCB (when a READ TI macro instruction is issued), a read request (that is, a pending read initial operation) is recorded for the line group. When an attention interruption comes from one of the display stations, the pending read initial operation is started for that display station, and the read request is cleared for the line group. If READYQ is specified a test for any device end interruptions is made before recording a read request for the line group. If one is found READYQ processing will be initiated and

the READ TI will be posted as stated under the section DEVICE END INTERRUPTS and BTAM READYQ. If no device end has been recorded then the read request will be recorded as usual.

CHGNTRY Macro Instruction

A special form of the CHGNTRY macro instruction is used in handling attention interruptions from the local 3270 display station. A CHGNTRY macro instruction with the SKIP operand is used to have a display station skipped, so that a read initial operation is not started (that is, an EXCP macro instruction is not issued) for that display station when a READ TI macro instruction is issued, even though an attention interruption has occurred. (If, when a READ TI macro instruction is issued, all display stations in the line group are to be skipped, a pending read initial operation is recorded for the line group.) A CHGNTRY macro instruction with the ACTIVATE operand is used to have a display station activated, so that a read initial operation is started when a READ TI macro instruction is issued, and an attention interruption has occurred. If, when the display station is activated, a pending read initial operation has been recorded for the line group and an attention interruption has been recorded for that display station, the pending read initial operation is started (that is, an EXCP macro instruction is issued).

Note: This form of the CHGNTRY macro instruction affects only read initial operations; it does not affect other local 3270 read or write operations.

Name	Operation	Operand
[symbol]	CHGNTRY	dcbaddr, ATTLLST, listposition,, { SKIP ACTIVATE }

dcbaddr
specifies the address of the DCB associated with the line group that includes the local 3270 display station to be skipped or activated.

ATTLLST
specifies that the relative line numbers of the local 3270 display stations in the line group associated with the DCB are to be treated as an attention list.

listposition
specifies the relative line number of the local 3270 display station to be skipped or activated.

SKIP
specifies that the local 3270 display station is to be skipped; that is, an attention interruption that has been recorded for the display station is to

be ignored if a READ TI macro instruction is issued.

ACTIVATE

specifies that the local 3270 display station is to be activated; that is, a read initial operation is to be started if an attention interruption has been recorded for the display station when a READ TI macro instruction is issued.

Return Codes: When this form of the CHGNTRY macro is used, the return codes have the following meanings:

<u>Code</u>	<u>Meaning</u>
X'00'	The requested action was or has been performed successfully.
X'08'	The requested action was not performed, because the RLN was too high or BTAM found an invalid control block.

RESETPL Macro Instruction

A special form of the RESETPL macro instruction is used for the local 3270 display system to cancel a read initial operation that is pending (because a READ TI macro instruction was issued when an attention interruption had not occurred).

Name	Operation	Operand
{symbol}	RESETPL	decbaddr[,ATTENT]

decbaddr

specifies the address of the DECB for the pending read initial operation that is to be canceled.

ATTENT

specifies that only the instructions required to cancel a pending read initial operation for a local 3270 display system are to be generated by this macro instruction.

If no second operand is specified, instructions are generated to determine the line type, and the proper instructions for that line type are executed.

Note: If POLLING or ANSRING is specified as the second operand, instructions for the local 3270 display system are not generated.

Programming Notes: The RESETPL macro instruction does not halt read initial operations that have been started; it does prevent the outstanding READ TI macro instruction from having any subsequent attention interruptions serviced. If a read initial operation is pending (that is, it has not

been started), the IOBs are marked free, and a completion code of X'48' is posted in the ECB. If a read initial operation has been started, the IOBs for active devices are not marked free, and a completion code is not posted in the ECB by the RESETPL macro instruction.

Follow the READ TI macro instruction with a WAIT macro instruction with the ECB-LIST operand (or a TWAIT macro instruction) that specifies multiple ECBs, the ECB in the DECB specified by the READ TI macro instruction and another ECB. To allow the RESETPL macro instruction to be issued to cancel the read initial operation, post the other ECB. Provided that the RESETPL macro instruction gave a return code of X'00' or X'04', follow the RESETPL macro instruction with a WAIT macro instruction that specifies the ECB in the DECB specified by the READ TI macro instruction. A completion code of X'7F' indicates that a read initial operation had been started and that it completed successfully. Other READ and WRITE macro instructions can then be issued.

Return Codes: When this form of the RESETPL macro instruction is used, the return codes have the following meanings:

- 00 This code is set when the pending read initial operation for the specified DECB was canceled successfully.
- 04 This code is set when a read initial operation for the specified DECB was started and will complete normally (see "Programming Notes" above).
- 08 This code is set for an illegal request. BTAM found that the specified DECB is not associated with a local 3270 display system.
- 0C This code is set for an unsuccessful request, the DCB was not open or BTAM found an invalid control block.
- 10 This code is set when the last macro instruction for the line group was not a RDTI

READ AND WRITE OPERATIONS

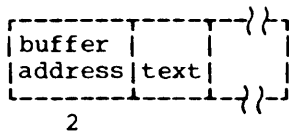
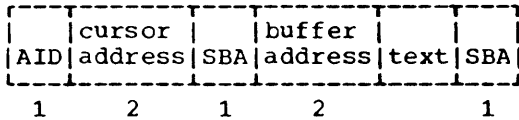
See "Read and Write Macro Instructions" and Figure 20 in the general section "Line Control and Message Transmission," see "READ Macro Instructions" and "WRITE Macro Instructions" under the heading "Local IBM 3270 Display System" in the section "Local Read and Write Operations," and see Appendixes B and G.

Read Operations

Read Modified Fields for Operator Input: To read a message from a display station after an attention interruption has come from it as the result of some action by the

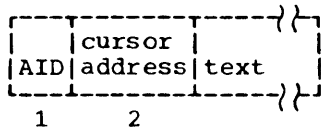
display station operator, use the READ TI macro instruction. The rln operand specifies which display station in the line group should be checked first for an attention interruption.

- If the operator pressed the ENTER key or pressed a PF key, READ TI macro instruction causes a normal read. If the device buffer was formatted, the buffer specified by the inoutarea operand contains a message with the format:



If the operator selected detectable fields with the selector pen, the message block has the same format except that it contains no text.

If the device buffer was unformatted, the buffer specified by the inoutarea operand contains a message with the format:



The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TMP macro instruction may be used to continue reading data.

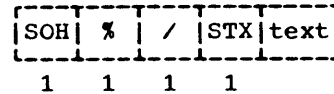
- If the operator pressed the CLEAR key or pressed a PA key or if a card was extracted from the badge reader, the READ TI macro instruction causes a short read. The buffer specified by the inoutarea operand contains:



- If a card or cards were read by means of the operator identification card reader, the buffer specified by the inoutarea operand contains a message in the same format as a message resulting from pressing the ENTER key. The cursor address contains the address of the

last character read from the identification card (the LRC character).

- If a test request message was entered and EROPT=T was not specified in the DCB, the buffer specified by the inoutarea operand contains:



The relative line number of the display station from which the message was read is placed into the low-order byte of the DECPOLPT field of the DECB.

To cancel the read initial operation requested by a READ TI macro instruction (if the operation is pending), issue the RESETPL macro instruction specifying the DECB from the READ TI macro instruction.

Do not issue a READ TI macro instruction specifying a DCB associated with a line group that contains only printers, since the local 3270 printer cannot generate attention interruptions.

An example of a READ macro instruction for reading modified fields for operator input is:

```
READ DECB1,TI,DCB1,INBUF1,300,,2
```

Read Modified Fields: To read a message from a device independently of action by the display station operator, use the READ TM macro instruction.

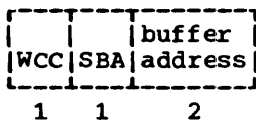
If the AID byte in the message contains neither C'-' nor C'Y' or if the first byte is an SOH, the buffer specified by the inoutarea operand contains a message with one of the formats described under "Read Modified Fields for Operator Input." Otherwise, the contents of the buffer are unpredictable; the message is probably non-existent or incomplete.

The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TMP macro instruction may be used to continue reading data. The relative line number of the device from which the message was read is placed into the DECPOLPT field of the DECB.

An example of a READ macro instruction for reading modified fields is:

```
READ DECB1,TM,DCB1,INBUF1,300,,3
```

Read Modified Fields from Position: To read a message from a device, use the READ TMP macro instruction with the entry operand specifying an area that contains:



The WCC should inhibit reset of modified data tags, and the buffer address should indicate where the read modified operation is to start. Data transfer begins with the first modified field at or following the buffer address specified.

Input messages are the same as those described under "Read Modified Fields."

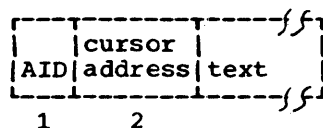
An example of a READ macro instruction for reading modified fields from position is:

```
READ DECBI,TMP,DCBI,INBUF1,300,ENTRY1,1
```

Read Buffer: To read a message from a device, use the READ TB macro instruction. If the device buffer was formatted, the buffer specified by the inoutarea operand contains a message with the format:



If the device buffer was unformatted, the buffer specified by the inoutarea operand contains a message with the format:

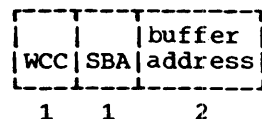


The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TBP macro instruction may be used to continue reading data. The relative line number of the device from which the message was read is placed into the DECPOLPT field of the DECB.

An example of a READ macro instruction for reading a buffer is:

```
READ DECBI,TB,DCBI,INBUF1,300,,2
```

Read Buffer from Position: To read a message from a device, use the READ TBP macro instruction with the entry operand specifying an area that contains:



The WCC should inhibit reset of modified data tags (if their setting is wanted in the input message), and the buffer address should indicate where the read buffer operation is to start.

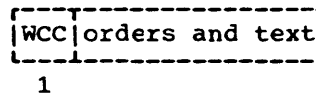
Input messages are the same as those described under "Read Buffer."

An example of a READ macro instruction for reading a buffer from position is:

```
READ DECBI,TBP,DCBI,INBUF1,300,ENTRY2,3
```

Write Operations

Write Buffer: To write a message to a device, use the WRITE TI macro instruction with the inoutarea operand specifying a buffer that contains:



For information about the WCC and the orders and text that may follow it, see IBM 3270 Information Display System, Library Users Guide. An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs.

An example of a WRITE macro instruction for writing a buffer is:

```
WRITE DECBI,TI,DCBI,OUTBUF,300,,2
```

Erase and Write Buffer: To clear its buffer to nulls (binary zeros) and write a message to a device, use the WRITE TS macro instruction with the inoutarea specifying a buffer that has the same contents as described under "Write Buffer."

An example of a WRITE macro instruction for erasing and writing a buffer is:

```
WRITE DECBI,TS,DCBI,OUTBUF,300,,2
```

Erase and Write Alternate Buffer: To place the selected device (display or printer) into a mode of using the alternate buffer size for that device, clear the buffer to nulls (binary zeros) and write a message to a device, use the WRITE TSA macro instruction with the inoutarea specifying a buffer that has the same contents as described under "Write Buffer". This macro instruction does not apply for devices attached to a 3272 Control Unit.

An example of a WRITE macro instruction for invoking alternate buffer size, erasing, and writing a buffer is:

```
WRITE DECBI,TSA,DCBI,OUTBUF,300,,2
```

Programming Notes:

1. The alternate buffer mode will remain in effect until an Erase/Write is sent (via a Write TS macro, Test Request processing OPEN macro or LOPEN macro) from the host. Operation of the CLEAR key for displays and Power-On for both displays and printers will also set the buffer size to its default buffer to be erased to nulls.
2. The default and alternate buffer sizes for the 3278 Display Station or 3287 or 3289 Printers are:

	Model			
	1	2	3	4
Default	480	1920	1920	1920
Alternate	960	1920	2560	3440

3. All commands except Erase/Write and Erase/Write Alternate are based on the current (last invoked) buffer size. Any application that invokes alternate buffer size should return the device to its default buffer size before the application terminates. This will reduce the possibility of starting a second application (which may not be aware of alternate buffer sizes) with the device in a buffer size other than the one expected.

Write Structured Field: This operation transfers a data stream containing structured fields to a specified device. The Write Structured Field macro instruction is supported in OS/VS1 and OS/VS2 MVS and applies only to those local devices with structured field capability, that is, devices attached to a 3274 Model 1D Control Unit. Devices unable to support structured fields, that is, those attached to a 3272 or 3274 Model 1B Control Unit, will reject the Write Structured Field with a command reject error. This instruction will not cause any specific device action; however, the structured field contains a type field which will cause a device action. For more information on structured fields see IBM 3270 Information Display System Library Users Guide.

Except for the optype, operands for this macro are the same as those for the WRITE Erase macro instruction. The optype for the WRITE Structured Field macro is TSF.

An example of the WRITE Structured Field is:

```
WRITE DECBI,TSF,DCBI,OUTBUF,
300,,2
```

Programming Note: Any application program which uses Write Structured Field mode for a local 3274 Model 1D should, before terminating, return the device to its local 3270 (default) mode. This will avoid starting a second application program, which may not use Write Structured Field, with the device in Write Structured Field mode. To reset Write Structured Field mode, issued an Erase Write Alternate or Erase/Write command with a WCC of .1..

BTAM OPEN and LOPEN routines and Request-for-Test (RFT) processing will issue an Erase/Write command with a WCC of .1.. .., thus resetting Write Structured Field mode for local 3274 Model 1D devices.

Erase Unprotected Fields: To set all unprotected fields in the buffer of a device to nulls (binary zeros), use the WRITE TUS macro instruction with the inout-area operand specifying any real address and the inoutlength operand specified as one.

For more information about the results of this operation, see IBM 3270 Information Display System, Library Users Guide.

An example of a WRITE macro instruction for erasing all unprotected fields is:

```
WRITE DECBI,TUS,DCBI,OUTBUF,1,,2
```

Note: Write operations to local 3270 printers are not posted complete until device end is sent by the device. Separate DECBS are recommended for printer operations to allow the application 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.

PREPARE TO READ OPERATIONS

The Prepare to Read operation applies to local 3274 Model 1D devices in OS/VS1 and OS/VS2 MVS. Prepare to Read does not apply to the 3272, 3274 Model 1B, or remote 3270 Control Units.

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

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

To use Prepare to Read select commands, the devices attached to a 3274 Model 1D must be specified (in OS/VS1 or OS/VS2 MVS operating systems) as supporting the Prepare to Read select commands. Specifying FEATURE=PTREAD in the IODEVICE macro sets an indicator in the Unit Control Block (UCB). If the indicator is on, BTAM constructs channel programs containing the Prepare to Read select commands. If the indicator is off, BTAM continues to use the local 3270 (default) select command. The result is that BTAM application programs do not require modification to use the Prepare to Read select commands. For more information, see Appendix D, System Generation.

Only the following Read/Write macros can generate channel programs containing the Prepare to Read Select commands:

Read Macro Instructions

Read Modified from Position (TMP)

If Prepare to Read has been specified, the following channel program will be built for Read Modified from Position:

1. Select Read Modified from Position Command
2. Write Command (to set buffer address)
3. Read Modified Command

Read Buffer (TB)

If Prepare to Read has been specified, the following channel program will be built for Read Buffer:

1. Select Read Buffer Command
2. Read Buffer Command

Read Buffer from Position (TBP)

If Prepare to Read has been specified, the following channel program will be built for Read Buffer from Position:

1. Select Read Buffer from Position Command
2. Write Command (to set buffer address)
3. Read Buffer Command

Write Macro Instructions

Write Initial (TI)

If Prepare to Read has been specified, the following channel program will be built for Write Initial:

1. Select Write Command
2. Write Command

The remaining Read/Write macro instructions use the local 3270 select command regardless of Prepare to Read specification.

DEVICE END INTERRUPTS AND BTAM READYQ

When a device of a local 3270 Display System recorded as powered off, is powered on, a non-related device end I/O interruption occurs. If the DCB has been opened for the line group containing this device and the not ready condition has been recorded the device end will be recorded, indicating the device ready condition.

If the READYQ option was specified, the current or next READ TI macro instruction for this line group will have its DECB posted with a completion code indicating the read initial operation was cancelled by a RESE TPL, the DECFLAGS field will contain a X'F0' indicating READYQ was activated by a device becoming ready (which cancels the RD TI), and the relative line number (rln) of the device is placed in the DECPOLPT field, informing the user which device has become ready. If attention interrupts have been received from the line group they are handled before device end interrupts so the user application must be prepared to issue a READ TI macro for each attention interrupt received plus one READ TI for each device as it becomes ready.

When the user application has been notified that a device that was turned off at open time has become ready it would be appropriate to issue an LOPEN macro or an ERASE/ WRITE macro to that device.

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

See the general section "Error Recovery Procedures and Error Recording," and see Appendix B.

RETRY OPTIONS

See the local 3270 retry options in the general section "Suggested Retry Options for Local Read and Write Operations."

ONLINE TESTING

See "Online Testing for Local 3270 Display System" in the general section "Online Testing."

To receive standard IBM maintenance for a local 3270 display system, the online testing facility must be available.

SYSTEM GENERATION

See Appendix D for information about the operands that must be included in the IOCTRL and IODEVICE system generation

macro instructions when generating an operating system that includes BTAM support for the remote 3270 display system or the local 3270 display system or both.

For information about other operands and other system generation macro instructions and about the system generation process, see the OS/VS1 or OS/VS2 system generation reference manual.

ONLINE TESTING

Test programs for the 3270 display system can be run under OS/VS using the Online Test Executive Program (OLTEP). For local 3270 control units and devices, an online test program and a BTAM application program can be executed concurrently; only the control unit or device being tested is unavailable to the application program. For remote 3270 control units and devices, a BTAM application program must end use of the line before an online test program can be executed.

For more information about OLTEP, see OS/VS OLTEP, GC28-0636.

CONVERSION

For information about converting from 2260 display stations to the 3270 display system, see IBM 2260 BTAM and 2260 GAM to IBM 3270 BTAM Conversion Guide, GC27-6975.

BTAM supports binary synchronous communication with the 3650 Retail Store System. The 3650 system is designed to provide retail merchandisers with a store system that controls terminals handling point-of-sale transactions, data entry and inquiry, report printing, and merchandise marking applications. A store can operate either on an interactive or batch basis with applications in the host computer.

Devices in the system include the 3651 Store Controller Model 50, 3275 Model 3, 3653 Point of Sale terminal, and the 3657 Ticket-Unit terminal. With BTAM support, the 3650 system attaches to an IBM 3704 or 3705 Communications Controller that runs in emulation mode. For detailed information on the 3650 Retail Store System, see IBM 3650 Subsystem Definition Programmer's Guide, GC30-3023.

IBM 3660 SUPERMARKET SYSTEM

BTAM supports binary synchronous communication with the 3660 Supermarket System. The 3660 system provides complete supermarket checkout functions, as well as specific data capture and store support procedures. Grocery merchandise is marked with the Universal Product Code Symbol (A and E). Devices in the system include the 3666 Checkout Scanner, the 3663 Supermarket Terminal, and the 3651 Store Controller Model 60. When loaded with data (such as

prices), operator authorizations, and customer data (for check authorization, if desired), the 3660 system can perform all checkout functions on a closed in-store loop system. With BTAM support, the 3660 system attaches to an IBM 3704 or 3705 Communications Controller that runs in emulation mode. For detailed information on the IBM 3660 Supermarket System, see IBM 3660 Supermarket System Programmer's Guide, GC30-3025.

IBM 3735 PROGRAMMABLE BUFFERED TERMINAL

The 3735 Programmable Buffered Terminal is a stand-alone programmable terminal. The 3735 contains a communication interface and the controls necessary to use the BTAM BSC facilities to transmit properly assembled and structured Form Description programs to any terminal in the network and to receive messages and data from the 3735 terminals.

Detailed information about the facilities that the 3735 provides is found in IBM 3735 Programmer's Guide, GC30-3001, which describes the methods and facilities necessary to design, write, and generate form description programs.

IBM 3741 MODEL 2 DATA STATION, IBM 3747 DATA CONVERTER

BTAM supports binary synchronous communication with the IBM 3741 Data Station and the IBM 3747 Data Converter equipped with the BSC adapter for point-to-point switched or nonswitched lines. For information concerning the IBM 3741 Data Station and the IBM 3747 Data Converter, see the IBM 3741/3742 Data Stations Reference Manual, GA21-9151, and the IBM 3747 Data Converter Reference Manual, GA21-9153. Support under

BTAM is supplied by specifying UNIT=BSC1 (nonswitched point-to-point), UNIT=BSC2 (switched point-to-point), or BSC3 (multipoint) in the IODEVICE macro instruction. A detailed description of the macro instructions supported, ID verification, error recovery, and other programming information may be found in IBM 3740 BTAM/TCAM Programmer's Guide, GC21-5071.

IBM 3750 SWITCHING SYSTEM (WORLD TRADE USERS ONLY)

BTAM supports binary synchronous communication with the IBM 3750 Switching System on a nonswitched point-to-point line. BTAM

programming information for the 3750 is in 3750 Switching System Application Programmer's Guide, GC11-6000.

IBM 3780 DATA COMMUNICATION TERMINAL

The IBM 3780 is similar in operation to the IBM 2770 (see the section "IBM 2770 Data Communication System -- Programming Considerations"). The 3780, however, does not necessarily have component selection. A 3780 without component selection will fold and print component selection characters received on a point-to-point line. It will not respond if component selection characters are received on a multipoint line; therefore, each polling and addressing character sequence should be three characters long and omit the component selection character. A 3780 with component selection, required with the 3781 Card Punch, is similar in operation to a 2770 with component selection.

Information about the 2770 in the sections "BSC Read and Write Operations" and

"Online Testing" generally applies to the 3780.

For more information about 3780 system configurations, transmission codes, and operating characteristics (including polling and selection and message formats), see Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.

Note: BTAM supports the space compression/expansion feature of the 3780 only to the extent of passing messages containing compressed data. The application program must do the processing required to make the space compression/expansion feature operational.

IBM 5275 DIRECT NUMERICAL CONTROL STATION

BTAM supports the 5275 Direct Numerical Control Station in an identical manner to the 3275 Display Station. If the 5275 is connected on nonswitched lines, use the nonswitched 3270 discussion in "IBM 3270 Information Display System (Remote)" in this general section, "BSC Device-Dependent Considerations." If the 5275 is connected on switched lines, use the switched 3275 discussion. Additional information on programming the 3270 is described in Introduction to Programming the IBM 3270, GC27-

6999; in general, this information applies also to the 5275. The 5275 itself is described in 5275 Direct Numerical Control Station Introduction and Component Description, GA34-0016.

Only EBCDIC transmission code is supported with the 5275.

This information is for planning purposes only until the 5275 is available.

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

BTAM provides facilities called error recovery procedures (ERP) that diagnose a variety of error conditions that occur during message transmission, and attempt to recover from those conditions that are considered recoverable, so that transmission can continue. In addition to the diagnostic and recovery capabilities, error recovery procedures:

- Provide to the user program information about errors from which ERP could not recover, so that the program can act accordingly.
- Accumulate in special data sets, called the Statistical Data Recorder (SDR) and the Outboard Recorder (OBR), counts of certain kinds of errors; these are useful to the Customer Engineer in analyzing line and equipment troubles. Note: Timeout, data check, and intervention required are not recorded in OBR or SDR; they can optionally be recorded in the LERB.
- Notify the operator at the central computer console (or some other console, if the system has the Multiple Console Support facility) of certain kinds of errors.

In addition to these ERP-provided functions, BTAM provides the capability of accumulating in a set of counters, collectively called a line error recording block (LERB), running totals of certain kinds of error conditions, and printing these totals at the central computer console (or some other console, if the system has the Multiple Console Support facility). These capabilities are provided by the LERB and LERPRT macro instructions.

Error recovery procedures handle errors arising from conditions at remote stations, on communications lines, and at transmission control units. The action with which ERP routines respond to an error condition depends on the kind of error, the type of I/O command (Poll, Read, Write, etc.) being executed when the error occurred, and the type of remote station involved. ERP routines analyze the error condition, considering each of these factors, and performs the appropriate action. Error conditions are considered to be in one of two categories: irrecoverable and not irrecoverable. Errors that are not irrecoverable may be temporary or permanent. An irrecoverable error is one that is inherently incapable of being corrected by program action; that is, some form of human inter-

vention is required. (An exception to this is the buffer-unavailable condition, which may occur when dynamic buffering is in use and the buffer pool becomes depleted. This condition (indicated by bit 4 in DECFLAGS), though not recoverable by ERP, can be anticipated by the user program, which can request retransmission of the lost message.)

ERP does not attempt to recover such errors. When an irrecoverable error occurs, ERP sets indicators denoting the nature of the error in the data event control block (DECB) for the line involved in the I/O operation, notifies the operator at the central computer console by message IEA000I (see Appendix C for the format of this message), and records the occurrence of the error in the Outboard Recorder (OBR).

If an error is not inherently irrecoverable, ERP routines attempt to recover from it, usually by reissuing the I/O command for which the error occurred or issuing other appropriate I/O commands to clear the condition. If ERP routines succeed in clearing the condition, the error is said to be temporary. BTAM records its occurrence in the Statistical Data Recorder (SDR), and the user program continues normally. If ERP routines are unsuccessful in clearing the condition, the error is said to be permanent. BTAM records its occurrence in the Outboard Recorder (OBR), sets bits indicating the nature of the error in the DECB for the line involved in the I/O operation, and notifies the operator at the central computer console, by means of message IEA000I. Note: Timeout, data check, and intervention required are not recorded in OBR or SDR; they can optionally be recorded in the LERB.

BTAM error recovery procedures attempt to clear error conditions up to two times, for operations on start-stop lines, and up to seven times on BSC lines. Thus, permanent error conditions are indicated in the DECB after the I/O operation has been attempted three times (start-stop) or eight times (BSC). Irrecoverable errors are indicated in the DECB after only one attempt, since such errors are not retried.

Once an error condition has occurred and the ERP facility is trying to recover from it, any subsequent errors that may occur during the retries are not indicated in the DECB to avoid obscuring the original error condition. Neither are subsequent errors recorded in the SDR, OBR, or LERB, nor are they indicated to the console operator. This ensures that only errors occurring during I/O operations issued by the user program are recorded.

Error recovery procedures are divided into basic functions and additional functions.

BASIC FUNCTIONS

- When an error occurs during an I/O operation that does not involve transmission of message text (these are called nontext errors), ERP retries the operation.
- ERP records each occurrence of a temporary error in the Statistical Data Recorder, and each occurrence of a permanent or irrecoverable error in the Outboard Recorder.
- ERP provides, in the data event control block for the line involved, information on (1) permanent errors, (2) irrecoverable errors, and (3) errors occurring during transmission of message text (called Read Text and Write Text errors) for which ERP does not attempt recovery. Recovery of Read Text and Write Text errors is an additional ERP function, described below.
- ERP sends a message to the operator at the console of the central computer (or some other console, if the system has the Multiple Console Support facility) for each permanent and irrecoverable error.

Basic ERP functions are optional for line groups involving start-stop terminals, but they are required for line groups involving BSC stations. To obtain these basic functions for start-stop terminals, code EROPT=E in the DCB macro for the line group, or omit the EROPT operand. To omit the basic functions, code EROPT=N (valid only for start-stop line groups).

Coding EROPT=N prevents certain ERP routines from being included in the system, with a consequent saving in storage space. It is recommended that EROPT=N be coded for line groups for AT&T 83B3 and WU 115A terminals, for the reasons given in the explanation of the EROPT operand in Figure 7.

ADDITIONAL FUNCTIONS

- When an error occurs during execution of a Read Text command (called a Read Text error), ERP will optionally retry the operation unless dynamic buffering is in use. This function is optional for the IBM 1050, 2740 (with the Record Checking feature), and 2260; it is unavailable for other start-stop ter-

minals; and it is always required for BSC stations.

- When an error occurs during execution of a Write Text command (a Write Text error), ERP will optionally retry the operation unless dynamic buffering is in use. This function is optional for start-stop terminals, and results in an additional copy of the message text for each retry (except for messages sent to a card punch or tape punch of an IBM 1050 with the Line Correction feature, or to a 2260 with the Line Address feature); it is always required for BSC stations.
- ERP will optionally record, in the line error recording block, each occurrence of a data check, timeout, or intervention required error.

These three additional functions are specified as follows. To provide recovery attempts for Read Text errors (start-stop terminals), code EROPT=R in the DCB macro for the line group involved; to provide for recovery from Write Text errors (start-stop terminals), code EROPT=W. These EROPT parameters are ignored if dynamic buffering is specified for the line group, or if they are specified for a type of terminal for which recovery is not available. (It is not necessary to explicitly specify recovery attempts for Read Text and Write Text errors for BSC stations; the function is always performed for BSC.)

To provide for recording of errors in the line error recording block, code EROPT=C in the DCB macro, code a LERB macro to define the LERB, and code the name of the LERB macro in the LERB operand of the DCB.

Note: Errors occurring during transmission of test messages by the online test facility are not recorded in the LERB.

LERB (Line Error Recording Block) Macro Instruction

LERB defines for each line in a line group an area of storage called a line error recording block. This block consists of a group of counters in which are kept cumulative totals of data check, intervention required, and nontext time-out errors, and of the number of transmissions. There are two sets of these counters for each communications line. The contents of one set, called the threshold counters, are incremented each time a transmission or an error occurs, until one of the counters (excluding the transmission counter) reaches its threshold value, which is determined by the LERB macro. When the threshold value is reached, the contents of all four threshold counters are printed at the console

of the central computer (message IEC801I; see Appendix C for the format). The contents of the threshold counters are added to the other set of counters, called cumulative counters or accumulators, and the threshold counters are reset to zero.

Note: When the transmission counter reaches its threshold, the contents of the threshold counters are added to the cumulative counters (or accumulators) and the threshold counters are reset to zero. Message IEC801I is not printed on the console.

The contents of the accumulators are printed, and both sets of counters are optionally reset to zero when you issue a LERPRT (Line Error Recording Print) macro instruction.

Name	Operation	Operand
symbol	LERB	nlines[,{([transmct] [,datack],[,intreq] [,nontto])}...]

symbol
is the name of the first line error recording block defined by this macro. It must be specified; it must also be coded in the LERB operand of the DCB macro for the line group.

nlines
specifies the number of lines composing the line group for which the LERB macro is defining these blocks. The allowable range of values is 1 to 255.

transmct
specifies the transmission count threshold: the number of consecutive transmissions that when reached causes the contents of the threshold counters to be added to the cumulative counters and the threshold counters are reset to zero. Message IEC801I does not appear on the console when the transmission threshold is reached. You may specify from 1 to 255; if you omit the operand, 255 is assumed.

datack
specifies the data check error count threshold: the number of data checks that when reached causes the contents of the threshold counters to be printed, and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 10 or the value of transmct, whichever is lower, is assumed.

intreq
specifies the intervention required error count threshold: the number of intervention required errors that when

reached causes the contents of the threshold counters to be printed, and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 5 or the value of transmct, whichever is lower, is assumed.

nontto
specifies the nontext timeout error count threshold: the number of nontext timeout errors that when reached causes the contents of the threshold counters to be printed and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 5 or the value of transmct, whichever is lower, is assumed.

Of these operands, only nlines is mandatory. If you code only this operand, the default threshold counts of 255, 10, 5 and 5 are assumed for each of the lines in the line group. If you code threshold counts for one line, the same counts are assumed for all successive lines for which you do not code threshold counts; the counts are effective until you code another set of threshold counts.

Example:

```
LG1LERB LERB 10,,,(200,20,,7),,,
          (240,20,25,10)
```

This LERB macro provides line error recording blocks for ten lines:

- The first two lines have assumed threshold values of 255 (transmct), 10 (datack), 5 (intreq), and 5 (nontto).
- The next three lines have explicit transmct, datack, and nontto threshold values of 200, 20, and 7, and an assumed intreq threshold value of 5.
- The remaining five lines have explicit threshold values of 240, 20, 25, and 10.

Omit the parentheses if you omit all the suboperands (transmct, etc.) for a line, as has been done for lines 1, 2, 4, 5, 7, 8, 9 and 10 in the example. Also omit trailing commas (as for lines 7, 8, 9, and 10 above).

LERPRT (Line Error Recording Print) Macro Instruction

LERPRT causes the current contents of the cumulative counters (accumulators) for a line to be printed at the console of the central computer or, in systems with Multiple Console Support, at some other user-designated console. (The message number is IEC802I; see Appendix C for the format.) As explained under the LERB macro, the contents of the four threshold counters for the line are added to the accumulator each

time one of the threshold counters reaches its threshold value. The contents of the threshold counters are also added to the accumulators (and the threshold counters are reset) when you issue the LERPRT macro. You may also specify in LERPRT that the threshold counters and accumulators for the line be reset to zero.

CAUTION: Issue a LERPRT macro for a line only if the line group containing that line is open.

Name	Operation	Operand
[symbol]	LERPRT	dcbaddr[,rln][,cid] [,CLEAR=YES] [,CLEAR=NO]

dcbaddr

specifies the address of the DCB for the line group containing the line for which the contents of the accumulators are to be printed.

rln

specifies the relative line number of the line involved. rln cannot exceed the number of lines contained in the line group (as indicated in the nlines operand of the LERB macro). If you omit this operand, all nonzero values of the accumulators for all lines in the group are printed.

cid

(applicable only to systems having the Multiple Console Support facility)

specifies the address of a byte containing the identification of the console at which the message is to be printed. If you omit this operand when the system includes the multiple console support facility, the message is printed at consoles having a routing code of 8. If you specify an invalid console ID, the message is printed at consoles having a routing code of 1.

CLEAR=YES

specifies that the threshold counters and accumulators be reset to zero after the contents of the accumulators are printed. If you omit this operand, CLEAR=YES is assumed.

CLEAR=NO

specifies that the threshold counters and accumulators are not to be reset after the contents of the accumulators are printed.

ERROR DETECTION AND ANALYSIS

To determine what action to take when an error occurs requires that the error condi-

tion be analyzed. This is true whether error recovery is to be performed by ERP routines or by the user program. If BTAM ERP is used, all error analysis and recovery attempts are performed before the Read or Write operation is posted as complete in the event control block (ECB) for the line. All user program analysis and recovery attempts, whether in addition to BTAM ERP actions or in place of them, occur after the Read or Write operation is posted as complete. Thus the user program analysis and recovery routine should receive control following each completion of a Read or Write operation.

As mentioned previously, BTAM sets various error indicators in the DECB. Some of these are set before ERP routines gain control, others are set by the ERP routines. The DECB fields containing error indicators are:

- **DECSDECB.** This is the address of the event control block, which is the first fullword of the DECB. The first byte contains the completion code for the operation.
- **DECFLAGS.** A one-byte field containing flags that BTAM sets at the conclusion of a Read or Write operation. Some of these flags represent not errors but exceptional conditions of interest to the user program. These flags are set regardless of whether the operation was completed successfully.
- **DECERRST.** A one-byte field containing flags indicating the kind of I/O error that occurred. These flags are set only in the event of a true I/O error (as opposed to an exceptional condition) as indicated in the event control block by a completion code of X'41'.
- **DECCSWST.** A halfword field containing the status indicators set in the channel status word (CSW) at the conclusion of the Read or Write operation.
- **DECSNSO.** A one-byte field containing the sense information returned by the transmission control unit at the conclusion of the I/O operation that resulted in a Unit Check error.

Refer to Appendix B for the meaning of the various bits settings in these fields.

USER PROGRAM ANALYSIS PROCEDURE

A recommended procedure for checking the results of a Read or Write operation is as follows.

First determine whether the operation was completed successfully or unsuccessfully. A completion code of X'7F' in the event control block (DECSDECB) indicates

successful completion, and a code of X'41' indicates unsuccessful completion. A third completion code, X'48', indicates that the Read or Write operation was halted as a result of issuing a RESETPL or CLOSE macro while the operation was in progress.

Normal Completion

If the completion code is X'7F', check the DECFLAGS field. If it contains all zeros, no exceptional condition has occurred, and the program can execute whatever Read or Write operation would normally follow the completed operation. If DECFLAGS does not contain all zeros, check each of the appropriate flags to determine the exceptional condition, then take suitable action.

Abnormal Completion

If the completion code is X'41', determine the nature of the I/O error as follows. Check the DECFLAGS and DECERRST fields. If DECERRST does not contain all zeros, check the appropriate bits from among bits 0-4. (Bit 2 is applicable unless BTAM ERP routines are not present in the system; bit 3 is applicable only if the TCU is an IBM 2701.)

If both DECFLAGS and DECERRST contain all zeros, check the bits in the CSW status field, DECCSWST, in the order indicated in the priority column of Figure 59. If the Unit Check bit is on, check the sense bits in DECSSENS0 in the order indicated in Figure 60. Then take appropriate action. These last two fields, DECCSWST and DECSSENS0, contain the same information as was checked by the ERP routines, if the system includes the ERP facility.

The preceding discussion mentioned checking the "appropriate" bits; this simply means those bits that could possibly be set in a given situation. For example, it would be appropriate to check bit 0 of DECFLAGS only if you were analyzing the result of an operation on a BSC line, as bit 0 is not used in start-stop operations.

Many factors are involved in determining what action is suitable in response to a given kind of error or exceptional condition.

For exceptional conditions occurring in conjunction with a successful completion of a Read or Write operation, the question is not of retrying an operation that failed, but rather of determining what kind of Read or Write operation should be executed next.

Priority	CSW Bit	Condition	Remarks
1	45	Channel Control Check	
2	46	Interface Control Check	
2	44	Channel Data Check	
3	32	Attention	
3	33	Status Modifier	
3	34	Control Unit End	
3	35	Busy	
4	38	Unit Check	Check Sense Bits
5	47	Chaining Check	
6	42	Program Check	Program Error
6	43	Protection Check	Program Error
7	39	Unit Exception	Possible Error
8	41	Incorrect Length	Possible Error

Figure 59. User Program Status Analysis

Priority	Sense Bit	Condition
1	3	Equipment Check
2	6	Lost Data
3	7	Timeout
4	1	Intervention Required
5	2	Bus Out
6	4	Data Check
7	5	Overrun
8	0	Command Reject

Figure 60. User Program Sense Byte Analysis

Some of the factors to be considered in determining what program action to take in the event of an error or exceptional condition are the type of Read or Write operation (for example Read Initial, Write Continue) and the specific command (for example Poll, Read Text, Write Response) being executed when the error occurred, and the type of remote station or line involved. The characteristics of the application determine other factors to be considered. In any event, it is inadvisable to pursue the same procedure that the BTAM ERP facility does, for any given error condition. The tables and explanations are intended to show what BTAM ERP has already done to recover or to guide you in writing your own recovery procedures; they are not intended to suggest user analysis actions after error recovery has been attempted. The next section suggests, for BSC lines, appropriate macros to issue after BTAM posts exceptional or error conditions.

In general, BTAM error recovery procedures anticipate all of the possible conditions from which recovery may be possible, and in each case takes appropriate action to achieve recovery. For this reason, use of the BTAM-provided error recovery procedures is highly recommended, since it saves much coding effort for the application programmer. In many applications, the appropriate action for the user program to take when ERP fails to recover is simply to ignore the error condition and resend the same message text with the next Read or Write operation, or to defer further operations on the line until an equipment failure or abnormal condition can be corrected. Usually, it is appropriate for the program to notify the operator at the central computer console or teleprocessing console of the condition.

The remainder of this chapter provides detailed explanations of what actions BTAM error recovery procedures perform for each of the error and exceptional conditions, and for each of the commands and types of remote stations for which the error can occur. This information is provided for users wishing to write their own error analysis and recovery routines, for use either in place of or in addition to the BTAM-provided ERP facility. As mentioned previously, an error analysis routine usually must consider the type of Read or Write operation and the type of channel command on which the error or exceptional condition occurred. Three fields in the DECB contain this information:

- DECTYPE indicates the kind of Read or Write operation being executed.

Note: Certain error conditions cause BTAM ERP to modify the DECTYPE field,

so that this field has a different value at the end of the Read or Write operation from the value it initially had. No other user-specified DECB field is modified by ERP.

- DECCMCO indicates the specific type of channel program command on which the error occurred.
- DECTPCOD indicates the TP Operation Code associated with the command. Each BTAM channel command contains this code, in byte five of the channel command word (CCW). This code, which is not present in CCWs for other (non-TP) environments, has no effect on channel operations. It is in effect an extension of the command code, and identifies the purpose of the command.

The meanings of the bits in each of these fields are given in Appendix B. In these descriptions, and in the descriptions of the BTAM ERP actions below, references are made to types of I/O operations, equipment conditions, and indicators that are not defined elsewhere in this publication, as they relate to the operational details of specific types of transmission control units and related equipment. Understanding of these references and writing of an error recovery routine requires a knowledge of the functional complexities of this equipment, which is beyond the scope of this publication to impart. Information on transmission control units may be found in the publications listed under that heading in the Preface of this manual.

BTAM ERP ERROR ANALYSIS AND RECOVERY ACTIONS

BTAM routines analyze error and exceptional conditions by examining the status field of the channel status word (CSW) and, in the case of a Unit Check error, by examining the sense information provided by the TCU at the end of the I/O operation.

More than one indication may be set in the CSW when an error condition occurs. Generally, only one of these indicators properly describes the condition; other indicators reflect secondary effects. Similarly, TCU errors can cause more than one sense bit to be set. To ensure that the primary condition is recognized and acted upon, priority schemes determine the order in which status and sense bits are tested.

Tables and explanations of actions are shown separately for start-stop and BSC operations, as the actions differ considerably for these two categories.

"Should-Not-Occur" Errors

Among the error conditions causing BTAM to pass control to ERP are those which are undefined for a particular command or combination of hardware conditions. An overrun error, for example, has no meaning for a Write command.

In some instances, however, ERP can recover even from undefined errors. Consider, for example, a lost data error occurring on a Dial command issued for a start-stop line. Although this condition, like lost data on a Write command, is not defined (and is therefore not mentioned in the publications for the IBM 2701, 2702, and 2703), ERP does try to recover by reexecuting the Dial command (see Figure 65 and ERP action 20.) After two unsuccessful retries, ERP disables the line, notifies the operator, and records the error in the Outboard Recorder.

Note that ERP does not attempt recovery when a lost data error occurs on a Write command; rather, ERP immediately indicates a "should-not-occur" error has occurred. In Figure 65, Write falls in the "all other commands" category; see action 11.

Figure 61 shows for various commands the error conditions OS/VS BTAM ERP considers to be "should-not-occur" errors and therefore do not attempt recovery.

START-STOP ERROR RECOVERY PROCEDURES

In Figures 64-71, parentheses following the name of a command contain the TP operation code of the command, as appearing in the DECTPCOD field of the DECB at completion of a Read or Write operation.

Error Condition	Write	Read	Inhibit	Prepare	Sense	Disable	Enable	Dial	Poll
Lost Data	•			• ¹		•	•		•
Timeout	• ²		•		•				
Intervention Required					•	•	•		
Bus Out Check		•	•	•	•	•	•		
Data Check	• ³			•	•	•	•	•	•
Overrun	•			•	•	•	•	•	• ⁴
Unit Exception				• ⁶	•	•	• ⁶	• ⁵	

¹Should-not-occur error only for start-stop lines.

²Not a should-not-occur error for any Write command that sends transparent text (BSC lines) or for a 2260/2848 (start-stop lines).

³Should-not-occur error only for Write operations on BSC lines.

⁴Should-not-occur error only if TCU is a 2701 for start-stop lines; always a should-not-occur error for BSC lines.

⁵Should-not-occur error only if TCU is a 2701.

⁶Not a should-not-occur error if TCU is a 2703, and user program issued RESETPL macro.

Figure 61. Should-Not-Occur Error Conditions Posted by BTAM ERP

CSW Bit	Condition	Remarks		Action
45	Channel Control Check		1	1
46	Interface Control Check		2	1
44	Channel Data Check		2	1
32	Attention		3	2
33	Status Modifier		3	2
34	Control Unit End		3	2
35	Busy		3	2
38	Unit Check	Check Sense Bits	4	See Figure 63
47	Chaining Check		5	3
42	Program Check	Program Error	6	4
43	Protection Check	Program Error	6	4
39	Unit Exception	Possible Error	7	See Figure 64
41	Incorrect Length	Possible Error	8	11

Figure 62. Status Analysis -- Start-Stop

Priority	Sense Bit	Condition	Action
1	3	Equipment Check	5
2	6	Lost Data	See Figure 65
3	7	Timeout	See Figure 66
4	1	Intervention Required	See Figure 67
5	2	Bus Out	See Figure 68
6	4	Data Check	See Figure 69
7	5	Overrun	See Figure 70
8	0	Command Reject	See Figure 71

Figure 63. Sense Byte Analysis -- Start-Stop

Command	Action
Write or Poll (03)	
2741	7
83B3 or 115A (Telegraph adapter type I)	8
all other terminals	9
Read	
Read response to addressing (06)	
Terminal adapter type I	
2740	10
all other terminals	11 (SNO)
All other types of adapters	11 (SNO)
Read response to text (20)	
2740 with station control & checking, or 2260	
Write text retries specified (EROPT=W)	12
Write text retries not specified	13
all other terminals	11 (SNO)
All other Reads	11 (SNO)
All other commands	11 (SNO)

Figure 64. Unit Exception -- Start-Stop

Command	Action
Read	
Read ID response (07)	14
Read response to addressing (06)	15
Read Text (11)	
If residual count = 0	
83B3 or 115A (Telegraph adapter type I)	16
all other terminals	17
If residual count not = 0	18 (SNO)
Read response to text (20)	19
Dial (01)	20
All other commands	11 (SNO)

Figure 65. Lost Data -- Start-Stop

Command	Action
Write	
2260 or 2265 (Terminal Adapter Type III)	21
all other terminals	22 (SNO)
Read (or Inhibit)	
Read response to addressing (06)	
83B3	23
all other terminals	24
Read ID response (07)	25
Read response to polling (05)	
First command is Disable	23
First command is not Disable	25
Read Index (0A)	
First command is Disable	23
First command is not Disable	26
Read text (11)	27
Read response to text (20)	
Write text retries specified (EROPT=W)	
Operation is Write Continue	13
Operation is not Write Continue	
First command is Disable	28
First command is not Disable	24
Write text retries not specified	13
Disable (01)	29
Enable (01)	
preceded by Disable	24
not preceded by Disable	30 (SNO)
Prepare (01)	25
Dial (01)	31
Poll (03)	32
All other commands	33 (SNO)

Figure 66. Timeout -- Start-Stop

Command	Action
Write or Break (23)	34
Read (or Inhibit)	
Read Index (0A)	26
Read response to polling (0A)	
Operation is Write TTV	35
Operation is not Write TTV	36
Read response to text (20)	
Read text retries specified (EROPT=R)	
Operation is Write Continue	37
Operation is not Write Continue	24
Read text retries not specified	37
All other Reads	39
Dial (01)	40
Prepare (01)	
2741	61
All other terminals	36
Poll (03)	24
All other commands	41 (SNO)

Figure 67. Intervention Required -- Start-Stop

Command	Action
Write	
prior to Write text	46
Write text (11)	
Write text retries specified (EROPT=W)	
Failing command is last command or next command is not Read response to text	46
Next command is Read response to text	45
Write text retries not specified	44
following Write text	45
Dial (01)	46
Poll (03)	46
All other commands	11 (SNO)

Figure 68. Bus Out Check -- Start-Stop

Command	Action
Write	
83B3, 115A, or TWX Model 33,35 Telegraph adapter type I or II	
Error occurred during text transfer	
Write retries specified (EROPT=W)	24
Write retries not specified	39
Error did not occur during text transfer	24
World Trade Telegraph terminals (WTT Adapter)	37
Read (or Inhibit)	
Read ID response (07)	48
Read response to polling (0A) or any other Read command prior to text transfer	
First command in channel program is Disable	49
First command in channel program is not Disable	50
Read text (11)	
Dynamic buffering is used	51
Dynamic buffering is not used	
Read text retries specified (EROPT=R)	
Operation is Read Repeat	24
Operation is Write Continue Conversational	53
All other operations	54
Read text retries not specified	51
Read response to text (20)	
Operation is Write Continue	48
All other operations	52 (SNO)
Remote terminal is 2740 with 2760 feature, and operation is Write TIO, TCO, or TVO	55
Poll (03)	56
All other commands	52 (SNO)

Figure 69. Data Check -- Start-Stop

Command	Action
Read	
Read Index (0A)	57
Read response to text (20)	58
All other Reads	
Dynamic buffering is used	59
Dynamic buffering is not used	
Read text retries are specified (EROPT=R)	60
Read text retries are not specified	59
Poll (03)	
TCU is 2702	57
TCU is not 2702	11 (SNO)
Inhibit (11)	
Read text retries are specified (EROPT=R)	60
Read text retries are not specified	59
All other commands	11 (SNO)

Figure 70. Overrun -- Start-Stop

Command	Action
Poll (03)	11 (SNO)
All other commands	9

Figure 71. Command Reject -- Start-Stop

ERROR RECOVERY ACTIONS FOR START-STOP OPERATIONS

In the descriptions of the actions performed by BTAM error recovery procedures, the phrase "if applicable," applied to the action of recording occurrences in the line error recording block (LERB), means that the error is so recorded only if it is a data check, intervention required, or non-text time-out error, and if you have specified that errors be recorded by appropriately coding the EROPT and LERB operands of the DCB macro instruction.

The phrase "ERP notifies the console operator" means that ERP writes message IEA000I to the console of the central computer or to some other console if the Multiple Console Support facility is in use. This allows the operator to take whatever action is necessary to correct the condition. See Appendix C for the format of this message.

The phrase "ERP posts the operation as complete-with-error" means that ERP sets the post flag in the ECB for the line to indicate conclusion of the operation, and sets a completion code of X'41' in the ECB, to indicate that an I/O error occurred.

- 1 If the Channel Check Handler (CCH) of the Recovery Management Support facility has been included in the operating system during system generation, ERP forces a permanent error condition by setting the Unit Check and Equipment Check sense bits in the sense byte. This causes ERP for this condition to notify the console operator. Operations can proceed normally for line groups using other channels. If the CCH facility is not included in the operating system, the condition is recorded in the System Environment Recorder (SER) and the Supervisor enters Wait state, as the error is too serious to allow further operations.

- 2 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 3 If the error occurred on a Read command, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR. If the error did not occur on a Read command, ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 4 ERP notifies the console operator. The error is probably a program error.
- 5 ERP notifies the console operator. The error is caused by control unit failure, and the communications line involved should be considered inoperative.
- 6 ---
- 7 ERP restarts the channel program at the Write or Poll command. On the third occurrence of the error, ERP posts the operation complete-with-error and records the occurrence in the OBR.
- 8 ERP executes a Break command, then restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error indicates that data is being received from the line without a command.
- 9 ERP executes a Read Skip command. If the Read Skip is successful, ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error indicates that data is being received from the line without a command.
- 10 The Unit Exception condition is normal for a 2740 and indicates receipt of a positive or negative response (circle Y or circle N). If circle Y was received, ERP turns off the Unit Exception bit and restarts the channel program at the next command. If circle N was received, ERP posts normal completion with or without error. Receipt of a character other than circle Y or circle N is a should-not-occur condition; ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error.
- 11 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 12 ERP executes a Write Continue channel program to resend the same message text. On the third occurrence of this error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error indicates a buffer overflow.
- 13 ERP posts the operation complete-with-error.
- 14 ERP restarts the channel program at the failing command. On the third occurrence of this error, ERP disables the line, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 15 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 16 ERP executes a Break command, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error indicates that the input message was larger than the input area specified in the READ macro.
- 17 ERP executes a Read Skip command, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error indicates that the input message was larger than the input area specified in the READ macro.
- 18 ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error.
- 19 ERP executes a Read Skip command, posts the operation complete-with-error, and notifies the console operator, and records the occurrence in the OBR.

20 ERP restarts the channel program at the Dial command. On the third occurrence of the error, ERP disables the line, posts the operation complete-with-error, notifies the console operator, and records the error in the OBR.

The error indicates that the Dial command was sent to a line that was already in the "off-hook" condition.

21 ERP builds and executes a Reset channel program. The error occurrence is recorded in the LERB (if applicable).

22 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR. The occurrence is recorded in the LERB (if applicable).

23 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

24 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

25 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

26 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

27 ERP posts the operation complete-with-error. The error indicates that no text was received or that the elapsed time between successive text characters exceeded about 28 seconds (the intercharacter timeout interval).

28 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete-with-error.

29 ERP restarts the channel program at the failing command. On the third

occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

The error indicates that the data set (modem) is failing to disconnect.

30 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator.

31 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence of the error is recorded in the LERB (if applicable).

The error indicates that the remote station is not answering, when dialed, in the time allotted.

32 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

The error indicates that no response was received from the remote station.

33 ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable).

34 ERP sets the retry count to maximum and posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable).

The error indicates that the addressed line has not been enabled.

35 ERP restarts the channel program at the third command to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

36 ERP restarts the channel program at the beginning. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

- 37 ERP posts the operation complete-with-error and notifies the console operator. The occurrence is recorded in the LERB (if applicable).
- 38 ---
- 39 ERP posts the operation complete-with-error and notifies the console operator.
- 40 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator.
- The error indicates that the Automatic Calling Unit power is off or that the addressed line is not connected to an Autocall adapter.
- 41 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator. The occurrence is recorded in the LERB (if applicable).
- 42 ---
- 43 ---
- 44 ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error is a parity error within either the command or the text data.
- 45 ERP records the error occurrence in the SDR and restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error is a parity error within either the command or the text data.
- 46 ERP records the error occurrence in the SDR and restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- The error is a parity error within either the command or the text data.
- 47 ---
- 48 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 49 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 50 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 51 ERP posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 52 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator. (See Note 1.)
- 53 ERP restarts the channel program at the Read Conversational part. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 54 ERP builds and executes a Read Repeat channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 55 ERP restarts the channel program at the Write EOA PRE o command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)
- 56 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error. Each occurrence is recorded in the LERB (if applicable).
- 57 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with--

error and notifies the console operator.

- 58 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator.
- 59 ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 60 ERP builds and executes a Read Repeat channel program and records the error occurrence in the SDR. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.
- 61 ERP issues Write EOT and restarts the channel program at the failing command.

Note 1: The error is one of the following:

- A VRC (parity) error was detected in one or more of the received characters.
- An LRC error was detected; that is, the LRC character received from the remote station did not match the LRC value generated by the transmission control unit.
- A negative response was received as a response to text.
- The communication line was in the "space" condition at stop-bit time, indicating that the TCU was out of synchronism.

BSC ERROR RECOVERY PROCEDURES

The BTAM Channel End/Abnormal End Appendage receives control from the supervisor following an I/O interruption or after an ERP routine issues a SVC 15 with no retry specified (that is, with bit 2 of IOBFLAG1 set to zero). When it receives control, the appendage makes an analysis of such things as the CSW information, the condition code, the operation in progress, the response received, etc., in order to determine the specific action to be performed based on the conditions existing.

The appendage passes control back to the supervisor as follows:

- At 0 + Register 14 - the channel program is posted complete, and the request element is made available. This is the so-called "normal return."

- At 4 + Register 14 - the channel program is not posted complete, but the request element is made available. This return is made for online test or when a SAD (Set Address) or Enable error occurs.
- At 8 + Register 14 - the channel program is not posted complete, and its request element is placed back on the request queue so the program can be retried. This return is used when the channel program is to be restarted.

The supervisor then determines where control is to be passed next; if the IOB exception bit is on (bit 5 in IOBFLAG1=1) and the DCB indicates that basic error recovery procedures are provided (that is, bit 7 in DCBERROP=0) for this line group, control is passed to the BSC ERP control routine. The control routine also receives control from the supervisor following an I/O interruption when an ERP channel program has been initiated, and the ERP routines, rather than the Channel End/Abnormal End Appendage, are to analyze the results of the operation.

The BSC ERP control routine determines which ERP routine is to receive control, causes it to be loaded if necessary, and transfers control to it. Recovery actions, as indicated in the tables following, are then performed.

ERROR RECOVERY ACTIONS FOR BSC OPERATIONS

By using Figure 72, you can locate the set of conditions for which you wish to determine the ERP action. In some cases, it will be necessary to consult more than one table to trace the complete sequence of the actions, as when ERP sets up a special return code to indicate the existence of a specific situation as it goes through a multistep recovery procedure.

Figure	Description
72	Status Analysis
73	Channel Data Check
74	Equipment Check
75	Command Reject
76	Sense Byte Analysis
77	Bus Out
78	Overrun
79	Intervention Required
80	Data Check
81	Lost Data
82	Timeout
83	Unit Exception
84	Special Return Codes
85	Error Post Actions

Figure 72. Index to BSC ERP Tables

You should note that the tables summarize the actions performed by ERP; details such as incrementing the retry count and testing for a need for LERB recording are not shown.

In the tables, the values in parentheses following each command, for example, Read Text (11), is the TP operation code for that command, in hexadecimal.

Where the phrase "proceed with error posting" appears, see Figure 86 to determine the actions taken by the Error Post routine.

Figure 73 serves as an entry point for tracing the ERP-initiated recovery actions. In many cases, it refers to another table for further definition of recovery actions. Figures 85 and 86 indicate the result of control being passed to the Special Return routine and the Error Post routine, respectively.

Two examples of the use of the tables are as follows:

Example 1: The accumulated block check character (bcc) does not match the bcc received following the ETB or ETX ending a text block, on a Read Text command, causing a status indication of Channel End/Device End/Unit Check, with Data Check indicated in the sense byte.

Step 1 - Refer to the Unit Check section of Figure 73. Since Channel End and Device End are on, you are referred to Figure 77 to find further actions based on the results of the ERP analysis of the sense information.

Step 2 - Refer to Figure 77, which refers you to Figure 81.

Step 3 - Refer to Figure 81, where the ERP actions for various commands are described. Since the error occurred on a Read Text command, the ERP action taken depends on whether or not dynamic buffering is being used. Assuming that it is not, ERP will build a channel program to write NAK and then transfer-in-channel (TIC) back to the Read Text command. ("Failing CCW" refers to the CCW on which the interrupt occurred.)

Example 2: A transmission causes an incoming ETB or ETX character to be distorted so that it is not recognized as a control character, the bcc characters are

considered data characters and sent into main storage, exhausting the count in the Read Text CCW. The status indication is Channel End/Device End/Unit Check, with Lost Data indicated in the sense byte.

Step 1 - Same as in example 1.

Step 2 - Refer to Figure 77, which, for a Lost Data condition, refers you to Figure 82.

Step 3 - Refer to Figure 82, which indicates that, assuming dynamic buffering is not being used, a special return code (X'82') is set up in IOBWORK + 1 (one of the two locations in the Input/Output Block (IOB) where special codes are placed for later analysis by the Special Return routine). Then a channel program is generated to read the ENQ that the transmitting station will send when it does not receive a response to the block of text. When the channel program ends as a result of receiving the ENQ, the ERP Special Return routine is entered.

Step 4 - Refer to Figure 85, Part B, which indicates that ERP builds a channel program to write NAK and then restarts the original channel program at the Read Text command.

LOCAL 3270 DISPLAY SYSTEM ERROR RECOVERY PROCEDURES

ERRORS DETECTED BY THE DEVICE OR CONTROL UNIT AND CHANNEL DATA CHECK ERRORS

Error Conditions

Figure 87 lists error conditions according to the bits that are on in the CSW and sense byte.

Recovery Actions

Figure 88 indicates (by error condition and failing command) the recovery actions taken by BTAM error recovery procedures.

The recovery actions are:

1. Permanent error
The request is marked as a permanent

error condition. The operation is terminated, the error is logged in the system error log, and a message is issued to the console operator indicating the failing device, operation, and conditions.

2. Should not occur
The request is marked as a nonrecoverable error condition. The operation is terminated.
3. Nonrecoverable error
The request is marked as a nonrecoverable error condition. The operation is terminated. The error is logged in the system error log.
4. Intervention required
If the failing status includes Channel End, action 2 is taken. If Channel End is not present, a Write CCW is created to resend the Write Control character to print the buffer contents.
5. Retry failing CCW
Restart the channel program on the failing CCW. When the retry count exceeds the number for the error condition and command, action 1 is taken.
6. Retry channel program
Restart the channel program on the first CCW in the chain. When the retry count exceeds three, action 1 is taken.
7. Busy
The request is held until the device is ready. Then the operation is retried.

ERRORS DETECTED BY THE CHANNEL (EXCEPT CHANNEL DATA CHECK ERRORS)

Error Conditions

Figure 89 lists error conditions according to the bits that are on in the ERPCODES field, which is byte seven of the Error Recovery Procedure Interface Block (ERPIB) built by the Channel Check Handler.

Note: Channel data checks are handled as though they were device-detected errors. Channel control checks and interface control checks are processed only if the Channel Check Handler is in the system.

Recovery Actions

Figure 90 indicates (by error condition and failing command) the recovery actions taken by BTAM error recovery procedures. If a diagnostic command is found in the failing CCW chain, action 1 is taken.

The recovery actions are:

8. Permanent error
The device is unable to recover and the request is marked as a permanent error condition. The operation is terminated and the error is logged in the system error log. Unless the failing device is the console, a message is issued to the console operator identifying the failing device, operation, and conditions.
9. Should not occur
A message is issued to the console operator. Action 8 is taken.
10. Nonrecoverable error
A message is issued to the console operator. Action 8 is taken.
11. Retry failing command
The failing command is retried. If the retry count exceeds three, action 8 is taken. Note: a Write command is retried only if the second character in the data stream is an SBA. This is to ensure cursor integrity.
12. Conditional retry
Retry the failing command (taking action 8 if the retry count exceeds three), unless the command is the first in a CCW string (in which case action 10 is taken).
13. Sequence code 5 procedure
If the CSW was stored after a Start I/O instruction was issued, follow sequence code 3 procedure. If the CSW was stored after an I/O interruption and the device status is valid, take action 11.
14. TIC command
If the command preceding the TIC is command-chained, retry the failing CCW pointed to by the TIC (taking action 8 if the retry count exceeds three). If the command preceding the TIC is data-chained, retry the first CCW with data chaining in the CCW string.

Status	Action
Attention Status Modifier Control Unit End Busy	<ol style="list-style-type: none"> 1. Set "Should Not Occur" bit in DECERRST. 2. Update statistics table. 3. See Figure 86 for further actions.
Start I/O Condition Code = 1 (CSW Stored)	<ol style="list-style-type: none"> 1. Update statistics table. 2. See Figure 86 for further actions.
Channel Data Check	<ol style="list-style-type: none"> 1. Update statistics table. 2. See Figure 74 for further actions.
Program Check Protection Check Chaining Check	<ol style="list-style-type: none"> 1. Set indicator to cause recording of occurrence in the Outboard Recorder (OBR). 2. Notify the console operator.
Unit Check	<ol style="list-style-type: none"> 1. If Channel End and Device End status bits are both off, update the statistics table. See Figure 86 for further actions. 2. If Channel End and Device End status bits are <u>not</u> both off, analyze the sense information. See Figure 77 for actions resulting from this analysis.
Start I/O Condition Code = 3 (Not operational)	<ol style="list-style-type: none"> 1. Write operator message - "IEC0804A xxx CONTROL UNIT NOT OPERATIONAL, REPLY CONT OR POST" where xxx is the line address of the line involved. 2. If reply is "CONT" retry the failing channel program; if reply is "POST", post ECB complete with permanent I/O error; if reply is neither "CONT" nor "POST", repeat the message.
Unit Exception	This status bit can be turned on by equipment or by BTAM. See Figure 84 for action taken when this status bit is on.

Figure 73. Status Analysis -- BSC

Command	Action
Write Text (11)	Indicate a permanent I/O error, then proceed with error posting.
Any Write except Write Text	If retry limit (7) has been reached, proceed with error posting; if not, retry the failing CCW.
Read Text (11) (Dynamic Buffering Not Used)	If retry limit (7) has been reached, proceed with error posting; if not, Write NAK and TIC to the failing CCW.
Read Text (11) (Dynamic Buffering Used)	Indicate a permanent I/O error, then proceed with error posting.
Read ENQ (0B)	If retry limit (7) has been reached, proceed with error posting; if not, retry the failing CCW.
Read Response to ENQ (0C)	Write ENQ and TIC to failing CCW.
Read Response to Text (25)	Write ENQ and TIC to failing CCW.
All other Reads	Set up special return code X'80' in IOBWORK+1 and generate a channel program to read a response.

Figure 74. Channel Data Check -- BSC

Command	Action
Write Text (11) (Dynamic Buffering not used)	Write ENQ and TIC to the CCW following the failing CCW (the Read Response to Text command).
Read Text (11) (Dynamic Buffering not used).	Set up special return code X'80' in IOBWORK+1 and generate a Read response channel program.
Read or Write Text (11) (Dynamic Buffering used)	Indicate permanent I/O error, then proceed with error posting.
Any command not during text transfer (that is, any TP-Op code other than 11)	If retry limit has been reached, indicate a permanent I/O error and proceed with error posting; if not, restart the channel program.

Figure 75. Equipment Check -- BSC

Command	Action
Read Response to Text (25)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, set up special return code X'04' in IOBERRCT+1 and write DLE ENQ.
Any command with a Special Return code of X'04'	Check for Channel End and Device End status only. If both bits are on, clear the special return indicator and return to the supervisor; if not, indicate permanent I/O error, then proceed with error posting.
All other commands	Indicate permanent I/O error, then proceed with error posting.

Figure 76. Command Reject -- BSC

Sense Bit	Condition	Action
0	Command Reject	See Figure 76
1	Intervention Required	See Figure 80
2	Bus Out Check	See Figure 78
3	Equipment Check	See Figure 75
4	Data Check	See Figure 81
5	Overrun	See Figure 79
6	Lost Data	See Figure 82
7	Timeout	See Figure 83

Figure 77. Sense Byte Analysis -- BSC

Command	Action
Dial (01)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, disable the line and TIC to the first CCW.
Poll (03)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the first CCW and start polling with the failing station.
Write -- Prior to Text Transfer (TP-Op Code less than 10)	Check to see if this is a Write EOT command (TP-Op Code (02)). If so, restart the channel program at the failing CCW; if not, restart the channel program at the CCW following the failing CCW.
Write -- During Text Transfer (11)	Check to see if the residual count is equal to the original count. If so, restart the channel program at the failing CCW; if not indicate a permanent I/O error, then proceed with error posting.
All other Writes	Check to see if this is the last CCW in the channel program. If so, restart the channel program at the failing CCW; if not, restart the channel program at the CCW following the failing CCW.
Any Read Command	Check to see if the residual count is equal to the original count. If so, restart the channel program at the failing CCW; if not, indicate a permanent I/O error then proceed with error posting.
All other commands	Restart the channel program at the failing CCW.

Figure 78. Bus Out -- BSC

Command	Action
Read ENQ (08)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read Response to ENQ (0C)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the CCW preceding the failing CCW.
Read Response to Text (25)	If retry limit (7) has been reached, indicate I/O error, then proceed with error posting; if not, generate a channel program to Write ENQ, then TIC to the failing CCW.
Read Text (11)	If dynamic buffering is specified, indicate a permanent I/O error, then proceed with error posting; if not (and if retry attempts have not been exhausted), generate a channel program to Write NAK, then TIC to the failing CCW.
All other commands	Set "should not occur" bit in DECERRST, indicate a permanent I/O error, then proceed with error posting.

Figure 79. Overrun -- BSC

Command	Action
	Each of the actions described below is preceded by a test to see whether the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed. Except for the Dial command, the actions listed below apply only to a non-switched line. If an error occurs on any command other than Dial, for a switched line, ERP indicates a permanent error and proceeds with error posting.
Dial (01)	Generate a channel program to perform a Disable, then TIC to the failing CCW.
Prepare (01)	Restart the channel program at the first CCW.
Poll (03)	Restart the channel program at the first CCW and start polling with the failing station.
Read Text (11) (Dynamic Buffering Not Used)	Set up special return code X'81' in IOBWORK+1 and generate a Read ENQ channel program. If this a Read Initial operation on a multipoint line, set up for the Special Return routine to retry the Read using the second Read command (that is, the one following the Read Index); if not a Read Initial on a multipoint line, set up for it to retry the Read using the failing CCW.
Read Text (11) (Dynamic Buffering Used)	Indicate a permanent I/O error, then proceed with error posting.
Write Text (11)	If the residual count is equal to the original count, restart the channel program at the failing CCW; if not, indicate a permanent I/O error, then proceed with error posting.
All other commands	Restart the channel program at the failing CCW.

Figure 80. Intervention Required -- BSC

Command	Action
Read ENQ (0B)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read Response to ENQ (0C)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the CCW preceding the failing CCW.
Read ID Response (07)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the preceding CCW if it is a Write ID ENQ CCW. If other than a Write ID ENQ CCW, restart at the failing CCW.
Read Response to Text (25)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, generate a channel program to Write ENQ, then TIC to the failing CCW.
Read Text (11) (Dynamic Buffering Not Used)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, set up special return code X'82' in IOBWORK+1 and generate a channel program to Write NAK, then TIC to the failing Read CCW. (TIC to the previous CCW if it is a Read response to text; otherwise, TIC to the Read text CCW.)
Read Text (11) (Dynamic Buffering Used)	Indicate a permanent I/O error has occurred then proceed with error posting.
All other commands	Set "should not occur" bit in DECERRST, indicate a permanent I/O error has occurred, then proceed with error posting.

Figure 81. Data Check -- BSC

Command	Action
	Unless otherwise specified, each of the actions described below is preceded by a test to see whether the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed.
Dial (01)	Generate a channel program to perform a Disable, then TIC to the failing CCW.
Prepare (01)	Restart the channel program at the failing CCW.
Read ENQ (0B)	If the CCW is part of a Write Reset channel program set the retry count to seven, indicate permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read ID Response (07)	Set up a special return code X'83' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Read Response to ENQ (0C)	Generate a channel program to Read Skip and set up for later restart of the channel program at the CCW preceding the failing CCW.
Read Response to Text (25)	Set up special return code X'83' in IOBWORK+1 and generate a channel program to perform a Read Skip.
Read Text (11) (Dynamic Buffering Not Used)	Set up special return code X'82' in IOBWORK+1 and generate a channel program to Read ENQ (with count =2). If this command is part of a Write Conversational channel program, set up for the special return routine to restart the channel program at the CCW preceding the failing CCW instead of at the failing CCW. If this command is part of an Autopoll operation, set up special return of X'82' in IOBWORK+1 and generate a Write NAK, then TIC to the failing Read Text.
Read Text (11) (Dynamic Buffering Used)	The retry limit test is not performed. Set special return code X'1A' in IOBERRCT+1, set the retry count to seven, and generate a channel program to perform a Read Skip.
All other commands	The retry limit test is not performed. Set the "should not occur" bit (in DECERRST) and generate a channel program to perform a Read Skip. Set the retry count to seven and set special return code X'1A' in IOBERRCT+1.

Figure 82. Lost Data -- BSC

Command	Action
	Unless otherwise specified, each of the actions described below is preceded by a test to see if the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed.
Dial (01) Disable (01)	Restart the channel program at the failing CCW.
Read Response to EOT (0B) Read ENQ (0B)	<p><u>For a failing read ENQ command:</u></p> <p>If the operation is a Read Initial (X'01') for a switched line, restart the channel program at the Read ENQ command; if a Read Initial for a nonswitched line, restart the channel program at the preceding command.</p> <p>If the operation is a Read Initial Inquiry (X'19') restart the channel program at the preceding command.</p> <p>If the operation is a Read Inquiry (X'15'), and no retries are requested, set the retry count to seven, then proceed with error posting. If it is a Read Inquiry and retries are requested, restart the channel program at the failing command.</p> <p>If the Read ENQ appears in an operation other than one of the foregoing, restart the channel program at the failing command.</p> <p><u>For a failing Read Response to EOT command, (The operation is a Write Reset):</u></p> <p>Set the retry count to seven and proceed with error posting.</p>
Read Response to ENQ (0C)	Restart the channel program at the CCW preceding the failing CCW.
Read Response to polling (0A)	Set up a special return code of X'82' in IOBWORK+1 and generate a Write NAK, then TIC to the Read Text CCW that follows the read index. (The data area address of the Read Text CCW is decremented by 1, and the length is incremented by 1.) For dynamic buffering, set the RETRY count to 7, and proceed with error posting.
Read Response to Addressing (06)	Restart the channel program at the first CCW.
Read ID Response (07)	<p>If the maximum retry count <u>has not</u> been reached: If this is the calling station, restart the channel program at the CCW preceding the failing CCW; if it is the answering station, restart at the failing CCW.</p> <p>If the maximum retry count <u>has</u> been reached: If this is the calling station, set special return code X'8C' in IOEWORK+1 and Write DLE EOT, then disable the line; if this is the answering station, disable the line and TIC to the Enable CCW.</p>

Figure 83

Timeout -- BSC

Command	Action
Read Text (11) (Dynamic Buffering Not Used)	If this is a polling operation, adjust CCW address and count (if necessary) to keep index byte location from being overlaid, set up return code X'82' in IOBWORK+1 and generate a channel program to Write NAK, then TIC to the failing Read Text; if not autopoll operation, setup return code X'81' and generate a channel program to read ENQ followed by NOP.
Read Text (11) (Dynamic Buffering Used)	The retry limit test is not performed. Indicate a permanent I/O error, then proceed with error posting.
Read Response to Text (25)	Set special return code X'86' in IOBWORK+1 and generate a channel program to write ENQ, then TIC to the failing CCW.
Write Transparent Text (11)	Generate a channel program to Write DLE ENQ, then TIC to the failing CCW.
All other commands	The retry limit test is not performed. Set "should not occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.

Figure 83 (Part 2 of 2). Timeout -- BSC

Command	Action
Write ENQ (03)	If NAK or RVI was received, set improper response (X'40') in DECFLAGS and restart the channel program at the CCW preceding the failing CCW. If neither NAK nor RVI was received, set special return code X'84' in IOBWORK+1 and generate a channel program to Read ENQ, with count=2.
Write Response to ENQ (08)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Write Text (11)	Set special return code X'88' in IOBWORK+1 and generate a channel program to Read Response (with count=2) and TIC to the failing CCW.
Write Response to Text (08)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Write EOT (21)	If the operation is a write reset, restart the channel program at the next CCW; if not, set special return code X'87' in IOBWORK+1 and generate a channel program to Read Response, with count=2.
Write WACK (01)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count = 2).
All other Writes	Set "should-not-occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.
Poll (03)	If the failing CCW is the first one in the channel program, set special return code X'88' in IOBWORK+1 and generate a channel program to Read ENQ (with count=2); if not, set up to start polling, beginning with the failing station, and restart the channel program at the first CCW.
Any Read command	If this is the first time through ERP, perform the action described below for the specific type of Read command; if not, turn off the 'ERP-in-control' indicator and restart the channel program. If two consecutive RVIs are received, proceed with error posting.

Figure 84. Unit Exception -- BSC

Command	Action
Read ENQ (05)	If EOT was received, proceed with error posting; if not, determine whether the retry limit (7) has been reached. If it has, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read Response to ENQ (0C)	If the retry limit (7) has not been reached, and NAK or RVI was received, indicate improper response (X'40') in DECFLAGS and restart the channel program at the preceding CCW. If the retry limit has not been reached, and neither NAK nor RVI was received, set special return code X'84' in IOBWORK+1 and generate a channel program to Read ENQ, with count=2. If retry limit has been reached, indicate a permanent I/O error, then proceed with error posting.
Read ID Response (07)	<p>If the operation is a Write Connect: If ID NAK or an invalid ID was received, restart the channel program at the CCW preceding the failing CCW. If after 7 retries ID NAK is still received, post the operation normally. If after 7 retries an invalid ID is still received, Write DLE EOT and disable the line.</p> <p>If the operation is a Read Connect: If an invalid ID was received, restart the channel program at the failing CCW, until the retry count of 7 is reached; thereafter, disable the line. For any other condition, or any other character received, set 'should-not-occur' bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.</p>
Read Text (11)	<ol style="list-style-type: none"> 1. If ENQ was not the last character received, set "should-not-occur" bit (in DECERRST, indicate a permanent I/O error, then proceed with error posting. 2. If ENQ was received and dynamic buffering is being used, proceed with error posting. 3. If ENQ was the only character received, dynamic buffering is not being used, the CCW preceding the failing CCW is a Write Response to Text (TP-Op code 08), and the retry limit (7) has not been reached: Restart the channel program at the CCW preceding the failing CCW (that is, at the Write Response to Text CCW). 4. If ENQ was the only character received, dynamic buffering is not being used, the CCW preceding the failing CCW is a Read Response to Text (TP-Op Code 25), and the retry limit has not been reached: Generate a channel program to Write NAK and TIC to the CCW preceding the failing CCW (that is, at the Read Response to Text CCW, which is part of a Write Conversational channel program). 5. If ENQ was the only character received, dynamic buffering is not being used, and the CCW preceding the failing CCW is neither a Write Response to Text (08) or a Read Response to Text (25): Set the "should-not-occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.

Figure 84 (Part 2 of 3). Unit Exception -- BSC

Command	Action
Read Text (11) (continued)	<p>1. If ENQ preceded by one or more characters was received, dynamic buffering is not being used, and the retry limit has not been reached:</p> <p>Generate a channel program to Write NAK and TIC to the failing CCW.</p> <p>7. In 3, 4, and 6 above, if all conditions are met except that the retry limit has been reached, indicate a permanent I/O error, then proceed with error posting.</p> <p>8. If the original count, minus one, does not equal the residual count and an SOH % message was received, proceed with error posting.</p> <p>9. If STX ENQ was received and dynamic buffering was not specified, indicate X'01' in DECFLAGS, set special return code X'89' in IOBWORK+1, and generate a channel program to Write NAK and TIC to the failing CCW. If dynamic buffering was specified, proceed with error posting after indicating X'01' in DECFLAGS.</p>
Read Response to Text (25)	<p>If NAK was received: If dynamic buffering is used, proceed with error posting; if not, restart the channel program at the Write Text CCW.</p> <p>If ENQ was received: If the operation is a Write Inquiry, indicate contention (X'10') in DECFLAGS and restart the channel program at the failing CCW. If the operation is not a Write Inquiry, set special return code X'86' in IOBWORK+1 and generate a channel program to Write ENQ and TIC to the failing CCW.</p> <p>If neither ENQ nor NAK was received, set special return code X'86' in IOBWORK+1 and generate a channel program to Write ENQ and TIC to the failing CCW.</p>
All other commands	Set "should-not-occur" bit (in DECERRST), indicate a permanent I/O error has occurred, then proceed with error posting.

Figure 84 (Part 3 of 3). Unit Exception -- BSC

BTAM uses special return codes in two locations in the Input/Output Block (IOB) to cause ERP to perform the required functions in certain circumstances. The ERP control routine examines the code in IOBERRCT+1, and takes the actions shown in Part A of this table. The Special Return routine examines the code in IOBWORK+1 and takes the actions shown in Part B of this table.

Part A. Actions for special Return Codes in IOBERRCT+1

Code	Set:	Action
X'04'	When ERP writes DLE ENQ after detecting Command Reject.	If the Write DLE ENQ channel program completes normally, restart the user channel program at the Read Response to Text CCW that had previously ended with Unit Check and Command Reject. If the channel program completes with error, indicate a permanent I/O error, set the retry count to the maximum, 7, and proceed with error posting.
X'14'	(indicates that a special return code has been set in IOBWORK+1)	After the ERP-initiated channel program completes, the Special Return routine examines IOBWORK+1 to determine what further action to take, as shown in Part B of this table.
X'1A'	When ERP issues a Read Skip CCW after detecting a Lost Data condition, or when an error has occurred on an ERP Write CCW.	After the ERP-initiated channel program completes, ERP proceeds as follows: <ol style="list-style-type: none"> 1. If a timeout occurred following a Read Skip CCW, restart the channel program at the CCW that had ended with Unit Check and Lost Data indicated. 2. If an ERP CCW ends with sense bits other than Lost Data or Timeout on, set up to issue an I/O error message. 3. If an ERP Write CCW ends with an error, turn on the ERP-in-control indicator in the IOB, place code X'1A' in IOBWORK+1, and restart the channel program that had ended with the error that caused ERP to be initiated. 4. If an ERP CCW that is not a Write or a Read Skip ends with a Timeout, restart the ERP channel program at the beginning.
X'00'	<ol style="list-style-type: none"> 1. Set before posting completion or returning to the supervisor. 2. Set when a Write DLE ENQ CCW was performed successfully after a Read Response to Text CCW ended with Command Reject (possibly indicating that the BSC adapter in the TCU was still in transparent mode). 	The ERP control routine proceeds with its own analysis of the condition code, status, and sense information to determine the action to be taken, rather than being forced to pass control to a specific ERP routine.

Figure 85. (Part 1 of 5). Special Return Codes -- BSC

Part B. Actions for Special Return Codes in IOBWORK+1		
Code	Set:	Action
X'80'	When Equipment Check is detected on a Read Text CCW.	<p>If ENQ was received:</p> <p>(for multipoint line) Clear the special return indicators, resend the last-sent acknowledgment, and TIC to the failing CCW.</p> <p>(for nonmultipoint line) Clear the special return indicators and restart the channel program at the failing CCW.</p> <p>If ENQ was not received:</p> <p>(if retry limit has been reached) Clear the special return indicators, then proceed with error posting.</p> <p>(if retry limit has not been reached) Restart the ERP channel program at the beginning.</p>
X'81'	<ol style="list-style-type: none"> 1. When a Timeout on a Read Text CCW is detected (multipoint line). 2. When an Intervention Required error occurs on a Read Text CCW. 	<p>If ENQ was received:</p> <p>Clear the special return indicators and generate a channel program to Write NAK and TIC to the failing CCW.</p> <p>If ENQ was not received:</p> <p>(if retry limit has been reached) Clear the special return indicators and proceed with error posting.</p> <p>(if retry limit has not been reached) Restart the ERP channel program at the beginning.</p>
X'82'	<ol style="list-style-type: none"> 1. When lost data occurs on a Read Text CCW. 2. When data check occurs on a Read Text CCW. 3. When time out occurs on a Read Text CCW. 4. When time out occurs on a Read Index CCW. 	<p>For Autopoll, if the recovery was successful, clear special return indicators and indicate normal completion. Otherwise polling will be restarted. For non-autopoll, clear special return indicators and generate a channel program to Write ENQ and TIC to the failing CCW.</p> <p>For Autopoll, if the recovery was successful, clear special return indicators and indicate normal completion. Otherwise polling will be restarted. For non-autopoll, clear special return indicators and indicate normal completion.</p> <p>For Autopoll, if recovery was successful, clear special return indicators and indicate normal completion. Otherwise polling will be restarted. For non-autopoll, clear special return indicators and indicate normal completion.</p> <p>If recovery was successful, clear special return indicators and indicate normal completion. If unsuccessful, polling will be restarted.</p>
X'83'	When Lost Data occurs on a Read ID Response or Read Response to Text CCW.	Clear special return indicators and generate a channel program to Write ENQ and TIC to the failing CCW.
X'84'	When Unit Exception is indicated on a Read response to ENQ or Write ENQ CCW.	<p>If the operation is Write Inquiry (X'16'):</p> <p>If line is nonswitched, restart the channel program at the failing command.</p> <p>If line is switched:</p> <p>If ENQ was received, turn on the contention bit (bit 3) in DECFLAGS and clear the return indicators.</p> <p>If DLE EOT was received, proceed with error posting.</p>

Figure 85 (Part 2 of 5). Special Return Codes -- BSC

Code	Set:	Action
X'84' Cont'd		<p>If the operation is not Write Inquiry:</p> <p> If ENQ was received:</p> <p> If the operation is not Write Initial (X'02'), set "should-not-occur" bit (bit 1) in DECERRST, clear the special return indicators, and proceed with error posting.</p> <p> If the operation is a Write Initial:</p> <p> If the line is switched or multipoint, set "should-not-occur" bit (bit 1) in DECERRST, clear the special return indicators, and proceed with error posting.</p> <p> If the line is nonswitched:</p> <p> If this is not the primary station, turn on the contention bit (bit 3) in DECFLAGS, restore the CSW information in the CSW, clear the special return indicators, indicate no more retries are to be made, and return control to the supervisor.</p> <p> If this is the primary station and the retry limit has not been reached, restart the user channel program from the beginning.</p> <p> If this is the primary station and the retry limit has been reached, clear the special return indicators and proceed with error posting.</p> <p> If ENQ was not received:</p> <p> If the operation is Write Connect (X'1C'), execute a Read Skip command.</p> <p> If the operation is not Write Connect (X'1C'), restart the channel program at the failing command.</p>
X'85'	(not used)	
X'86'	1. When Unit Exception has been indicated on a Read Response to Text CCW.	<p>If a NAK is the only character received and dynamic buffering is being used:</p> <p> Clear the special return indicator and proceed with error posting.</p> <p>If a NAK is the only character received and dynamic buffering is not being used:</p> <p> Clear the special return indicator and restart the channel program at the Write Text CCW.</p>

Figure 85 (Part 3 of 5). Special Return Codes -- BSC

Code	Set :	Action
X'86' Cont'd	<p>2. When timeout has occurred on a Read Response to Text.</p>	<p>If a NAK preceded by other characters is received:</p> <p>Clear the special return indicator and proceed with error posting.</p> <p>If the proper ACK (ACK-0 or ACK-1) is received:</p> <p>Clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor.</p> <p>If something other than NAK, ACK-0, or ACK-1 is received:</p> <p>(If the retry limit (7) has been reached) Clear the special return indicators, then proceed with error posting.</p> <p>(If the retry limit has not been reached) Restart the ERP channel program.</p> <p>If the wrong ACK is received:</p> <p>(If dynamic buffering is being used) Clear the special return indicators and proceed with error posting.</p> <p>(If dynamic buffering is not being used) Clear the special return indicators and restart the channel program at the Write Text CCW.</p> <p>If the failing CCW is a Write ENQ:</p> <p>If the maximum retry count (7) has not been reached, restart the channel program at the Write ENQ CCW. If the maximum count has been reached, proceed with error posting.</p> <p>If the correct alternating acknowledgment was received: post normal completion.</p> <p>If the wrong alternating acknowledgment was received: If dynamic buffering was used, proceed with error posting; if not used, and the failure occurred during a timeout situation, restart the channel program at the Write Text CCW; if not used and the failure did not occur during a timeout situation, resend the ENQ character.</p> <p>If NAK was received, restart the channel program at the Write Text CCW.</p> <p>If EOT or RVI was received, clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor.</p> <p>If some character other than one of the foregoing was received, restart the channel program at the Write ENQ CCW.</p>

Figure 85 (Part 4 of 5). Special Return Codes -- BSC

Code	Set:	Action
X'87'	When Unit Exception is indicated on a Reset operation (TP-Op code 21)	<p>If ENQ was received:</p> <p>Clear the special return indicators and restart the user channel program at the failing CCW.</p> <p>If ENQ was not received.</p> <p>(Failing CCW is the last CCW in user channel program) Restart channel program at failing CCW.</p> <p>(Failing CCW is not last CCW in user channel program) Restart channel program at the following CCW (Disable).</p>
X'88'	When Unit Exception is indicated on a Poll or Write Text CCW.	Clear the special return indicators and restart the user channel program at the failing CCW.
X'89'	When Unit Exception is indicated when STX ENQ is received.	<p>If EOT was received, set the retry count to 7, clear the special return indicators, and proceed with error posting.</p> <p>If EOT was not received and the first character is not STX, set "should-not-occur" bit in DECERRST.</p> <p>If the first two characters are STX ENQ, and the maximum retry count has not been reached, restart the channel program at the failing CCW; if the retry count has been reached, proceed with error posting.</p> <p>If the first two characters are not STX ENQ, clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor.</p>
X'8A'	When Unit Exception is indicated on a first ERP Write CCW.	Execute a Read Skip CCW and set special return code X'8A' in IOBWORK+1 after the original return code is saved. After executing the Read Skip, restore the original command and return codes.
X'8B'	When Data Check is indicated on a Read Text CCW (TP-Op code 11)	<p>If ENQ was received, and the maximum retry count has not been reached, restart the ERP channel program (Write NAK and TIC to the Read CCW).</p> <p>If ENQ was not received, clear the special return codes and proceed with error posting.</p>
X'8C'	When timeout is indicated on a Read ID Response CCW (TP-Op code 07)	Clear the special return indicators and proceed with error posting.

Figure 85 (Part 5 of 5). Special Return Codes -- BSC

Condition	Action
Retry count is at limit (7)	Set up new polling or addressing characters, if applicable, and restore the original status and sense information to the IOB (this indicates the nature of the original error that occurred during the user channel program and that caused ERP to be initiated). If a Timeout error has occurred but the message is to be suppressed, pass control to the BTAM channel end appendage; if not, pass control to the operating system message writer.
Retry count is not at limit and	
1. Failing CCW is not an ERP CCW.	Same as for action when retry count is at the limit.
2. A special return code is present and the failing CCW is a Write CCW.	Set special return code X'1A' in IOBERRCT+1 to force control to be returned to the Error Post routine when the next interrupt occurs, indicate that ERP is in control (X'24' in IOBFLAG1), and restart the channel program at the CCW following the failing CCW. Indicate that ERP is in control (X'24' in IOBFLAG1), and restart the ERP channel program at the beginning.
3. Lost Data is indicated in the sense byte following execution of an ERP channel program for a Write Connect operation.	Indicate that ERP is in control (X'24' in IOBFLAG1) and restart the channel program at the CCW that ended with the error that caused ERP to be initiated.
4. An error other than Lost Data has occurred following execution of an ERP channel program for a Write Connect operation.	Same as action for condition 2, above.
5. An ERP Write CCW has ended with an error.	Return control to the ERP Control routine.
6. The interruption occurred on an ERP CCW other than Write, and no sense errors were indicated in the sense byte (ignoring Lost Data).	Same as for action when the retry count is at the limit.
7. The interruption occurred on an ERP CCW other than Write, and the sense byte indicates that an error other than Timeout or Lost Data occurred.	Same action as for condition 4, above.
8. The interruption occurred on an ERP CCW other than Write, the sense byte indicates Timeout and the Skip bit of the failing CCW is on.	Restart the failing ERP channel program at the beginning.
9. The interruption occurred on an ERP CCW other than Write, the sense byte indicates Timeout, and the Skip bit of the failing CCW is <u>not</u> on.	Set up the fields used in printing operator-awareness messages generated by terminals for SYS1.LOGREC. The control is passed to the Teleprocessing Recorder.
10. SOH % E or SOH % C message was received.	

Figure 86. Error Post Actions -- BSC

CSW Bits								Sense Bits								Error Condition
A	CUE	B	CE	DE	UC	UE	CDC	CR	IR	BOC	EC	DC	US	CC	OC	
					X					X						1
					X				X							2
					X			X								3
	(X)		X	X	X					X						4
	(X)		X	X	X										X	5
	(X)		X	X	X							X				6
	(X)		X	X	X							X	X			7
	(X)		X	X	X									X		8
	(X)		X	X		X										9
	(X)		(X)	X		X										10
	(X)	(X)	(X)	X	X				X		X		X			11
	(X)	(X)	(X)	X	X				X							12
	(X)	(X)	(X)	X	X						X		X			13
	(X)	(X)	(X)	X	X							X				14
	(X)	(X)	(X)	X	X							X	X			15
	(X)	(X)	(X)	X	X									X		16
	(X)	(X)	(X)	X	X										X	17
X		(X)			X						X					18
X		(X)			X							X	X			19
							X									20

X indicates that the bit is on.
(X) indicates that the bit may be on if stacking is done by the channel.

Figure 87. Local 3270 Error Conditions (According to CSW and Sense Byte)

- | | |
|--------------------------|----------------------------|
| STATUS | SENSE |
| A = Attention | CR = Command Reject |
| CUE = Control Unit End | IR = Intervention Required |
| B = Busy | BOC = Bus Out Check |
| CE = Channel End | EC = Equipment Check |
| DE = Device End | DC = Data Check |
| UC = Unit Check | US = Unit Specify |
| UE = Unit Exception | CC = Control Check |
| CDC = Channel Data Check | OC = Operation Check |

Failing Command	Command Code																				
		1	2	3	4 ¹	5 ¹	6 ¹	7 ¹	8 ¹	9	10	11	12	13	14	15	16	17	18	19	20
Write	01	6	1	2	3	2	3	3	3	7	2	1	4	1	3	3	3	3	3	3	3
Erase write	05	6	1	2	6	2	6	6	6	7	2	1	4	1	6	6	6	3	3	3	6
Erase Write Alternate ²	0D	6	1	2	6	2	6	6	6	7	2	1	4	1	6	6	6	3	3	3	6
Read buffer	02	6	1	2	3	3	6	3	6	7	2	3	3	3	3	3	3	3	3	3	6
Read modified	06	6	1	2	3	3	6	3	6	7	2	3	3	3	3	3	3	3	3	3	6
Write Structured Field ³	11	6	1	2	6	2	3	3	5	7	2	3	3	3	3	3	5	3	3	3	5
Select ⁴	0B	6	1	2	3	3	6	6	6	7	7	6	3	6	6	6	6	3	3	3	6
Erase all unprotected	0F	6	1	2	3	3	6	6	6	7	7	3	3	3	6	3	6	3	3	3	6

¹This error condition may be the result of stocking in the channel.

²Does not apply to 3272 devices.

³Does not apply to the 3272 or 3274 Model 1B devices.

⁴The error condition and action also includes the Prepare to Read select commands (command codes X'1B', X'2B', X'3B', X'4B') when directed to a 3274 Model 1D Control Unit. If Prepare to Read select commands are directed to 3272 or 3274 Model 1B devices, a Command Reject error occurs and no recovery is attempted.

Figure 88. Local 3270 Recovery Actions (By Error Condition and Failing Command)

```

xx.. .... Termination code
00.. .... Code 0
01.. .... Code 1
10.. .... Code 2
11.. .... Code 3
..xx x... reserved
.... .xxx Sequence code
.... .000 Code 0
.... .001 Code 1
.... .010 Code 2
.... .011 Code 3
.... .100 Code 4
.... .101 Code 5
.... .110 Code 6
.... .111 Code 7

```

Note: For a definition of these codes, refer to OS/VS1 Recovery Management Support Logic, SY27-7239.

Figure 89. Local 3270 Sequence and Termination Codes in Byte 7 of ERPIB

Sequence Code Termination Code	0			1			2			3			4			5			6			7			
	Failing Command	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Write	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9
Erase/Write	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9
Erase/Write Alternate ¹	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9
Read Buffer	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9
Read Modified	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9
Select ²	9	9	9	11	9	12	11	9	11	9	9	9	11	11	12	13	13	13	13	9	9	9	9	9	9
Erase All Unprotected	9	9	9	11	9	12	11	9	11	9	9	9	11	11	12	13	13	13	13	9	9	9	9	9	9
Write Structured Field ³	9	9	9	11	9	12	11	9	11	11	11	11	11	11	11	12	13	13	13	9	9	9	9	9	9

¹Does not apply to the 3272 Control Unit.

²The error condition and action also includes the Prepare to Read select commands (X'1B', X'2B', X'3B', X'4B') when directed to 3274 Model 1D devices.

³Does not apply to 3272 or 3274 Model 1B devices.

Note: If data chaining is in effect on the failing CCW, the failing command is the first CCW with data chaining in the CCW string.

Figure 90. Local 3270 Recovery Actions (By Error Condition and Failing Command)

This Page Intentionally Left Blank

SUGGESTED RETRY OPTIONS FOR READ AND WRITE OPERATIONS

After a user program Read or Write operation has completed, the program must decide what the next operation should be. This depends largely on the result of the preceding operation -- whether it was completed normally, with or without some exceptional condition, or abnormally; and if the latter, what kind of error caused the abnormal completion. The tables in this chapter suggest, for various completion codes and ending conditions, the next READ or WRITE macro it might be appropriate for the user program to issue.

Retry options applying only to the local or switched 3275 (BSC2) are grouped in a set of tables separate from the tables covering other devices used on BSC2 lines.

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	41	DECFLAGS: 40	NAK received in response to ENQ	1
	7F	DECFLAGS: C0	WACK received in response to ENQ	1,3, or 6
	41	ENQ in DECREASESPN	ENQ received in response to ENQ (MODE=CNTRI in DCB macro) retried seven times	1
	41	DECFLAGS: 10	ENQ received in response to ENQ (MODE≠CNTRL in DCB macro)	5
11	41	Data check	(for TIV, TIVX, TTV, TTVX only) Text was received with error	2
25	41	DECFLAGS: 20	Wrong acknowledgment received in response to text	3
	7F	DECFLAGS: 40	EOT received in response to text	1
	7F	DECFLAGS: 42	RVI received in response to text	7
	7F	DECFLAGS: C0	WACK received in response to text	3 or 6
	41	DECFLAGS: 40	NAK received in response to text	4
Retry Options:				
<ol style="list-style-type: none"> 1. Issue a WRITE Initial (TI) macro. 2. Issue a READ Repeat (TP) macro. 3. Issue a WRITE Inquiry (TQ) macro. 4. Issue a WRITE Continue (TT) macro. 5. Issue a READ Initial (TI) or READ Initial Inquiry (TIQ) macro. 6. Issue a WRITE Reset (TR) macro. 7. Continue normally. 				

Figure 91. Retry Options for Write Operations (Nonswitched Point-to-Point Line [BSC1])

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0B	41	Timeout	ENQ not received	1
	7F	DECFLAGS: 10	Received [ID] ENQ did not match expected [ID] ENQ	6
	41	Timeout	[ID] ENQ not received	5
11	41	Lost data, data check, or overrun	Text was received with error	2 or 4
	41	Timeout	No text received	1,3 or 4
	41	DECFLAGS: 01	STX ENQ received in lieu of text	4
	7F	----	DLE EOT from switched line	4

Retry Options:

1. Issue a WRITE Break (TB) macro to disconnect the line.
2. Issue a READ Repeat (TP) macro.
3. Issue a READ Inquiry (TQ) macro.
4. Issue a WRITE Disconnect (TD) macro to disconnect the line.
5. If expanded ID verification is in use, BTAM automatically disconnects the line and reissues the READ Connect macro. If expanded ID verification is not in use, issue a WRITE Break (TB) macro.
6. If expanded ID verification is in use, BTAM automatically disconnects the line. You may therefore reissue the READ Connect macro. If expanded ID verification is not in use, issue a WRITE Break (TB) macro.

Figure 92. Retry Options for Read Operations -- Answering (Switched Point-to-Point [BSC2] except 3275)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	41	Timeout	No response received to ENQ	1
0B	41	Timeout	ENQ not received	1
07	7F	DECFLAGS: 10	Invalid ID received in response to ID ENQ	2
	41	Timeout	No response received to [ID] ENQ	1
	41	Lost data, data check, or overrun	Text was received with error	1 or 3
11	41	Timeout	No text received	1,2, or 4

Retry Options:

1. Issue a WRITE Disconnect (TD) macro to disconnect the line.
2. Issue a WRITE Break (TB) macro to disconnect the line.
3. Issue a READ Repeat (TP) macro.
4. Issue a READ Inquiry (TQ) macro.

Figure 93. Retry Options for Read Operations -- Calling (Switched Point-to-Point [BSC2] except 3275)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	41	DECFLAGS: 20	Wrong acknowledgment received in response to ENQ	1 or 4
	41	DECFLAGS: 40	NAK received in response to ENQ	1 or 4
07	7F	DECFLAGS: 10	Invalid ID received in response to ID ENQ	If original operation is Write TI, option 4. If original operation is Write TC, option 5
	7F	DECFLAGS: 40	NAK or ID NAK received in response to ID ENQ	4 or 5
	7F	DECFLAGS: C0	WACK received in response to ID ENQ	4 or 5
	41	Timeout	No response received to [ID] ENQ	If original operation is Write TI, option 4 If original operation is Write TC, option 5
11	41	Lost data, data check, or overrun	(for TIV, TIVX, TTV, TTVX only) Text was received with error	3 or 4
25	41	DECFLAGS: 20	Wrong acknowledgment received in response to text	1 or 4
	41	NAK in DECRESPI	NAK received in response to text	2 or 4
	41	Timeout	No response received to text	1 or 4
Retry Options:				
<ol style="list-style-type: none"> 1. Issue a WRITE Inquiry (TQ) macro. 2. Issue a WRITE Continue macro to retransmit the text. 3. Issue a READ Repeat (TP) macro. 4. Issue a WRITE Disconnect (TD) macro to disconnect the line. 5. Reissue the WRITE Connect (TC) macro. 				

Figure 94. Retry Options for Write Operations (Switched Point-to-Point Line [BSC2] except 3275)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0B	41	Timeout	ENQ not received - the 3275 Dial has no data to send	1,2, or 3
07	7F	DECFLAGS: 10	Received ID ENQ did not match expected ID ENQ	4
	41	Timeout	ID ENQ not received	5
11	7F		EOT received after ETB in lieu of text	6
	7F	DECFLAGS: 02	Sense/status message received	7
	41	Lost data, data check, or overrun	Text was received with error	1 or 8
	41	Timeout	No text received	1 or 9
	41	DECFLAGS: 01	STX ENQ received in lieu of text	1

Retry Options:

1. Issue a WRITE Disconnect (TD) macro to disconnect the line.
2. Issue a WRITE Inquiry (TQ) macro.
3. Issue a WRITE Reset Monitor (TRM). Follow by issuing RESETPL if no response is received in a reasonable time.
4. If expanded ID verification is in use, BTAM automatically disconnects the line. Reissue the READ Connect (TC) macro. If expanded ID verification is not in use, issue a WRITE Disconnect (TD) macro.
5. If expanded ID verification is in use, BTAM automatically disconnects the line and reissues the READ Connect macro. If expanded ID verification is not in use, issue a WRITE Disconnect (TD) macro.
6. Issue WRITE Reset Monitor (TRM) followed by READ Continue (TT) to read the sense/status message.
7. Issue READ Continue (TT) to acknowledge receipt of the sense/status message and receive EOT.
8. Issue a READ Repeat (TP) macro.
9. Issue a READ Inquiry (TQ) or READ Inquiry Monitor (TQM) macro.

Figure 95. Retry Options for Read Operations -- Answering (Switched Point-to-Point [BSC] 3275 only)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
OB	41	Timeout	ENQ not received - the 3275 Dial has no data to send	1,2, or 3
07	7F	DECFLAGS: 40	ID NAK received	4
	7F	DECFLAGS: 10	Invalid ID received in response to ID ENQ	1
	41	Timeout	No response received to ID ENQ	1
11	7F	area+length -residual count: EOT	EOT received after ETB in lieu of text	4
	7F	DECFLAGS: 02	Sense/status message received	5
	41	Lost data, data check, or overrun	Text was received with error	1 or 6
	41	Timeout	No text received	1 or 7

Retry Options:

1. Issue a WRITE Disconnect (TD) macro to disconnect the line.
2. Issue a WRITE Inquiry (TQ) macro.
3. Issue a WRITE Reset Monitor (TRM). Follow by issuing RESETPL if no response is received in a reasonable time.
4. Issue WRITE Reset Monitor (TRM) followed by READ Continue (TT) to read the sense/status message.
5. Issue READ Continue (TT) to acknowledge receipt of the sense/status message and receive EOT.
6. Issue a READ Repeat (TP) macro.
7. Issue a READ Inquiry (TQ) or READ Inquiry Monitor (TQM) macro.

Figure 96. Retry Options for Read Operations -- Calling (Switched Point-to-Point [BSC2] 3275 only)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
07	7F	DECFLAGS: 10 Write TI	Invalid ID received in response to ID ENQ	1
	7F	DECFLAGS: 10 Write TC	Invalid ID received in response to ID ENQ	2
	7F	DECFLAGS: 40	ID NAK received in response to ID ENQ	3
	7F	DECFLAGS: C0	WACK received in response to ID ENQ	1
	41	Timeout Write TI	No response received to ID ENQ	1
	41	Timeout Write TC	No response received to ID ENQ	2
11	41	Lost data, data check, or overrun	(for TIV, TTV only) Text was received with error	1 or 4
	7F		For TIV, EOT was received in lieu of text. For TTV, EOT was received after ETB in lieu of text	3
25	7F	DECFLAGS: 20	Wrong acknowledgments received in response to text	1 or 5
	7F	DECFLAGS: C0 DECRESPI: WACK	WACK response to ENQ received	6
	7F	DECRESPI: EOT DECFLAGS: 40	EOT response to write text	3
	41	DECRESPI: NAK	NAK received in response to text	1 or 7
	41	DECRESPI: NAK Write TQ	NAK response to ENQ received	3
	41	Timeout	No response received to text	1 or 5

Retry Options:

1. Issue a WRITE Disconnect (TD) macro to disconnect the line.
2. Reissue the WRITE Connect (TC) macro.
3. Issue WRITE Reset Monitor (TRM) followed by READ Continue (TT) to read the sense/status message.
4. Issue a READ Repeat (TP) macro.
5. Issue a WRITE Inquiry (TQ) macro.
6. Reissue READ Inquiry or issue READ Inquiry Monitor (TQM). Note: Timeout is the usual condition for the switched 3275. Use READ TQM if you do not wish to time out. RESETPL can be issued to terminate READ TQM if no response is received in a reasonable time.
7. Issue a WRITE Continue (TT) macro to retransmit the text.

Figure 97. Retry Options for Write Operations (Switched Point-to-Point [BSC2] 3275 only)

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0A	41	Timeout	No index byte was received	1,2, or 4
09	7F	DECFLAGS: 04	Negative response to polling	1,2, or 4
03	41	Timeout	No terminal responded to polling	1,2, or 4
	48		Initial read terminated by RESETPL macro	1,2, or 4
11	41	DECFLAGS: 01	STX ENQ sent in lieu of text	2
	41	Lost data, data check, or overrun	Text was received in error	2 or 3
	41		ENQ response to Read Continue	7
	41	DECFLAGS: 40	Text was received ending with an ENQ	5
	7F	DECFLAGS: 02	Sense/status message was received	6

Retry Options:

1. Issue a READ Initial (TI) macro to poll the same or a different station.
2. Issue a WRITE Reset (TR) macro.
3. Issue a READ Repeat (TP) macro.
4. Issue a WRITE Initial (TI) macro.
5. Issue a READ Initial (TI) macro (using the polling entry of the remote 3270 device for which completion was posted) to receive the error status message.
6. Issue a READ Continue (TT) macro, and examine the sense/status bytes to determine what action to take.
7. Probably a device problem. Retry operation from READ Initial (TI). If problem recurs, issue WRITE Reset (TR) and continue working with other devices.

Figure 98. Retry Options for Read Operations (Nonswitched Multipoint Lines [BSC3])

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
06	7F	DECFLAGS: 04	NAK received in response to addressing	1,2, or 7
	7F	DECFLAGS: 42	RVI received in response to addressing	1,2, or 7
	7F	DECFLAGS: 42	RVI received (remote 3270)	2 or 8
	7F	DECFLAGS: C0	WACK received in response to addressing	1 or 2
	41	Timeout	No response received to addressing	1 or 2
11	41	Data check	(TIV, TIVX, TTV, TTVX only) Text was received with error	2 or 5
	41	DECFLAGS: 40	(TIV only) Text was received ending with an ENQ	8
25	7F	DECFLAGS: 20	Wrong acknowledgment received in response to text	2 or 4
	7F	DECFLAGS: 42	RVI received in response to text	2 or 6
	7F	DECFLAGS: C0	WACK received in response to text	2 or 4
	7F	DECFLAGS: C0	WACK received (remote 3270)	9
	41	DECFLAGS: 40	NAK received in response to text	2 or 3
	7F	DECFLAGS: 40 DECRESPTN: EOT	EOT received in response to text	8
	41	Timeout	No response received to text	2 or 4
Retry Options:				
<ol style="list-style-type: none"> 1. Issue a WRITE Initial (TI) macro to address the same or a different station. 2. Issue a WRITE Reset (TR) macro to terminate selection. 3. Issue a WRITE Continue (TC) macro. 4. Issue a WRITE Inquiry (TI) macro. 5. Issue a READ Repeat (TP) macro. 6. Continue normally. 7. Issue a READ Initial (TI) macro to poll another station. 8. Issue a READ Initial (TI) macro (using the polling entry of the remote 3270 device for which completion was posted) to receive the error status message. 9. If the write operation started a printer, issue a WRITE Reset (TR) macro to reset the line, and continue normally. 				

Figure 99. Retry Options for Write Operations (Nonswitched Multipoint Line [BSC3])

Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
7F	DECFLAGS: 01	OLTEP received control of the device following normal completion of the I/O operation	1, 2, 3, or 4
41	DECFLAGS: 01 (and other error flags)	OLTEP received control of the device following a permanent I/O error	2, 3, or 5
41	DECERRST: 80	Control unit not operational	1, 2, or 3
41	DECERRST: 00 DECSENSO: 01	Incorrect data stream	6
41	DECERRST: 10	Integrity of the device regeneration buffer is questionable	4
41	Other than above	I/O error	5
44		I/O request intercepted	7
48		Read TI canceled	1

Retry Options

1. Continue normally.
2. Inform the system operator, and request additional information.
3. Wait for some interval of time before trying the next I/O operation.
4. Issue a WRITE TS macro instruction to reconstruct the buffer contents.
5. Further use of the device is questionable, although prohibited. The problem program should consider the device unavailable and should consider requesting that diagnostics be run on the device.
6. Check that the data stream is correct (that is, buffer addresses are correct, order sequences are complete, and orders do not cause overrun).
7. The contents of the device buffer are doubtful, because (1) an error occurred following the completion of the previous I/O operation or (2) a request-for-test message was received from the device requesting that a test message be sent to another device. If the current operation is a write erase, it should be done. Otherwise, a WRITE TS macro instruction should be issued to reconstruct the buffer contents before doing the current operation.

Figure 100. Retry Options for Local 3270 Read and Write Operations

ONLINE TESTING

Online testing is an optional BTAM facility that permits the user to verify proper operation of terminals and of the communication lines that link them to the computer and to aid in diagnosing line or terminal troubles. Online testing centers around transmission of predefined standard test messages, the formats of which depend on the purpose of the test.

Online testing is performed during normal BTAM operation. Only the communication lines and terminals specified are involved; data transmission proceeds as usual on other lines. Operation of the program is affected only to the extent of the line time required for test transmissions and of the CPU time required to process requests for tests.

You may wish to perform certain kinds of online testing as a routine procedure, for example, to test line or terminal functioning at the beginning of each day, or at intervals during the day. Other kinds of tests are appropriate as diagnostic aids, and are normally performed as needed by the computer or terminal operator or IBM customer engineer.

In order to have the online testing facility available, you must code T among the EROPT operands of the DCB macro instruction for the line group.

Online testing is implemented somewhat differently for start-stop lines and for binary synchronous lines.

ONLINE TESTING FOR START-STOP COMMUNICATIONS LINES

For start-stop communication lines, test requests may be initiated only at remote terminals. The tests requested may involve message switching, comparing the contents of a test message to a predefined character sequence in main storage, sending a string of characters to a specified terminal, or checking the IBM SELECTRIC typing element mechanism of a terminal printer.

Start-stop online tests are initiated by transmission of test request messages, the format of which is:

99999	xx	type	ADDR	SELECT	text	END CHAR
-------	----	------	------	--------	------	-------------

5 2 1 1or2 1or2 Variable 1
Field length (bytes)

99999

identifies this message as a test request.

xx

defines the type of test to be executed (see "Test Type Codes," below).

type

specifies the type of terminal from which the test is being requested. Applicable type codes are:

Code	Device
0	2741
1	1030 card reader
2	1050
3	1060
4	2740 (with or without 2760 attached)
5	1030 badge reader or manual entry unit
6	2260 (Remote) and 2265 (Remote)

TO ADDR

specifies the address of the terminal to which the message is to be sent (for 2760 tests, specifies function to be performed).

TO ADDR is a one-byte field for the IBM 1030 Card Reader and 1050, 2740, and 2741 terminals; it contains the addressing character for the selected terminal. For those 2740 and 2741 terminals not using addressing characters (that is, all terminals not equipped with station control), this field should contain a space character (in the hexadecimal representation of the transmission code pattern for space) except when a 2760 frame change or scan point test is performed.

TO ADDR is a two-byte field for the 1030 badge readers and manual entry units, 1060 terminals, 2260 and 2265 terminals; it contains a two-byte code indicating which addressing characters BTAM is to send on the line.

For 1030

Code	Addressing Character
02	B
03	C
04	D
.	.
.	.
.	.
26	Z

Note: Codes of 01 and 10, representing A and J, may not be used, as A and J are invalid 1030 addresses.

For 1060

<u>Code</u>	<u>Addressing Character</u>
01	A
02	B
03	C
.	.
.	.
.	.
26	Z

For 2760

TO ADDR contains the F-character that specifies the 2760 function to be performed. Figure 22 lists the F-characters and their meanings.

For 2848 (2260) and 2845 (2265)

TO ADDR is used to select the 2848 or 2845 display control unit. The address of a display control unit can be any USASCII noncontrol character (that is, any character in columns 3-7 in the USASCII code chart), therefore allowing 96 possible display control addresses.

<u>Actual Unit Address</u>	<u>Code</u>
b ...b ₁	
0100000	01
0100001	02
...	...
1111111	96

Note: The TO ADDR code applicable to a particular display control unit can be determined from one of its attached display stations by specifying the Request Address test (test type 09) in the test message.

UNIT SELECT

Note: Unit select is not applicable to 1030, 2740, or 2741 tests; therefore, text can start in this position.

For 1050 and 1060 (1 Character)

UNIT SELECT specifies the particular component of the selected terminal that is to receive the message, that is, 1052, 1053, 1055, 1062 Printer 1 or 2, etc. The appropriate unit select code can be determined from the publication pertaining to the terminal. For 2760 tests, this field contains the A₁, A₂ characters that specify amount of filmstrip movement. See Figure 24.

For 2260 or 2265 (or 1053 Attached to the 2848 or 2845) (2 characters)

2260 and 2265 Display Stations and 1053 Printers are selected by transmitting a predefined code in these character positions. The device selection code can be one of 25 USASCII noncontrol characters.

<u>Actual Unit Address</u>		<u>Code</u>
b ...b ₁		
1000000		01
1000001	2260 and 1053 attached to	02
...	2848	...
1011000		25
1011001	2265 attached to 2845	26
1011001	1053 attached to 2845	27

Note: The UNIT SELECT code applicable to a particular 2260 display station can be determined from that display station itself by utilizing the Request Address test (test type 09).

END CHARACTER

1030 =	EOB
1050 =	EOT
1060 =	EOB
2740 =	EOT
2741 =	EOT
2760 =	EOT
2848 =	ETX

Note: The test message is transmitted from a 1060 terminal by utilizing the data and transaction keys. The EOB character is entered by depressing the teller A or B key.

TEST TYPE CODES

01 Message Switching

This test receives a message from the requesting terminal and transmits it to the terminal (on the same line) specified in the test message. Note: The length of the message to be switched cannot exceed the length of the data area specified in the READ macro for the line over which the test is requested.

02 Tilt

This test sends the tilt test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

03 Rotate

This test sends the rotate test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

message is sent to the terminal specified in the TO ADDR field:

CMP VLD-*

04 Twist

This test sends the twist test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

The character printed in the position of the asterisk will be the last character against which a comparison could be made. Exception: The message sent to a 1060 after a valid comparison is:

05 Stored Compare

This test provides a means to compare the received message with a particular character sequence in main storage. The message in main storage is compatible with the transmitting capabilities of the terminals involved.

CMP VLD

If the request was received properly, but an insufficient count was specified in the READ and thus no characters could be compared, a / character is printed in the asterisk position.

The test message to be compared with the character sequence in main storage is transmitted from the terminal and consists of the numbers 0 through 9 followed by the alphabet (A through Z). The incoming test message must specify the comparison characters in the same order as they appear in the sequence in main storage although not all of them need be specified.

- 2. If the comparison to the stored message is invalid, the data received is message-switched to the terminal specified in the TO ADDR field.

The length of the test message cannot exceed the length of the data area specified in the READ macro that will receive the message. The data area must be long enough to contain the header information (99999, etc.), the characters to be compared, and the end character.

Note: The Stored Compare test is not applicable for the 1030 manual entry unit or badge reader.

06 All Characters Test

Exceptions:

- 1. When transmitting from any 2740 terminal, a space character must precede the comparison data. This space character is in addition to the space character in the TO ADDR field.
- 2. The stored compare test for a 1060 is requested by entering the following message:

This test provides the standard All Characters test for IBM Customer Engineer terminal checkout and serves as a start-up message. Special characters are not used in the terminal test. Characters received at the terminal are:

For 1030, 1060, 2848 (2260 and 1053):

Numbers: 0-9, and alphabet: A-Z.

For 1050, 2740, 2741:

Numbers: 0-9, alphabet a-z (lower case), and alphabet A-Z (upper case).

9 9 9 9 9 0 5 3 4 2 1 0 EOB

07 SELECTRIC Analyzer Test

Comparison is then made to this message. Responses to this request are printed only at the requesting terminal.

This test provides an exercise to analyze the capability of the SELECTRIC typewriter carrier mechanism to perform within specifications. When this test is requested, BTAM sends to the terminal a predefined message that exercises the carrier mechanism. This test is not applicable to a 1053 Printer attached to a 2848 or 2845 Display Control.

Messages received at the terminal are:

- 1 If the comparison to the stored message is valid, the following

08 Write at Line Address Test (2260 and 2265)

This test provides line selectivity checkout by using the first two characters after the UNIT SELECT field as a new display line code. This can be followed by data which is to be switched to the terminal and displayed on the display station screen at the selected line. The codes and associated display lines are:

<u>Code</u>	<u>Display Line</u>
01	1
02	2
03	3
.	.
.	.
12	12

09 Request Address Test (2260 and 2265)

This test allows the operator at a display station to determine the display control and display station address applicable to that station.

The TO ADDR and UNIT SELECT fields are not utilized in this test message since the test itself provides these fields to the requesting terminal. ETX can be sent immediately after the TYPE field.

BTAM returns to the requesting display station a 9 character message giving the addressing information for that station. The format is:

DC+DVxxyy

DC+DV indicates that the message contains the requested addressing information; xx and yy are the display control and device (that is, display station) addresses.

Note: This test provides only the TO ADDR and UNIT SELECT codes of the requesting display station. It is not a means of getting these codes for some other display station.

10 Frame Change Test (2760)

This test enables an IBM Customer Engineer to request that a filmstrip be moved to a new frame. The request for a frame change test is entered on the 2740 keyboard. BTAM uses the data in this message to generate the appropriate frame change message and sends it to the 2760. The Customer Engineer visually verifies the correctness of the film movement.

11 Scan Point Test (2760)

This test performs a filmstrip movement and then allows the Customer Engineer to probe the screen and have the horizontal and vertical coordinates of the probed response points printed on the 2740 printer. The request for a scan point test is entered at the 2740 keyboard. BTAM generates a frame change message and sends it to the 2760. The Customer Engineer then probes one or more response points, depending on the mode specified in the test request message. BTAM sends to the 2740 a message containing the coordinates of the response points probed.

See "Online Testing" under "IBM 2760 Optical Image Unit -- General Information," for further information on 2760 online tests.

TERMINAL TEST RESTRICTIONS

1. A remote terminal may send a test request message only when the operation in effect for the line is a Read Initial or Read Conversational operation.
2. No READ macro can include the Reset option. For example, a READ TI or TV can be issued, but not a READ TIR or TVR, for a line over which test requests may be received. The line connection must be maintained during the terminal test (the Reset option causes BTAM to break the connection).
3. The user program input area must be long enough to accommodate the entire test message. The response to polling must be read into the first byte of this area. If dynamic buffering is used there is an additional restriction: the data area of the first buffer in the chain must contain all of the characters in the test request.
4. To request a test from a 1030 badge reader, the badge reader must be wired to read out the entire 10 columns of the badge (refer to publications about the IBM 1030).
5. The transaction code received from a 1030 is not included as part of the test request.
6. All 1030 tests require a 1033 Printer on the same line as the requesting terminal. The printer address must be specified in the TO ADDR field.

7. The terminal tests will not test 1035 Badge Readers or 1030 Badge Readers in a 1035 environment.
8. If insufficient storage is available for the test pattern, the request will be switched to the terminal specified by the TO ADDR field.

address, selection address) is entered by means of the keyboard, and (3) the TEST REQUEST key is pressed to frame the text with control characters (SOH % / STX and ETX). The selection address is omitted for the switched 3275; see "Formats of RFT, Test, and Console Messages" in this section. For a System/370, it is sent by means of an ONLTST (Online Test) macro instruction coded within the user program, as explained below.

ONLINE TESTING FOR BINARY SYNCHRONOUS COMMUNICATIONS LINES

Online tests for BSC lines may be requested by the central computer, by remote stations, or both, depending upon the type of test and the line and station configuration. There are 34 types of tests, not all of which apply to all configurations.

Online testing is available for all types of remote BSC stations. For System/370-to-System/370 operation, both computers may run under BTAM with the online test facility, or one may run under BTAM and the other under an online diagnostic program. Operation between System/370 and an 1800, 2715, 2770, 2780, 2972, or remote 3270 requires the System/370 to run under BTAM or an online diagnostic program. For System/370 to System/3 or 1130 operation, the System/370 must run under BTAM, and the System/3 or 1130 must run under an online diagnostic program.

In System/370-to-System/370 operation, either computer may initiate online tests. In operations between the central computer and a System/3, 1130, 1800, 2770, or 2792, the central computer cannot initiate the online test except for a test type of 0. In operations between the central computer and a 2715, only the 2715 can initiate an online test. In operations between the central computer and a remote 3270, any remote terminal on the same line can initiate an online test of the remote 3270.

When the central computer initiates the test with a 2780, the 2780 mode switch must be set to either Print or Punch position if the 2780 is on a point-to-point line.

Tests are requested at a remote station by sending to the central computer a message having a special format, called a request-for-test (RFT) message. The method of sending the RFT message differs for the various types of remote station. For a 2780, the RFT message is punched in a card. For an 1130 or System/3, the message is sent by a diagnostic program. For a remote 3270, (1) the cursor is positioned at the top left of an unformatted screen (by pressing the CLEAR key and then the RESET key, for example), (2) the text of the RFT message (test type, number of times, length of

To request a test at the central computer, the programmer codes an ONLTST macro instruction in the program at the point at which the test is to be performed. The ONLTST macro generates the proper RFT message and sends it to the remote computer or terminal specified in the ONLTST macro.

The format of the RFT message is the same whether it is sent by the central computer or is received by the central computer from a remote computer or terminal. The format is shown below under "Formats of RFT, Test, and Console Messages.)

Transmission of an RFT message is followed by one or more transmissions of test messages. The RFT message contains a field called the X field, which contains a code indicating the type of test to be performed. The code, from 00 to 34, governs the sequence of I/O operations comprising the test and determines the content of the test message.

TYPES OF TESTS

Type 00

For this type of test the requesting station sends an RFT message, immediately followed by a test message, or a sequence of test messages, the content of which is user-specified. The test message is sent the number of times specified in the Y field of the RFT message, which may be from one to 99. For example, if you specify a Y value of 5, the requesting station sends the RFT message, followed by five consecutive transmissions of the same test message. The computer or terminal that receives the RFT and test message responds with an acknowledgment after each message.

For this type of test, the requesting station may be the central computer except when the remote station is a 2715 Model 1, or any type of remote station. When the requesting station is a 2770, 2780, or 2972, however, the operation differs somewhat. First, the test message is sent not as a separate message following the RFT message, but as a part of the RFT message itself. Second, the Y field of the RFT message can only be coded as one, since the RFT mes-

sage, including the message text, is sent only once.

Another restriction applies when the station receiving the RFT message is a 2770 or 2780. The job switch (2770) or mode switch (2780) must be set to permit the RFT message to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

The requesting station may not be a remote 3270 display station.

Type 01

For this type of test, the requesting station sends an RFT message that includes user-specified text characters. The station receiving the RFT message acknowledges it, prepares a test message containing the text characters from the RFT message, and sends the test message the number of times specified in the Y-field of the RFT message -- from 1 to 99. The station receiving the test messages (that is, the station that sent the test request) responds with an acknowledgment after each test message.

For this type of test, the requesting station may be the central computer only if the remote station is a System/370. The requesting station may be any type of remote station. If the requesting station is a 2770 or 2780, its job switch (2770 or mode switch (2780)) must be set to permit the test messages returned from the central computer to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

Note that in type 00 tests, the requesting station also sends the test messages, and receives acknowledgments in reply, but in type 01 tests the requesting station receives test messages in reply.

Types 02-34

For these types of tests, the requesting station sends an RFT message. Unlike tests of types 00 and 01, the RFT message neither contains nor is followed by a test message. Instead, the X field of the RFT message indicates to the receiving station which of 33 BTAM-defined standard test messages it is to return to the requesting station. When BTAM receives the RFT message, it examines the X and Y fields, selects the test message designated by X, and sends it Y times. The contents of test messages for each type of test are given below under "Formats of RFT, Test, and Console Messages."

For this type of test, the requesting station may be the central computer only if the remote station is a System/370 using OS/VS BTAM. The requesting station may also be any type of remote station. If the requesting station is a 2770 or 2780, its job switch (2770) or mode switch (2780) must be set to permit the test messages sent from the central computer to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

Note: Set the 2780 'Online Test' switch to the on position. This will suppress the generation of an STX character preceding the RFT message.

BTAM RESPONSES TO REQUEST-FOR-TEST MESSAGES

BTAM recognizes and responds to any RFT messages received from a remote computer or terminal provided that:

1. The online test facility is available (you have coded T among the EROPT options in the DCB macro for the line group).
2. The RFT message was received on a Read Initial (TI) operation. If the device to be tested is part of a remote 3270 display system, the RFT message may have been received on a Read Continue (TT) operation or (for BSC2) on a Read Connect (TC) operation.
3. The length of the input area specified by the READ macro is at least 300 bytes for test types 02-34. If buffering is used, the entire 300-byte area must be contained within one buffer. For test types 02-34, if the area is less than 300 bytes, BTAM returns an EOT instead of a test message. The EOT ends the test before any test messages are sent. For test types 00 and 01, no check is made to determine the length of the input area; instead the length specified in the READ macro is used. Ensure that the area is large enough to accommodate the text data in the RFT message or the test message that follows the RFT message. Otherwise lost data and timeout errors will result.
4. The RFT message was received without error.

The remote computer or terminal may send an RFT message only when the BTAM program has a Read Initial operation pending on the line over which the RFT message will be received, unless the device to be tested is part of a nonswitched remote 3270 display

system, which can send an RFT message on a Read Continue operation. If the remote 3270 is a switched 3275, the RFT message can be received on a Read Initial, Read Continue, or Read Connect operation. When BTAM recognizes the message received by a Read Initial operation as an RFT message, the Read operation is not posted complete as it is for non-RFT messages. Instead, control is given to the online test logic, which examines the RFT message, generates the requested test message in the area specified in the READ macro, and sends the test message to the requesting computer or terminal (or other specified destination, for multipoint lines). If the RFT message specified a type 00 test, only a response is returned to the requesting computer or terminal, as explained previously. Following transmission of the test message the requested number of times, the online test logic sends an EOT character for non-switched lines, or DLE EOT (and disables the line) for switched lines, then restarts the program at the Read Initial operation that received the RFT message.

When an RFT message is received for a nonswitched remote 3270 display station on a Read Continue operation, BTAM gives control to the online test logic, which generates and sends the test message and then posts the Read Continue operation complete and places an EOT in the input area specified in the read operation.

When a RFT message is received for a switched remote 3275 display station on a Read Continue operation, the test message is transmitted the specified number of times and is followed by Write Reset. The response to the Write Reset is tested by BTAM with possible results as follows:

- If the response is ENQ (the switched 3275 has text data to send), BTAM restores the Read Continue operation.
- If the response is DLE EOT (a disconnect signal), BTAM sets up the Read Continue, does not issue it, puts DLE EOT in the user's buffer, and posts the operation complete with a X'7F'.
- If there is no response (timeout -- no more data to send), BTAM retries up to 25 times. If there is still no response, BTAM sets up the Read Continue but does not issue it, puts the sense and CSW status information in DECSNS0 and DECCSWST fields of the DECB (to indicate timeout), turns on the "3275 Dial RFT Error" bit (DECERRST, bit 7), and posts the operation complete with a X'41'.
- If there is an error other than timeout, BTAM retries up to seven times.

If the condition persists, the timeout actions described above are followed.

Notes:

1. Test mode will not be entered until the RFT message is received correctly and positively acknowledged and until the proper positive response (ACK-0) to selection or line bid is received. If a positive response to selection (ACK-0) is not received initially or after seven retries, the test will be terminated.
2. Once test mode has been entered, if one or more WACK responses are received, the transmitting station will respond to each WACK with an ENQ, until the regular positive response is received. The number of WACKs that will be accepted is 25; if more than this number are received consecutively, the online test is terminated.
3. When a test message is requested for a remote 3284 or 3286 printer, the RFT message should specify that the test message be sent only once. This avoids wasting line time, since the test message appears only once on a remote 3270 printer even though attempts are made to send it more than once when the Y field of the RFT message is greater than one.

BTAM INITIATION OF REQUEST-FOR-TEST MESSAGES

As mentioned previously, you may initiate online tests by coding the ONLTST macro instruction in your program. ONLTST causes the online test logic to prepare an RFT message, send it, send or receive test messages (depending on test type), receive or send appropriate acknowledgments, and accumulate and display on the central computer console the results of the test. The ONLTST macro is described below. Message formats for each type of test are given under "Formats of RFT, Test, and Console messages."

ONLTST (Online Test) Macro Instruction

The ONLTST macro instruction is used to send a request-for-test (RFT) message on a binary synchronous communication line. It provides the information necessary to build the RFT message, generates the linkage to the online test routine, and causes the RFT message to be sent.

The Write operation executed by the ONLTST macro is similar to a Write Initial operation; the ONLTST macro must therefore be used in the same manner. That is, it may appear in your program only where a

Write Initial macro could appear. ONLTST may be issued only when the computer or terminal that is to receive the RFT message is capable of recognizing it as such and acting accordingly. For example, if the computer that is to receive the RFT message is operating under BTAM, ONLTST may be issued only when the corresponding operation at the receiving computer is a Read Initial or Read Connect operation for which the input area length is at least 300 bytes.

Upon completion of an online test on a switched point-to-point line, BTAM breaks the line connection.

After issuing an ONLTST macro, you must issue a WAIT or TWAIT macro (or otherwise test for completion of the online test) before starting any other Read or Write operation for the line.

After execution of the ONLTST macro, control is returned to the next sequential instruction in the user program.

Note: ONLTST cannot be issued to initiate a test between the central computer and an IBM 2715 or a nonswitched or switched remote IBM 3270.

Name	Operation	Operands
[symbol]	ONLTST	DECB=decb address, X=type of test, Y=no. of transmissions, DCB=dcb address, AREA=rft message area [,TEXT=user text area, LENGTH=user text length] [,ENTRY=list address] [,RLN=line number]

DECB specifies the address of the data event control block for the line on which the online test is to be performed.

X specifies the type of test to be performed. Permissible values of X and their meanings are tabulated below, under "Formats of RFT, Test, and Console Messages."

Y specifies the number of times the test message is to be transmitted. Y may be from 1 to 99.

DCB specifies the address of the data control block for the line group.

AREA specifies the address of the area from which the RFT message is to be sent. The online test routine formats the RFT message in this area and also reads into it the responding test messages, for test types 01-19. For type 00, BTAM moves the data comprising the test message into this area. For test type 00 or 01, this area must be large enough to receive the expected test message. For test types 02-22, this area must be at least 300 bytes long. If buffering is used, the entire area must be contained within one buffer.

TEXT specifies the address of the user-defined test message where X (test type) equals 0 or 1. For non-transparent text, you must begin and end the text with the appropriate framing characters (STX and ETX); for transparent text, you supply only DLE STX at the beginning of the text; BTAM provides the DLE ETX at the end of the message. Some amount of text data must be specified when the X operand is 0 or 1. For other values of X, this operand is not required, and is ignored if coded. The contents of this area are not destroyed by ONLTST and may be used for successive tests

LENGTH specifies the number of text characters in the RFT message, where TEXT is specified. This operand must be coded if the TEXT operand is coded.

ENTRY specifies the address of the addressing or ID list (OPENLST, DIALST, or BSClst types). The list must contain only one entry. A calling list, not an answering list, must be specified if the line is switched point-to-point. This operand is not used for online tests on point-to-point lines.

RLN specifies the relative line number of the line within the line group on which the test is to be performed.

- Notes:**
1. No ONLTST macro may be issued for a line until a data event control block has been established for that line by means of a READ or WRITE macro in list or standard format.
 2. In an online test between a System/370 and a 2770, test messages sent to the 2772 control unit cannot exceed a length of 128 bytes, unless the 2772 has the Expanded Buffer feature, in which case the maximum length is 256 bytes.

Return codes: After an ONLTST macro is issued, BTAM sets register 15 to zero if no error was detected. If an abnormal condition is detected, the online test operation is not started, and control is returned to your program at the instruction following the ONLTST macro. A return code in register 15 indicates the error. Bits 0 through 23 are zero; bits 24 through 31 contain one of the following error codes in hexadecimal notation. (Code 0C is issued for the ONLTST macro itself; the other codes result from errors occurring when the online test routine executes a Write operation.)

- 04 Busy. The specified line is busy with a previously requested Read or Write operation.
- 08 Invalid RLN. The relative line number specified in ONLTST is zero or exceeds the number of lines in the line group.
- 0C Invalid test type or transmission count. The value specified by the X operand is undefined, or the value specified by the Y operand exceeds 99.
- 10 The skip bit of the addressing list entry specified by the ENTRY operand is on.
- 14 A line error occurred during Open.
- 18 Online test facility was not specified in the EROPT operand of the DCB macro.

Note: All nonzero return codes indicate that no I/O operation was initiated; therefore the program must not issue a WAIT or TWAIT macro for an ONLTST macro that resulted in a nonzero return code.

Completion Codes: On completion of an online test operation, a completion code is set in the high-order byte of the event control block for the line being tested. The code, in hexadecimal notation, indicates the nature of the completion:

- 7F Normal completion: Channel end and Device end.
- 41 Operation completed with I/O error: The DECB for the line does not contain error indicators when this occurs. The operator at the computer executing BTAM receives a message indicating the nature of the error. It is suggested that the user program check the completion code and if it is 41, issue a Write-to-operator-with-reply (WTOR) macro to permit the operator to determine what further action should be performed (for example, retry the online test by reissuing the ONLTST macro, or indicate to the user program that no further Read or Write operations can be

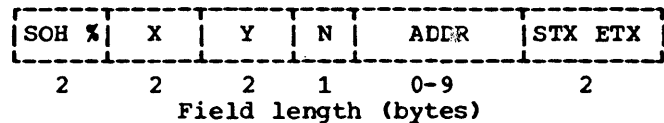
performed on that line). In the latter case the operator can, after the error condition has been cleared, notify the program that I/O operations may be resumed.)

FORMATS OF RFT, TEST, AND CONSOLE MESSAGES

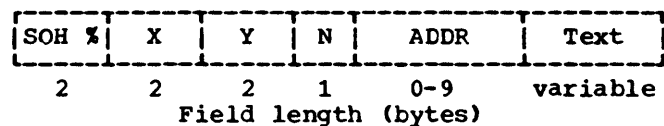
Request-For-Test Messages

An RFT message has one of three formats.

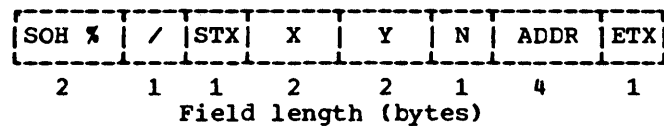
For type 00 tests except for RFT messages from a 2770 or 2780:



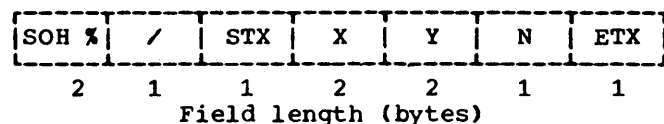
For type 01 tests and for type 00 RFT messages from a 2770 or 2780 and type 02-22 tests requested from a station other than a remote 3270:



For type 23-34 tests requested from a non-switched remote 3270:



For type 23-34 tests requested from a switched 3275:



- SOH % identifies the message as an RFT message.
- / identifies the message as an RFT message from a remote 3270.
- X specifies the test type (00-34). X is a two-byte zoned decimal field. Tests 23-28 are for switched and nonswitched 3270 devices using EBCDIC transmission code; tests 29-34 are for switched and nonswitched 3270 devices using ASCII transmission code.
- Y specifies the number of times (0-99) the test message is to be sent. Y is

a two-byte zoned decimal field. If X equals 0, and the remote station is a 2770, 2780, or 2972, Y must equal 1, because these stations transmit only the RFT message, not separate test messages. If the test message is to be sent to a remote 3284 or 3286 printer, Y should equal one, since the test message appears only once on a remote 3270 printer.

N

specifies the length (0-9) of the ADDR field. Code N as 0 and omit the ADDR field for tests over point-to-point lines, unless component selection characters are desired in the ADDR field. Always code N as 0 for the switched 3275 display station; for the nonswitched multipoint 3270 configuration, code N as 4.

ADDR

contains the address of the station or device to which the test message is to be sent, or (for 2770), component selection characters (DC1, DC2, or DC3). For a multipoint configuration, the ADDR field contains the selection address of the unit to which the test message is to be sent. (For the remote 3270, for example, 61C1 would be entered as the hexadecimal form of the selection address for EBCDIC device 1 on control unit 1. See Figures 48 and 49 for other remote 3270 control unit and device addresses.) This need not be the same unit that sent the RFT message. For a point-to-point configuration, the ADDR field contains the required component selection sequence, for example, ESC x, where x indicates the component to be selected. This sequence is limited to two characters. The ADDR field is not present if N=0.

Text

is the data and framing characters to be sent when X (test type) equals 00 or 01. For nontransparent text the data characters must be framed by STX or ETX. For transparent text the data characters must be framed by DLE STX and DLE ETX.

Test Messages

The contents of test messages are determined by the X field (test type) of the RFT message that initiates transmission of the test message. The values of X, the contents of the corresponding test message, and the configurations for which the test types are valid, are as follows:

X=00 For this test type, the test message is sent Y times, except for an RFT message from a 2770 or 2780, in which case the text is sent as part of the RFT message, not separately

(RFT messages from a 2770 or 2780 must specify a Y value of 1). The RFT and test messages are acknowledged by DLE, ACK-1 if received without errors, by NAK if a data check is detected, and are not responded to at all if any other ending condition is detected. The RFT message and the following test messages (or included text data) can be received from any type of remote BSC station: System/370, System/3, 1130, 1800, 2715, 2770, 2780, and 2972.

Note: If this test type is specified, the size of the input area specified by the Read Initial operation that receives the RFT message (by means of the DECB length parameter) must be large enough to receive the entire RFT message, including the text portion.

X=01

For this test type, the content of the test message is identical to the text portion of the RFT message, including the framing characters. The text is transmitted Y times. This message may be sent to any type of remote BSC station: System/370, System/3, 1130, 1800, 2715, 2770, 2780, and 2972.

For the remaining test types, the text of the test message is predefined by the on-line test routine.

X=02

Transparent EBCDIC Message:

DLE STX ...Text... DLE ETX

The text consists of all 256 EBCDIC codes in collating sequence order. This message may be sent to a System/370, System/3, 1130, 1800, 2715, and 2770.

X=03

Transparent USASCII Message:

DLE STX ...Text... DLE ETX

The text is in USASCII code (high-order bit always zero), and consists of all 128 USASCII codes in collating sequence order. This message may be sent only to a System/370.

X=04

Normal EBCDIC Message:

STX SYN SYN ...Text... ETX

The text is in EBCDIC code, and consists of the 245 non-data link control characters. The characters excluded are SOH, STX, ETX, ETB, EOT, ENQ, ACK, NAK, SYN, US, DLE. This message may be sent to a System/370, System/3, 1800, and 2770. (The text includes several terminal control characters, such as CR, HT, VT, and FF, that, when sent

to an output device, cause the associated function to occur, if the device is capable of performing that function. For example, the HT or FF characters in text sent to a terminal printer will cause the printer to execute the horizontal tab and forms feed operations, if the printer is so equipped.)

X=05 Normal USASCII Message:

STX SYN SYN ...Text... ETX

The text is in USASCII code and consists of the 117 non-data link control characters. The excluded characters are the same as for X=04. This message may be to a System/370, System/3, 1800, 2770, and 2780. (The text includes several terminal control characters, such as CR, HT, VT, and FF, that, when sent to an output device, cause the associated function to occur, if the device is capable of performing that function. For example, the HT or FF characters in text sent to a terminal printer will cause the printer to execute the horizontal tab and forms feed operations, if the printer is so equipped.)

X=06 Alphameric USASCII Message:

STX SYN SYN A B C D E F G H I J K
L M N O P Q R S T U V W X Y Z 0 1 2
3 4 5 6 7 8 9 ETX

This message may be sent to a System/370, System/3, 1800, 2770, and 2780.

X=07 USASCII Printer Message:

STX ESC Q A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX

This message is used to test the IBM 2780 printer. It may also be sent to a System/370, 1800, and 2972; these stations treat the ESC Q sequence (printer selection code) as data.

X=08 USASCII Punch Message:

STX ESC 4 A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX

This message is used to test the IBM 2780 card punch. It may also be sent to a System/370, 1800, and 2972; these stations treat the ESC 4 sequence (punch selection code) as data.

X=09 Transcode Printer Message:

STX ESC / A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX

This message is coded in Transcode and is used to test the IBM 2780 printer. It is valid only for a 2780 on a switched line or a non-switched multipoint line.

X=10 Transcode Punch Message:

STX ESC 4 A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX

This message is coded in Transcode and is used to test the IBM 2780 card punch. It is valid only for a 2780 on a switched line or a non-switched multipoint line.

X=11 Transcode Message:

STX SYN SYN A B C D E F G H I J K L
M N O P Q R S T U V W X Y Z 0 1 2 3
4 5 6 7 8 9 ETX

This message is coded in Transcode and may be used to test either the card punch or the printer of an IBM 2780. It is valid only for a 2780, on any type of line configuration.

X=12 EBCDIC Printer Message:

This message has the same content as the Transcode printer message, X=09, except coded in EBCDIC. This message is used to test the IBM 2780 printer. It may also be sent to a System/370, 1130, 1800, and 2972; these stations treat the ESC / sequence (printer selection code) as data.

X=13 EBCDIC Punch Message:

This message has the same content as the Transcode punch message, X=10, except coded in EBCDIC. This message is used to test the IBM 2780 card punch. It may also be sent to a System/370, 1130, 1800, and 2972; these stations treat the ESC 4 sequence (punch selection code) as data.

X=14 EBCDIC Alphameric Message:

This message has the same content as the Transcode message, X=11, except coded in EBCDIC. This message may be used to test either the card punch or the printer of an IBM 2780.

It may also be sent to a System/370, System/3, 1130, 1800, and 2770.

X=15 EBCDIC Weak Pattern Message

STX SYN SYN ...text... ETX

The text consists of 74 NUL (X'00') characters, followed by six SYN (X'32') characters. This message may be sent to a System/370, System/3, 1130, 1800, 2770, and 2780.

(This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for switched lines) or business machine clock (for switched or nonswitched lines).)

X=16 EBCDIC Weak Pattern Message

STX SYN SYN ...text... ETX

The text consists of 40 bytes of X'AA', followed by 40 bytes of X'55'. This message may be sent to a System/370, System/3, 1130, 1800, 2770, and 2780.

(This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for nonswitched lines).)

X=17 Transcode Weak Pattern Message

STX SYN SYN ...text... ETX

The text consists of 80 SOH (X'00') characters. This message may be sent only to a 2780. This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for switched lines) or business machine clock (for nonswitched lines).

X=18 Transcode Weak Pattern Message

STX SYN SYN ...text... ETX

The text consists of 40 N's (X'15'), followed by 40 ESC (X'2A') characters. This message may be sent only to a 2780.

(This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for nonswitched lines).)

X=19 EBCDIC Weak Pattern Message (DLE SYN Insertion)

DLE STX ...text... DLE ETX

The text consists of 280 NUL (X'00') characters, followed by 10 SYN (X'32') characters. This message may be sent to a S/370, System/3, 1130, 1800, and 2715.

X=20 Transparent EBCDIC Message

DLE STX ...text... DLE ETX

The text consists of the characters U through Z, 0 through 9, and X'00' through X'3F' (a total of 80 characters). This message may be sent to a S/370, 1800, 2770, and 2780.

X=21 Transparent EBCDIC Message

DLE STX ...text... DLE ETX

The text consists of the characters A through Z, 0 through 9, and X'00' through X'53' (a total of 120 characters). This message may be sent to a S/370, 1800, 2770 and 2780.

X=22 Transparent EBCDIC Message

DLE STX ...text... DLE ETX

The text consists of the characters A through Z, 0 through 9, and X'00' through X'6B' (a total of 144 characters). This message may be sent to a S/370, 1800, 2770, and 2780.

The two SYN characters following the STX in nontransparent test messages are present to allow space for a component selection address, if required in a point-to-point configuration. If a component selection address is not required in the message, the SYNs are transmitted, but are deleted by the receiving station.

X=23 3270 Basic Test Message (EBCDIC)

This test message checks all alphameric characters at a display station or printer. It checks the sue of the WCC to sound the audible alarm and allows attribute field specification to be checked at a display station. It starts a printer, printing 40 characters to a line.

X=24 3270 Model 1 Align Test Pattern (EBCDIC)

This test pattern checks position alignment for the 480-character display station. It also checks the WCC for sounding the audible alarm. It starts a printer, printing 40 characters to a line.

X=25 3270 Model 2 Align Test Pattern (EBCDIC)

This test pattern checks position alignment for the 1920-character display station. It also checks the WCC for sounding the audible alarm. It starts a printer, printing 80 characters to a line.

results of an online test. Messages are in one of two formats: For messages reporting the results of BTAM-transmitted test messages, or of a BTAM-transmitted RFT message specifying a test type (X field) of 00:

```
-----
| IEC807I cuu ONLINE TEST xx yy tt nn
|           ii...ii
|-----
```

X=26 3270 Orders Test Message (EBCDIC)

This test message checks 3270 orders (for example, SF and SBA), checks the WCC for sounding the audible alarm, and uses high and normal intensities. It starts a printer, printing 64 characters to a line.

For messages reporting the results of test messages received by BTAM from a remote computer or terminal:

```
-----
| IEC808I cuu ONLINE TEST xx yy tt ll dd
|-----
```

X=27 3270 Universal Character Set Test Pattern (EBCDIC)

This test pattern, which is mainly intended for the printer, checks several solid lines of alphameric print containing the universal character set. It checks the WCC for starting the printer and prints 132 characters to a line (honoring NL and EOM orders). (If issued to a display station, it checks the WCC for sounding the audible alarm.)

The meanings of the message fields are:

- cuu indicates the address of the line (channel and unit)
- xx indicates the test type (X field of the RFT message).
- yy indicates the number of transmissions. For IEC807I messages, this value is obtained from the N field of the RFT message. For IEC808I messages, this value is accumulated by the online test routine as each test message is received by BTAM.
- tt indicates the number of occurrences of timeout errors.
- nn is the number of NAK responses to BTAM-transmitted test messages.
- ii is the terminal identification sequence. This is printed for tests on multipoint lines.
- ll indicates the number of occurrences of lost-data errors.
- dd indicates the number of occurrences of data check errors.

X=28 3270 NL/EOM Test Pattern (EBCDIC)

This test pattern, which is mainly intended for the printer, checks the end of message (EOM) order and multiple new-line (NL) orders. It checks the WCC for starting the printer and prints 132 characters to a line. (If issued to a display station, it checks the WCC for sounding the audible alarm.)

ONLINE TESTING FOR LOCAL 3270 DISPLAY SYSTEM

Online tests for local 3270 devices are requested from local display stations. Figure 103 summarizes the options. There are six types of tests, not all of which apply to all devices.

X=29-34 3270 Test Messages and Patterns (ASCII)

These test messages and patterns correspond to types 23-28. ASCII transmission code is used instead of EBCDIC.

Online testing between a System/370 computer and a local 3270 device requires that BTAM or an online diagnostic program be running in the computer. Only the local 3270 display system can initiate an online test. Tests are requested by sending a request-for-test (RFT) message to the computer. For a local 3270, (1) the cursor is

Figure 100 shows the types of online tests that can be used for each type of remote station, except 3270 display stations and printers.

Figure 102 shows the types of online tests that can be used for each remote 3270 display station or printer. Information pertaining to the 3275 applies both to the nonswitched 3275 and to the switched 3275 (equipped with the dial feature).

Console Messages

The online test facility prints on the console typewriter of the central computer the

positioned at the top left of an unformatted screen (by pressing the CLEAR key and then the RESET key, for example), (2) the text of the RFT message (test type, number of times, length of address, channel and unit address) is entered by means of the keyboard, and (3) the TEST REQUEST key is pressed to precede the text with control characters (SOH % / STX).

TYPES OF TESTS

Types 23-28

The local 3270 display station sends an RFT message, which neither contains nor is followed by a test message. The X field of the RFT message indicates which of six BTAM-defined standard test messages it is to return to a local 3270 device. The Y field indicates how many times the test message is to be sent. The ADDR field indicates which local 3270 device is to receive the test message. The device receiving the test message must be associated with the same DCB as the device sending the RFT message. The contents of test messages are given below.

APPLICATION RESPONSE TO RFT MESSAGES

The application program must be prepared to handle an RFT message. When a read initial operation puts the control characters SOH%/ (X'016C61') in the application program's input area, the program should first get the relative line number (RLN) of the 3270 device whose TEST REQUEST key was pressed (this RLN is contained in byte 3 of the DECPOLPT field of the DECB). It should then either:

- Issue another Read Initial to the line group that includes the 3270 device; or
- Issue a Write Initial instruction for a length of 1 byte to write the control character that unlocks the keyboard. (This operation ensures that the contents of the display screen are not changed.)

The application program should not issue an Erase Write instruction.

When it receives an RFT message, the application program also should check to see if the message is valid or invalid. If the message is valid, the residual count in the DECB indicates that 3 bytes of data were read. If the message is invalid, the residual count indicates that more than 3 bytes were read.

BTAM RESPONSE TO RFT MESSAGES

BTAM recognizes and responds to any RFT message received from a local 3270 display station provided that:

1. The online test facility is available (that is, T was specified among the EROPT options of the DCB macro instruction for the group of local 3270 devices).
2. The RFT message was received on a read initial operation (that is, a READ TI macro instruction was issued).
3. The input area is at least 300 bytes long (that is, the inlength operand of the READ macro instruction was at least 300). If buffering is used, the entire 300-byte area must be contained within one buffer.
4. The RFT message was received without error.

When BTAM recognizes an RFT message, control is given to the online test logic, which examines the message, generates the requested test message in the input area for the read operation, and sends the test message the requested number of times. If the test message was sent to the same device from which the RFT message was received, the read initial operation is restarted. If the test message was sent to a different device, the next I/O operation to the receiving station is posted complete with a completion code of X'44', indicating that the buffer contents are unpredictable. Device buffers are reset to default size by online test logic.

FORMATS OF RFT, TEST, AND CONSOLE MESSAGES

Request-for-Test Message

An RFT message from a local 3270 display station has the format:

SOH %	/	STX	X	Y	N	ADDR
2	1	1	2	2	1	3
Field length (bytes)						

- SOH % identifies the message as an RFT message.
- / identifies the message as an RFT message from a local 3270.
- X specifies the test type (23-28). X is a two-byte zoned decimal field.

Y specifies the number of times (1-99) the test message is to be sent. Y is a two-byte zoned decimal field.

N specifies the length (3) of the ADDR field.

ADDR contains three characters (0-9,A-F) that indicate the channel and unit address of the device that is to receive the test message.

Test Messages

The contents of test messages are determined by the X (test type) field of the RFT message that initiates the sending of the test message.

X=23-28 3270 Test Messages and Patterns EBCDIC

These test messages and patterns correspond to types 23-28 for remote 3270 display stations and printers. See the descriptions of test messages 23-28 above under "Online Testing for Binary Synchronous Communications Lines."

Figure 103 shows the types of online tests that can be used for each local 3270 display station or printer.

Console Messages

See the description of console messages above under "Online Testing for Binary Synchronous Communications Lines."

X's indicate the test type available for each remote 3270 device.					
Test Type ²	Content of Test Message	Display		Printer ¹	
		Buffer 480	Buffer 1920	Buffer 480	Buffer 1920
23/29	3270 Basic	X	X	X	X
24/30	3270 Align	X		X	
25/31	3270 Align		X		X
26/32	3270 Orders	X	X	X	X
27/33	3270 Universal Character Set			X	X
28/34	3270 NL/EOM Printer			X	X

¹For the 3284 model 3 attached to a 3275 only tests 27, 28, 33 and 34 are applicable

²Tests 23-28 are EBCDIC and 29-34 are the ASCII transmission code equivalents

Note: For devices with alternate buffer sizes, the default size is used for the tests.

Figure 102. Summary of BSC Online Test Options for Remote 3270 Devices

X's indicate the test types available for each local 3270 device.					
Test Type	Content of Test Message	Local CU			
		Display		Printer	
		Buffer 480	Buffer 1920	Buffer 480	Buffer 1920
23	3270 Basic	X	X	X	X
24	3270 Align	X		X	
25	3270 Align		X		X
26	3270 Orders	X	X	X	X
27	3270 Universal Character Set			X	X
28	3270 NL/EOM			X	X

Note: For devices with alternate buffer sizes, the default size is used for the tests.

Figure 103. Summary of Online Test Options for Local 3270 Devices

This Page Intentionally Left Blank

APPENDIX A: FORMAT OF TERMINAL LISTS

This appendix illustrates each of the various kinds of terminal lists given under "Defining and Modifying Terminal Lists," elsewhere in this manual.

Each terminal list consists of one or more entries, each representing a remote station or a specific component of a remote station. Terminal lists vary in format; the illustrations in this appendix show how each type is organized.

Note: Terminal lists are not used for the local 3270 display system.

In lists of the OPENLST and WRAPLST type, each entry contains a control byte, illustrated in Figure 103. The bits in this control byte have the following meaning:

<u>Bit Position</u>	<u>Meaning</u>
0	If on, indicates that the entry is the last in the list.
1	If on, indicates that the entry is to be skipped when polling or addressing. If off, indicates an active entry. This bit is turned on and off with the CHGNTRY macro.
2	If on, indicates that the list is a wraparound list.
3-7	List entry number. Each entry is numbered successively starting with 1. This field limits to 31 the number of terminal or component entries for a list created by the DFTRMLST macro. This field is not presently used by BTAM, but is reserved for later use. Large lists can be created by coding a series of DFTRMLST macro instructions of the OPENLST type. If a wrap-around list is desired, code a series of DFTRMLST macros of the OPENLST type, and follow the last in the series by the instruction DC HL2'-n', where n is the number of bytes occupied by the terminal list entries.

Note: In the examples the polling and addressing characters and the identification sequences are shown as alphabetic and numeric characters, but you must code them in the DFTRMLST macro as the hexadecimal representation of the appropriate transmission code bit patterns.

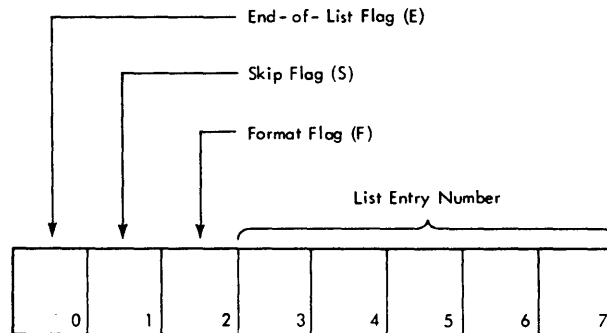


Figure 103. Format of Control Byte for OPENLST and WRAPLST Entries

OPENLST Format

Each entry in an open list (polling or addressing) consists of a one (1030) or two (all others) byte field for the polling or addressing characters plus the control byte. Examples for 1050 and 1030 are shown in Figure 105.

WRAPLST Format

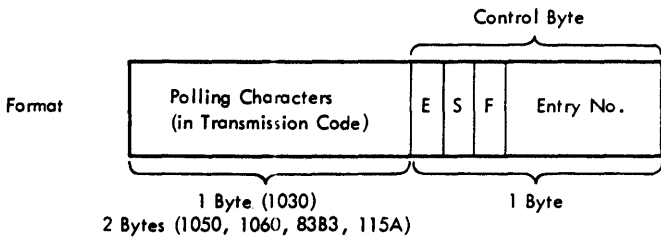
Wraparound polling lists differ from open lists in two ways:

1. Format bit (bit 2 in control byte) is on in the last entry.
2. A two-byte field follows the last entry and contains a negative binary value used by the polling restart routine to find the start of the list.

An example is shown in Figure 106.

DIALST Format

Terminal lists for stations on switched lines are illustrated in Figure 107.



Example IBM 1050

A	5	0	0	0	0	0	0	0	1
A	6	0	1	0	0	0	0	1	0
B	5	0	0	0	0	0	0	1	1
B	6	0	0	0	0	0	1	0	0
C	5	0	1	0	0	0	1	0	1
E	0	1	0	0	0	0	1	1	0

Example IBM 1030

D	0	0	0	0	0	0	0	1
E	0	0	0	0	0	0	1	0
F	1	0	0	0	0	0	1	1

Figure 105. Open Polling or Addressing List (OPENLST): Format and Examples

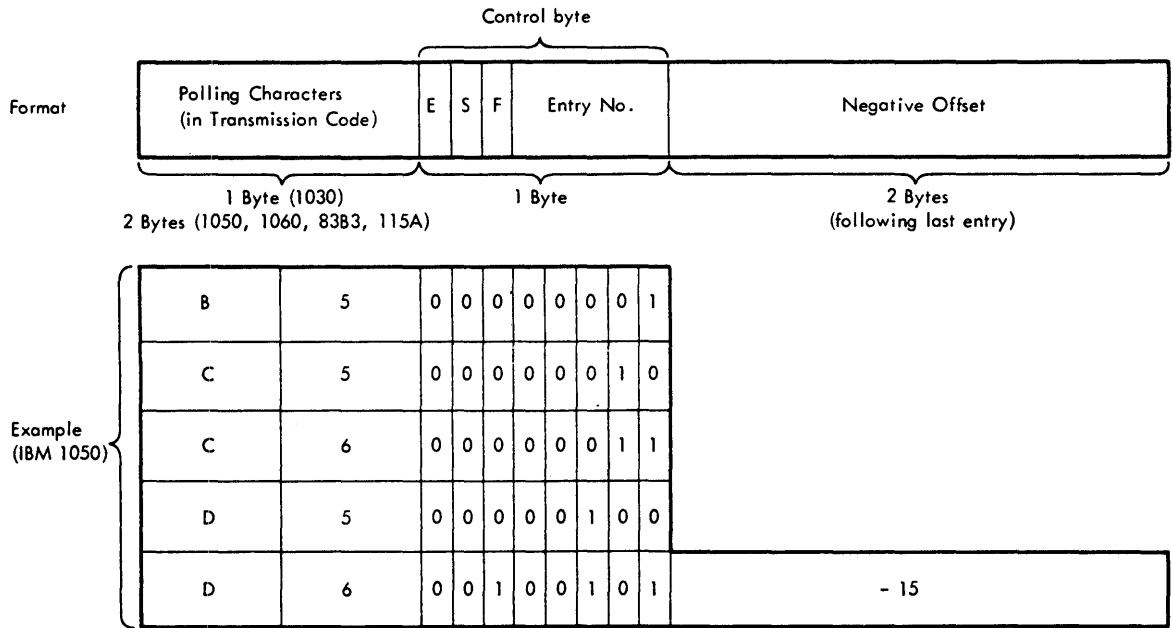


Figure 106. Wraparound Polling List (WRAPLST): Format and Example

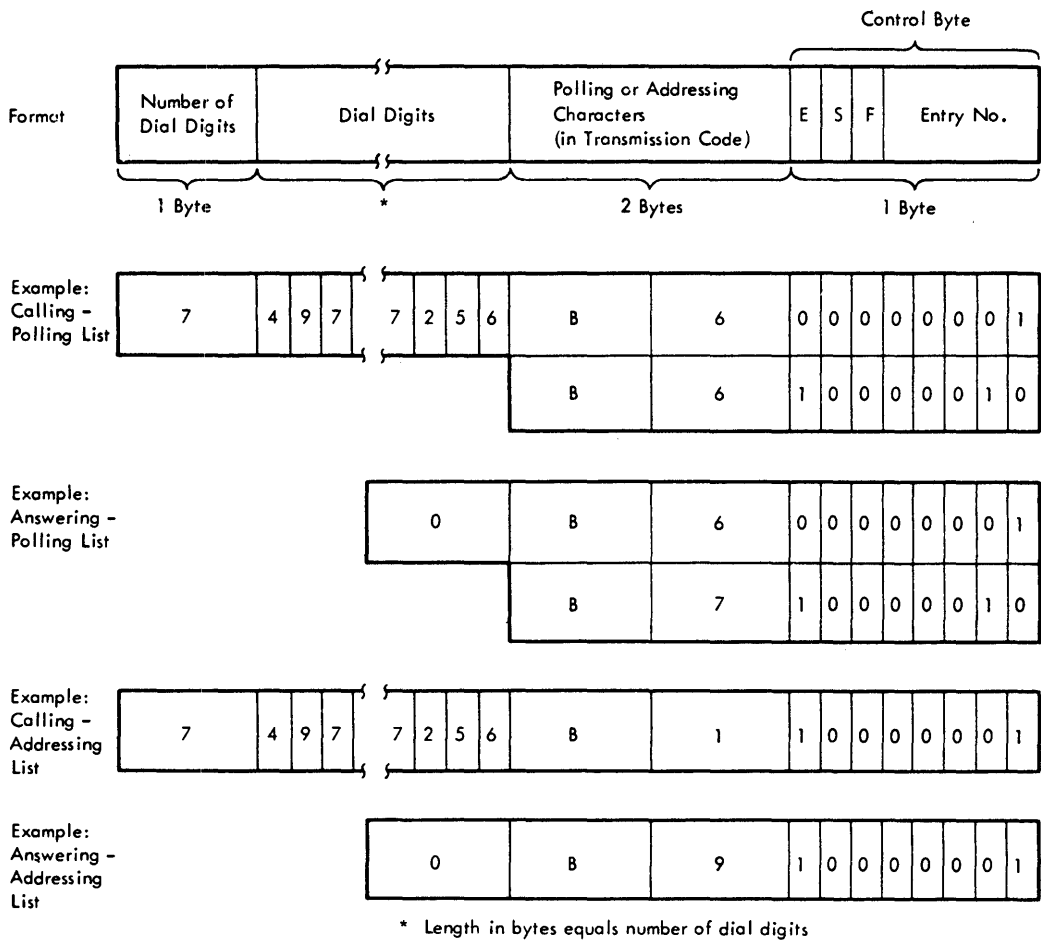


Figure 107. Dial List (DIALST): Format and Examples

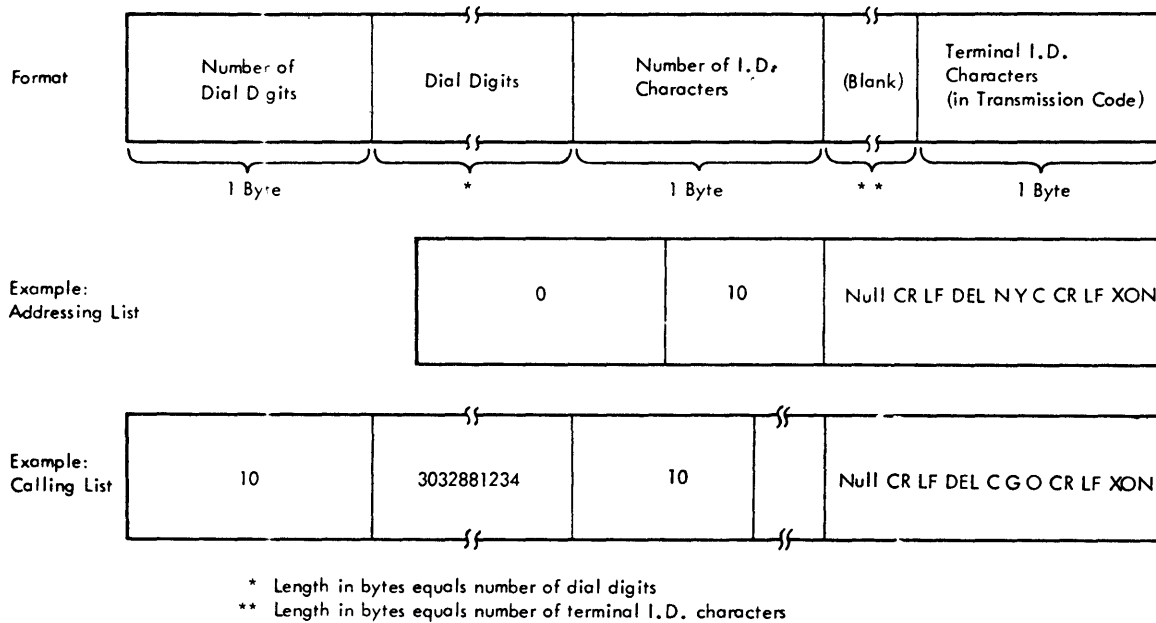


Figure 108. Identification List (IDLST): Format and Example

IDLST Format

Terminal lists for TWX terminals (Models 33 or 35) are illustrated in Figure 108.

SSALST and SSAWLST, AUTOLST and AUTOWLST Format

Terminal lists for all stations for which Autopoll is employed are illustrated in Figure 109.

- TE the total number of entries in list (1-253)*
- AE the total number of active entries in list (0-253)*
- NNN entry width $P_i + I_i$ (2-9)
- W wraparound flag (on for SSAWLST and AUTOWLST, off for SSALST and AUTOLST)

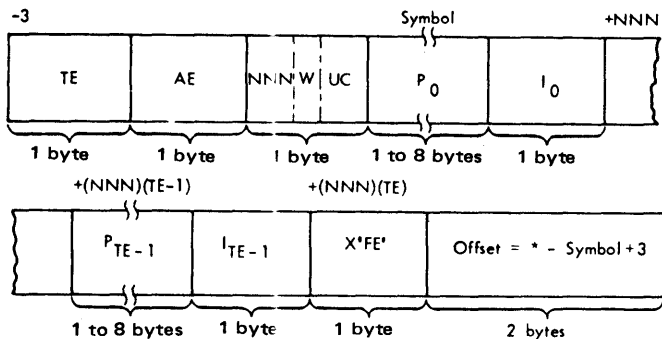


Figure 109. Open and Wraparound Autopoll Lists for Start-Stop (SSALST, SSAWLST) and BSC (AUTOLST, AUTOWLST): Format

UC usage count (0-15). The usage count indicates the total number of polling operations using the terminal list at any one time.

Pi polling characters (1 or 2 bytes). The value X'FE' must not be used as a polling character.

Ii index (1-253)*

X'FE' scan stop byte used to find end of list.

OFFSET two-byte field used to find heading of list from end of list.

* TE, AE, and Ii can be as high as 253, but at the time of publication, the assembler imposes additional restrictions.

BSCLST Format

Terminal lists for BSC communication over a switched line with ID verification are illustrated in Figure 110.

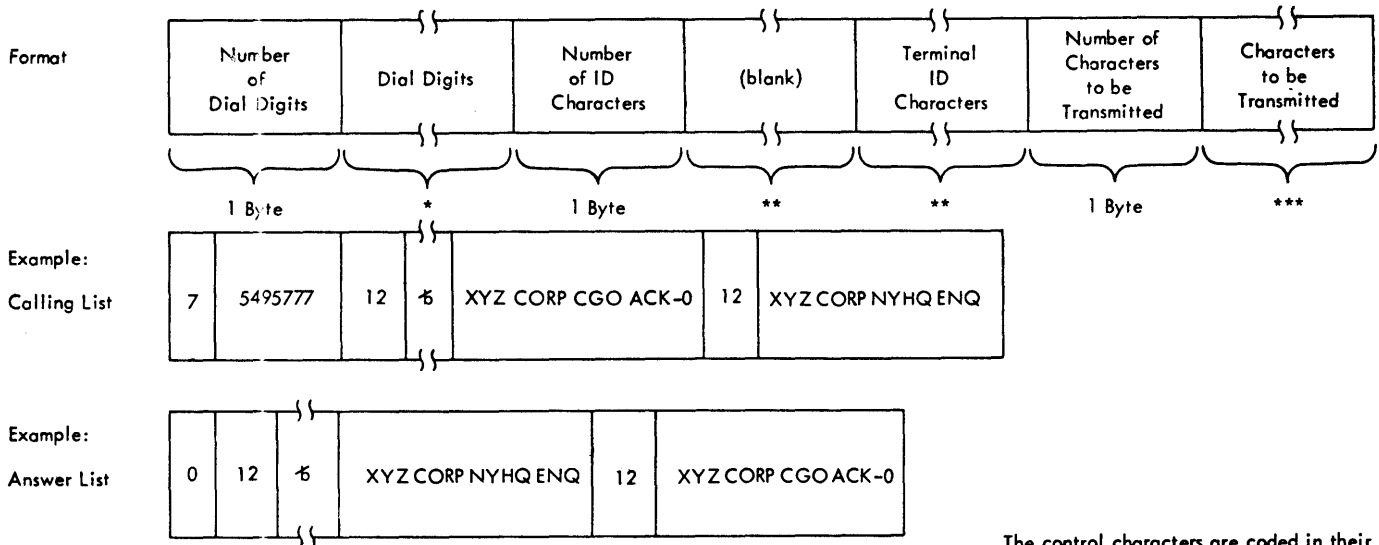
WTALST Format

Terminal lists for World Trade telegraph terminals are illustrated in Figure 111.

SWLST Format

The format and contents of the header and entries of a calling and answering list of the SWLST form is as follows (see Figure 112).

<u>Field</u>	<u>Contents</u>
(HEADER)	
Pointer to Sequence Matching Received Sequence:	Address (right-adjusted) of the last authorized ID sequence that was received prior to completion of the READ Connect or WRITE Connect operation. (Byte 0 contains X'FF' to indicate that the list is of the SWLST form.)
Number of List Entries:	Number (binary) of entries in the list (that is, the number of different authorized ID sequences that will be honored).
Entry Length:	Number of bytes (binary) in each entry in the list. This number is specified by the entry-length operand of the DFTRMLST macro, and should equal the number of bytes required to accommodate the longest expected ID sequence, plus the user-data field (0 or 4), plus one (for the control byte).
Read-In Area Length:	Number (binary) of characters in the longest expected ID sequence. This number will have a minimum value of 2, to accommodate a two-character sequence such as DLE EOT.

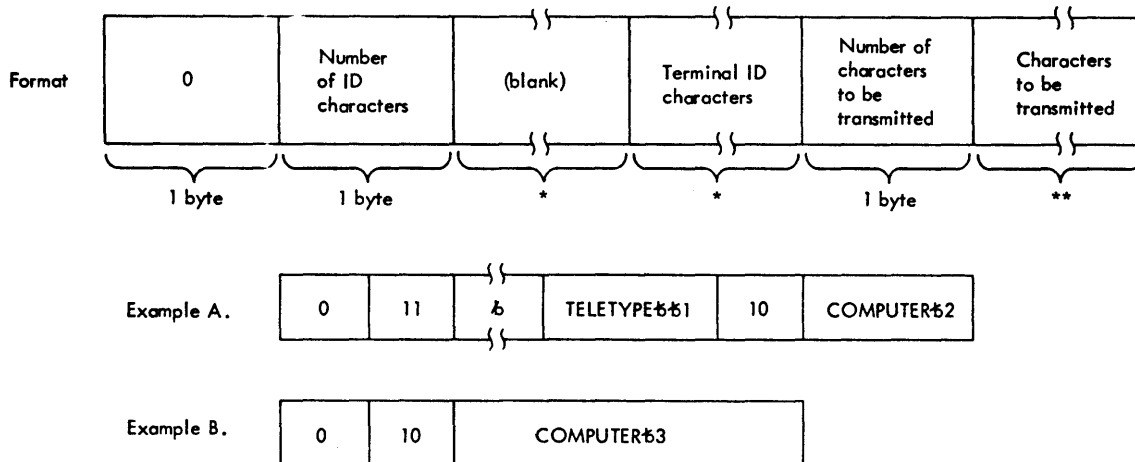


The control characters are coded in their hexadecimal equivalents shown below:

- * Length in bytes equals number of dial digits.
- ** Length in bytes equals number of terminal ID characters.
- *** Length in bytes equals number of characters to be transmitted.

ENQ - X'2D'
ACK-0 - X'1070'

Figure 110. BSC Dial List (BSCLST) with ID Verification: Format and Examples



- *length in bytes equals the number of terminal ID characters to be received.
- **length in bytes equals the number of computer ID characters to be transmitted.

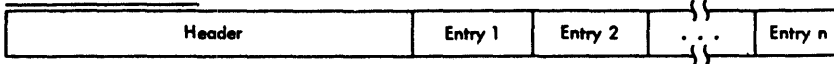
If the transmission code used with the WT terminals is the International Telegraph Alphabet Number 2, these terminal lists would be defined by coding:

Example A:
DFTRMLST WTTALST,0,11,0110091001150D1004043D,10,0E03070D1C01100A0439

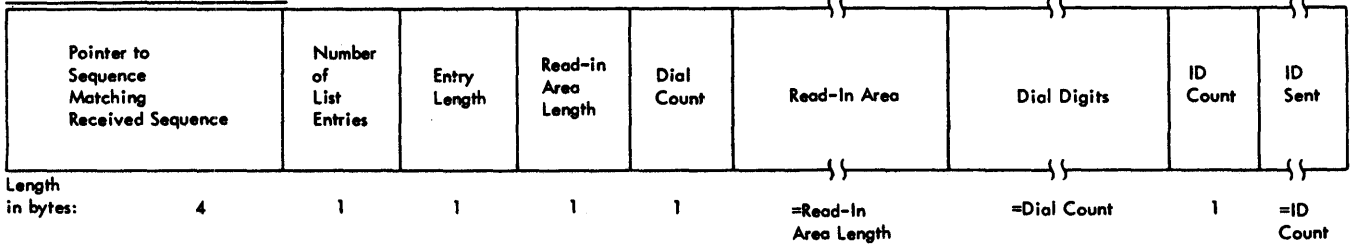
Example B:
DFTRMLST WTTALST,0,0,0,10,0E03070D1C01100A0430

Figure 111. WT Terminal List (WTTALST): Format and Examples

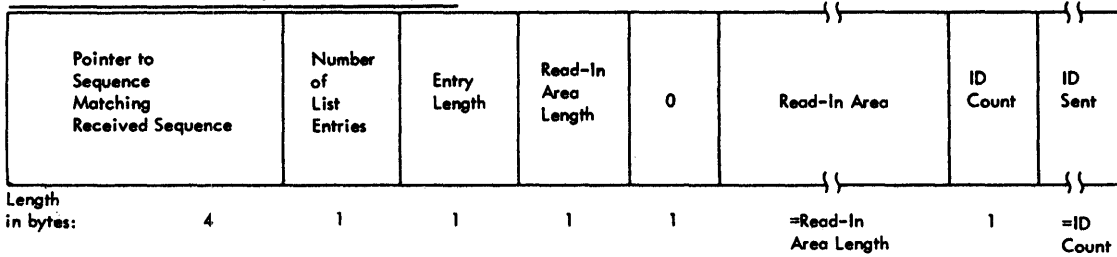
General Format of List:



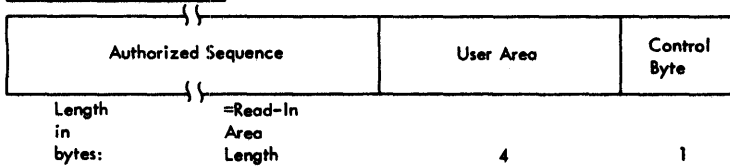
Header of Auto-Dial Calling List:



Header of Manual-Dial Calling List or Answering List:



Entry including User Area:



Entry omitting User Area:

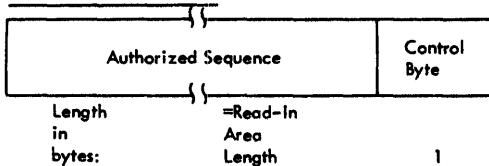


Figure 112. Calling and Answering Lists for Expanded ID Verification (SWLST): Format

- Dial Count:** For an automatic dialing list: number (binary) of dial digits to be used in calling the remote station. For a manual dialing list or an answering list: 0.
- Read-In Area:** Area into which the ID response is read from the remote station. The length of this field is determined by the longest possible sequence that can be received, but no less than two bytes.
- Dial Digits:** The dial digits (binary), for an automatic dial calling list. For an answering list or a manual-dial calling list, this field is omitted.
- Id Count:** Number of characters (binary) in the sequence defined in the ID sent field.

Id Sent For a calling list, this field contains the characters of the ID-ENQ sequence to be sent to the remote station. For an answering list, this field contains the ID ACK-0 sequence to be sent to the remote station when the control byte value of the entry containing the received ID ENQ sequence is 0. It is recommended that the first two characters of each ID sequence be identical, to provide greater identification reliability.

(ENTRY)
Authorized Sequence: The characters composing an authorized sequence that can be received. The size of this field is usually the length of the Read-In Area. Since this length is never less than two, the size of this field is less than the size of the Read-In Area when an answering list is defined with only one entry, containing the single ENQ character. Authorized sequences can be of different lengths; each sequence is left-adjusted in the Authorized Sequence field.

User Area (optional): May contain a user-specified relocatable expression for each list entry. This four-byte field is included in each entry if you specify the userlength operand of the DFTRMLST macro as 4; otherwise, the field is omitted.

Control Byte: A value, specified in the controlvalue operand of the DFTRMLST macro, indicating the action BTAM is to perform when an authorized ID sequence is received. The value may be 0, 1, or 2. (See description of the DFTRMLST macro for the significance of these values.)

APPENDIX B: DATA EVENT CONTROL BLOCK

The format of the data event control block (DECB) is shown in Figure 113. Its contents are described below.

DECSDECB

standard four-byte ECB. Only the first byte of this field is of concern to the BTAM programmer. This byte can contain the following hexadecimal completion codes:

<u>Hex Value</u>	<u>Meaning</u>
00	After READ OR WRITE macro instruction issued, before WAIT.
80	WAIT macro instruction issued; event not complete.

Note: As long as the wait bit is on, the contents of some DECB fields are unpredictable (the fields are used internally by BTAM); the contents of the DECB should therefore be considered meaningful only after the Read or Write operation has been completed (that is, the completion bit is on).

- 7F Normal completion: The Read or Write operation has ended with indications of Channel End-Device End and either Unit Exception or Incorrect Length, or both, if they are normal conditions (for example, Unit Exception indicating end-of-transmission or negative response to polling). The user program should examine the bits in DECFLAGS to determine the status of the operation.
- 41 Complete with I/O error; the program should examine the bits in DECERRST to determine the kind of error.
- 44 The I/O request was rejected, because (1) a device error was detected after the last I/O operation on the device was posted complete or (2) a request-for-test message was received from a local 3270 display station requesting that a test message be sent to another local 3270 device. The buffer contents are unpredictable.
- 48 Enable Command Halted or I/O Operation Purged: Indicates one of the following:
- An Enable command (automatic answering function for a switched line) was terminated by Halt I/O as a result of a RESETPL macro instruction (second operand omitted or specified as ANSRING).
 - An Enable command was terminated as a result of closing (CLOSE macro instruction) a line group with Enable commands outstanding.
 - An I/O operation was purged at Channel End interrupt time as a result of closing the line group while I/O operations were still in progress.
 - A Read Initial operation for World Trade telegraph has ended with a Halt I/O command because a RESETPL macro instruction was issued (second operand omitted).

- A Read Initial operation for the local 3270 display system was canceled, because a RESETPL macro instruction was issued. This may be caused by a device becoming ready if READYQ=0 was specified in the DCB.

DECTYPE Operation type:

first byte: (In any combination)

- bit 0 - current operation is a Read operation using Autopoll
- bit 1 - RJE (Remote Job Entry) requested WTO timeout message suppression
- bits 2-3 - (reserved)
- bit 4 - write inquiry operation
- bit 5 - 'entry' coded as 'S'
- bit 6 - 'area' coded as 'S'
- bit 7 - 'length' coded as 'S'

second byte:

- bit 0 of this second byte specifies Reset for Read Initial and Reset (TIR), Write Initial and Reset (TIR), Read Continue and Reset (TTR), Write Continue and Reset (TTR), Read Conversational and Reset (TVR), Write Conversational and Reset (TVR), Read Repeat and Reset (TPR), Write at Line Address and Reset (TLR), and Write Erase and Reset (TSR).
- bits 1 and 2 are reserved.
- bits 3, 4, 5, 6, and 7

Hex

Value	Operation
00	Write Break (TB)
01	Read Initial (TI)
02	Write Initial (TI)
03	Read Continue (TT)
04	Write Continue (TT)
05	Read Conversational (TV)
06	Write Conversational (TV)
07	Write Erase Alternate (TSA)
07	Read Repeat (TP), or Read Continue with Identification Exchange (TE) (WT terminal)
08	Write Positive Acknowledgment (TA)
09	Read Skip (TS)
09	Read Inquiry Monitor (TQM)
0A	Write Negative Acknowledgment (TN), Write Reset (TR), Write Disconnect (TN) (TWX)
0B	Read Buffer (TB)
0B	Write Reset Monitor (TRM)
0C	Write at Line Address (TL), Write Initial Optical (TIO), Write Initial Transparent Block (TIE)
0D	Write Initial Conversational (TIV), Read Continue with Leading Acknowledgment (TTA)
0E	Write Erase (TS), Write Invitational Optical (TCO), Write Continue Transparent Block (TTE)
0F	Write Continue Conversational (TTV)
10	Write Disconnect (TD) (BSC)
11	Read Connect (TC), Read Modified (TM)
12	Write Initial Transparent (TIX), Write Conversational Optical (TVO), Write Unprotected Erase (TUS)
13	Read Continue with Leading Graphics (TTL), Read Buffer from Position (TBP)
14	Write Continue Transparent (TTX)
15	Read Inquiry (TQ)
16	Write Inquiry (TQ)
17	Read Repeat with Leading Graphics (TPL)
18	(Reserved)
19	Read Initial Inquiry (TIQ), Read Modified from Position (TMP)
1A	Write Wait Before Transmitting (TW)

1B Read Interrupt (TRV)
 1C Write Connect (TC)
 1D Write Initial Conversational Transparent (TIVX)
 1E Read Connect with Tone (TCW)
 1F Write Continue Conversational Transparent (TTVX)

DECLNGTH
 Buffer length or message area length.

DECONLTT
 When BSC online test is in control of the line, the 0 and 1 bits have the following meaning:
 bit 0=0 indicates online test was requested by RFT
 =1 indicates that online test has been requested by the ONLTST macro
 bit 1 (meaningful only when online test has been initiated) =0
 =0 if test messages are sent by BTAM;
 =1 if test messages are received by BTAM

DEDCBAD
 Address of associated DCB.

DECAREA
 Address of the message area or first buffer. The high-order byte of this field must always contain zero.

DECSSENSO
 Sense information, as set by the control unit, when the CSW status (DECCSWST) indicates a unit check.

Bit	Meaning
0	Command reject
1	Intervention required
2	Bus out check
3	Equipment check
4	Data check
5	Overrun or unit specify (local 3270)
6	Lost data or control check (local 3270)
7	Timeout or operation check (local 3270)

DECSSENS1
 (Reserved)

DECCOUNT
 Residual count from the CSW for the last CCW that was executed.

DECCMCO
 Command Code (one byte) identifies the type of command upon which the error occurred.

Code (Hex)	Command	Code (Hex)	Command
00	Test I/O	13	Set Address 0
01	Write	17	Set Address 1
02	Read	1B	Set Address 2
03	I/O NOP	1E	Address Prepare
04	Sense	1F	Set Address
05	Diagnostic Write (Auto Wrap)	23	Set Mode
06	Prepare	27	Enable
09	Poll	29	Dial
0A	Inhibit	2F	Disable
0D	Break	41	Write Break
0E	Search	42	Read Clear
12	Diagnostic Read	D4	Release
		F4	Reserve

DECENTRY

Address of the terminal list entry specified in the entry operand of the READ or WRITE macro instruction, prior to a Read or Write operation; after the operation it contains the address of the polling list entry that was last polled.

DECFLAGS

Status flags that may be set regardless of whether there was an I/O error (that is, the completion code in the DECSDECB may be either 7F or 41).

- Bit 0: For start-stop operations, this bit is reserved. For BSC operations, it indicates that a WACK (Wait-before-transmit) was received, if bit 1 is also on. If bit 1 is not on, bit 0 indicates that an error status message was received. (An error status message begins with SOH % S and provides status information about a remote station.) If a WACK has been received, the user program should respond by sending ENQ (or EOT, if transmission is to be ended), unless the WACK was received in response to selection (multipoint line), in which case the user program should retransmit the selection characters, that is, reissue the WRITE macro.
- Bit 1: For start-stop operations, this bit is reserved. For BSC operations, it indicates that some response other than ACK-0 or ACK-1 was received into the DECRESFN field. Examination of the response will determine which action should be taken to reestablish proper communication. (This bit is set when WACK (see also bit 0) is received or when RVI (see also bit 6) is received.
- Bit 2: For start-stop operations, this bit is reserved. For BSC operations, it means that an incorrect alternating acknowledgment was received: ACK-1 received when ACK-0 was expected, or vice versa. If this bit is on and the completion code for the operation is 7F (that is, no line transmission error occurred), a complete message may have been lost.
- Bit 3: The ID received from a TWX 33/35 or a BSC station did not equal the expected ID as defined in the terminal list specified in the WRITE TI, WRITE TC, or READ TC macro instruction, or the index received as a result of an Autopoll operation did not match the index byte in any of the active entries in the polling list. For BSC (nonswitched line) this bit, when on, indicates that contention has occurred and this is not the control station. The control station should retry this WRITE and this (remote) station should issue a READ Initial. For World Trade telegraph terminals, this bit indicates that contention occurred, or that the ID

received from a terminal did not equal the expected ID as defined in the terminal list specified in the READ TF macro instruction. Test the TP code in the DECB to determine which condition occurred.

Bit 4: No buffer was available upon completion of a dynamic buffering Read command. The last buffer is posted complete and the remainder of the message is read from the communications line (under control of a dynamic buffering Read Skip command), but the data is not placed into storage.

Bit 5:

- The end of the terminal list has been reached, or all the skip bits are on. This is an indication that:

1. A negative response to polling has been received from the terminal represented by the last active (non-skipped) entry in an open polling list (OPENLST, SSALST, AUTOLST);
2. A negative response to polling has been received following a RESETPL macro instruction of the POLLING type (second operand omitted or specified as POLLING);
3. All of the entries in a wraparound polling list (WRAPLST) are inactive (all skip bits are on).

Note: Condition 3 can occur only as a result of one or more skip bits being turned on after initiation of a programmed polling operation with a wraparound polling list. If all skip bits were on at the time that the READ macro instruction was executed, no I/O operation would be initiated.

- Negative response to addressing has been received.
- The last message sent by a World Trade telegraph terminal ended with EOT or a timeout.
- Power is off or other Intervention Required condition exists for 2741.

Bit 6: WT Terminals: Message ended with WRU signal.
BSC Stations: RVI sequence received (see also bit 1).
2741: Write operation was ended by terminal interrupt.
Remote 3270: If bit 6 is on, but bit 1 is not on, an error status message was received. (An error status message for a remote 3270 device begins with SOH % R and provides sense and status information about the device.)

Bit 7: WT Terminals: Contention condition was encountered.
BSC Stations: STX ENQ sequence was received.
Local 3270: OLTEP is using the device to run diagnostics.

For local 3270, X'F0' in the DECFLAGS field indicates that a local 3270 has become ready (READYQ was specified). DECPOLPT+3 will contain the rln of the ready device.

DECRLN

Relative line number.

DECRESPN

Start-stop: First byte: one-character response to addressing
Second byte: one-character LRC/VRC response to text
BSC: two-character response to addressing, ENQ, or text. Exception: responses to text for Write TIV, TIVX, TTV, and TTVX are read into the input area designated by the WRITE macro.

DECTPCOD

TP Op. code. Bits 2-7 of these codes identify types of channel commands that are not identifiable by the command code alone. Bits 0 and 1 are used in conjunction with, but independent of, bits 2-7, as described below.

Bit 0: Indicates the final command in the channel program (not necessarily the last command executed).

Bit 1: The command just executed was the first Read Text or Write Text CCW to be executed in a channel program using dynamic buffering.

Bits 2-7:

<u>Hex Value</u>	<u>Meanings</u>
00	Any command issued by Online Test routine.
01	Disable, when the disable is the first command of a channel program; dial, enable, prepare, write pad characters, or write wait-before-transmitting; or sense (World Trade telegraph terminals).
02	Write EOA EOT EOT EOT sequence prior to selection, write EOT sequence prior to polling or addressing, write response to text, write EOA and 15 idle characters (Basic 2740), or Write EOA PRE o (2740/2760).
03	Write polling or addressing character or write / (/ is the broadcast addressing character) (2740 with Station Control), turn-around sequence (TWX), CPU-ID sequence (TWX or BSC), Poll command with SSALST, SSAWLST, AUTOLST, or AUTOWLST, or write inquiry (ENQ).
04	Write space (2740 with Station Control), write 2848 command (2260R), write FIGS (83B3), write 1 (1030), write WRU, Identification, pad, or LTRS characters (World Trade terminals), or Sense (2740).
05	Read response to polling.
06	Read response to addressing.
07	Read ID response (TWX or BSC).
08	Write end of addressing character following addressing (on 1030, 1050, 1060, 2260R, or 2740). Write response to inquiry. Write response to text (BSC). Write EOB (2760/2740).
09	NOP or TIC following Poll in the polling list: SSALST, SSAWLST, AUTOLST, or AUTOWLST.
0A	Read index (Autopoll) or read response to polling (programmed polling).
0B	Read inquiry (BSC only).
0C	Read response to inquiry (BSC only).
10	Write at line address (2260R).
11	Read or write text. Write frame change sequence (2760/2740).
12	Read skip or TIC command for dynamic buffering.
13	Write end-of-transparent text (DLE ETX) characters (BSC).
14	(Reserved)
20	Read response to text (start-stop).

- 21 All reset commands.
- 22 Read skip.
- 23 Write break.
- 24 Any command issued during OPEN, LOPEN, or CLOSE (Set Address, Enable, Disable, and Set Mode commands).
- 25 Read Response to text (BSC).

DECERRST

Error status flags that may be set if an I/O error has occurred (that is, a completion code of 41 is placed in DECSDECB).

- Bit 0: The START I/O instruction resulted in a condition code of 3, indicating that the control unit or the specified line is not operational.
- Bit 1: An error condition that should not occur (is undefined for the particular command or device) has occurred.
- Bit 2: An error condition occurred on an I/O operation initiated by the error recovery routines: (1) as part of an intermediate recovery procedure, (2) as part of a diagnostic write/read procedure (2701 only), or (3) as part of a disconnect procedure for a switched line.
- Bit 3: A diagnostic write/read operation terminated in error, indicating a control unit failure (2701 only). An error occurred that makes the integrity of the device regeneration buffer doubtful (local 3270 only).
- Bit 4: A Disable command was issued to a switched line by the error recovery routines after detecting a permanent error on that line.

Note: If this bit is on after execution of error recovery procedures, the user program must execute an initial-type Read or Write operation, in order to reestablish the line connection.

Bits 5-6: (Reserved)

- Bit 7: Switched 3275 RFT error bit: The RFT message has been transmitted the specified number of times, followed by a Write Reset. The response to the Write Reset indicates that an error has occurred.

DECCSWST

contains the status bits from the CSW for the last CCW that was executed.

DECADRPT

pointer to the addressing list entry used in the previous operation.

DECPOLPT

for programmed polling, contains the address of the current entry in the polling list. For Autopoll, the high-order byte contains the index to the current polling list entry. The remaining bytes contain the address of the polling list (that is, the address of the first entry therein). For BSC online test operations, contains the address of the area in which user-specified text data is placed (for test messages). For local 3270 read operations, contains the relative line number of the device from which the message was read or the device that became ready (READYQ).

DECWLNQ

length of the data area in leading-graphics or conversational operations or when using READ TWC.

DECWAREA

address of the data area in leading-graphics and conversational operations or when using READ TWC. The high-order byte of this field must always contain zeros.

Fields Defined by User

It may be useful for the user program to maintain application-dependent information about the line and about the stations connected to the line. This may conveniently be done by appending to each DECB a sequence of fields containing the needed information, which might typically include:

- Line status: A one-byte field that indicates the status of the line; for example, active or inactive. The inactive bit might be set after a certain number of transmission errors have accumulated, to indicate to the message control routine that no further Read and Write operations are to be executed using that line.
- Address of User Terminal Table: This table would contain a series of fixed-length entries, one for each terminal, containing terminal information such as whether or not the terminal is active, and the addresses of the terminal list entries for that terminal.
- Terminal Count: A count of the number of terminals connected to the line.
- Processing Routine Address: Contains the address of the next routine to be given control for the line. For example, this field would contain the address of a line analysis routine to be given control upon completion of a Read or Write operation.

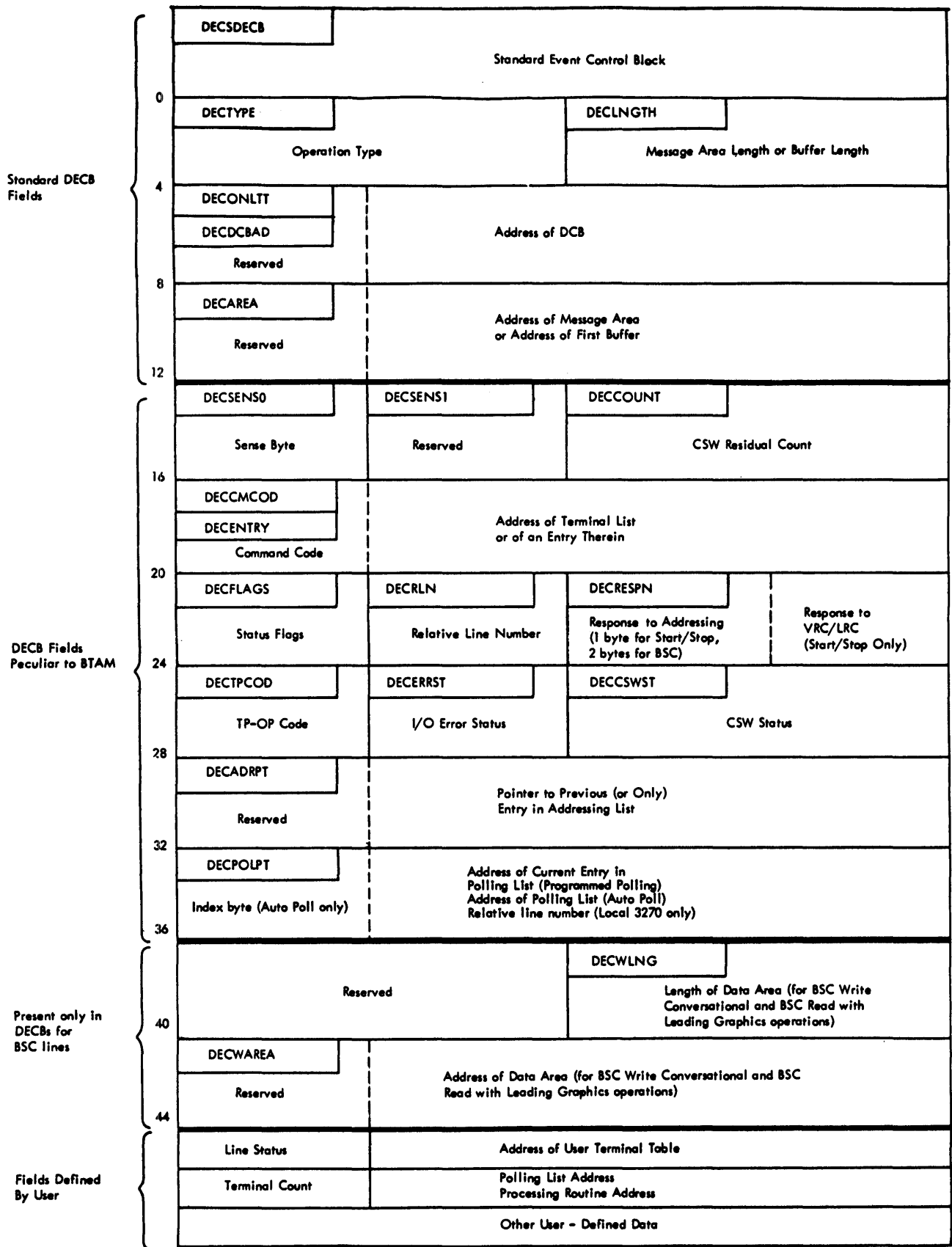


Figure 113. Format of Data Event Control Block

This Page Intentionally Left Blank

APPENDIX C: BTAM ERROR MESSAGES AND ABEND CODES

This appendix explains each of the BTAM-related error messages that may be printed during program execution at the console of the central computer, or at some other console, if the system includes the Multiple Console Support facility, or in the assembler listing in the SYSPRINT data set during program assembly. Also given are Abend codes 090 - 099 which may be issued during opening of a BTAM DCB.

Both system-generated and user-generated messages are described herein. System-generated refers to those messages printed at a console (or in an assembly listing) by the operating system; these begin with a standard identification, for example, IEC801I.

User-generated refers to those messages that are sent by the user (for example, the operator of a remote station) to the central computer for routing to the user program, console, or an error file on a system residence device.

ASSEMBLY ERRORS

These messages are produced by the assembler program during expansion of supervisor and data management macro instructions. They appear on the assembler listing in the SYSPRINT data set. See OS/VS VM370 Assembler Programmer's Guide for return codes.

IHB002 INVALID xxx OPERAND SPECIFIED-yyy

Explanation: An operand whose position or name is xxx was specified as yyy. The specified operand is invalid.

System Action: The macro instruction was partially expanded; expansion stopped on detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the invalid operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB072 LERB REQUESTED - EROPT=C ASSUMED

Explanation: LERB was coded in the DCB but EROPT=C (indicating a request for line error recording) was not coded.

System Action: The macro instruction was expanded normally with line error recording provided. Severity code=*

Programmer Response: Probable user error. Delete the LERB operand if line error recording is not wanted. If line error recording is wanted, code EROPT=C. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB073 LERB OMITTED - EROPT=C IGNORED

Explanation: Line error recording was requested by EROPT=C but no LERB address was given.

System Action: The macro instruction was expanded normally with no line error recording provided. Severity code=*.

Programmer Response: Probable user error. Delete EROPT=C if line error recording is not wanted. If line error recording is wanted, code a LERB address. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB074 EROPT=N - LERB IGNORED

Explanation: LERB was coded in the DCB but error recovery procedures were not requested (EROPT=N).

System Action: The macro instruction was expanded normally with no line error recording provided. Severity code=*.

Programmer Response: Probable user error. Delete the LERB operand if line error recording is not wanted. If line error recording is wanted, code EROPT=C. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB075 TABLENAME OPERAND REPEATED - XXX

Explanation: In the ASMTTAB macro instruction, a table name operand was coded more than once. XXX is the repeated operand.

System Action: The macro instruction was expanded normally. Severity code=*.

Programmer Response: Probable user error. Remove the duplicate operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB076 MACRO NAME FIELD BLANK - NAME REQUIRED

Explanation: A name must be specified in the name field for this macro instruction.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Code a name in the name field of the macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB078 XXX OPERAND REGISTER NOTATION INVALID - YYY

Explanation: For the XXX operand, the operand was not enclosed in parentheses or specified an invalid register. YYY is the invalid notation.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Correct the register notation or specify a valid register and reassemble. If the problem recurs, do the following before calling IBM for programming support:

IHB079 FIRST OPERAND REGISTER NOTATION REQUIRED

Explanation: Register notation is required for the first operand.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Specify a register notation for the first operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB080 ONE ECBLIST OPERAND ONLY REQUIRED

Explanation: The ECBLIST operand was omitted or more than one supplied.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error, Supply one and only one ECBLIST operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB085 DEVD = xx CODED - EROPT = y IGNORED

Explanation: In a DCB macro instruction, one of the following occurred:

- Both DEVD=BS and EROPT=N were coded. However, EROPT=N is invalid for binary synchronous devices. Error recovery procedures are required.
- Both DEVD=WT and EROPT=R, W, or T were coded. However, EROPT=R, W, or T is invalid for World Trade telegraph terminals.

System Action: The macro instruction was expanded normally. The EROPT operand was ignored. Severity code= *.

Programmer Response: Probable user error. Remove the EROPT operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB100 X OR Y PARAMETER NOT WITHIN ALLOWABLE VALUE RANGE.

Explanation: In the ONLTST macro instruction, either the X or Y operand specified an incorrect value. The X operand must specify a value from 00 through 22, and the Y operand must specify a value from 01 through 99.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Correct the X or Y operand in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB103 TEXT OR LENGTH MISSING WHEN X = 0 OR X = 1.

Explanation: In the ONLTST macro instruction, although the X operand specified 0 or 1, either the TEXT or the LENGTH operand was missing. Whenever the X operand specifies 0 or 1, the TEXT and LENGTH operands must also be specified.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Include both the TEXT and LENGTH operands in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB104 TEXT OR LENGTH MISSING.

Explanation: In the ONLTST macro instruction, either the TEXT or the LENGTH operand was missing. If one of these two operands is specified, the other operand must also be specified.

System Action: The macro instruction was not expanded. Severity code = 1.

Programmer Response: Probable user error. Include both the TEXT and the LENGTH operands in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB105 X GREATER THAN 1. TEXT AND LENGTH PARAMETERS IGNORED.

Explanation: In the ONLTST macro instruction, although the X operand specified a value greater than 1, the TEXT and LENGTH operands were also specified. Whenever the X operand specifies a value greater than 1, the TEXT and LENGTH operands should not be specified.

System Action: The macro instruction is expanded normally, and the TEXT and LENGTH operands are ignored.

Programmer Response: Probable user error. Remove the TEXT and LENGTH operands from the ONLTST macro instruction. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB107 DIALCOUNT AND DIALCHARS NOT IN AGREEMENT.

Explanation: In the DFTRMLST macro instruction, the length of the telephone number specified in the dialcount operand is not the same as the number of dial digits specified in the dialchars operand.

System Action: The dial digits are generated as specified in the dialchars operand without regard to the length specified in the dialcount operand. Severity code = 4.

Programmer Response: Probable user error. Correct the dialcount or dialchars operand in error. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB108 POLLING CHARACTERS ARE IMPROPER.

Explanation: In the DFTRMLST macro instruction, the number of entries specified in the polling list was greater than 253 or one of the polling characters in an entry was hexadecimal FE, a value that must not be used as a polling character.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Correct the polling list. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB109 LENGTH OF POLLING CHARACTERS PER ENTRY IS IMPROPER.

Explanation: In the DFTRMLST macro instruction, the entries in the polling list are not all of the same length.

System Action: All entries are truncated or expanded to equal the length of the first entry. Severity code = 4.

Programmer Response: Probable user error. Correct the polling list so that all the entries are of the same length. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB110 DIAL CHARACTERS INVALID IN WTLIST.

Explanation: In a DFTRMLST macro instruction specifying a list type of WTLIST, dial digits were specified. However, a list type of WTLIST should be used only where manual dialing is intended, and no dial digits can be specified.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Remove the dial digits from the DFTRMLST macro and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB111 LENGTH OR ADDRESS OF TONE OMITTED.

Explanation: In a DFTRMLST macro instruction specifying an answering list of the WTLIST type, either the length or the address of the data tone characters was omitted. However, both operands must be included for a list of this type.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Make sure that both length and adress operands are specified for an answering list of the WTLIST type. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB113 IDCOUNT AND IDSENT DO NOT AGREE.

Explanation: In a DFTRMLST macro instruction, the value specified for the idcount operand does not equal the number of characters specified by the idsent operand.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the idcount value and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB114 IDCOUNT IS TOO LARGE.

Explanation: In a DFTRMLST macro instruction, the value specified for idcount is greater than 16 (for a calling list of the AD or MD type), or is greater than 17 (for an answering list of the AN type).

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the idcount value and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB115 INVALID TYPE ATTRIBUTES.

Explanation: In a DFTRMLST macro instruction, an invalid type attribute was specified for one of the operands.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the operand and reassemble. If the problem recurs, do the following before calling IBM for programming support.

- Have the associated program listing available

IHB116 AUTHORIZED SEQUENCE IS MISSING

Explanation: In a DFTRMLST macro instruction, either a control value or a user data area was specified without an authorized sequence having been specified.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Either specify an authorized sequence or eliminate the control value or user data area. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB117 PARENTHESIS IS MISSING.

Explanation: In a DFTRMLST macro instruction, the authorized sequence was not enclosed in parentheses.

System Action: The macro was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Enclose the authorized sequence in parentheses and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

I/O ERROR MESSAGE

This message is printed at the console of the central computer following an error that BTAM error recovery procedures have failed to correct.

IEA000I aaa,I/O ERR,tb,cccc,ddee,ffgghhhh

IEA000I

is the standard message code for the operator. The internal component name is IEA, the serial number is 000, and the action code is I (meaning information); immediate operator action is not required.

The following information is typed in hexadecimal (except I/O ERR):

aaa

is the address of the communication line on which the error occurred.

I/O ERR

is the message text, indicating the occurrence of an I/O error.

bb

is the command code of the failing command in the channel program. (See the DECCMCOB field in Appendix B for code values and meanings.)

cccc

is the status bytes of the channel status word (CSW) as specified in the Input/Output Block (IOB).

dd

is the first sense byte as specified in the IOB.

ee

is the sense information resulting from issuing diagnostic Write or Read commands if the commands resulted in a unit check (IBM 2701 only), or for BSC lines, is the operation type (DECTYPE field, byte 5) of the DECB associated with the failing READ/WRITE operation. (See the second byte of the DECTYPE field in Appendix B for code values and meanings).

ff

is the TP operation code of the failing command in the channel program. (See the DECTPCOB field in Appendix B for code values and meanings.)

gg

for BSC lines, is the flag byte (DECFLAGS, byte 24) of the DECB associated with the failing READ/WRITE operation. (See the DEC-FLAGS field in Appendix B for code values and meanings.) For non-BSC lines, this field (gg) is not used.

hhhh

is the first two polling or addressing characters of the terminal list. If only one polling character is used, it is left-justified in this field. (For IBM 2740 Model 2: When this message is issued for an addressing error, the first character (hh..) is the address of the terminal, and the second character (..hh) indicates the kind of error that occurred on the previous Write operation. The meanings of the codes are given in the IBM 2740 -- General Information section of the Start-Stop Read and Write Operations chapter. When the message is issued for a polling error, only one character, the polling character, appears at this point in the message. For remote 3270: only the control unit is identified.)

LINE ERROR RECORDING MESSAGES

These messages indicate the number of errors occurring for a given line.

Message IEC801I prints the contents of each of the four error threshold counters, indicating the number of data check, intervention required, or non-text time-out errors that have occurred since the last time the error threshold counters were reset. This message is printed whenever the threshold count has been reached for any of the three types of errors, or when the number of transmissions reaches the threshold

count. (See the LERB (Line Error Recording Block) macro instruction for further information.)

Message IEC802I prints the contents of each of the four cumulative counters (accumulators), indicating the total number of data check, intervention required, and nontext timeout errors, and number of transmissions that have been accumulated since the cumulative counters were last reset. This message is printed whenever the user program issues a LERPRT macro instruction.

```
-----  
IEC801I aaa THRESHOLD TRANS=bbb DC=ccc IR=ddd TO=eee  
-----
```

IEC801I
is the standard message code for the operator. The internal component name is IEC, the serial number is 801, and the action code is I, meaning information; immediate operator action is not required.

aaa
is the address of the communication line on which the error occurred (printed in hexadecimal).

THRESHOLD
is the message text.

TRANS=bbb
is the number of transmissions that have occurred on this line (in decimal).

DC=ccc
is the number of data check errors that have occurred on the line during the indicated number of transmissions (in decimal).

IR=ddd
is the number of intervention required errors that have occurred on the line during the indicated number of transmissions (in decimal).

TO=eee
is the number of nontext timeout errors that have occurred on the line during the indicated number of transmissions (in decimal).

```
-----  
IEC802I aaa LINE TOTALS TRANS=bbbbbbbb DC=cccc IR=dddd TO=eeee  
-----
```

IEC802I
is the standard message code for the operator. The internal component name is IEC, the serial number is 802, and the action code is I, meaning information; immediate operator action is not required.

aaa
is the address of the communications line on which the errors occurred.

LINE TOTALS
is the message text, indicating the total number of errors on the specified line.

TRANS=bbbbbbbb
is the total number of transmissions that have occurred on the line since the accumulators were reset (in decimal).

DC=cccc
is the total number of data check errors that have occurred on the line during the indicated number of transmissions (in decimal).

IR=dddd

is the total number of intervention required errors that have occurred on the line during the indicated number of transmissions (in decimal).

TO=eceee

is the total number of nontext time-out errors that have occurred on the line during the indicated number of transmissions (in decimal).

CONTROL UNIT OR DEVICE INOPERATIVE MESSAGE

This message is issued whenever an IBM 2701, 2702, or 2703 becomes inoperative. Usually, this message is printed during opening of a line group associated with the inoperative TCU. It appears when an I/O operation for some line connected to that TCU is attempted. When the message appears, the central computer operator should determine the reason for the condition and reactivate the TCU. For a local 3270, this message indicates that an I/O error was the result of the Erase Write done during Open processing.

```
-----  
IEC804A aaa {CONTROL UNIT} {CONT OR POST}  
            {DEVICE} NOT OPERATIONAL. REPLY {CONT, POST OR DROP}  
-----
```

IEC804A

is the standard message code for the operator. The internal component name is IEC, the serial number is 804, and the action code is A, meaning operator action is required.

aaa

is the address of the communications line or device.

CONTROL UNIT NOT OPERATIONAL. REPLY CONT OR POST

is the message text for a control unit, indicating the response for the operator. Reply either CONT or POST. If the operator replies CONT, the I/O operation for which this message was printed will be retried. If the retry is unsuccessful, the message will be reissued. If successful, the operation will continue. If the reply is POST, the operation will be posted complete-with-error and the "not operational SIO" bit (bit 0) will be turned on in the DECERRST field of the DECB for the line.

Note 1: For VSI the verbage may be abbreviated.

Note 2: If a reply is not entered before the requesting job is cancelled, the system may enter wait state.

DEVICE NOT OPERATIONAL. REPLY CONT, POST OR DROP

is the message text for a local 3270. If the operator replies CONT, the I/O operation for which this message was issued is retried. If the retry is unsuccessful, the message is reissued. The operation continues if the retry is successful. If the operator replies POST, the operation is posted complete-with-error. The DROP reply discontinues issuance of the message for the duration of open processing, and all I/O errors will be posted complete-with-error, for this line group.

```
-----  
IEC809I aaa CONTROL UNIT NOT OPERATIONAL  
-----
```

IEC809I

is the standard message code for the operator. The internal component name is IEC, the serial number is 809, and the action code is I, meaning no operator action is required.

aaa

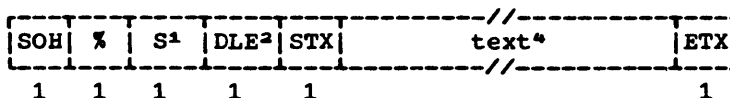
is the address of the communications line.

CONTROL UNIT NOT OPERATIONAL
is the message text.

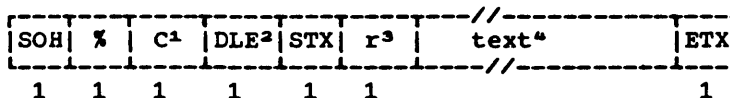
REMOTE BSC STATION ERROR MESSAGES

BTAM allows remote BSC stations to send error information to the central computer. Currently, the BSC stations that can send these messages are the 2715 (second and third formats only), the 2770 (first format only), and the remote 3270 (fourth format only). This information is routed to the user program, to the central computer console, or to an error file on a system residence device. The format of the error message depends on which of these destinations is desired:

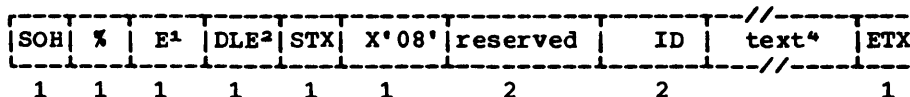
1. Error information to be sent to the user program must appear in a message having this format:



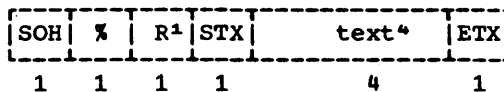
2. Error information to be sent to the central computer console must appear in a message having this format:



3. Error information to be sent to the system error file must appear in a message having this format:



4. Error information from a remote 3270 display system to be sent to the problem program and to be recorded as T-type records in SYS1. LOGREC must appear in a message having this format:



Notes:

- ¹This character must be uppercase (EBCDIC or USASCII).
- ²DLE need be present only for transparent text; however, in the third format (system error file), the text must begin in the eleventh byte.
- ³r is the routing code that specifies the console to which this message is to be routed.
- ⁴The text of each message depends on the format:
 - Format 1 - The text is user provided; its length depends on the size of the user's buffer.
 - Format 2 - The text must consist of printable characters. The length of the text must be either 17 or 60 characters; extra characters will be automatically truncated.

Format 3 - The text is bit significant. The length may be from 35 to 210 characters, but must be a multiple of 35 (i.e., it can be 35, 70, 105, 140, 175, or 210 characters). If necessary, the text should be padded with 'FF' bytes.

Format 4 - For the nonswitched remote 3270, the text includes the control unit and device addresses of the remote 3270 device from which the message was received and two sense/status bytes. For the switched 3275, the text includes only two sense/status bytes.

After sending the error message to the user program, console, or error file, BTAM restarts the user-program Read operation with which the error message was received. The remote station then may send another error message, a regular message, or EOT.

Upon receiving an error message with a Read Initial Operation on a multipoint line, BTAM preserves the Autopoll index byte in the first byte of the input area. Therefore, following each Read Initial operation on a multipoint line, the user program should check the second byte of the input area for an EOT character.

ERROR STATUS MESSAGES (IBM 2770)

One of the following five error messages may be sent by the 2770 terminal operator, as specified by the error recovery procedure for the 2770.

- Checkpoint Restart:
Last Restart Point: SOH % S STX 0 X₁ X₂ C SP SP
Specific Restart Point: SOH % S STX 2 X₁ X₂ C text...

(The first format causes BTAM to begin retransmission at the point from which the previous transmission began, or at any other point decided by the user program when it detects the 'C' preceding the two space characters. The second format allows the terminal operator to indicate to the user program where he wishes for retransmission to begin. The text can be up to 50 characters long and can contain any information the user program needs to identify the point at which transmission is to begin. This might be, for example, a page number or form number.

- Customer Engineer
Attention Required: SOH % S STX 0 X₁ X₂ D Z₁ Z₂
- Job Restart: SOH % S STX 0 X₁ X₂ M SP SP
- Format Error: SOH % S STX 0 X₁ X₂ F SP SP

In these formats:

X₁ is the station address

X₂ is the component address

text is any information the terminal operator wishes to send to identify to the user program the point from which retransmission is to begin.

Z₁ is the station address. This is the same as X₁ if the component requires attention by a Customer Engineer, but the station is operational; it is the address of an alternate station if the send-

ing station requires CE attention; and it is a SP character if no alternate station is available or desired.

Z₂ is the component address of an alternate component at the sending station or at an alternate station.

When BTAM recognizes an error status message, it posts the operation complete with a completion code of X'7F' and turns on bit 0 of DECFLAGS.

TERMINAL ERROR STATUS MESSAGE (IBM 2715)

This message, in one of four formats, provides the results of a scan of the error file of an IBM 2715 Transmission Control. The scan occurs when the error threshold for one of the area stations connected to the 2715 is exceeded (threshold value is eight) or when manually requested at the 2715, the 2740 attached to the 2715, or the central computer. BTAM prints the message on the master console, the teleprocessing console, or the system maintenance console, depending on the routing code included in the error scan message sent by the 2715. (The routing code does not appear in the message printed on the console.)

In the four formats below:

cuu is the address of the communications line (channel and unit) (EBCDIC).

xx is the address of the area station for which the error scan is reported (hexadecimal).

tttt is the time (0001-2400) the error scan occurred (decimal).

ww is the address of a particular adapter within the 2715 (hexadecimal).

Other fields in the message are indicated under individual formats below.

```
-----  
IEC815I cuu xx tttt yy ERS z  
-----
```

Explanation: This message reports the results of an error scan by the 2715 when five or more of the eight errors involved a particular one of the devices attached to the area station.

yy is the address of the device for which the errors occurred (hexadecimal).

z is the number of errors (from 5 to 8) that occurred for the device (decimal).

Operator Response: None.

```
-----  
IEC815I cuu xx tttt THRESHLD  
-----
```

Explanation: This message indicates that the threshold value of eight has been reached for the area station whose address is xx, but no one device attached to the station accounted for as many as five of the errors.

Operator Response: None

```
-----  
IEC815I ccc  xx tttt yy eeee zzzz yy eeee zzzz yy eeee zzzz yy eeee  
          zzzz  
-----
```

Explanation: This message is issued whenever an error scan for a particular area station is manually requested at the 2715, 2740 attached to 2715, or the central computer. The message appears twice in succession. Each indicates the nature of four errors; the two messages together provide this information for the eight most recent occurrences for area station xx.

yy is the address of a device (hexadecimal).

eeee is the error data for device yy (hexadecimal).

zzzz is the time (0001-2400) the error data was recorded on the 2715 disk (decimal).

Operator Response: None.

```
-----  
IEC815I cuu  ww tttt eeeeeee zzzz eeeeeee zzzz eeeeeee zzzz eeeeeee  
          zzzz  
-----
```

Explanation: This message is issued whenever an error scan for a particular 2715 adapter is manually requested at the 2715, 2740 attached to 2715, or the central computer. The message appears twice in succession. Each indicates the nature of four errors; the two messages together provide this information for the eight most recent error occurrences for adapter ww.

eeeeeee is the error data for adapter ww (hexadecimal).

zzzz is the time (0001-2400) the error data was recorded on the 2715 disk (decimal).

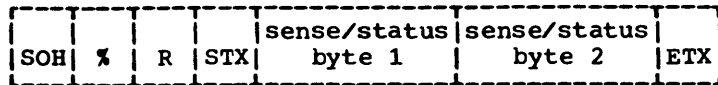
Operator Response: None.

ERROR STATUS MESSAGE (REMOTE IBM 3270)

An error status message from a nonswitched remote 3270 device has the format:

```
-----  
|SOH| % | R |STX| cu | device |sense/status|sense/status|  
|    |   |   |    |address|address|  byte 1  |  byte 2  |ETX|  
-----
```

An error status message from a switched 3275 display station has the format:



cu address

is the address of the control unit of the device from which the message was received (see Figure 47 in the section "IBM 3270 Display System -- Programming Considerations").

device address

is the address of the device from which the message was received (see Figure 49 in the section "IBM 3270 Display System -- Programming Considerations").

sense/status byte 1
has the format:

<u>Bit</u>	<u>Definition</u>
0	Setting depends on bits 2-7 (see Figure 50 in the section "IBM 3270 Display System -- Programming Considerations")
1	Setting always 1
2	Reserved
3	Reserved
4	Device Busy (DB)
5	Unit Specify (US)
6	Device End (DE)
7	Transmission Check (TC)

sense/status byte 2
has the format:

<u>Bit</u>	<u>Definition</u>
0	Setting depends on bits 2-7 (see Figure 50)
1	Setting always 1
2	Command Reject (CR)
3	Intervention Required (IR)
4	Equipment Check (EC)
5	Data Check (DC)
6	Control Check (CC)
7	Operations Check (OC)

For more information about the sense/status bytes, see IBM 3270 Information Display System, Component Description.

When BTAM recognizes an error status message from a remote 3270 device, the operation is posted with a completion code of X'7F', and bit 1 is turned off and bit 6 is turned on in the DECFLAGS field of the DECB. BTAM sends the error status message to the application program's input/output area and records the information as T-type records in SYS1.LOGREC.

Figure 114 indicates suggested actions according to the contents of the sense/status bytes in error status messages. (The figure applies both to nonswitched remote 3270 display systems and to the switched 3275.) The suggested actions are:

1. Execute a new address selection sequence, and retransmit the message starting with the command sequence that was being executed when the error occurred. If the operation is not successful after two retries, consider the error nonrecoverable, and take action 9.
2. Do the same as in action 1, except take action 10 after two retries.

3. Do the same as in action 1, except retransmit the entire failing chain of commands.
4. If possible, reconstruct the entire screen buffer image, and retry the failing chain of commands (within the BSC sequence of operations). If the screen buffer cannot or need not be reconstructed, retry the operation anyway. If the operation is not successful after three retries, consider the error nonrecoverable, and take action 9.
5. Do the same as in action 4, except reconstruct the buffer of the "from" device specified in the copy command. If the operation is not successful after three retries, consider the error nonrecoverable, and take action 10.
6. Wait for the display operator or system operator to ready the printer. Retry the printout by issuing a write command with the WCC and no data stream. Or take action 4.
7. Wait for the display operator or system operator to ready the "from" device specified in the copy command. Take action 2.
8. Examine the data stream to determine the cause of the nonrecoverable programming error. For command reject this may be caused by an EWA command issued to a 3271 or 3275, or by a WSF command issued to a 3271, 3275, on 3274 Model 1C (without WSF capability).
9. Request maintenance on the malfunctioning device. After repair, try to reconstruct the screen buffer image (using an erase/write command to correct a missing or multiple cursor condition in the buffer). Retry the failing chain of commands as in the previous action.
10. Request maintenance on the malfunctioning device (the "from" device specified in the copy command). After repair, try to reconstruct the screen buffer image (using an erase/write command to correct a missing or multiple cursor condition in the buffer). Retry the failing chain of commands as in the previous action.
11. If a new printout is required, take action 6. The error occurred during a printout and indicates either a character generator readout error or a print mechanism hang. There is no data error.
12. If a new printout is required, take action 4.
13. Periodically issue a specific poll (nonswitched) or read (switched) to obtain the Device End indication that is sent by the device to the TCU when the device goes not busy.
14. Periodically issue a specific poll (nonswitched) or read (switched) to obtain the Device End indication that is sent by the device to the TCU when the device goes not busy. Take action 4.
15. Do the same as in action 14, except take action 1 when the "from" device specified in the copy command goes not busy.
16. If the failing command is (1) a write command with a data stream or more than one byte or (2) one of a chain of commands that contains a previous write command without an SEA order immediately following the WCC, take action 4. Otherwise, take action 3. If the problem is still not corrected, take action 9.
17. An unauthorized attempt was made to copy data from a device. The device address in the error status message is the address of the "to" device specified in the copy command.

18. Do not take any action; this is not an error condition. This code signals that a previously detected busy device has gone not busy, a previously detected not ready device has gone ready, or a previously detected not available device is now available. The device end bit may be set alone or with other error bits.
19. This condition occurs if the operator of a formatted 3277 has performed a backtab or erase input in rapid succession. Ignore Device End and treat the condition as Device Busy only. Take action 13.

Error Condition			Suggested Action
Sense/Status Bytes	Bit(s) Set	Unit(s)	
X'4050'	IR	3271, 3275	6
X'4060' ¹	CR	3271, 3275	8
X'40C1'	OC	3271, 3275	8
X'40C2'	CC	3271	4
X'40C3'	CC, OC	3271	2
X'40C4'	DC	3271, 3275	4
X'40C6'	DC, OC	3271	2
X'40D1'	IR, OC	3271	7
X'4A40'	DB, DE	3271	19
X'4C40'	DB, US	3271, 3275	14
X'4E40'	DB, US, DE	3271, 3275	4
X'C140'	TC	3275	16
X'C240'	DE	3271, 3275	18
X'C250'	IR, DE	3271, 3275	6
X'C2C4'	*DC, DE	3271, 3275	4
X'C2C8'	EC, DE	3275	11
X'C2D8'	IR, EC, DE	3275	11
X'C4C1'	OC, US	3271	17
X'C4C4'	DC, US	3271, 3275	4
X'C4C5'	DC, OC, US	3271	5
X'C6C4'	DC, US, DE	3271, 3275	12
X'C6C8'	EC, US, DE	3271	11
X'C6D8'	TR, EC, US, DE	3271	11
X'C840'	DB	3271, 3275	13
X'C8C1'	DB, OC	3271	15

¹May be caused by the EWA command issued to a 3271 or 3275, or by a WSF command issued to a 3271, 3275, or 3274 Model 1C (without Write Structured Field capability).

Figure 114. Suggested Actions According to Remote 3270 Error Status Message

BTAM ABEND CODES

090 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found that a device other than a communications device was allocated to the data control block (DCB) being opened; that is, the device class code in the unit control block (UCB) for the device allocated to the data control block was not equal to hexadecimal 40.

Programmer Response: Either the UNIT parameter of the DD statement for the communications device is incorrect or unit control block generated during system generation is invalid. Check for improper specification of the UNIT parameter of the DD statement or the UNIT operand of the IODEVICE macro instruction. After correcting the error, execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

091 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported type of transmission control unit specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the IOCONTRL macro instruction used in generating the system. After correcting the error, execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement, and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

092 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported type of terminal control or terminal adapter specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the ADAPTER operand in the IODEVICE macro instruction used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement, and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

- 093 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.
- An Open routine found an invalid or unsupported type of terminal specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.
- Programmer Response: Check for improper specification of the UNIT operand in the IODEVICE macro instruction used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following before calling IBM for programming support:
- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
 - Have the associated job stream and program listing available.
- 094 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.
- An Open routine found an invalid or unsupported optional feature or mode of operation specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.
- Programmer Response: Check for improper specification of the FEATURE operand in the IODEVICE macro used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following before calling IBM for programming support:
- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
 - Have the associated job stream and program listing available.
- 095 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.
- An Open routine found that the lines allocated to the line group did not have identical types of terminals or lines, or that the terminals did not have the identical features.
- Programmer Response: Determine which line group contains different types of terminals or lines and redefine its lines through DD statements or a new system generation. If the problem recurs, do the following before calling IBM for programming support:
- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
 - Have the associated job stream and program listing available.
- 096 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.
- An Open routine found that dynamic buffer allocation had been specified in the DCBBFTEK field of the data control block (DCB). However, the Open routine could not dynamically allocate buffers because the data control block specified neither the address of a buffer pool control block (in the DCBBUFEB field) nor the number and length of the buffers (in the DCBBUFNO and DCBBUFL fields).
- Programmer Response: Correct the error by (1) providing a buffer pool and specifying the address of its control block in the

DCBBUFCB field, (2) specifying the number and length of the buffers in the DCBBUFNO and DCBBUFL fields, or (3) handling buffering in the user program and deleting the BFTEK=D operand in the DCB macro instruction or the DCB parameter of the DD statement. Then execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

097 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine required an additional entry in the device I/O directory; however, the directory was already full. Since the last system start, the maximum number of device types have been allocated. Normally, the maximum number is 16 (30 for VS2 Release 2).

Programmer Response: Do the following before calling IBM for programming support:

- * Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

098 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

Although Dual Communication Interface B or Dual Code Feature B was specified in the data control block (DCB), the transmission control unit is not an IBM 2701 or the unit control block (UCB) established at system generation time did not specify that the 2701 is equipped with the Dual Communication Interface or Dual Code feature.

Programmer Response: Probable user error. Correct the DCB macro that defined the data control block that erroneously specified the Dual Communication Interface B or Dual Code B, reassemble, and re-execute the job step. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

099 Explanation: The error occurred during execution of a BTAM OPEN macro instruction. An OPEN routine found that dynamic buffering was being initiated while the BTAM application was running in virtual mode. Dynamic buffering is not supported for virtual mode in VS2 Release 2.

Programmer Response: Specify ADDRSPC=REAL in the JCL JOB or EXEC statement and rerun the job.

APPENDIX D: SYSTEM GENERATION MACRO INSTRUCTIONS

This appendix explains the operands that must be included in the system generation macro instructions when generating an operating system that includes BTAM. Only those macro instructions and operands directly related to BTAM are given; for other macro instructions required for generating a system and an explanation of the generation process, see the OS/VS1 or OS/VS2 system generation reference manual. In particular, refer to explanations of the DATAMGT, and IODEVICE macro instructions.

DATAMGT Macro Instruction

DATAMGT causes one or more optional access methods to be included in your operating system. To include BTAM, code:

Name	Operation	Operand
	DATAMGT	ACSMETH=BTAM

IODEVICE Macro Instruction

IODEVICE describes to the operating system the characteristics of an input/output device and its operating system requirements. For BTAM, IODEVICE identifies the type of device, that is, remote station, that is connected to a communications line, or the type of line configuration and the type of transmission control unit (TCU). You therefore code one IODEVICE macro for each line, regardless of how many remote stations are connected to the line. For BTAM support for the local 3270 display system, IODEVICE identifies the type of local 3270 device connected to a 3272 control unit. You code one IODEVICE macro for each local 3270 device.

To use BTAM Prepare to Read support, an I/O generation must include local 3270 devices with Prepare to Read capability. Prepare to Read capability is indicated by specifying PTREAD on the FEATURE operand. FEATURE=PTREAD applies only to displays and printers attached to 3274 Model 1D Control Units.

CAUTION: Even though an access method has been previously specified in a complete system generation, device-dependent code cannot be added to access method support merely by doing an I/O generation for the

device. A complete system generation is required if BTAM device support for the local 3270s being added to the system is desired.

Only the operands shown are applicable for a BTAM system.

Name	Operation	Operand
[symbol]	IODEVICE	UNIT=type, ADDRESS=address, TCU=type, MODEL=model, ADAPTER=type [,FEATURE=(feature ₁ , feature ₂ ,...)] [,SETADDR=type] [,OBRCNT=n]

UNIT

specifies the type of remote terminal (start-stop) or type of line configuration (BSC) associated with the line address given by the ADDRESS operand. For the local 3270 display system, specifies the type of local device with the device address given by the ADDRESS operand. Valid UNIT parameters are:

- For start-stop:
 - 1030
 - 1050
 - 1060
 - 2260
 - 2740
 - 2741P (2741 using PTTC code)
 - 2741C (2741 using Correspondence code)
 - 83B3
 - 115A
 - TWX
 - WTTA (World Trade Telegraph Terminals)
- For BSC:
 - BSC1 (for nonswitched point-to-point line)
 - BSC2 (for switched point-to-point line)
 - BSC3 (for nonswitched multipoint line)

- For local 3270 display system:

3277
3284
3286

ADDRESS

specifies the three-digit address of the line over which the type of station given by UNIT is to communicate. For the local 3270 display system, specifies the three-digit address of the local 3270 device given by the UNIT operand. Valid parameters are within the range 000-6FF, inclusive (hexadecimal).

TCU

specifies the type of transmission control unit: 2701, 2702, or 2703.

Notes: If an IBM 3705 Communications Controller with an Emulation Program is attached, the control unit definition in the IODEVICE macro instruction remains TCU=2701, TCU=2702, or TCU=2703, as appropriate. Certain features of the 2701, 2702, and 2703 are not supported by the 3705 Emulation Program; these features are listed in IBM 3705 Communications Controller Emulation Program Generation and Utilities, Guide and Reference Manual, GC30-3002.

MODEL

specifies whether the local 3270 device given by the UNIT operand is a model 1 (480-character buffer) or model 2 (1920-character buffer). Code MODEL=1 or MODEL=2.

Note: The MODEL operand applies only to the local 3270 display system. Only model 1 devices (3277, 3284, 3286) may be connected to a model 1 3272 control unit. Model 1 or model 2 devices or both may be connected to a model 2 3272 control unit.

ADAPTER

specifies the type of TCU terminal control and terminal adapter associated with the line address given by the ADDRESS operand. Code one of the following values:

IBM1

for IBM 1050, 1060, 2740 or 2741 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type I, and either: (1) an appropriate data set, or (2) an IBM Line Adapter.

- IBM 2702 or 2703 through an IBM Terminal Control, Type I, and either: (1) a Data Set Line Adapter and an appropriate data set, or (2) an IBM Line Adapter.

IBM2

for IBM 1030 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type II, and either: (1) an appropriate data set, or (2) an IBM Line Adapter.
- IBM 2702 or 2703 through an IBM Terminal Control, Type II, and either: (1) a Data Set Line Adapter and an appropriate data set, or (2) an IBM Line Adapter.

IBM3

for IBM 2260/2848 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type III and an appropriate data set.

IBMT

for IBM 1050 communicating with:

- IBM 2701 through an IBM Telegraph Adapter.
- IBM 2703 through an IBM Terminal Control, Type I, and a Telegraph Line Adapter.

TELE1

for AT&T 83B3 or Western Union 115A communicating with:

- IBM 2701 through a Telegraph Adapter, Type I.
- IBM 2702 or 2703 through a Telegraph Terminal Control, Type I, and a Telegraph Line Adapter.

TELE2

for WU TWX (Model 33 or 35) communicating with:

- IBM 2701 through a Telegraph Adapter, Type II, and an appropriate data set.
- IBM 2702 or 2703 through a Telegraph Terminal Control, Type II, and a Data Set Line Adapter and an appropriate data set.

TELEW

for World Trade Telegraph terminal communicating with:

- IBM 2701 through a World Trade Telegraph Adapter.
- IBM 2702 or 2703 through a World Trade Telegraph Adapter and a Telegraph Line Adapter.

BSCA

for IBM System/370, System/3, 1130, 1800, 2715, 2770, 2780, or 2972 communicating with:

- IBM 2701 through a Synchronous Data Adapter, Type II, and an appropriate data set.
- IBM 2703 through a Synchronous Terminal Control and an appropriate data set.

This Page Intentionally Left Blank

FEATURE

specifies certain optional features with which the transmission control unit (TCU) or remote station is equipped. For the local 3270 display system, specifies certain features with which the local 3270 display station is equipped. Code each of the applicable parameters:

AUTOCALL

if the TCU (2701, 2702, or 2703) to which the remote station is connected is equipped with the Autocall feature and the line is connected to the TCU terminal adapter by means of an Automatic Calling Unit and an appropriate data set. When these conditions are met, and you specify AUTOCALL, the channel programs generated for the line whose address is specified by the ADDRESS operand can automatically dial the remote stations, using the telephone number you specify in the terminal list associated with the line.

AUTOANSR

if the data set (modem) connecting the access line specified by the address operand to the TCU is a switched line over which calls are to be answered. AUTOANSR must be coded regardless of whether the line is equipped with an automatic answering unit. When you specify AUTOANSR, the channel programs generated for the specified line will automatically initiate message transmission when a remote station calls the computer.

AUTOPOLL

if the automatic polling facility of the TCU is to be used. This facility is a standard feature of the IBM 2703, and an optional feature (called Auto Poll) of the IBM 2702. For the IBM 2701, this parameter is valid only for lines connected through the Synchronous Data Adapter Type II. If you specify AUTOPOLL, the Read Initial channel programs generated for the specified line will be so arranged that a negative response from a remote station causes the TCU to automatically poll the next station in the terminal list without signalling an I/O interrupt. If you omit AUTOPOLL, Read Initial operations will employ programmed polling with each negative response from a remote station causing an I/O interrupt. Only those Read Initial operations that send polling characters are affected.

This parameter is valid only for non-switched multipoint lines to which are connected the IBM 1030, 1060, 1050, 2740, or any BSC stations, (as specified by the UNIT operand), as only

these types of stations can be polled using the Auto Poll facility. The AUTOPOLL operand must be coded for BSC stations on multipoint lines, and may be coded for the foregoing start-stop terminals.

If UNIT=2740 is specified, you must also code in the FEATURE operand, either SCONTROL or SCONTROL and CHECKING (in addition to the AUTOPOLL parameter).

DUALCOMM

if the TCU (IBM 2701 only) to which the line specified by ADDRESS is connected is equipped with the Dual Communication Interface special feature. This feature allows program selection (in the DCB macro) of either of two data sets (modems) over which transmission is to occur (BSC lines only).

DUALCODE

if the TCU (IBM 2701 only) is equipped with the Dual Code special feature. This feature allows program selection (in the DCB macro) of the transmission code to be used on the communication line (BSC lines only).

For IBM 2740 Terminals only:

CHECKING

if UNIT=2740 is specified and the terminal is equipped with the Record Checking special feature.

SCONTROL

if UNIT=2740 is specified and the terminal is equipped with the Station Control special feature. This parameter and the AUTOCALL, AUTOANSR, OIU, and XCONTROL parameters are mutually exclusive.

XCONTROL

if UNIT=2740 is specified and the terminal is equipped with the Transmit Control special feature and the Dial Up special feature. You also must indicate the Dial-up special feature in the FEATURE operand by the AUTOCALL or AUTOANSR parameter, or both, as appropriate. This parameter and the OIU parameter are mutually exclusive.

OIU

if the UNIT=2740 is specified the terminal is equipped with an IBM 2760 Optical Image Unit. This parameter and the SCONTROL and XCONTROL parameters are mutually exclusive.

For Local 3270 Devices Only:

- One of the following character generator options:

DOCHAR
if the device has a domestic monospace character generator. If the FEATURE operand is not coded, this value is assumed.

ASCACHAR
if the display station has an ASCII A monospace character generator.

ASCBCHAR
if the display station has an ASCII B monospace character generator.

FRCHAR
if the device has a French monospace character generator.

GRCHAR
if the device has a German monospace character generator.

KACHAR
if the device has a Katakana monospace character generator.

UKCHAR
if the device has a United Kingdom monospace character generator.

- One of the following keyboard options (if a keyboard is present):

EBKY3277
if the display station has an EBCDIC typewriter keyboard.

ASKY3277
if the display station has an ASCII typewriter keyboard.

DEKY3277
if the display station has a data entry keyboard.

OCKY3277
if the display station has an operator console keyboard.

- One of the following keyboard options (if a keyboard is present):

KB66KEY
if the display station has a 66-key keyboard (that is, has no program function keys).

KB78KEY
if the display station has a 78-key keyboard (that is, has program function keys).

KB70KEY
if the display station has a Katakana character generator and a 70-key data entry keyboard.

KB81KEY
if the display station has a Katakana character generator and an 81-key EBCDIC typewriter keyboard.

- As many of the following options as required:

SELPEN
if the display station has a selector pen.

NUMLOCK
if the display station has the numeric lock feature.

AUDALRM
if the display station has a keyboard and an audible alarm.

MAGCDRD
if the display station has a magnetic card reader adapter.

PTREAD
if the device is attached to a 3274 Model 1D Control Unit. This operand applies to OS/VS1 and OS/VS2 MVS operating systems; it is not applicable to displays and printers attached to 3272 and 3274 Model 1B Control Units.

SETADDR
specifies which of the four Set Address (SAD) commands is to be issued to the transmission control unit (IBM 2702 only) for operations on the line specified by the ADDRESS operand. The SAD command selects the appropriate line speed for the type of terminal connected to the line. The association between the specific command (Sadzer, Sadone, Sadtwo, or Sadthree) and the corresponding line speed is established by internal connections within the 2702; this is done by the customer engineer when the 2702 is installed. You must code this operand in the TCU to which the line is connected is a 2702; if it is a 2703, the SAD commands will be ignored. Do not code this operand if the TCU is a 2701, as a command reject will be signalled when the line group is opened.

Code:	<u>If the SAD command for the line is:</u>
0	Sadzer
1	Sadone
2	Sadtwo
3	Sadthree

**OBRCNT (For IBM 2715 Transmission Control
Unit only)**

specifies the number of area stations
connected to the 2715s on the line
represented by the IODEVICE macro.
(This value is used to compute the
space required on SYS1.LOGREC for
error data received from the 2715.)

This Page Intentionally Left Blank

APPENDIX E: CODE CHARTS FOR BSC AND THE LOCAL 3270 DISPLAY SYSTEM

This appendix includes code charts for EBCDIC, USASCII, and six-bit TRANSCODE.

Six Bit Transcode

Code Positions →	0	1	0	1	0	1
	0	0	0	1	1	0
	1	0	1	0	1	1
2 3 4 5						
0 0 0 0	SOH 12-9-1	& 12	- 11	0 0		
0 0 0 1	A 12-1	J 11-1	/ 0-1	1 1		
0 0 1 0	B 12-2	K 11-2	S 0-2	2 2		
0 0 1 1	C 12-3	L 11-3	T 0-3	3 3		
0 1 0 0	D 12-4	M 11-4	U 0-4	4 4		
0 1 0 1	E 12-5	N 11-5	V 0-5	5 5		
0 1 1 0	F 12-6	O 11-6	W 0-6	6 6		
0 1 1 1	G 12-7	P 11-7	X 0-7	7 7		
1 0 0 0	H 12-8	Q 11-8	Y 0-8	8 8		
1 0 0 1	I 12-9	R 11-9	Z 0-9	9 9		
1 0 1 0	STX 12-9-2	SPACE No Punch	ESC 0-9-7	SYN 9-2		
1 0 1 1	. 12-8-3	\$ 11-8-3	, 0-8-3	# 8-3		
1 1 0 0	x 12-8-4	* 11-8-4	% 0-8-4	@ 8-4		
1 1 0 1	BEL 9-7	US 11-9-8-7	ENQ 0-9-8-5	NAK 9-8-5		
1 1 1 0	SUB 9-8-7	EOT 0-9-8-7	ETX 12-9-3	EM 11-9-8-1		
1 1 1 1	ETB 0-9-6	DLE 12-11-9-8-1	HT 12-9-5	DEL 12-9-7		

Standard Representation of USASCII

Rows	Columns	0	1	2	3	4	5	6	7
	b ₇ b ₆ b ₅	000	001	010	011	100	101	110	111
	b ₄ b ₃ b ₂ b ₁								
0	0 0 0 0	NUL	DLE	SP	0	@	P	`	p
1	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
2	0 0 1 0	STX	DC2	"	2	B	R	b	r
3	0 0 1 1	ETX	DC3	#	3	C	S	c	s
4	0 1 0 0	EOT	DC4	\$	4	D	T	d	t
5	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
6	0 1 1 0	ACK	SYN	&	6	F	V	f	v
7	0 1 1 1	BEL	ETB	'	7	G	W	g	w
8	1 0 0 0	BS	CAN	(8	H	X	h	x
9	1 0 0 1	HT	EM)	9	I	Y	i	y
10	1 0 1 0	LF	SUB	*	:	J	Z	j	z
11	1 0 1 1	VT	ESC	+	;	K	[k	{
12	1 1 0 0	FF	FS	,	<	L	\	l	
13	1 1 0 1	CR	GS	-	=	M]	m	}
14	1 1 1 0	SO	RS	.	>	N	^	n	~
15	1 1 1 1	SI	US	/	?	O	_	o	DEL

Data Link Control Functions

Function	Characters Used In:		
	EBCDIC	USASCII	TRANSCODE
ACK-0	DLE, X'70'	DLE, 0	DLE, - (hyphen)
ACK-1	DLE, X'61'	DLE, 1	DLE, T
WACK	DLE, X'6B'	DLE, ;	DLE, Z
RVI	DLE, X'7C'	DLE, <	DLE, 2

EBCDIC

Bit Positions 0 and 1 →		00				01				
Bit Positions 2 and 3 →		00	01	10	11	00	01	10	11	
Bit Positions 4, 5, 6, and 7	0000	①	②	③	④	⑤	⑥	⑦	⑧	
	0001	NUL	DLE	DS		SP	&	-	⑬	1
	0010	SOH	DC1	SOS						2
	0011	STX	DC2	FS	SYN					3
	0100	ETX	DC3							4
	0101	PF	RES	BYP	PN					5
	0110	HT	NL	LF	RS					6
	0111	LC	BS	EOB	UC					7
	1000	DEL	IL	PRE	EOT					8
		CAN								
	9	9	9	9	9	9	9	9		
	12	11	0		12	12	11	12		
					0	11	0	0		
Zone Punches										

Bit Positions 0 and 1 →		10				11				
Bit Positions 2 and 3 →		00	01	10	11	00	01	10	11	
Bit Positions 4, 5, 6 and 7	0000					⑨	⑩	⑪	⑫	8-1
	0001	a	j			A	J	⑭	1	1
	0010	b	k	s		B	K	S	2	2
	0011	c	l	t		C	L	T	3	3
	0100	d	m	u		D	M	U	4	4
	0101	e	n	v		E	N	V	5	5
	0110	f	o	w		F	O	W	6	6
	0111	g	p	x		G	P	X	7	7
	1000	h	q	y		H	Q	Y	8	8
	1001	i	r	z		I	R	Z	9	9
	12	12	12	12	12	11				
	0	11	0	0	0	0				
Zone Punches										

Bit Positions 0 and 1 →		00				01				
Bit Positions 2 and 3 →		00	01	10	11	00	01	10	11	
Bit Positions 4, 5, 6, and 7	1001		EM							8-1
	1010	SMM	CC	SM		ç	l	⑮	:	8-2
	1011	VT	CU1	CU2	CU3	.	\$,	#	8-3
	1100	FF	IFS		DC4	<	*	%	@	8-4
	1101	CR	IGS	ENQ	NAK	()	_	'	8-5
	1110	SO	IRS	ACK		+	;	>	=	8-6
	1111	SI	IUS	BEL	SUB	!	~	?	"	8-7
	9	9	9	9	12	11				
	12	11	0		0	0				
Zone Punches										

Bit Positions 0 and 1 →		10				11				
Bit Positions 2 and 3 →		00	01	10	11	00	01	10	11	
Bit Positions 4, 5, 6, and 7	1010									8-2
	1011									8-3
	1100									8-4
	1101									8-5
	1110									8-6
	1111									8-7
	12	12	12	12	9	9	9	9		
	0	11	0	0	0	11	0	0		
Zone Punches										

- | | | | | |
|---------------|----------------|-----------|---------|------------|
| ① 12-0-9-8-1 | ④ 12-11-0-9-8- | ⑦ 11 | ⑩ 11-0 | ⑬ 0-1 |
| ② 12-11-9-8-1 | ⑤ No Punches | ⑧ 12-11-0 | ⑪ 0-8-2 | ⑭ 11-0-9-1 |
| ③ 11-0-9-8-1 | ⑥ 12 | ⑨ 12-0 | ⑫ 0 | ⑮ 12-11 |

USASCII	EBCDIC Equivalents
RS	IRS
ETB	EOB
ESC	PRE
FS	IFS

APPENDIX F: WORLD TRADE TELEGRAPH TERMINALS: TRANSLATION TABLE MODIFICATIONS

Because the International Telegraph Alphabet No. 2 and the Figure Protected Code ZSC3 (See Figure 115) vary from country to country, the BTAM-supplied translation tables RCTW, RCT3, SCTW, and SCT3 may not fit a given installation. Therefore, four macro instructions, TRSLRCTW, TRSLRCT3, TRSLSCTW, and TRSLSCT3 are provided to modify these tables to produce new tables for use with the TRNSLATE macro instruction. These macros both modify the tables and cause them to be assembled into the user program, so it is not necessary to use the ASMTRTAB macro instruction.

TRSLRCTW and TRSLRCT3 Macro Instructions

Name	Operation	Operand
symbol	{ TRSLRCTW } { TRSLRCT3 }	Fx=code, ...

symbol

is the name to be given to the modified table (that is, the name that

will be specified in the TRNSLATE macro instruction). If symbol is omitted, the original name, IECTRCTW or IECTRCT3, is the name of the modified table.

TRSLRCTW

specifies that table RCTW is to be modified and assembled.

TRSLRCT3

specifies that table RCT3 is to be modified and assembled.

Fx=code

specifies what modification is to be made. F stands for figures shift, x represents the number of the code combination to be translated. The permissible values of x are:

For TRSLRCTW: 1, 28 3, 6, 7, 8, 10 through 14, 19, 22, 24, 26, and 32.

For TRSLRCT3: 1, 5, 8, 9, 11, 12, 14, 15, 17 through 20, 22, 24, 26, and 32.

Example: If the transmission code used by a WT terminal is the International Telegraph Alphabet No. 2, combination 6 in figures shift, representing the % character, does not exist in table RCTW. Therefore, you would modify table RCTW by coding

TRSLRCTW F6=6C

where 6C is the hexadecimal representation of the % character in EBCDIC.

TRSLSCTW and TRSLSCT3 Macro Instructions

Name	Operation	Operand
symbol	TRSLSCTW TRSLSCT3	Xyy=Fx, ...

symbol

is the name to be given to the modified table (that is, the table name that will be specified in the TRNSLATE macro instruction). If symbol is omitted, the original name, IECTSCTW or IECTSCT3, is the name of the modified table.

TRSLSCTW

specifies that table SCTW is to be modified and assembled.

Code Combination No.	Elements 12 345	Lettershift		Figureshift		
		Hex Code	Character	Hex Code	Character	
					ITA2	ZSC3
1	11 000	18	A	38	-	+
2	10 011	13	B	33	?	6
3	01 110	0E	C	2E	:	8
4	10 010	12	D	32	Who are you	
5	10 000	10	E	30	3	-
6	10 110	16	F	36	N/A	4
7	01 011	08	G	28	N/A	0
8	00 101	05	H	28	N/A	?
9	01 100	0C	I	2C	8	Bell
10	11 010	1A	J	3A	Bell	2
11	11 110	1E	K	3E	((
12	01 001	09	L	29))
13	00 111	07	M	27	.	7
14	00 110	06	N	26	,	:
15	00 011	03	O	23	9	:
16	01 101	0D	P	2D	0	9
17	11 101	1D	Q	3D	1	N/A
18	01 010	0A	R	2A	4	/
19	10 100	14	S	34	'	'
20	00 001	01	T	21	5	.
21	11 100	1C	U	3C	7	1
22	01 111	0F	V	2F	=	=
23	11 001	19	W	39	2	3
24	10 111	17	X	37	/	N/A
25	10 101	15	Y	35	6	5
26	10 001	11	Z	31	-	N/A
27	00 010	02		22	CR	CR
28	01 000	08		28	LF	LF
29	11 111	1F		3F	LTRS	LTRS
30	11 011	18		38	FIGS	FIGS
31	00 100	04		24	Space	Space
32	00 000	00		20	N/A	N/A

Note: N/A = Not assigned
CR = Carriage return
LF = Line feed
LTRS = Letters shift
FIGS = Figures shift

Figure 115. World Trade Telegraph Codes ITA2 and ZSC3

TRSLSCT3

specifies that table SCT3 is to be modified and assembled.

Xyy=Fx

specifies what modification is to be made. yy is the hexadecimal representation, in EBCDIC, of the character to be translated. x is the number of the code combination for the character to which yy is to be translated. (F stands for figures shift.) The permissible values of yy are : 2A, 3F, 4A through 50, 5A through 61, 6A through 6F, and 7A through 7F.

Example: If the transmission code used by a WT terminal is the ITA No. 2, and if you wish to translate an EBCDIC % character (hexadecimal 6C in EBCDIC) to an ITA No. 2 % character (combination 6 in figures shift), you would code:

TRSLSCTW X6C=F6

Similarly, if you wish to translate an EBCDIC * character (hexadecimal 5C in EBCDIC) to a % character, you would code:

TRSLSCTW X5C=F6

And if you wish both the % and * characters to be translated to % characters, you would code:

TRSLSCTW X6C=F6,X5C=F6

Note: You can code the same macro several times, each with a different name, to create as many translation tables as needed. This permits several terminals using the same transmission code, but varying character arrangements to operate in the same installation.

Each operand of each BTAM macro instruction can be coded in one or more ways, as indicated in the table to the right. Listed below are the meanings of each of the column headings in the table. The same information is given for each of the user-table-defining macro instructions for the IBM 2715; see the table following the BTAM Macro Instructions table.

Abbreviations Used in Macro Instruction Tables

<u>Abbreviation</u>	<u>Meaning</u>
	You may code the operand as:
Sym	Any symbol valid in the assembler language.
Dec Dig	Decimal digits, within the range shown in the macro instruction description. (The sequence of digits is assembled as a single integer, not as individual digits.)*
Register	Register notation; that is, a number of a general register, enclosed in parentheses. You must previously have loaded the specified register with the value or address indicated in the operand description. The value or address must be right-adjusted in the register, with all high-order bits set to zero. You may specify registers 2-12 symbolically (CTREG5), or with an absolute expression (5). Registers 0 and 1 can only be specified absolutely: (0), (1).
Char	Any character self-defining term, coded without the framing characters, C' '.
Dec Char	Concatenated decimal digits (each digit is individually assembled in binary format).*
Hex Char	Concatenated hexadecimal digits, coded without the framing characters, X' '.
Code	One of the coded values as given in the individual macro instruction description.
RX-type	Any address that is valid in an RX-type instruction (e.g., LA).
Rel Exp	A relocatable expression (acceptable as an A-type or V-type address constant by the assembler).
Abs Exp	Any absolute expression as defined by the assembler: self-defining terms (decimal, hexadecimal, binary, character), length attributes, absolute symbols, paired relocatable terms in the same control section (CSECT), and arithmetic combinations of absolute terms.

 *The distinction between Dec Dig and Dec Char may be illustrated by two examples:

19 coded where Dec Dig is specified is assembled as binary 1 0011;
 19 coded where Dec Char is specified is assembled as binary 0000 0001 0000 1001.

267 coded where Dec Dig is specified is assembled as binary 1 0000 1011; 267 coded where Dec Char is specified is assembled as binary 0000 0010 0110 0000 0111.

BTAM Macro Instructions (1 of 6)

Macro Instruction	Operand	Sym	Register			RX- type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
			Dec Dig	(2-12)	(1)							
ASMTRTAB	tablename											X
CHGNTRY	listaddr**		X				X					
	dcbaddr***		X				X					
	listtype											X
	listposition		X					X				
	numchars**		X					X				
	action											
CLOSE	dcb						X					
	MF=											X
	listname		X	X	X							
DCB	DSORG=											X
	MACRF=											X
	DDNAME=	X										
	BUFNO=							X				
	BUFL=							X				
	BUFCB=						X					
	EXLST=						X					
	BFTEK=											X
	LERB=						X					
	EROPT=											X
	DEVD=											X
	MODE=											X
	CODE=											X
	READYQ=***							X				X
DFTRMLST	listtype											X
	xx										X	
	yy										X	
	dialcount		X									
	dialchars								X			
	numsent		X									

* see macro description for allowable values
 ** does not apply to local 3270 display system
 *** applies only to local 3270 display system

BTAM Macro Instructions (2 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*	
				(2-12)	(1)	(0)								
DFTRMLST (Cont'd)	sentchar											X		
	numcnsent		X											
	cntrlseq											X		
	tidseq											X		
	numrec		X											
	ridseq											X		
	AN													**
	MD													**
	AD													**
	entrylength		X											
	userlength		X											
	idcount		X											
	idsent												X	
	authsequence												X	
	controlvalue		X											
	userdata								X					
LERB	nlines								X					
	transmct									X				
	datack									X				
	intreq									X				
	notto									X				
LERPRT	dcbaddr	X		X	X									
	rln			X		X				X				
	cid			X				X						
	CLEAR=												X	
LOPEN	dcbaddr	X		X										
ONLST	DECB=			X	X		X							
	X=			X					X					
	Y=			X					X					
* see macro description for allowable values														
**as shown														

BTAM Macro Instructions (3 of 6)

Macro Instruction	Operand	Syn	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
	DCB=			X			X						
	AREA=			X			X						
	TEXT=			X			X						
	LENGTH=			X					X				
	ENTRY=			X			X						
	RLN=			X					X				
OPEN	dcb							X					
	MF=												X
	listname			X	X		X						
READ (list form, MF=L)	decbaddr	X											
	optype												X
	dcbaddr							X					
	inoutarea							X					
	inarea							X					
	outarea							X					
	inoutlength								X				
	inlength								X				
	outlength								X				
	entry							X					
	rln								X				
	MF=L												**
READ (Execute form, MF=E)	decbaddr			X	X		X						
	optype												X
	dcbaddr			X			X						
	inoutarea			X			X						'S'
	inarea			X			X						'S'
	outarea			X			X						
	inoutlength			X					X				'S'
	inlength			X					X				'S'
* see macro description for allowable value													
** as shown													

BTAM Macro Instructions (4 of 6)

Macro Instruction	Operand	Sym	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
			Dec Dig	(2-12)	(1)							
READ (Cont'd)	outlength			X				X				
	entry			X		X						'S'
	rln			X				X				
	MF=E											**
READ (Standard form)	decbaddr	X										
	optype											X
	dcbaddr			X				X				
	inoutarea			X				X				'S'
	inarea			X				X				'S'
	outarea			X				X				
	inoutlength			X				X				'S'
	inlength			X				X				'S'
	outlength			X				X				
	entry			X				X				
	rln			X				X				
RELBUF	dcbaddr			X	X		X					
	bufferaddr			X								
REQBUF	dcbaddr			X	X		X					
	returnreg			X								
	count			X		X		X				
RESETPL	decbaddr			X	X			X				
	POLLING											**
	ANSRING											**
	ATTENT											**
	HIO2740											**
TRANSLATE	dcbaddr			X			X					
	tablename			X			X					
	area			X			X					
	length			X		X		X				'S'
* see macro description for allowable values												
** as shown												

BTAM Macro Instructions (5 of 6)

Macro Instruction	Operand	Sym	Register			RX- type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
			Dec Dig	(2-12)	(1)							
TRSLRCTW	Fnn=											X
TRSLRCT3	Fnn=											X
TRSLSCTW	Xyy=											X
TRSLSCT3	Xyy=											X
TWAIT	returnreg			X								
	ECBLIST=			X		X						
WAIT	count			X		X		X				
	ECB=			X	X	X						
	ECBLIST=			X	X							
WRITE (List form, MF=L)	decbaddr	X										
	optype											X
	dcbaddr							X				
	inoutarea							X				
	inarea							X				
	outarea							X				
	inoutlength								X			
	inlength								X			
	outlength								X			
	entry								X			
	rln									X		
	MF=L											**
WRITE (Execute form, MF=E)	decbaddr			X	X		X					
	optype											X
	dcbaddr			X			X					
	inoutarea			X			X					
	inarea			X			X					'S'
	outarea			X			X					
	inoutlength			X					X			'S'
inlength			X					X			'S'	

* see macro description for allowable value
 ** as shown

BTAM Macro Instructions (6 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
WRITE (Cont'd)	outlength			X					X				
	entry			X			X						
	rln			X					X				
	MF=E												**
WRITE (Standard form)	decbaddr	X											
	optype												X
	dcbaddr	X						X					
	inoutarea	X						X					
	inarea	X						X					'S'
	outarea	X						X					
	inoutlength	X							X				'S'
	inlength	X							X				'S'
	outlength	X							X				
	entry	X						X					
	rln	X								X			
* see macro description for allowable value													
** as shown													

2715 User-Table Macro Instructions (Part 1 of 4)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
AS	ID=							X					
	ASGROUP=	X											
	DEGROUP= tgroupname deunumber	X						X					
ASCTR	ID=							X					
	HIGHCTR=							X					
	ROUTE=												X
	LOG												**
	AS LOG												**
	EXTALRM												**
	NEXTAS							X					
ASLIST	device												X
	NORM=							X					
	LENGTH= data length gdlight2							X					
	DIGIT= entrypos compvalue gdlight3							X					
	ENTRY=												X
	MSG=								X				
	INQDISP=							X					
	MODULUS= entrypos data length gdlight4							X					
	SELTRAN=												X
	CONFIGUR	COFE=											
PC=													X
GDU=													X
FUNCERR=								X					
ENDERR=								X					

* see macro description for allowable value
 ** as shown

2715 User-Table Macro Instructions (Part 2 of 4)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
CONFIGUR (cont.)	MONERR=							X					
	GETID=							X					
	STORID=							X					
	IDCOUNT=							X					
	INQDISP=												X
CTRGROUP	ctrno							X					
	sro							X					
	ctest							X					
	ID=							X					
	SROENAB=												X
	CTINIT=												X
CTRLIST	DEVCOD=												X
	CTRADR=												X
	CTRRD=												X
	CTTEST=												X
	CTROP=												X
	MSG=									X			
CTRSCHED	sched							X					
DEULIST	LENGTH=		X										
	DIGIT= entrypos compvalue							X X					
	MSG=									X			
	MODULUS= entrypos data length							X X					
	DIGIT2= entrypos compvalue							X X					
DISPGUID	DISPMSG=									X			
	SUPPRES=												X
GDUAS	ID=							X					
	GDUNUMB=							X					
* see macro description for allowable values													

2715 User-Table Macro Instructions (Part 3 of 4)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX- type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
GDULIST	PARAMNO=							X					
	NORGUID=							X					
	DISPMMSG=	X											
	IDENT=							X					
	MSG=									X			
	ENTRY=												X
GDUTRANS	TRCODE=							X					
	TRLIST=	X											
PARAMNUM	PLN=							X					
	PARMLST=	X											
PARMLIST	CKLNTH= data length gdlight							X X					
	CKMONKY=												X
	CKMOD11= data length entrypos gdlight							X X X					
	CKRANGE= firstpos lastpos compvalue							X X X					
	LOWGUID=							X					
	HIGUID=							X					
	RNGETST=												X
	CKMOD10= data length entrypos gdlight							X X X					
	CKOR= data pos checkchar							X				X	
	ORGJID=							X					
	CKAND= startpos endpos checkchar							X X				X	
	ANDGUID=							X					

* see macro description for allowable values

2715 User-Table Macro Instructions (Part 4 of 4)

Macro Instruction	Operand	Sym	Register			RX- type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
			Dec Dig	(2-12)	(1)							
PARMLIST (cont.)	CKNONUM= startpos endpos gdlight							X X X				
	CKNUM= startpos endpos gdlight							X X X				
	TRANSL=											X
	IDENT=											X
	STEND	no operands										
TGROUP	TCn= tcode E	X										As Shown
TRANSLAT	TRANSCH=									X		
	TRANSTXT=								X			
TRLIST	TRID=							X				
	ROUTE=											X
	LOG											As shown
	NULL											As shown
	asaddr							X				
	TEXT=											X
	INQDISP=											X
	DEMOD10=											X
	DEMOD11=											X
GDU=											X	
* see macro description for allowable values												

This Page Intentionally Left Blank

APPENDIX H: BTAM CHARACTER SET AND CODE CORRESPONDENCE CHART

This chart shows the character set and bit patterns for the Extended Binary Coded Decimal Interchange Code (EBCDIC), and the character sets and transmission code bit patterns for each of the remote station types supported by BTAM.

The chart may be used to determine the bit patterns, as contained in main storage bytes, for each of the various characters sent or received by a specific type of station, and to determine the relationships, as established by the arrangement of the IBM-provided translation tables, among the character sets for the various types.

For convenience in referring to particular chart locations, the chart's columns and rows are given reference numbers. Combined, these numbers enable reference to a particular chart location; for example, location 21/17, the intersection of row 21 and column 17, contains NL.

Arrangement of Chart

The chart contains a group of three columns for the EBCDIC character set and a group for each of the various terminal character sets. Within the EBCDIC group, column 3 contains the 256 bit patterns comprising the code. For those bit patterns to which characters are currently assigned, the characters appear in column 1 (graphics) and column 2 (line controls and device controls). All currently assigned characters are shown, regardless of whether they are in the character sets of any of the types of remote stations represented in the remainder of the chart.

Each of the remaining groups (columns 4 through 33) contains the characters comprising the character set of a specific station type, along with the transmission code bit patterns. Column 34 repeats the EBCDIC code presented in column 3, for ease of reference.

In the EBCDIC group, the bit patterns and characters are arranged in collating sequence from hexadecimal 00 to hexadecimal FF. In the remainder of the chart, the locations of bit patterns and characters are determined by the arrangement of the translation tables.

Character Sets

This chart shows only the characters comprising the commonly used character set

options. The options represented in the chart are:

<u>Terminal</u>	<u>Option</u>
IBM 1030	Standard and "H" options
IBM 1050	System/370 option
IBM 1060	Standard option
IBM 2260	Standard option
IBM 2740	System/370 option
AT&T 83B3	"A" and "C" options
WU 115A	
WU TWX	Standard option

IBM 1030 graphics and AT&T 83B3/WU 115A graphics that differ for the respective options are indicated in the chart by S and H, and A and C, respectively. Graphics not so marked are the same in both options.

Transmission Codes

The notations in the code columns of the chart for the various types of stations represent the System/370 byte bit pattern equivalents of the applicable transmission codes. The applicable transmission codes are:

<u>Terminal</u>	<u>Code</u>
IBM 1030	Perforated tape and transmission code.
IBM 1050	Perforated tape and transmission code
IBM 1060	Perforated tape and transmission code
IBM 2260	IBM 2260 transmission code
IBM 2740	Perforated tape and transmission code (BCD code)
AT&T 83B3	5-level Baudot code
WU 115A	5-level Baudot code
AT&T TWX	8-level TWX code

Representation of Characters and Bit Patterns

Appearance of a character and its associated bit pattern in a character set signifies that the appropriate IBM-provided

translation tables effect either incoming translation (that is, translation of that character to the corresponding EBCDIC character), or outgoing translation (that is, translation of the corresponding EBCDIC character to that character), or both. How the bit pattern appears indicates which of these cases applies:

1. Where the hexadecimal representation of the bit pattern appears in brackets, only incoming translation is performed.
2. Where the bit pattern is enclosed in parentheses, only outgoing translation is performed.
3. Where the bit pattern is not enclosed by brackets or parentheses, both incoming and outgoing translation are performed.

Because each unique bit pattern for a terminal character can be represented only once in an "incoming" translation table, the character associated with the bit pattern can be translated to only one EBCDIC character. The converse is not true, however; any one transmission code bit pattern can be placed any number of times within an "outgoing" table. Therefore, any number of EBCDIC characters can be translated to the terminal character represented by that bit pattern.

Appearance of two bit patterns opposite a single character signifies that the character has both an upper-case (or figures shift) and a lower-case (or letters shift) bit pattern, and that both forms of the character are translated to the same EBCDIC character. (Exception: In the code column for TWX terminals, where two bit patterns appear, the left-hand one is the even-parity pattern, and the right-hand one is the non-parity pattern.)

Example: The bit pattern of the NL character appears in location 21/9. Both the lower- and upper-case bit patterns of this character are translated to the EBCDIC NL character when they appear in an incoming message. When an EBCDIC NL character appears in an outgoing message, BTAM translates it to the lower-case form of the NL character.

Where more than one EBCDIC character requires translation to the same character in a terminal character set, the terminal character appears an equivalent number of times in the column (for example, locations 0/23, 6/23, 7/23, 23/23, and 50/23 all contain the LTRS character).

Where a character appears in both the graphics and the controls columns for a

terminal type, its function depends on whether it is sent when the line is in control mode or in text mode. Depending on the type of terminal and the mode, the character may perform a control function, print as a graphic, or both. For details, see the reference manuals for the various terminal types.

Nonequivalent Characters

Designing the system to accommodate terminal types having different character sets and control functions has resulted in several instances where dissimilar characters have been "equated" in translation tables. This accounts for the appearance in certain rows of this chart of nonequivalent characters, for example, in rows 3, 38, and 50.

In other instances, the same or similar functions have different names among the various terminal types; for example, HT and Tab in row 5 are equivalent, as are DEL and Rubout in row 7. In a few cases, terminals using the same transmission code have different meanings assigned to the identical bit pattern; for example, bit pattern 79 in the transmission code has the meaning PF for an IBM 1050, and Subtract for an IBM 1060.

Substitutions

Where blank positions appear in the terminal character set portion of the chart, there is no equivalent character for the EBCDIC character or bit pattern at the left of the chart. Where these blanks appear, the SUB character is to be assumed (they were omitted to make the chart more readable). That is, in each translation table that handles incoming messages, each position representing an invalid transmission code bit pattern (that is, one not used by a character in the terminal's character set) contains the EBCDIC code (3F) for the SUB character. In each translation table that handles outgoing messages, the transmission code bit pattern for a substitute graphic is contained in each of the following positions:

- Each position that represents an invalid EBCDIC bit pattern (a pattern to which no EBCDIC character has been assigned).
- Each position that represents a bit pattern for a character having no equivalent in the destination terminal's character set.

For the IBM 1050, 2260, and 2740, and the AT&T 83B3 and WU 115A, this substitute character is a colon (:). For the IBM 1030 and 1060, and the WU TWX, it is a slash (/).

General Notes

1. Standard abbreviations are used to represent the control characters. The full names of the characters are given in a following section entitled "Control Characters." For descriptions of these characters, see the reference manuals for the various terminals.
2. Where a "circle" character (ⓑ, ⓓ, etc.) appears in parentheses adjacent to a control character, it is an alternate name for that control character.
3. Notes pertaining to specific characters or bit patterns are indicated by superscript numerals next to the character or bit pattern. The notes appear below, and indicate the chart locations to which they apply.
4. Most of the characters in the "S" and "H" character set options (1030) and in the "A" and "C" character set options (83B3, 115A) are identical. Where they differ between the options, the translation tables "favor" the "S" option and the "A" option, as illustrated in the chart. If messages from an "H" option 1030 are sent only to another "H" option 1030, the translation table may be used as is, and similarly, for the 83B3/115A, with respect to the "C" option. If messages from terminals with the "H" or "C" option are to be exchanged with other terminal types, you may wish to modify the tables.
5. Some TWX terminals send even-parity transmission code bit patterns; others send non-parity bit patterns. All bit patterns sent by non-parity machines have a "1" in the low-order bit position (that is, the position that serves as the parity bit in even-parity machines). The RCT2 translation table translates either a non-parity or an even-parity bit pattern to the EBCDIC bit pattern for the corresponding character. The SCT2 translation table always sends even parity.

Control Characters

ACK	Positive Acknowledgment
ⓑ	End-of-block (same as EOB)
BEL	Bell
BS	Backspace
BYP	Bypass
ⓒ	End-of-transmission (same as EOT)

Control Characters (cont)

CAN	Cancel
CC	Cursor control
CR	Carriage (carrier) return
CU1	Customer Use 1
CU2	Customer Use 2
CU3	Customer Use 3
ⓓ	Machine end-of-address (same as EOA)
DC1	
DC2	Device controls
DC4	
DEL	Delete
DLE	Data link escape
DS	Digit select
EM	End of medium
ENQ	Enquiry
EOA	End-of-address
EOB	End-of-block
EOC	End of card
EOFC	End of first card
EOM	End-of-message
EOT	End-of-transmission
ETB	End-transmission-block
ETX	End-of-text
FF	Forms feed
FIGS	Figures shift
FS	Field separator
HT	Horizontal tabulate
IFS	Interchange file separator
IGS	Interchange group separator
IL	Idle
IRS	Interchange record separator
IUS	Interchange unit separator
LC	Lowercase snift
LF	Line feed

Control Characters (cont)

LF-CR	Line feed-carriage return
LTRS	Letters shift
MZ	Minus zero
Ⓝ	Negative response to polling, addressing, or LRC/VRC
NAK	Negative acknowledgment
NL	New line
NUL	Null
PF	Punch off
PN	Punch on
PRE	Prefix
PZ	Plus zero
RES	Restore
RM	Record mark
RS	Reader stop
Ⓢ	Start-of-address
SI	Shift in
SM	Set mode
SO	Shift out

Control Characters (cont)

SOH	Start-of-header
SMM	Start-manual-message
SOS	Start-of-significance
SP	Space
STX	Start-of-text
SUB	Substitute
SYN	Synchronous idle
Tab	Tabulate (horizontal)
TM	Tape mark
TpAuxOff	Tape auxiliary off
TpAuxOn	Tape auxiliary on
UC	Upper-case shift
VT	Vertical Tabulate
WRU	'Who Are You?'
X-Off	Transmitter off
X-On	Transmitter on
Ⓨ	Positive response to polling, addressing, or LRC /VRC

The character set and code correspondence charts (pages H-5 thru H-12) follow the index. To improve their usability, move them to follow page H-4.

Notes:

¹Left bracket translates to EBCDIC hex 79; no EBCDIC character has been assigned to this bit pattern (location 121/3, 121/25).

²No graphic prints in the "A" character set option (location 90/22).

³Backslash translates to EBCDIC hex E1; no EBCDIC character has been assigned to this bit pattern (locations 225/3, 225/25).

⁴IBM 1031 sends the numeric 0 as a hex 20; 1033 receives the numeric 0 as a hex 15 (location 240/4).

⁵Right bracket translates to EBCDIC hex 49; no EBCDIC character has been assigned to this bit pattern (locations 73/3, 73/25).

DIC	IBM 1030				IBM 1050			IBM 1060			IBM 2260						IBM 2740			AT&T 83B3 W U 115A			WU TWX				WTIA (ITA2)		WTIA (ZSC3)			EBCDIC	Ref.	
	Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	2260		1053		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)				
		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control	Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Even	Non		Graphic			Control
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
	48 49 ⁵ 4A 4B		EOFC	76	¢	Ⓢ	A0 76		Ⓢ	(76)	.		4E	.		(4E)	¢	Ⓢ	A0 76			27]		BB 74 75								48 49 4A 4B	72 73 74 75
	4C 4D 4E 4F			< (+ 			84 93 E1 B7				< (+ 		5C 48 4B FE	< (+ 		(5C) (48) (4B) (FE)	< (+ 		84 93 E1 B7	A ¹ c ¹ /2		3E	< (+ 		3C 14 D4 7B	3D 15 D5 7B	(+		3E 31	(+		3E 4D 4E 4F	76 77 78 79	
	50 51 52 53	& H ⁺		61	&		61	+		(61)	&		46	&		(46)	&		61	&		28	&		65 65							50 51 52 53	80 81 82 83	
	54 55 56 57																															54 55 56 57	84 85 86 87	
	58 59 5A 5B	\$		57	!		D7 57	\$		(57)	!		44	!		(44)	!		D7 57	A ² c ¹ /4 \$		36 32	!		84 24 85 25							58 59 5A 5B	88 89 90 91	
	5C 5D 5E 5F) 			90 95 B7 F6)) 		4A 49 5B FC) 		(4A) (49) (5B) (FC)) 		90 95 87 F6	A ¹ c ³ /4 A ² c ³ /8		29 2F) 		55 95 DD DD	55 95 DD DD)		29)		5C 5D 5E 5F	92 93 94 95	
	60 61 62 63	- /	Ⓢ	40 23	- /	Ⓢ	40 23	- /	Ⓢ	(40) 23	- /		4D 4F	- /		(4D) (4F)	- /	Ⓢ	40 23	- /		38 37	- /		84 F5	85 F5	- /		38 37	- /	30 2A	60 61 62 63	96 97 98 99	
	64 65 66 67																															64 65 66 67	100 101 102 103	
EOM	68 69 6A 6B		Ⓢ	37	■		37	■		(37)	■		41 4C	■		(41) (4C)	■	Ⓢ	37	A ¹ c ⁷ /8		26	■		35 35	■				26	■		68 69 6A 6B	104 105 106 107
	6C 6D 6E 6F			∨ 		Ⓢ	3B C9 AE A3	∨ 			∨ 		45 BF 5E 5F	∨ 		(45) (BF) (5E) (5F)	∨ 	Ⓢ	88 C0 8E A3	A ² c ⁵ /8		33	∨ 		A5 FA 7D FC	A5 FB 7D FD	?		33	?	25	6C 6D 6E 6F	108 109 110 111	
	70 71 72 73																																70 71 72 73	112 113 114 115
	74 75 76 77																																74 75 76 77	116 117 118 119
EOA	78 79 ³ 7A 7B	s ¹ H ⁼ s ² H ¹	EOA(Ⓢ)	16 (20)	;	EOA(Ⓢ)	88 16	;	EOA(Ⓢ)	16 (20)	;	EOA(Ⓢ)	5A 43	;	EOA(Ⓢ)	(5A) (43)	;	EOA(Ⓢ)	88 16	A ¹ c ¹ /8 CR		2E (02)	[D8 5C C5	D8 5D C5	:		2E	:	23	78 79 7A 7B	120 121 122 123	
	7C 7D 7E 7F			@		EOA(Ⓢ)	20 8D 82 96	@		(20)	@		E0 47 5D	@		(E0) (47) (5D)	@	EOA(Ⓢ)	20 8D 82 96	A ¹ c Bell		34	@ 		03 E4 8D 44	03 E5 8D 45	,		34 2F	,	34 2F	7C 7D 7E 7F	124 125 126 127	
	80 81 82 83	A B C		(62) (64) (67)	a b c		62 64 67	A B C		(62) (64) (67)	a b c		(A1) (A2) (A3)	A B C		(A1) (A2) (A3)	a b c		62 64 67	A B C		(18) (13) (0E)	A B C		(82) (42) (C3)	(83) (43) (C3)	A B C		(18) (13) (0E)	A B C	80 81 82 83	128 129 130 131		
	84 85 86 87	D E F G		(68) (68) (60) (6E)	d e f g		68 68 4D 6E	D E F G		(68) (68) (60) (6E)	d e f g		(A4) (A5) (A6) (A7)	D E F G		(A4) (A5) (A6) (A7)	d e f g		68 68 4D 6E	D E F G		(12) (10) (16) (0B)	D E F G		(22) (A3) (63) (E2)	(23) (A3) (63) (E3)	D E F G		(12) (10) (16) (0E)	D E F G	84 85 86 87	132 133 134 135		
	88 89 8A 8B	H I		(70) (73)	h i		70 73	H I		(70) (73)	h i		(A8) (A9)	H I		(A8) (A9)	h i		70 73	H I		(05) (0C)	H I		(12) (93)	(13) (93)	H I		(05) (0C)	H I	88 89 8A 8B	136 137 138 139		
	8C 8D 8E 8F																																8C 8D 8E 8F	140 141 142 143

Ref.	EBCDIC			IBM 1030			IBM 1050			IBM 1060			IBM 2260						IBM 2740			IBM 7770/7772			AT&T 8383 W U 115A					
	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	2260			1053			Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)	Character		Code (Hex)
	Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control		Graphic	Control	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
144			90																											
145	i		91	J		(43)		i		J	(43)	J		(AA)	J		(AA)	i		43				J		(1A)				
146	k		92	K		(45)		k		K	(45)	K		(AB)	K		(AB)	k		45				K		(1B)				
147	l		93	L		(46)		l		L	(46)	L		(AC)	L		(AC)	l		46				L		(1C)				
148	m		94	M		(49)		m		M	(49)	M		(AD)	M		(AD)	m		49				M		(1D)				
149	n		95	N		(4A)		n		N	(4A)	N		(AE)	N		(AE)	n		4A				N		(1E)				
150	o		96	O		(4C)		o		O	(4C)	O		(AF)	O		(AF)	o		4C				O		(1F)				
151	p		97	P		(4F)		p		P	(4F)	P		(B0)	P		(B0)	p		4F				P		(1G)				
152	q		98	Q		(51)		q		Q	(51)	Q		(B1)	Q		(B1)	q		51				Q		(1H)				
153	r		99	R		(52)		r		R	(52)	R		(B2)	R		(B2)	r		52				R		(1I)				
154			9A																											
155			9B																											
156			9C																											
157			9D																											
158			9E																											
159			9F																											
160			A0																											
161			A1																											
162	s		A2	S		(28)		s		S	(28)	S		(B3)	S		(B3)	s		28				S		(14)				
163	t		A3	T		(29)		t		T	(29)	T		(B4)	T		(B4)	t		29				T		(15)				
164	u		A4	U		(2A)		u		U	(2A)	U		(B5)	U		(B5)	u		2A				U		(1C)				
165	v		A5	V		(2A)		v		V	(2A)	V		(B6)	V		(B6)	v		2A				V		(1D)				
166	w		A6	W		(2C)		w		W	(2C)	W		(B7)	W		(B7)	w		2C				W		(1E)				
167	x		A7	X		(2F)		x		X	(2F)	X		(B8)	X		(B8)	x		2C				X		(1F)				
168	y		A8	Y		(31)		y		Y	(31)	Y		(B9)	Y		(B9)	y		31				Y		(1G)				
169	z		A9	Z		(32)		z		Z	(32)	Z		(BA)	Z		(BA)	z		31				Z		(1H)				
170			AA																											
171			AB																											
172			AC																											
173			AD																											
174			AE																											
175			AF																											
176			B0																											
177			B1																											
178			B2																											
179			B3																											
180			B4																											
181			B5																											
182			B6																											
183			B7																											
184			B8																											
185			B9																											
186			BA																											
187			BB																											
188			BC																											
189			BD																											
190			BE																											
191			BF																											
192		PZ	C0																											
193	A		C1	A		62		A	PZ	75	Restore	(75)	A		A1	A	(A1)	A		E2	A		[C9]	A		18				
194	B		C2	B		64		B		E2		(62)	B		A2	B	(A2)	B		E4	B		[D1]	B		19				
195	C		C3	C		67		C		E4		(64)	C		A3	C	(A3)	C		E7	C		[D9]	C		13				
196	D		C4	D		68		D		E7		(67)	D		A4	D	(A4)	D		E8	D		[CA]	D		12				
197	E		C5	E		68		E		EB		(68)	E		A5	E	(A5)	E		EB	E		[D2]	E		10				
198	F		C6	F		6D		F		ED		(68)	F		A6	F	(A6)	F		ED	F		[DA]	F		16				
199	G		C7	G		6E		G		EE		(6E)	G		A7	G	(A7)	G		EE	G		[CB]	G		08				
200	H		C8	H		70		H		F0		(70)	H		A8	H	(A8)	H		F0	H		[D3]	H		05				
201	I		C9	I		73		I		F3		73	I		A9	I	(A9)	I		F3	I		[DB]	I		0C				
202			CA																											
203			CB																											
204			CC																											
205			CD																											
206			CE																											
207			CF																											
208		MZ	D0																											
209	J		D1	J		43		J	MZ	54	Message	(54)	J		AA	J	(AA)	J		C3	J		[89]	J		1A				
210	K		D2	K		45		K		54		(43)	K		AB	K	(AB)	K		C5	K		[91]	K		1E				
211	L		D3	L		46		L		54		(45)	L		AC	L	(AC)	L		C6	L		[99]	L		07				
212	M		D4	M		49		M		54		(46)	M		AD	M	(AD)	M		C9	M		[8A]	M		09				
213	M		D5	N		4A		N		54		(49)	N		AE	N	(AE)	N		CA	N		[92]	N		06				
214	O		D6	O		4C		O		54		(4A)	O		AF	O	(AF)	O		CC	O		[9A]	O		03				
215	P		D7	P		4F		P		54		(4C)	P		B0	P	(B0)	P		CF	P		[8B]	P		0D				

Control	IBM 1050		IBM 1060		IBM 2260		IBM 1053		IBM 2740		IBM 7770/7772		AT&T 8383 W U 115A		WU TWX		WTTA (ITA2)		WTTA (ZSC3)		EBCDIC			
	Character		Character		Character		Character		Character		Character		Character		Character		Character		Character		Character		Code (Hex)	Ref.
	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	Graphic	Control	37	
	Q		Q		Q		Q		Q		Q		Q		Q		Q		Q		Q		D8	216
	R		R		R		R		R		R		R		R		R		R		R		D9	217
																							DA	218
																							DB	219
																							DC	220
																							DD	221
																							DE	222
																							DF	223
		RM																					E0	224
	S		S		S		S		S		S		S		S		S		S		S		E1	225
	T		T		T		T		T		T		T		T		T		T		T		E2	226
	U		U		U		U		U		U		U		U		U		U		U		E3	227
	V		V		V		V		V		V		V		V		V		V		V		E4	228
	W		W		W		W		W		W		W		W		W		W		W		E5	229
	X		X		X		X		X		X		X		X		X		X		X		E6	230
	Y		Y		Y		Y		Y		Y		Y		Y		Y		Y		Y		E7	231
	Z		Z		Z		Z		Z		Z		Z		Z		Z		Z		Z		E8	232
																							E9	233
																							EA	234
																							EB	235
																							EC	236
																							ED	237
																							EE	238
																							EF	239
	0		0		0		0		0		0		0		0		0		0		0		F0	240
	1		1		1		1		1		1		1		1		1		1		1		F1	241
	2		2		2		2		2		2		2		2		2		2		2		F2	242
	3		3		3		3		3		3		3		3		3		3		3		F3	243
	4		4		4		4		4		4		4		4		4		4		4		F4	244
	5		5		5		5		5		5		5		5		5		5		5		F5	245
	6		6		6		6		6		6		6		6		6		6		6		F6	246
	7		7		7		7		7		7		7		7		7		7		7		F7	247
	8		8		8		8		8		8		8		8		8		8		8		F8	248
	9		9		9		9		9		9		9		9		9		9		9		F9	249
																							FA	250
																							FB	251
																							FC	252
																							FD	253
																							FE	254
																							FF	255

APPENDIX I: TRANSMISSION CODE CHART

This chart may be used in reading transmission code as found in main storage. In the leftmost column of each section of the chart is shown the hexadecimal representations of the 256 bit patterns that can appear in a System/370 byte. Opposite each bit pattern is the character represented by that bit pattern in the EBCDIC character set and in the character sets for each of the types of terminals listed in the remaining columns of the chart. (The specific character set options represented in the chart are the same as those listed in Appendix H.) For example, before translation to EBCDIC a hexadecimal 04 appearing in main storage would represent the digit 2, if the bit pattern was received from an IBM 1030, 1050, 1060, or 2740; the character EOT, if from a 2260; or a Space character, if from an 83B3, 115A, or World Trade telegraph terminal. The absence of a character in the column headed AT&T TWX signifies that the bit pattern 04 is undefined for TWX terminals.

Example: In order to translate

1601E4CC A5011515 150201CA B1E70190

as found in storage, first separate the characters into pairs:

16 01 E4 CC A5 01 15 15

15 02 01 CA B1 E7 01 90

If this sequence was received from an IBM 1050, it represents the characters:

EOA SP B O S SP 0 0

0 1 SP N Y C SP *

so that the message entered at the 1050 terminal was, in part,

BOS 0001 NYC *

S/370 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260				IBM 2740		IBM 7770, 7772		AT&T 8383 WU 115A		WU TWX				WTTA				S/370 Byte (Hex)	
									2260		1053										ITA2		ZSC3					
	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch		Gr
00 01 02 03	NUL SOH STX ETX		SP		SP		SP				STX ETX		SOH STX ETX		SP			T O	CR	@		T O	CR	T O	CR			00 01 02 03
04 05 06 07	PF HT LC DEL	2		2		2					EOT ACK		EOT ACK	2				H N M	SP		SP	H N M	SP	H N M	SP			04 05 06 07
08 09 0A 0B		4		4		4					NL		NL	4		1 4 7		L R G	LF	P P		L R G	LF	L R G	LF			08 09 0A 0B
0C 0D 0E 0F	FF CR SO SI	6		6		6								6			SP	I P C V		0		I P C V		I P C V				0C 0D 0E 0F
10 11 12 13	DLE DC1 DC2 TM	8		8		8								8		2 5 8		E Z D B		H		E Z D B		E Z D B				10 11 12 13
14 15 16 17	RES NL BS IL	0	EOA	0	EOA	0	EOA				NAK		NAK	0	EOA	0		S Y F X		((S Y F X		S Y F X				14 15 16 17
18 19 1A 1B	CAN EM CC CU1				PN RS						CAN					3 6 9		A W J	FIGS	X		A W J	FIGS	A W J	FIGS			18 19 1A 1B
1C 1D 1E 1F	IFS IGS IRS IUS				Upshift								Upshift		CU3			U Q K		B		U Q K		U Q K		LTRS		1C 1D 1E 1F
20 21 22 23	DS SOS FS	@	n	@			Add						@		CU2 CU1 EOT			5 9		D D	EOT	5 9	CR	.	CR			20 21 22 23
24 25 26 27	BYP LF ETB (EOB) ESC (PRE)	S		s		S									EOB			/	SP	\$ \$		SP		?	SP			24 25 26 27
28 29 2A 2B		U		u		U												a) c 3/4 4 &		TPAuxOff TPAuxOff)	LF)	LF			28 29 2A 2B	
2C 2D 2E 2F		W		w		W												8 0 A A		4		8 0 :		9 8 =	Bell		2C 2D 2E 2F	
30 31 32 33		Y		y		Y												3 5 S A		FF FF	3 +	WRU	-	WRU			30 31 32 33	
34 35 36 37	PN RS UC EOT				RM													A' c Bell 6 c 1/4 /				1 6 /		1 5 4			34 35 36 37	
38 39 3A 3B					BYP													- 2 A Bell FIGS			- 2 c"	Bell FIGS	+ 3 2	FIGS			38 39 3A 3B	
3C 3D 3E 3F	DC4 NAK SUB		EOB		EOB PRE		EOB						EOB					7 1 A (c 1/2 LTRS		<		7 1 (1 (LTRS		3C 3D 3E 3F

S/370 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260 R				IBM 2740		IBM 7770, 7772		AT&T 8383 WU 115A		WU TWX		WTTA				S/370 Byte (Hex)	
									2260		1053										ITA2		ZSC3			
	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH	Gr	CH		Gr
40 41 42 43		SP	-	Ⓝ	-	Ⓝ	-	Ⓝ	-	SP	-	SP	-	Ⓝ	0											40 41 42 43
44 45 46 47			K	L	k	l	K	L	\$	%	\$	%	k	l												44 45 46 47
48 49 4A 4B			M	N	m	n	M	N	()	()	m	n	/	U	X				TrAuxOn	TrAuxOn				48 49 4A 4B
4C 4D 4E 4F	<	(O	P	o	p	O	P	/	-	/	-	o	p								2				4C 4D 4E 4F
50 51 52 53	&		Q	R	q	r	Q	R	0	1	0	1	q	r	S	V	Y				LF	LF				50 51 52 53
54 55 56 57						MZ		Message	4	5	4	5													54 55 56 57	
58 59 5A 5B						RES			8	9	8	9			T	W	Z									58 59 5A 5B
5C 5D 5E 5F						BS			<	=	<	=		BS							:	:				5C 5D 5E 5F
60 61 62 63																										60 61 62 63
64 65 66 67			B	C	b	c	B	C					b	c												64 65 66 67
68 69 6A 6B			D	E	d	e	D	E					d	e												68 69 6A 6B
6C 6D 6E 6F			F	G	f	g	F	G					f	g												6C 6D 6E 6F
70 71 72 73			H	I	h	i	H	I					h	i												70 71 72 73
74 75 76 77						PZ		Restore																		74 75 76 77
78 79 7A 7B						HT		Subit Tab						HT												78 79 7A 7B
7C 7D 7E 7F	@	'				Dwnshft								Dwnshft												7C 7D 7E 7F

S/370 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060	IBM 2260				IBM 2740		IBM 7770, 7772		AT&T 8383 W U 115A	WU TWX		WTTA				S/370 Byte (Hex)		
	Gr	CH	Gr	CH	Gr	CH		2260		1053		Gr	CH	CH	Gr		Gr	CH	ITA2		ZSC3				
								Gr	CH	Gr	CH								Gr	CH	Gr	CH		Gr	CH
80 81 82 83		a b c			=	SP																		80 81 82 83	
84 85 86 87		d e f g			<																			84 85 86 87	
88 89 8A 8B		h i			:											J M P			X-On X-On					88 89 8A 8B	
8C 8D 8E 8F					.	>																		8C 8D 8E 8F	
90 91 92 93		i k l			.											D Z K			HT HT					90 91 92 93	
94 95 96 97		m n o p)	EOA																		94 95 96 97	
98 99 9A 9B		q r			P N R S											L O R								98 99 9A 9B	
9C 9D 9E 9F					Upshif																			9C 9D 9E 9F	
A0 A1 A2 A3		s t			¢	?		A B C	A B C			¢	?						WRU WRU					A0 A1 A2 A3	
A4 A5 A6 A7		u v w x			S T			D E F G	D E F G			S T													A4 A5 A6 A7
A8 A9 AA AB		y z			U V			H I J K	H I J K			U V													A8 A9 AA AB
AC AD AE AF					W X			L M N O	L M N O			W X													AC AD AE AF
B0 B1 B2 B3					Y Z			P Q R S	P Q R S			Y Z								CR					B0 B1 B2 B3
B4 B5 B6 B7					I			T U V W	T U V W			I													B4 B5 B6 B7
B8 B9 BA BB					BYP LF			X Y Z	X Y Z			LF													B8 B9 BA BB
BC BD BE BF					EOB PRE			►Start MI	¢			EOB													BC BD BE BF

S/370 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260 R				IBM 2740		IBM 7770, 7772		AT&T 8383 W U 115A		WU TWX		WTIA				S/370 Byte (Hex)
									2260		1053										ITA2		ZSC3		
	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	Gr	Ch	
C0 C1 C2 C3	A B C	PZ			- J	Ⓝ								- J	Ⓝ	& (7770 only)									C0 C1 C2 C3
C4 C5 C6 C7	D E F G				K L									K L											C4 C5 C6 C7
C8 C9 CA CB	H I				M N									M N		A D G					X-Off				C8 C9 CA CB
CC CD CE CF					O P									O P											CC CD CE CF
D0 D1 D2 D3	J K L	MZ			Q R									Q R		B E H									D0 D1 D2 D3
D4 D5 D6 D7	M N O P				I									I											D4 D5 D6 D7
D8 D9 DA DB	Q R				RES NL										NL	C F I									D8 D9 DA DB
DC DD DE DF					BS IL									BS IL											DC DD DE DF
E0 E1 E2 E3		RM			+ A				@	@				+ A											E0 E1 E2 E3
E4 E5 E6 E7	U V W X				B C									B C											E4 E5 E6 E7
EB E9 EA EB	Y Z				D E									D E											EB E9 EA EB
EC ED EE EF					F G									F G											EC ED EE EF
F0 F1 F2 F3	0 1 2 3				H I									H I											F0 F1 F2 F3
F4 F5 F6 F7	4 5 6 7				- Ⓞ									- Ⓞ											F4 F5 F6 F7
F8 F9 FA FB	8 9				PF Tab									HT											F8 F9 FA FB
FC FD FE FF					Dwnshft DEL				- I	- I				Dwnshft DEL							? SI				FC FD FE FF

This Page Intentionally Left Blank

APPENDIX J: LINE AND STATION CONFIGURATIONS SUPPORTED BY BTAM

Start-Stop Communications

The types of remote start-stop (asynchronous) terminals that can communicate with a System/370 under BTAM control, and the kinds of communication lines that can be controlled, are described below. The communication lines must be connected to the computer via an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control.

1. Nonswitched lines (point-to-point or multipoint), using programmed polling:

IBM 1030 Data Collection System
IBM 1050 Data Communications System
IBM 1060 Data Communications System
IBM 2260 Display Station --
 IBM 2848 Display Control
 (Remote -- 2701 only)
IBM 2265 Display Station -- IBM 2845
 Display Control (Remote -- 2701
 only)
IBM 2740 Communications Terminal
 (Model 1): Basic; with Checking¹;
 with Station Control²; with Checking
 and Station Control²; or with Check-
 ing and IBM 2760 Optical Image Unit
 features (point-to-point only, if
 2740 is equipped with 2760 Optical
 Image Unit)
 (Model 2): Basic or with Checking¹
IBM 2741 Communications Terminal
Western Union Plan 115A Outstations
AT&T 83B3 Selective Calling Stations

2. Switched lines:

IBM 1050 Data Communications System
IBM 2740 Communications Terminal
 (Model 1): Dial; Dial, with Check-
 ing; Dial, with Transmit Control;
 Dial, with Checking and Transmit
 Control, or Dial, with Checking and
 IBM 2760 Optical Image Unit
 features.
IBM 2741 Communications Terminal
WU Model 33/35 Teletypewriter Exchange
Terminal (TWX)

3. Nonswitched multipoint lines using the Auto Poll facility (IBM 2702 or 2703 only):

IBM 1030 Data Collection System
IBM 1050 Data Communications System

¹Used as a regular terminal or as an operator's console, when the operating system includes the Multiple Console Support.

IBM 1060 Data Communications System
IBM 2740 (Model 1 and 2): with Sta-
tion Control² or with Station Con-
trol² and Checking features

Binary Synchronous Communications

The types of remote binary synchronous stations (computers or terminals) that can communicate with a central System/370 under BTAM control, and the kinds of communication lines that can be controlled, are as follows. The communication lines must be connected to the central computer via an IBM 2701 Data Adapter Unit or an IBM 2703 Transmission Control. An IBM 2701 (with Synchronous Data Adapter Type II) may be attached to either the multiplexer channel or a selector channel. An IBM 2703 (with Synchronous Base Type 1) must be attached to the multiplexer channel.

1. Nonswitched point-to-point and switched point-to-point lines:

IBM System/370³
IBM System/360
IBM System/360 Model 20
IBM System/3
IBM 1130 Computing System
IBM 1800 Data Acquisition and
Control System
IBM 2715 Transmission Control Unit
 (Model 1 attaches directly to multi-
 plexer channel of central computer;
 Model 2 communicates with central
 computer via IBM 2701 or 2703)
IBM 2770 Data Communications System
IBM 2780 Data Transmission Terminal
IBM 3650 Retail Store System
IBM 3735 Programmable Buffered
Terminal
IBM 3741 Data Station
IBM 3747 Data Converter

²Station Control feature cannot be used if the 2740 is also used as a console under Multiple Control Support.

³The remote System/370 may be a Model 135, 145, 155, 158, 165, 168, or 195.

IBM 3750 Switching System (World Trade users only; nonswitched point-to-point line only)

IBM 3780 Data Communication Terminal

2. Switched point-to-point lines:

IBM 3275 equipped with the dial feature

IBM 3650 Retail Store System

IBM 3660 Supermarket System

IBM 5275 Direct Numerical Control Station

3. Nonswitched multipoint lines:

IBM System/360 Model 20

IBM System/3

IBM 1130 Computing System

IBM 1800 Data Acquisition and Control System

IBM 2715 Transmission Control Unit (Model 1 attaches directly to multiplexer channel of central computer; Model 2 communicates with central computer via IBM 2701 or 2703)

IBM 2770 Data Communications System

IBM 2780 Data Transmission Terminal

IBM 2972 (Models 8 & 11) General Bank-ink Terminal System

Remote IBM 3270 Display System

IBM 3650 Retail Store System

IBM 3735 Programmable Buffered Terminal (Requires special feature)

IBM 3741 Data Station

IBM 3780 Data Communication Terminal

IBM 5275 Direct Numerical Control Station

- The remote stations must be attached to an acceptable data adapter or transmission control unit (IBM 2701, 2702, or 2703). (A local 2715 (Model 1) must be connected to the multiplexer channel.)

- All remote start-stop terminals that are connected to the same multipoint line, or are capable of communicating with the computer over any given switched line termination, must be of the same type and must be equipped with the same features. (Remote binary synchronous stations are not subject to this limitation.)

- All devices must be attached to the System/370 via the multiplexer channel except the IBM 2701 with Synchronous Data Adapter Type II, which may be attached via the selector channel (non-switched lines only), or to the multiplexer channel.

- No device may be operated in burst mode concurrently with the operation of BTAM except the 2701 attached via the selector channel.

- Execution of BTAM requires that the interval timer of the central computer be working.

- In a system in which BTAM is used in more than one partition, if the BTAM Read/Write module (IGG019MA) is resident, all device I/O modules that are shared by the BTAM-using partitions must also be made resident.

Note: When a 2740 is specified as a console during system generation, a second partition is created for BTAM; thus, the above requirement pertains to the system.

Local Communications

The local IBM 3270 Display System can communicate with a System/370 under BTAM control. The local 3270 display system is connected to the computer by means of a selector, multiplexer, or block multiplexer channel.

Machine and Programming Requirements

BTAM operates on any System/370 that meets the following requirements:

- The system must meet the minimum configuration of OS/VS.

- Use of the STIMER macro by the user is restricted during the time a BTAM Open (OPEN), Line Open (LOPEN) or Close (CLOSE) operation is in progress, because BTAM routines use STIMER.

Note: Terminals that are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to IBM-supplied products or programs may have on such terminals.

APPENDIX K: IBM 2980 CHARACTER SET AND TRANSMISSION CODE CHART

These charts show for each transmission code bit pattern the corresponding 2980 character, for each of the models of the 2980 (1, 2, and 4). Also shown is the EBCDIC character equivalent for that bit pattern.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
00	NUL						
01	SOH						
02	STX	STX	STX	STX	STX	STX	STX
03	ETX	ETX	ETX	ETX	ETX	ETX	ETX
04	PF			open chute			open chute
05	HT	HT	HT	HT	HT	HT	HT
06	LC				LC	LC	LC
07	DEL						
08							
09	RLF						
0A	SMM						
0B	VT						
0C	FF						
0D	CR						
0E	SO						
0F	SI						
10	DLE	DLE	DLE	DLE	DLE	DLE	DLE
11	DC1						
12	DC2						
13	TM						
14	RES			turn page light			turn page light
15	NL	NL	NL	NL	NL	NL	NL
16	BS						
17	IL	message light		message light	message light		message light
18	CAN						
19	EM						
1A	CC						
1B	CU1						
1C	IFS						

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
1D	IGS						
1E	IRS						
1F	IUS						
20	DS						
21	SOS						
22	FS						
23							
24	BYP*	BYP	BYP	BYP	BYP		
25	LF	pass-book index		pass-book index	pass-book index		pass-book index
26	ETB	ETB	ETB	ETB	ETB	ETB	ETB
27	ESC						
28							
29							
2A	SM						
2B	CU2						
2C							
2D	ENQ	ENQ	ENQ	ENQ	ENQ	ENQ	ENQ
2E	ACK						
2F	BEL						
30							
31							
32	SYN	SYN	SYN	SYN	SYN	SYN	SYN
33							
34	PN*						
35	RS						
36	UC	UC	UC	UC			
37	EOT	EOT	EOT	EOT	EOT	EOT	EOT
38							

* Also used as a Terminal Selection Character.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
39							
3A							
3B	CU3						
3C	DC4						
3D	NAK	NAK	NAK	NAK	NAK	NAK	NAK
3E							
3F	SUB						
40	SP*	SP	SP	SP	SP	SP	SP
41							
42							
43							
44							
45							
46							
47							
48							
49							
4A	ç						
4B	.	3	.	.	.	┘	
4C	<						
4D	(
4E	+						
4F	l						
50	&	validate I.D. char.	&	validate I.D. char	&	+	&
51							
52							
53							
54							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
55							
56							
57							
58							
59							
5A							
5B	\$	-	\$		\$		
5C	*	\$	&		*	ç	
5D)						
5E	;						
5F	┘						
60	-	F	-		-	-	
61	/	T	/		/	?	
62							
63							
64							
65							
66							
67							
68							
69							
6A							
6B	,	2	,	,	,		
6C	%						
6D	—						
6E	>						
6F	?						
70							
71							

* Also used as a Terminal Selection Character

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
72							
73							
74							
75							
76							
77							
78							
79							
7A	:						
7B	#	\$	%	&	'	"	+
7C	&						
7D	'						
7E	=						
7F	"						
80							
81	a						
82	b						
83	c						
84	d						
85	e						
86	f						
87	g						
88	h						
89	i						
8A							
8B							
8C							
8D							
8E							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
8F							
90							
91	j						
92	k						
93	l						
94	m						
95	n						
96	o						
97	p						
98	q						
99	r						
9A							
9B							
9C							
9D							
9E							
9F							
A0							
A1							
A2	s						
A3	t						
A4	u						
A5	v						
A6	w						
A7	x						
A8	y						
A9	z						
AA							
AB							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
AC							
AD							
AD							
AF							
B0							
B1							
B2							
B3							
B4							
B5							
B6							
B7							
B8							
B9							
BA							
BB							
BC							
BD							
BE							
BF							
C0							
C1	A	C	a	A	A	A	A
C2	B		b	B	B	B	B
C3	C	t	c	C	C	C	C
C4	D	$\frac{N}{B}$	d	?	D	D	D
C5	E	X	e	E	E	E	E
C6	f	$\frac{O}{B}$	f	F	F	F	F
C7	G	$\frac{-}{S}$	g	G	G	G	G

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
C8	H	<u>00</u>	h	#	H	H	6
C9	I	8	i	\bar{C}	I	I	0
CA							
CB*				M			.2
CC	\square						
CD							
CE	Υ						
CF							
D0*							R
D1	J	4	j	J	J	J	J
D2	K	5	k	K	K	K	K
D3	L	6	l	L	L	L	Q
D4	M	1	m	X	M	M	M
D5	N	0	n	N	N	N	N
D6	O	9	o	O	O	O	I
D7	P	+	p	P	P	P	H
D8	Q	R	q	$\frac{O}{B}$	Q	Q	5
D9	R	\bar{A}	r	$\frac{C}{V}$	R	R	-
DA							
DB							
DC							
DD							
DE							
DF							
E0*				$\frac{S}{V}$			4
E1				/			Y
E2	S	$\frac{T}{F}$	s	$\frac{S}{V}$	S	S	S

* A non-EBCDIC code.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
E3	T	\bar{B}	t	▲	T	T	T
E4	U	7	u	\bar{M}	U	U	U
E5	V	\bar{P}	v	▼	V	V	V
E6	W	Q	w	*	W	W	W
E7	X	\bar{M}	x	\bar{B}	X	X	X
E8	Y	\bar{D}	y	\bar{T}	Y	Y	3
E9	Z	V	z		z	z	z
EA							
EB*				l			l
EC							
ED							
EE							
EF							
F0	0	U	0	0	0)	.
F1	1	▲	1	1	1	=	L
F2	2	H	2	2	2	<	S
F3	3	\bar{F}	3	3	3	;	#
F4	4	\bar{M}	4	4	4	:	O
F5	5	\bar{L}	5	5	5	%	P
F6	6	\bar{C}	6	6	6	'	*
F7	7	\bar{O}	7	7	7	>	7
F8	8	\bar{M}	8	8	8	*	8
F9	9	\bar{C}	9	9	9	(9

* A non-EBCDIC code.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
FA	LVM						
FB							
FC							
FD							
FE	EO						
FF							

This Page Intentionally Left Blank

GENERAL CHARACTERISTICS

Data received from the IBM 50 Magnetic Data Inscrber (MDI) attachment to the IBM 2772 Multi-Purpose Control Unit contains MDI control characters. (For full information on the IBM 50, see IBM 50 Magnetic Data Inscrber Component Description.) The TPEDIT macro enables the user to edit this data. The Edit routine, entered from the TPEDIT macro in the user program, edits the data as specified, then returns control to the user program. You have the option of gaining temporary control (via a user-specified exit routine) to process error records. The Edit routine is written in reentrant code. If data is to be received from more than one MDI at a time, you must provide a separate parameter list for each of them.

When the user program issues a READ macro, it receives one block of data, which may contain one or more MDI logical records (or none). The Edit routine extracts one record from this block of data, edits it and gives it to the user program with a return code indicating whether or not the user program input area is empty. If the input area is not empty, the user program must reissue the TPEDIT macro to obtain another record. When control is returned to the user program with an indication that the input area is empty, the input area can be reused. If the input area contains a partial record, the available portion is edited into the work area, and maintained there. The Edit routine gives a return code indicating that the input area is empty but a record is not available. It is your responsibility to obtain the remainder of the record via READ macros. When control is given back to the Edit routine, the characters in the input area (until EOR is encountered) are treated as the remaining portion of the partial record.

TPEDIT Macro Instruction

The TPEDIT macro is used to specify the type of editing to be done on the input received from the IBM 50 MDI attachment to the IBM 2772.

Name	Operation	Operand
[name]	TPEDIT	MINLN=n[,REPLACE= {X'19' X'xx'}] [,EDIT={EDITD} EDITR}] [,RECFM={V U}] [,ERROPT={IGNORE}] {name}] [,VERCHK={NOCHK VOKCHK}] [,BUFFER={NO YES}]

MINLN

specifies the minimum acceptable length of an input record. For EDIT=EDITD, SOR and EOR codes are excluded from the length; for EDITR, SOR and EOR are included in the length.

REPLACE

specifies the code to be used as a replacement character whenever the Edit routine detects a 2772 replacement character (that is, the EBCDIC SUB character, X'3F') in the input. X'19' is chosen as the assumed value because it is an end-of-data (ED) signal for an IBM 50 MDI cartridge and therefore can never appear as a valid data byte. For REPLACE=X'xx' you specify xx as hexadecimal characters of your choosing. These choices may be made from the code chart in Figure 116, with exceptions as noted below.

Programming Note: BSC control characters should not be used as replacement characters if the data is to be transmitted via BSC facilities after editing.

Hexadecimal characters representing special purpose MDI codes that should not be used as replacement bytes are:

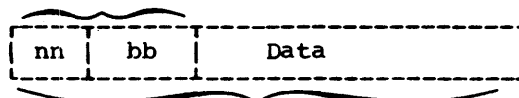
X'00' (LZ)	X'1E' (VOK)	X'74' (P4)
X'11' (DUP)	X'3C' (RM)	X'75' (P5)
X'12' (LZS)	X'71' (P1)	X'76' (P6)
X'18' (CAN)	X'72' (P2)	X'77' (P7)
X'1D' (GS)	X'73' (P3)	X'78' (P8)

at a time from the input area by scanning for the record delimiting codes (SOR and EOR). DUP codes are replaced by the character from the corresponding location of the record that was in the work area when control was last returned to the Edit routine. (This does not apply to the first record of a cartridge.) Left-zero fields are right-adjusted, with leading zeros inserted where necessary. Left-zero start codes, records containing a cancel code, and group separator codes do not appear in the output stream. Line control characters (ETB, ETX, STX, and DLE STX) are always deleted if found in the input area.

RECFM

specifies the format of the output from the Edit routine. If RECFM=V, a segment descriptor word is appended to each record as shown.

Segment Descriptor Word



Logical Record

where nn (2 bytes) is the length of the logical record and bb (2 bytes) is binary zeros reserved for system use.

This four-byte field is included in the record length returned to the user program via a parameter list.

Note: Allow for this four-byte field when determining the size of the work area (see section on Input to the TPE-DIT macro).

If RECFM=U, no segment descriptor word is appended to each record.

ERROPT

specifies whether a user error exit routine is provided to handle erroneous records.

IGNORE

an error exit routine is not provided. The error conditions are to be disregarded and the record is to be passed normally to the user program.

name

specifies the name of the user error exit routine to be entered when the Edit routine detects logical errors or replacement characters in the record.

VERCHK (valid only if ERROPT=name is coded) specifies whether the records are to be checked for verify-OK (VOK) codes.

If you specify VOKCHK and a record does not contain the verify-OK code, the record is passed to the error exit routine.

When the Edit routine encounters an erroneous record and control passes to this user-supplied routine, register 13 contains the address of a 72-byte register save area aligned on a full-word boundary, and register 1 contains the address of a two-word parameter list aligned on a fullword boundary. The parameter list is defined as follows:

Word	Contents
1	Record address
2	Address of record length

The record length includes the four-byte error description word appended, as shown, to the data record. In addition, if RECFM=V, the logical record length (nn) includes these four bytes when the record is passed to the error exit routine.

If RECFM=V is specified:

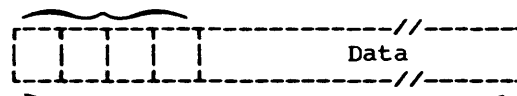
Error Description Word



Logical record

If RECFM=U is specified:

Error Description Word



Logical record

Information on the Error Description Word may be found under Error Record Identification. The error exit routine can be used to analyze and, if possible, correct the erroneous record. When control returns to the Edit routine via register 14, you must set register 15 to zero if you wish to bypass the error record. To direct the Edit routine to pass only that segment of the record in error and process the rest of the record normally, set register 15 to a nonzero value. Note that neither acceptance nor bypassing of the erroneous record changes its effects on subsequent records. The Edit routine removes the

error description word when control returns from the error exit routine.

BUFFER

specifies whether or not the user data is in BTAM buffers obtained through dynamic buffering operations. If you specify YES, the Edit routine edits all data in the input area until the area is empty.

Note: The entire buffer chain must have been posted complete in the DECB before you issue the TPEDIT macro.

Input to TPEDIT Macro

Register 1 must point to a four-word parameter list (aligned on a fullword boundary) containing:

<u>Word</u>	<u>Contents</u>
1	Input Address If you are using dynamic buffering, this address points to the first buffer in the chain. The Edit routine edits all records in the buffer chain before indicating that the input area is empty. If dynamic buffering is not used, this is the address of the data to be edited.
2	Input Length If dynamic buffering is used, this is the length of one buffer. If an I/O area is used, this is the length of the data to be edited.
3	Edit w 3rk area address The work area required by the Edit routine for a given parameter list is obtained in either of two ways. The work area can be provided by the Edit routine (via an unconditional GETMAIN), or you may provide it.

If the work area is to be provided by the EDIT routine, this word must contain binary zeros. The Edit routine issues a GETMAIN macro to obtain the required storage and places the address of the storage obtained in this word. If you provide the work area, this word contains the address of the area supplied. The amount of storage needed in addition to the fixed amount required is determined from:

- (1) the maximum record length.
- (2) whether a user exit exists (72 bytes for a register save area and 4 bytes for an EDW are required by the macro if an exit is specified).
- (3) whether RECFM=V.

The size (in bytes) of the work area may be determined from the formula:

$$S = 84 + 76E + R + 4V$$

Where:

S is the size (in bytes) of the work area.

E = 0 if ERROPT=IGNORE is coded

= 1 if ERROPT=NAME is coded

V = 0 if RECFM=U

= 1 if RECFM=V

R is the length of the longest record to be processed.

4 Maximum record length.
This is the length, in bytes, of the longest valid edited record. For EDIT=EDITD the length should exclude SOR and EOR codes; for EDIT=EDITR, the length should include SOR and EOR codes.

The value of the maximum record size should not include the four-byte segment descriptor word added to a variable length record.

Records that exceed the maximum record size are considered erroneous records. Register 13 must contain the address of a 72-byte register save area aligned on a fullword boundary.

Return Codes

After the Edit routine has edited a record, it provides in register 15 a return code indicating record availability and status of the input area, prior to returning control to the user program. The return codes and their meanings are as follows:

<u>Code (hex)</u>	<u>Meaning</u>
00	A record is available; input area is empty. The routine has edited the last logical record in the input area and is passing the record to the user program.
04	A record is available; input area is not empty. The routine has edited one logical record and is passing that record to the user program.
08	No record is available; input area is empty. The last record

in the input area was incomplete; that is, it was a partial record.

0C End-of-data (ED) code was detected.

For return codes 00 and 04, the record address and the address of the record length are given to the user program in a two-word parameter list aligned on a full-word boundary. The address of the parameter list is returned in register 1. The parameter list has the following format:

<u>Word</u>	<u>Contents</u>
1	Address of the record
2	Address of the record length

RECORDS CONTAINING ERRORS

This section describes what the Edit routine considers to be records containing errors. Once the Edit routine has determined a record to be in error, it passes that record to the user error exit routine, if ERROPT=name is specified in the TPEDIT macro statement. If an error exit routine is not specified, the erroneous record is returned to the user program.

The Edit routine maintains information about each record as it is being edited. This information is summarized in the Error Description Word (EDW) described below. When the EDW contains a nonzero value in either the Level Status (byte 0) or the Type Status (byte 1), the record is considered an erroneous record and the EDW is inserted between the four-byte record length field and the data portion if RECFM=V is specified. Otherwise, the EDW is appended to the start of the record to help you analyze the error. Figure 116 shows the format of the EDW.

Level Status (Byte 0)

The level status indicator identifies erroneous records that result from interrecord dependency and that cannot be identified in the type status byte. The level status is presented with each erroneous record and has one of the values shown in Figure 117.

A level status of other than zero is presented with erroneous records resulting from the following:

- The start-of-record (SOR) location has a character defined as an error.
- The record contains two or more data check bytes in succession.

- The record is longer than the user-specified maximum length record.
- The length of the record is not equal to the length of the first valid record of the same program level encountered on the MDI cartridge from which data is being obtained.
- The record has a data duplication dependency on a previous record having one of the foregoing errors.

The level status is set to zero whenever the Edit routine encounters (1) a record without one of the previous errors, (2) a canceled record, or (3) the first record of a cartridge.

Type Status (Byte 1)

The type status indicator identifies records in error because of SOR, EOR, length, field, and/or data check error conditions.

The type status is presented with each erroneous record and has a value of:

- 0 For any record that has no identifiable errors, but contains questionable data due to a level status of other than zero (see Level Status).
- 1 For any record that (1) has a SOR character of other than P1 through P8 or a GS code, or (2) has an EOR character of other than a VOK code when you have specified VERCHK=VOKCHK, or (3) has an EOR character of other than a VOK code or RM code when you have specified VERCHK=NOCHK.
- 2 For any record that has an incorrect length because it is:
 - Longer than the specified maximum, or
 - Shorter than the specified minimum (MINLN), or
 - Not equal to the length of the first valid record of the same program level encountered on the MDI cartridge from which data is being obtained.
- 4 For any record that has one or more field errors. A field error is a field or fields where duplication and/or left-zero justification functions did not occur due to an error condition.
- 8 For any record that has a data check error.

<p><u>Byte 0: Level status</u></p> <p>0 - for any error record that will not cause questionable data to be in the following records.</p> <p>1 - for any error record that may cause questionable data to be in the following records.</p> <p>2 - for any error record that (1) contains questionable data due to the error level of preceding record(s) and (2) may cause questionable data to be in the following records; and where the level status of the previous record was either 1 or 2.</p>
<p><u>Byte 1: Type status</u></p> <p>0 - No identifiable errors.</p> <p>1 - Start-of-record (SOR) or end-of record (EOR) in error.</p> <p>2 - Length error.</p> <p>4 - Field error.</p> <p>8 - Data check error.</p> <p><u>Note:</u> This field may contain combinations of these error types; for example, a C (hexadecimal) indicates a data check error and a field error.</p>
<p><u>Byte 2: Program Level</u></p> <p>1 - P1 5 - P5 E -none of the preceding levels. Start-of-record 2 - P2 6 - P6 (SOR) is in error. 3 - P3 7 - P7 4 - P4 8 - P8</p>
<p><u>Byte 3: Record Status</u></p> <p>U - Unverified record. V - Verified record. E - Neither U nor V. End-of-record (EOR) is in error.</p> <p><u>Note:</u> The error description record is in EBCDIC format. For example, a 2 is represented as X'F2'; a C is represented as X'C3'.</p>

Figure 117. Format of Error Description Word

The type status indicator can also have hexadecimal values of 3, 5, 6, 7, 9, A, B, C, D, E and F. These values indicate various combinations of SOR, EOR, length, field, and data check errors. For example, a value of A indicates a record with a data check error (8) as well as an incorrect length error (2).

Note: A data check error is indicated by the presence of 2772 replacement characters (that is, EBCDIC SUB characters, X'3F'), in the input.

Program Level (Byte 2)

This byte contains an indication of the start-of-record (SOR) character associated with this record. (See Figure 108 for values.)

Record Status (Byte 3)

This byte contains an indication of the end-of-record (EOR) character associated with this record. (See Figure 109 for values.)

EXAMPLES OF RECORDS CONTAINING ERRORS

Figure 118 shows some of the errors that may occur during processing and their effect on the error description word (EDW). For these records, the maximum record length is specified as 50, EDITR and VOKCHK are specified, and the hexadecimal REPLACE character is '5B' (\$). An asterisk in the records indicates the presence of a DUP code in the location before editing.

(Record 2).

19EV	***** V *O
\$111378 RECORD NUMBER 2AK	

(Record 3)

201V	P ***** V *O
1357987 RECORD NUMBER 3AK	

(Record 4)

081V	P ***** V O
1358977 REC\$RD NUMBER 4AK	

(Record 5)

131U	P 1358436 RECORD NUMBER 5M	R
------	----------------------------	---

(Record 6)

241V	P ***** V *O
1358436 RECORD NUMBER 6\$K	

(Input record 7)

233E	P 3998865 RECORD NUMBER 7A MAXIMUM 00001430 IN WAREH	V O O USEK
------	------------------------------------------------------	---------------------

(Error record 7)

(Error record 8)

21EV	V O O USEK
------	---------------------

(Input record 8)

(Error record 9)

081V	P 1367\$82 RECORD NUMBER 8AK	V O
------	------------------------------	--------

Resulting Error Description Word

Figure 118. Examples of Erroneous Records (IBM 50 MDI)

Record 1 was a valid record. It contained a program level 1 code and thus established the valid length for all program level 1 records received from the cartridge.

Record 2 has a data check in the SOR location. Level status is set to 1 because the SOR location might have contained a cancel code that would cause any data duplicated into the following record to be questionable.

Record 3 has no identifiable error but may contain questionable data because it contained DUP codes and follows a record with a level status of 1.

Record 4 has a data check error. Because it contained no DUP codes, the level status is set to 0.

Record 5 is shorter than first program level 1 records received from the cartridge (length error). This record also contains an RM code rather than a VOK code in the EOR location (VOKCHK was specified). Because the Edit routine cannot determine why the record is short, all data duplicated from this record is questionable; the level status is therefore set to 1.

Record 6 contains a DUP code that is beyond the last position of the preceding record.

Record 7 is longer than the maximum specified record length. Note that it is passed as two records. The first record indicates an EOR error and a length error; the second indicates an SOR error.

Record 9 has a data check error. Because it contained no DUP codes, the Level Status is set to zero.

PROGRAMMING CONSIDERATIONS

- All canceled records are bypassed and are not passed as erroneous records.
- All input records less than three bytes in length (SOR location, one data byte, EOR location) are treated as canceled records. An input record of this size may be the remaining portion of a record that was longer than the maximum user-specified record size.
- Data duplication occurs with the DUP code replaced by the character from the corresponding location of the previous record that was in the work area when control was last returned to the Edit routine.
- For any of the following conditions, data duplication does occur and the DUP code is replaced with the user-specified error replacement character and a field error is indicated:

The DUP code is encountered in the first record of a cartridge.

The DUP code is encountered in a record and the previous record was a canceled record.

The DUP code is encountered in a record and its position would cause duplication of the previous record's end-of-record delineator location or a position beyond the length of the previous record.

The DUP code is encountered in a record and its position would cause duplication of an error replacement character.

- For either of the following conditions, left-zero justification does not occur, the left-zero-fill code (LZ) is replaced with the user-specified error replace character, and a field error is indicated:

The left-zero-fill code (LZ) is encountered without its corresponding left-zero-start code (LZS).

The user-specified maximum record size is exceeded before the valid end of a left-zero field is encountered.

- If dynamic buffering is being used, the BSC control characters ETB and ETX should not be entered as data on IBM 50 MDI cartridges.

END-OF-CARTRIDGE CODE

A unique code, written by the IBM 50 MDI, is used to signal the 2772 control unit that all meaningful data on a cartridge has been read. For the MDI cartridge, the end-of-cartridge code is the ED character (X'19'), which is equivalent to the EBCDIC end-of-medium (EM) character (X'19').

After initiation of a Read operation the MDI continues to read data from the tape until it senses the ED character. When the MDI sends this character to the 2772, the 2772 signals the tape unit to rewind the tape and then transmits the data in its buffer to the central computer.

```

//TEST JOB MSGLEVEL=1
//STEP EXEC ASMF
//ASM.SYSLIB DD DSN=SYS1.MACLIB,DISP=OLD
//SYSIN DD *
CONFIGUR CORE=32,PC=YES,INQDISP=YES,GDU=YES,GETID=F0,STORID=F5,      X
IDCOUNT=8,MONERR=(4,5),FUNCERR=(6,7),ENDERR=(8,9)
*
*       THE CORE=32 OPERAND OF THE CONFIGUR MACRO
*       INDICATES THAT THE 2715 HAS 32K OF STORAGE
*       AVAILABLE. CODING PC=YES, INDICATES THAT
*       PULSE COUNTERS EXIST ON THIS 2790 SYSTEM.
*       INQDISP=YES INDICATES THAT INQUIRY DISPLAY
*       WILL BE USED ON THE 2790 SYSTEM. GDU=YES
*       INDICATES THAT 2798 GUIDANCE DISPLAY UNITS
*       ARE ON THIS 2790 SYSTEM. THE USER CAN
*       DEFINE EIGHT IDENTIFIERS. THE GET IDENTIFIER
*       CHARACTER IS THE EBCDIC CHARACTER 0 AND THE
*       STORE IDENTIFIER CHARACTER IS THE EBCDIC
*       CHARACTER 5. IF A MONITOR KEY CHECK FAILS,
*       ERROR GUIDANCE LIGHTS 4 AND 5 WILL BE
*       TURNED ON AT THE 2798. WHEN AN INVALID
*       FUNCTION IS RECOGNIZED, ERROR GUIDANCE
*       LIGHTS 6 and 7 WILL BE TURNED ON AT THE 2798.
*       WHEN A PREMATURE TERMINATION ERROR OCCURS,
*       ERROR GUIDANCE LIGHTS 8 AND 9 WILL BE
*       TURNED ON AT THE 2798.
*
*
*       THE FOLLOWING AS MACROS DEFINE 60 AREA
*       STATIONS WITH ID'S BETWEEN 0 AND 59
*       FROM WHICH TRANSACTIONS CAN BE ENTERED.
*       WORKOUT AND NORMAL ARE THE NAMES OF THE
*       TGROUP MACROS THAT DEFINE THE TRANSACTION
*       CODES THAT CAN BE USED FROM THE DATA ENTRY
*       UNITS ON THE SYSTEM. CONTROL IS THE NAME OF
*       THE TGROUP MACRO THAT DEFINES THE TRANSACTION
*       CODES THAT CAN BE USED FROM THE AREA STATIONS.
*       FOR EXAMPLE, THE AS MACRO DEFINING THE AREA
*       STATION WHOSE ADDRESS IS DECIMALLY REPRESENTED
*       BY ID=01 INDICATES THAT WORKOUT IS THE NAME
*       OF THE TGROUP MACRO DEFINING WHICH TRANSACTIONS
*       CAN BE USED BY THE 32 DATA ENTRY UNITS ON
*       THIS AREA STATION. THE TRANSACTIONS THAT CAN
*       BE USED BY THE DATA ENTRY UNITS ARE NOT THE
*       SAME AS THOSE THAT CAN BE USED BY THE AREA
*       STATIONS IN THIS TABLE LOAD.
*
*
*

```

AS ID=00,DEGROUP=(WORKOUT,32)
AS ID=01,DEGROUP=(WORKOUT,32)
AS ID=02,DEGROUP=(WORKOUT,4)
AS ID=03,ASGROUP=CONTROL,DEGROUP=(NORMAL,32)
AS ID=04,ASGROUP=CONTROL
AS ID=05,ASGROUP=CONTROL
AS ID=06,ASGROUP=CONTROL
AS ID=07,ASGROUP=CONTROL
AS ID=08,ASGROUP=CONTROL
AS ID=09,ASGROUP=CONTROL
AS ID=10,ASGROUP=CONTROL
AS ID=11,ASGROUP=CONTROL
AS ID=12,ASGROUP=CONTROL
AS ID=13,ASGROUP=CONTROL
AS ID=14,ASGROUP=CONTROL
AS ID=15,ASGROUP=CONTROL
AS ID=16,ASGROUP=CONTROL
AS ID=17,ASGROUP=CONTROL
AS ID=19,ASGROUP=CONTROL
AS ID=20,ASGROUP=CONTROL
AS ID=21,ASGROUP=CONTROL
AS ID=22,ASGROUP=CONTROL
AS ID=23,ASGROUP=CONTROL
AS ID=24,ASGROUP=CONTROL
AS ID=25,ASGROUP=CONTROL
AS ID=26,ASGROUP=CONTROL
AS ID=27,ASGROUP=CONTROL
AS ID=28,ASGROUP=CONTROL
AS ID=29,ASGROUP=CONTROL
AS ID=30,ASGROUP=CONTROL
AS ID=31,ASGROUP=CONTROL
AS ID=32,ASGROUP=CONTROL
AS ID=33,ASGROUP=CONTROL
AS ID=34,ASGROUP=CONTROL
AS ID=35,ASGROUP=CONTROL
AS ID=36,ASGROUP=CONTROL
AS ID=37,ASGROUP=CONTROL
AS ID=38,ASGROUP=CONTROL
AS ID=39,ASGROUP=CONTROL
AS ID=40,DEGROUP=(WORKOUT1,4)
AS ID=41,DEGROUP=(WORKOUT1,4)
AS ID=42,DEGROUP=(WORKOUT1,4)
AS ID=43,DEGROUP=(WORKOUT1,4)
AS ID=44,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS ID=45,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS ID=46,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS ID=47,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS ID=48,DEGROUP=(WORKOUT,4)
AS ID=49,DEGROUP=(WORKOUT,4)
AS ID=50,DEGROUP=(WORKOUT,4)

```

AS      ID=51,DEGROUP=(WORKOUT,4)
AS      ID=52,ASGROUP=CONTROL
AS      ID=53,ASGROUP=CONTROL
AS      ID=54,ASGROUP=CONTROL
AS      ID=55,ASGROUP=CONTROL
AS      ID=56,DEGROUP=(WORKOUT,4)
AS      ID=57,DEGROUP=(WORKOUT,4)
AS      ID=58,DEGROUP=(WORKOUT,4)
AS      ID=59,ASGROUP=CONTROL

```

```

*
*
*
*
*
*
*
*
*
*
*

```

THE FOLLOWING GDUAS MACROS INDICATE THAT THIS SYSTEM HAS FOUR AREA STATIONS WITH 2798 GUIDANCE DISPLAY UNITS ATTACHED. THE GDUNUMB OPERAND INDICATES THE NUMBER OF 2798S ON THAT PARTICULAR AREA STATION. THERE ARE A TOTAL OF 48 2798 GDUS ON THIS SYSTEM.

```

GDUAS   ID=00,GDUNUMB=16
GDUAS   ID=02,GDUNUMB=4
GDUAS   ID=43,GDUNUMB=16
GDUAS   ID=44,GDUNUMB=12

```

```

*
*
*
*
*
*
*
*
*
*
*

```

THE FOLLOWING TGROUP MACROS ASSOCIATE SPECIFIC TRANSACTION CODES WITH USER DEFINED TRANSACTIONS. THE TRANSACTION CODES CAN BE SPECIFIED FROM EITHER DATA ENTRY UNITS OR FROM AREA STATIONS BUT CAN NOT BE SPECIFIED FROM 2798 GUIDANCE DISPLAY UNITS.

```

CONTROL TGROUP TC1=BADGE,TC2=BADGE1,TC3=MANUAL,TC4=CARD,TC5=CARD1, X
          TC6=CARD2,TC7=BADGE,TC9=(EXPAND,E)

```

```

*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

```

THIS TRANSACTION GROUP IS ENTERED BY THE 2715 WHEN THE USER SELECTS TRANSACTION CODES 1-7 OR 9 FROM AREA STATIONS WHOSE ADDRESS IS DECIMALLY REPRESENTED BY IDS FROM 03-39, 44-47, 52-55, OR 59. IF TRANSACTION CODE 1 IS SPECIFIED BY THE USER AT ONE OF THE ABOVE AREA STATIONS, THE TRANSACTION DEFINED BY THE TRLIST MACRO WITH THE NAME BADGE IS ENTERED BY THE 2715. LIKEWISE, THIS TGROUP MACRO ASSOCIATES ALL ALLOWABLE TRANSACTION CODES THAT CAN BE SPECIFIED FROM THE ABOVE AREA STATION WITH A USER DEFINED TRANSACTION BEGINNING WITH A TRLIST MACRO. WHEN TRANSACTION CODE 9 IS SPECIFIED ON ONE OF THE ABOVE AREA STATIONS, THE USER MUST SPECIFY ONE MORE TRANSACTION CODES AS INDICATED BY THE FOLLOWING EXPAND

```

*
*           TGROUP MACRO. TRANSACTION CODES 91,92,93,
*           94,95, AND 96 ARE ASSOCIATED WITH A
*           DIFFERENT TRANSACTION. THIRTEEN
*           DIFFERENT TRANSACTIONS CAN BE SPECIFIED BY
*           OPERATORS ON THE ABOVE AREA STATIONS.
*           CODING E AS IN THE TC9 OPERAND INDICATES
*           THAT THE ADDITIONAL TRANSACTIONS POINTED
*           TO BY THE FOLLOWING EXPAND TGROUP MACRO
*           CAN BE SPECIFIED BY THE OPERATORS ON THE
*           ABOVE AREA STATIONS.
EXPAND    TGROUP TC1=EXP1,TC2=EXP2,TC3=EXP3,TC4=EXP4,TC5=EXP5,TC6=EXP5
NORMAL    TGROUP TC1=DEU1,TC2=DEU2,TC3=DEU3,TC4=DEU4,TC5=(EXPDEU,E),      X
          TC6=DEU6,TC7=ALARM,TC8=ALRMTX,TC9=TEXT
*
*           THIS TRANSACTION GROUP IS ENTERED BY THE
*           2715 WHEN A TRANSACTION CODE OF 1,2,3,4,51,
*           52,53,54,6,7,8, OR 9 IS SPECIFIED AT A DATA
*           ENTRY UNIT ON AN AREA STATION WHOSE ADDRESS
*           IS DECIMALLY REPRESENTED BY ID=03,44,45,46,
*           OR 47.
EXPDEU    TGROUP TC1=DEUEXP,TC2=DEUEXP,TC3=DEUEXP,TC4=DEUEXP
WORKOUT   TGROUP TC1=(ALRMESG,E),TC2=(ALRMESG,E),TC3=EXP,TC4=CPU,      X
          TC5=READ,TC6=READST,TC7=READSID,TC8=DISK,TC9=DISK
WORKOUT1  TGROUP TC1=RDIPSG,TC2=RDEPSGM,TC3=RDEPSGB,TC4=RDEPGPM,      X
          TC5=RDEPGPB,TC6=RDRSTIP,TC7=RDRSTEPH,                          X
          TC8=(EXPAND1,E),TC9=(EXPAND2,E)
EXPAND1   TGROUP TC1=RDSTIPM,TC2=RDSTEPH,TC3=RDSTEPB,TC4=RDSIDEPH,      X
          TC5=RDSIDGPM,TC6=RDSIDGPB,TC7=WRIPB,TC8=WRPEH,TC9=CPU
ALRMESG   TGROUP TC1=ALARM,TC2=ALRMTX,TC3=TEXT,TC4=EXPALM,             X
          TC5=EXPALMTX,TC6=EXPTX,TC7=CPU,TC8=CPU,TC9=CPU
EXPAND2   TGROUP TC1=RDGPEPAA,TC2=RDGPEPBB,TC3=RDSIEPAA,                X
          TC4=RDMSIEP,TC5=RDSIIPAA,TC6=RDSIIPAB

```

```

*
*           THE FOLLOWING ASCTR MACROS DEFINE THE AREA
*           STATIONS WITH PULSE COUNTERS ATTACHED. ONE
*           ASCTR MACRO MUST BE CODED FOR EVERY AREA
*           STATION WITH PULSE COUNTERS ATTACHED.

```

```

*
*           ASCTR    ID=01,HIGHCTR=23,ROUTE=(DISK,LOG)
*
*           THIS ASCTR MACRO INDICATES THAT 23 IS THE
*           HIGHEST COUNTER ON THE AREA STATION WITH
*           ID=01 ON WHICH EITHER COUNT TESTING OR
*           SCHEDULE READOUT FUNCTIONS ARE TO BE
*           PERFORMED BY THE 2715. OVERFLOW AND COUNT
*           TEST RESPONSE MESSAGES WILL BE ROUTED TO
*           THE 2715 DISK AND THE 2740.

```



```

*
* ASCTR ID=40,HIGHCTR=00,ROUTE=CPU,NEXTAS=42
* THIS ASCTR MACRO INDICATES THAT NO COUNTERS
* ON THE AREA STATION WITH ID=40 WILL USE
* COUNT TESTING OR SCHEDULE READOUT FUNCTIONS.
* OVERFLOW MESSAGES WILL BE TREATED AS
* PRIORITY DATA TO BE ROUTED TO THE CPU BY
* 2715. NEXTAS=42 INDICATES THE NEXT AREA
* STATION THAT HAS COUNTERS FOR WHICH COUNT
* TESTING OR SCHEDULE READOUT FUNCTIONS WILL
* BE PERFORMED HAS ID=42.
*
* ASCTR ID=41,HIGHCTR=00,ROUTE=CPU,NEXTAS=42
* ASCTR ID=42,HIGHCTR=2,ROUTE=(CPU,EXTALRM,ASLOG)
* ASCTR ID=48,HIGHCTR=00,ROUTE=CPU,NEXTAS=0
* CODING NEXTAS=0 IN THIS ASCTR MACRO
* INDICATES THAT THERE ARE NO MORE AREA
* STATIONS ON THE 2790 SYSTEM THAT HAVE
* COUNTERS THAT WILL USE COUNT TESTING OR
* SCHEDULE READOUT FUNCTIONS.
*
* ASCTR ID=58,HIGHCTR=00,ROUTE=CPU,NEXTAS=0
*
*
* THE FOLLOWING CTRGROUP MACROS DEFINE EVERY
* COUNTER FOR WHICH COUNT TESTING OR SCHEDULE
* READOUT MAY BE PERFORMED.
*
*
* CTRGROUP 1,1,14,ID=01,SROENAB=YES,CTINIT=NCT
* THIS CTRGROUP MACRO INDICATES THAT THE
* READOUT SCHEDULE USED WILL BE THE FIRST
* SCHEDULE (1 MINUTE) DEFINED BY THE
* CTRSCHEM MACRO FOR COUNTER 1 ON THE AREA
* STATION WITH ID=01. THE COUNT TEST
* SCHEDULE TO BE USED WILL BE THE FOURTEENTH
* SCHEDULE (183 MINUTES) DEFINED BY THE
* CTRSCHEM MACRO. SROENAB=YES INDICATES THAT
* SCHEDULE READOUT WILL BE AUTOMATICALLY
* STARTED AT ICPL TIME AT THE 2715 FOR THIS
* COUNTER. CTINIT=NCT INDICATES THAT NO COUNT
* TESTING WILL BE STARTED AT ICPL TIME BY THE
* 2715 FOR THIS COUNTER.
*
* CTRGROUP 2,2,7,ID=01,SROENAB=YES,CTINIT=NCT
* CTRGROUP 3,3,10,ID=01,SROENAB=YES,CTINIT=UNASP
* CTRGROUP 6,13,8,ID=01,CTINIT=UNASP
* CTRGROUP 7,5,9,ID=01,SROENAB=YES
* CTRGROUP 11,9,11,ID=01,SROENAB=YES,CTINIT=NULL
* CTRGROUP 16,4,8,ID=01,CTINIT=NCT
* CTRGROUP 17,6,1,ID=01,SROENAB=YES,CTINIT=NCT
* CTRGROUP 23,7,12,ID=01,SROENAB=YES,CTINIT=NCT
* CTRGROUP 1,1,0,ID=42,SROENAB=YES
* CTRGROUP 2,0,14,ID=42,CTINIT=NCT

```

CTRSCHED 1,2,3,4,4,3,2,1,3,1,6,90,83,183

THE CTRSCHEd MACRO DEFINES ALL THE SCHEDULES
THAT CAN BE USED FOR SCHEDULE READOUT OR
COUNT TESTING. EACH SCHEDULE IS DEFINED IN
MINUTES. FOURTEEN SCHEDULES ARE DEFINED HERE.

THE FOLLOWING GDUTRANS MACROS ASSOCIATE USER
DEFINED TRANSACTION CODES FOR THE 2798 GDU
WITH DEFINED TRANSACTIONS. THE TRCODE OPERAND
INDICATES THE TRANSACTION CODE. THE TRLIST
OPERAND INDICATES THE NAME OF THE TRLIST MACRO
THAT DEFINES THE CORRESPONDING TRANSACTION.

GDUTRANS TRCODE=00,TRLIST=TEST0
GDUTRANS TRCODE=01,TRLIST=TEST1
GDUTRANS TRCODE=02,TRLIST=TEST2
GDUTRANS TRCODE=03,TRLIST=TEST3
GDUTRANS TRCODE=04,TRLIST=TEST4
GDUTRANS TRCODE=05,TRLIST=TEST5
GDUTRANS TRCODE=06,TRLIST=TEST6
GDUTRANS TRCODE=07,TRLIST=TEST7
GDUTRANS TRCODE=08,TRLIST=TEST8
GDUTRANS TRCODE=10,TRLIST=ROUTE1
GDUTRANS TRCODE=11,TRLIST=ROUTE2
GDUTRANS TRCODE=12,TRLIST=TESTJOB1
GDUTRANS TRCODE=13,TRLIST=TESTJOB2
GDUTRANS TRCODE=14,TRLIST=TESTJOB3
GDUTRANS TRCODE=15,TRLIST=CARDORD
GDUTRANS TRCODE=16,TRLIST=UALMAINT
GDUTRANS TRCODE=19,TRLIST=INV1
GDUTRANS TRCODE=20,TRLIST=INV2
GDUTRANS TRCODE=21,TRLIST=INV3
GDUTRANS TRCODE=22,TRLIST=INV4
GDUTRANS TRCODE=23,TRLIST=INV5
GDUTRANS TRCODE=24,TRLIST=STOCK
GDUTRANS TRCODE=25,TRLIST=INPROC
GDUTRANS TRCODE=26,TRLIST=LEADTIME
GDUTRANS TRCODE=27,TRLIST=RATING
GDUTRANS TRCODE=28,TRLIST=SUPPLIER
GDUTRANS TRCODE=29,TRLIST=INTRANS
GDUTRANS TRCODE=30,TRLIST=LINE
GDUTRANS TRCODE=31,TRLIST=BIN
GDUTRANS TRCODE=32,TRLIST=RAWMAT
GDUTRANS TRCODE=33,TRLIST=ORDER
GDUTRANS TRCODE=34,TRLIST=QUALCON
GDUTRANS TRCODE=35,TRLIST=QUOTE

```

GDUTRANS TRCODE=36,TRLIST=LASTPUR
GDUTRANS TRCODE=37,TRLIST=ECONQTY
GDUTRANS TRCODE=38,TRLIST=CREDIT
GDUTRANS TRCODE=39,TRLIST=QUADEQN

```

```

*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

```

```

THE FOLLOWING PARAMNUM MACROS ASSOCIATE USER
DEFINED PARAMETER LIST NUMBERS WITH PARAMETER
LISTS TO BE USED BY THE 2715 TO CHECK DISPLAY
ENTRIES FROM THE 2798 GDU.
THE PLN OPERAND DEFINES THE PARAMETER LIST
NUMBER AND THE PARMLST OPERAND INDICATES THE
NAME OF THE PARMLST MACRO THAT DEFINES THE
CORRESPONDING PARAMETER LIST. THE USER SPECIFIES
WHICH PARAMETER LIST HE WISHES THE 2715 TO USE
WHEN CHECKING A 2798 DISPLAY ENTRY BY CODING
THE PARAMETER LIST NUMBER OF A DEFINED
PARAMNUM MACRO IN THE PARAMNO OPERAND OF THE
GDULIST MACRO.

```

```

PARAMNUM PLN=01,PARMLST=PAR1
PARAMNUM PLN=02,PARMLST=PAR2
PARAMNUM PLN=03,PARMLST=PAR3
PARAMNUM PLN=04,PARMLST=PAR4
PARAMNUM PLN=05,PARMLST=PAR5
PARAMNUM PLN=06,PARMLST=PAR6
PARAMNUM PLN=07,PARMLST=PAR7
PARAMNUM PLN=08,PARMLST=PAR8
PARAMNUM PLN=09,PARMLST=PAR9
PARAMNUM PLN=10,PARMLST=PAR10
PARAMNUM PLN=11,PARMLST=PAR11
PARAMNUM PLN=12,PARMLST=PAR12
PARAMNUM PLN=13,PARMLST=PAR13
PARAMNUM PLN=14,PARMLST=PAR14
PARAMNUM PLN=15,PARMLST=PAR15
PARAMNUM PLN=16,PARMLST=PAR16
PARAMNUM PLN=17,PARMLST=PAR17
PARAMNUM PLN=18,PARMLST=PAR18
PARAMNUM PLN=19,PARMLST=PAR19
PARAMNUM PLN=20,PARMLST=PAR20
PARAMNUM PLN=21,PARMLST=PAR21
PARAMNUM PLN=22,PARMLST=PAR22
PARAMNUM PLN=23,PARMLST=PAR23
PARAMNUM PLN=24,PARMLST=PAR24
PARAMNUM PLN=25,PARMLST=PAR25
PARAMNUM PLN=26,PARMLST=PAR26
PARAMNUM PLN=27,PARMLST=PAR27

```

PARAMNUM PLN=28,PARMLST=PAR28
 PARAMNUM PLN=29,PARMLST=PAR29
 PARAMNUM PLN=30,PARMLST=PAR30
 PARAMNUM PLN=31,PARMLST=PAR31
 PARAMNUM PLN=32,PARMLST=PAR32
 PARAMNUM PLN=33,PARMLST=PAR33
 PARAMNUM PLN=34,PARMLST=PAR34
 PARAMNUM PLN=35,PARMLST=PAR35
 PARAMNUM PLN=36,PARMLST=PAR36
 PARAMNUM PLN=37,PARMLST=PAR37
 PARAMNUM PLN=38,PARMLST=PAR38
 PARAMNUM PLN=39,PARMLST=PAR39

*
 *
 * THE FOLLOWING PARMLIST MACROS GENERATE THE
 * PARAMETER LISTS TO BE USED BY THE 2715 TO
 * CHECK DISPLAY ENTRIES FROM THE 2798 GDU.
 *

PAR1 PARMLIST CKMONKY=NO,TRANSL=YES
 * THIS MACRO INDICATES THE 2715 WILL NOT
 * CHECK THE DISPLAY ENTRY TO SEE IF THE MONITOR
 * KEY IS ON AND THE TRANSLATE FUNCTION MAY BE
 * USED.
 PAR2 .PARMLIST CKLNGLTH= (17,11)
 * THIS MACRO INDICATES THAT IF THE LENGTH OF A
 * 2798 GDU ENTRY LENGTH IS NOT 17 CHARACTERS
 * LONG (16 DATA CHARACTERS PLUS MONITOR KEY BYTE),
 * THEN OPERATIONAL GUIDANCE LIGHT 11 WILL BE
 * TURNED ON AT THE 2798 TO INDICATE AN ERROR.
 PAR3 PARMLIST CKMONKY=YES,CKMOD11= (15,2,13),IDENT=YES
 * THIS MACRO INDICATES THAT THE 2715 WILL CHECK
 * THE MONITOR KEY BYTE IN THE GDU ENTRY TO INSURE
 * THAT IT WAS ON. IF THE MONITOR KEY IS OFF, THE
 * ERROR GUIDANCE DEFINED BY THE MONERR OPERAND
 * OF THE CONFIGUR MACRO WILL BE DISPLAYED ON THE
 * 2798. THE MODULUS 11 CHECK WILL BE
 * PERFORMED BY THE 2715 IN POSITIONS
 * 15 AND 16. THE CHECK CHARACTER WILL BE IN
 * POSITION 17. IF THE MODULUS 11 VALUE DOES NOT
 * EQUAL THE CHECK CHARACTER THEN OPERATIONAL
 * GUIDANCE LIGHT 13 WILL BE TURNED ON AT THE
 * 2798 TO INDICATE AN ERROR. THE STORE OR GET
 * IDENTIFIER FUNCTION MAY BE USED IN THIS DATA
 * ENTRY.
 PAR4 PARMLIST CKMONKY=YES,CKMOD10= (15,2,13),IDENT=YES

```

PAR5      PARMLIST CKOR=(2,C1,C6,D2,D7,E4),ORGUID=9
*
*          THIS MACRO INDICATES THAN AN OR CHECK WILL BE
*          PERFORMED ON THE CHARACTER IN POSITION 2 OF THE
*          2798 GDU ENTRY FOR ONE OF THE FOLLOWING EBCDIC
*          CHARACTERS: A,F,K,P,OR U. IF THE CHARACTER IN
*          POSITION 2 IS NOT ONE OF THE SPECIFIED
*          CHARACTERS, THEN OPERATIONAL GUIDANCE LIGHT 9
*          WILL BE TURNED ON AT THE 2798 TO INDICATE AN
*          ERROR.
PAR6      PARMLIST CKOR=(3,C2,C7,D3,D8,E5),ORGUID=9
PAR7      PARMLIST CKOR=(4,C3,C8,D4,D9,E6),ORGUID=9
PAR8      PARMLIST CKOR=(5,C4,C9,D5,E2,E7),ORGUID=9
PAR9      PARMLIST CKOR=(6,C5,D1,D6,E3,E8),ORGUID=9
PAR10     PARMLIST CKOR=(7,E9,7B,F1,F6,7C),ORGUID=9
PAR11     PARMLIST CKOR=(8,7F,5A,F2,F7,61),ORGUID=9
PAR12     PARMLIST CKOR=(9,7E,5E,F3,F8,15),ORGUID=9
PAR13     PARMLIST CKOR=(10,7A,5C,F4,F9,25),ORGUID=9
PAR14     PARMLIST CKOR=(11,6F,4E,F5,F0,05),ORGUID=9
PAR15     PARMLIST CKOR=(12,5B,6B),ORGUID=9
PAR16     PARMLIST CKOR=(13,50,4B),ORGUID=9
PAR17     PARMLIST CKOR=(14,60,40),ORGUID=9
PAR18     PARMLIST CKOR=(15,5A),ORGUID=9,
*          CKAND=(2,14,F1,F2,F3,F4,F5,F6,F7,F8,F9,F0,7C,61,15),
*          ANDGUID=16
*
*          THIS MACRO INDICATES THAT AN OR CHECK WILL BE
*          PERFORMED ON THE CHARACTER IN POSITION 15 OF
*          THE 2798 ENTRY FOR AN ] CHARACTER. IF IT IS NOT,
*          OPERATIONAL GUIDANCE LIGHT 9 WILL BE TURNED
*          ON AT THE 2798 TO INDICATE AN ERROR. ALSO, AN
*          AND CHECK IS PERFORMED AND THE CHARACTERS IN
*          POSITIONS 2 THROUGH 14 MUST BE EXACTLY THE
*          FOLLOWING CHARACTERS: 1,2,3,4,5,6,7,8,9,0,@,1,
*          NEW LINE. IF THE AND CHECK IS NOT SATISFIED,
*          OPERATIONAL GUIDANCE LIGHT 16 WILL BE
*          TURNED ON AT THE 2798 TO INDICATE AN ERROR.
PAR19     PARMLIST CKOR=(16,05),ORGUID=9,
*          CKAND=(2,15,7F,7E,7A,6F,5A,5E,5C,4E,5B,50,60,25,6B,4B),
*          ANDGUID=16
PAR20     PARMLIST CKOR=(17,4E),ORGUID=9,
*          CKAND=(2,16,D8,D9,E2,E3,E4,E5,E6,E7,E8,E9,7B,40,6B,4B,
*          05),ANDGUID=16
PAR21     PARMLIST CKAND=(2,17,C1,C2,C3,C4,C5,C6,C7,C8,C9,D1,D2,D3,D4
*          D5,D6,D7),ANDGUID=16
PAR22     PARMLIST CKNUM=(2,17,12)
*
*          THIS MACRO INDICATES THAT CHARACTERS IN
*          POSITIONS 2 THROUGH 17 MUST BE NUMERIC. IF
*          ALL THE CHARACTERS IN THE FIELD ARE NOT
*          NUMERIC, OPERATIONAL GUIDANCE LIGHT 12
*          WILL BE TURNED ON AT THE 2798 TO INDICATE AN
*          ERROR.

```

```

PAR23  PARMLIST CKNONUM=(2,17,10)
*      THIS MACRO INDICATES THAT POSITIONS 2 THROUGH
*      17 WILL BE CHECKED TO INSURE THAT ALL
*      CHARACTERS ARE NON-NUMERIC. IF A NUMERIC
*      CHARACTER IS FOUND IN THE FIELD,
*      OPERATIONAL GUIDANCE LIGHT 10 WILL BE TURNED ON
*      AT THE 2798 TO INDICATE AN ERROR.
PAR24  PARMLIST CKRANGE=(2,17,73,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00), *
      HIGUID=9,LOWGUID=13
PAR25  PARMLIST CKMONKY=YES,CKLENGTH=(8,11),CKNUM=(7,8,12), *
      CKAND=(2,6,5B,00,00,00,4B),ANDGUID=10
PAR26  PARMLIST CKMONKY=YES,CKLENGTH=(17,11),CKMOD11=(6,11,13), *
      IDENT=YES,CKMOD10=(5,2,9),CKOR=(8,5C,60,F0,4E,40, *
      ORGUID=10
PAR27  PARMLIST CKLENGTH=(13,11),CKMOD10=(11,2,13),CKMONKY=YES, *
      IDENT=YES
PAR28  PARMLIST CKNUM=(2,4,12),CKNONUM=(5,6,10),CKRANGE=(7,8,81,81), *
      HIGUID=9,LOWGUID=13
PAR29  PARMLIST CKLENGTH=(3,11),CKOR=(2,C1,C2,C3,C4,C5),ORGUID=10, *
      CKNUM=(3,3,11)
PAR30  PARMLIST CKOR=(2,4E,60),ORGUID=10,CKNUM=(3,5,12)
PAR31  PARMLIST CKLENGTH=(6,11),CKRANGE=(2,6,10,09,55,00,00), *
      HIGUID=9,LOWGUID=13
PAR32  PARMLIST
PAR33  PARMLIST CKLENGTH=(7,11),CKNUM=(2,7,12),IDENT=YES
PAR34  PARMLIST CKLENGTH=(5,11),CKOR=(2,4E,60),ORGUID=10, *
      CKRANGE=(3,5,10,50,00), *
      HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR35  PARMLIST CKLENGTH=(11,11),CKOR=(4,C4,E3),ORGUID=10,CKAND=(2,3, *
      F2,D3),ANDGUID=10,CKRANGE=(5,11,55,88,00,73,80,39,78), *
      HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR36  PARMLIST CKLENGTH=(9,11),CKMOD11=(7,2,13),IDENT=YES
PAR37  PARMLIST CKMONKY=YES,IDENT=YES
PAR38  PARMLIST CKRANGE=(2,3,50,90), *
      HIGUID=9,LOWGUID=13,RNGETST=ERROR
*      THIS MACRO INDICATES THAT A RANGE CHECK WILL BE
*      PERFORMED ON POSITIONS 2 AND 3 TO CHECK THAT
*      THEIR VALUE LIES BETWEEN 00 AND 59. IF THE
*      VALUE OF THE FIELD IS HIGHER THAN 59,
*      OPERATIONAL GUIDANCE LIGHT 9 IS TURNED ON
*      AT THE 2798. IF THE VALUE OF THE FIELD IS
*      LESS THAN 0, OPERATIONAL GUIDANCE LIGHT 13
*      IS TURNED ON AT THE 2798. RNGETST=ERROR
*      INDICATES THAT THE 2715 WILL NOT ACCEPT
*      RANGE TEST BUT TREATS IT AS AN ERROR.
PAR39  PARMLIST CKLENGTH=(11,11),CKNUM=(8,11,12), *
      CKAND=(2,7,F2,F3,F9,F5,F1,P1),ANDGUID=13

```

```

*
* THE FOLLOWING DISPGUID MACROS DEFINE THE
* DISPLAY GUIDANCE MESSAGES THAT CAN BE
* DISPLAYED WHEN A TRANSACTION STEP IS ENTERED.
* THE USER INDICATES WHICH MESSAGE HE WANTS
* DISPLAYED AT THE 2798 FOR A STEP BY CODING
* THE NAME OF A DISPGUID MACRO IN THE DISPMSG
* OPERAND OF A GDULIST MACRO. CODING SUPPRES=NO
* IN ANY OF THE FOLLOWING DISPGUID MACROS
* INDICATES THAT WHENEVER THE DEFINED DATA IN
* THE PARTICULAR MACRO IS WRITTEN TO THE 2798
* DISPLAY BY THE 2715, THAT DATA WILL BE RETURNED
* TO THE 2715 ON THE NEXT ACTIVATION OF THE
* ENTER KEY UNLESS IT HAS BEEN CHANGED BY THE
* OPERATOR. CODING SUPPRES=YES OR OMITTING THE
* OPERAND INDICATES THAT WHENEVER THE DEFINED
* DATA IN THE PARTICULAR DISPGUID MACRO IS
* WRITTEN TO THE 2798 DISPLAY BY THE 2715, THAT
* DATA WILL NOT BE RETURNED TO THE 2715 ON THE
* NEXT ACTIVATION OF THE ENTER KEY.
*
*
DG1      DISPGUID DISPMSG='2=1ENTR TESTDATA'
DG2      DISPGUID DISPMSG='DEPRESS ENTER',SUPPRES=NO
DG3      DISPGUID DISPMSG='STEP 2'
DG4      DISPGUID DISPMSG='STEP 3'
DG5      DISPGUID DISPMSG='STEP 4'
DG6      DISPGUID DISPMSG='STEP 5'
DG7      DISPGUID DISPMSG='GET/STORE'
DG8      DISPGUID DISPMSG='3-1ENTR SERVCODE'
DG9      DISPGUID DISPMSG='3-2BLDG/COLUMN'
DG10     DISPGUID DISPMSG='MAT 1-1'
DG11     DISPGUID DISPMSG='MAT 2-2'
DG12     DISPGUID DISPMSG='SELECT LEVR TO 3'
DG14     DISPGUID DISPMSG='MAT 1-2      SL',SUPPRES=NO
DG15     DISPGUID DISPMSG='OLD PART'
DG16     DISPGUID DISPMSG='NEW PART'
DG17     DISPGUID DISPMSG='TRANSLATE'
DG18     DISPGUID DISPMSG='ENTER TEXT'
DG19     DISPGUID DISPMSG='LOCATE20-ORDER21'
DG20     DISPGUID DISPMSG='STOCK24-INPROC25'
DG21     DISPGUID DISPMSG='PRICE22-OTHER23'
DG22     DISPGUID DISPMSG='QUO35-LP36-QTY37'
DG23     DISPGUID DISPMSG='LT26-RAT27-SUP28'
DG24     DISPGUID DISPMSG='IT29-LIN30-BIN31'
DG25     DISPGUID DISPMSG='RM32-ORD33-QC34'
DG26     DISPGUID DISPMSG='WAIT FOR ANSWER'
DG27     DISPGUID DISPMSG='239511',SUPPRES=NO

```

DG28 DISPGUID DISPMSG='TOTAL PURCHASE'
 DG29 DISPGUID DISPMSG='3-2 TO ADR='
 DG37 DISPGUID DISPMSG='QUAD EQN A='
 DG38 DISPGUID DISPMSG='B='
 DG39 DISPGUID DISPMSG='C='

*
 *
 * THE FOLLOWING TRANSLAT MACROS EACH ASSOCIATE
 * A USER DEFINED TRANSLATE CHARACTER WITH UP TO
 * 14 CHARACTERS OF TEXT. THE USER CAN ONLY USE
 * THE TRANSLATE FUNCTION ON ANY TRANSACTION
 * STEP (GDULIST MACRO) THAT HAS A PARAMETER LIST
 * NUMBER (PARAMNO OPERAND) ASSOCIATED WITH A
 * A PARMLIST MACRO THAT HAS TRANSL=YES CODED.
 *

TRANSLAT TRANSCH=C3,TRANTXT='CE'
 TRANSLAT TRANSCH=C4,TRANTXT='DOCTOR'
 TRANSLAT TRANSCH=C6,TRANTXT='FIRE'
 TRANSLAT TRANSCH=C9,TRANTXT='IBM MAINT'
 TRANSLAT TRANSCH=D4,TRANTXT='MOVER REQUIRED'
 TRANSLAT TRANSCH=D9,TRANTXT=';N'
 TRANSLAT TRANSCH=E3,TRANTXT='TEL REPAIR'
 TRANSLAT TRANSCH=E5,TRANTXT='VENDING MACH'

* THE FOLLOWING MACROS DEFINE THE USER
 * TRANSACTIONS. EACH TRANSACTION BEGINS WITH A
 * TRLIST MACRO WHICH GENERATES THE TRANSACTION
 * LIST HEADER AND CONTAINS FROM 1 TO 16
 * MACROS: ASLIST, DEULIST, CTRLIST, GDULIST. FROM 1
 * TO 160 TRANSACTIONS MAY BE SPECIFIED BY THE
 * USER WITH TRID VALUES BETWEEN 0 AND 159.
 *

CPU TRLIST ROUTE=CPU,TRID=0
 DEULIST

* THE CPU TRANSACTION CONSISTS OF 1 STEP AND
 * WILL BE ROUTED TO THE CPU.

BADGE TRLIST ROUTE=(LOG),TRID=1
 ASLIST B,NORM=19

* THE BADGE TRANSACTION CONSISTS OF 1 STEP AND
 * WILL BE ROUTED TO THE 2740 ATTACHED TO THE 2715.
 * THE DATA ENTRY WILL BE A BADGE ENTERED AT THE
 * AREA STATION WITH GUIDANCE LIGHT 19 TURNED ON
 * WHEN THE TRANSACTION STEP IS ENTERED.

BADGE1 TRLIST ROUTE=(CPU,LOG),TRID=2,DEM0D10=YES,INQDISP=YES
 ASLIST B,NORM=31,MODULUS=(2,10,4),LENGTH=(11,2),INQDISP=7

* THE BADGE1 TRANSACTION CONSISTS OF 1 STEP AND
 * WILL BE ROUTED TO BOTH THE CPU AND THE 2740. A
 * MODULUS 10 CHECK WILL BE PERFORMED ON POSITIONS

* 2 THROUGH 10 AND WILL BE CHECKED WITH THE SELF-
 * CHECK CHARACTER IN POSITION 11. IF THE MODULUS
 * 10 CHECK FAILS, GUIDANCE LIGHT 4 WILL BE TURNED
 * ON. THE DATA ENTRY WILL BE A BADGE ENTERED AT
 * THE AREA STATION WITH GUIDANCE LIGHT 4 TURNED
 * ON WHEN THE TRANSACTION STEP IS ENTERED. THE
 * LENGTH OF THE DATA ENTRY WILL ALSO BE CHECKED
 * AND IF THE LENGTH IS NOT 11, GUIDANCE LIGHT 2
 * WILL BE TURNED ON. THIS TRANSACTION IS ALSO AN
 * INQUIRY DISPLAY TRANSACTION. GUIDANCE LIGHT 7 ON
 * THE AREA STATION WILL BE TURNED ON WHEN THIS
 * TRANSACTION IS RECEIVED BY THE 2715 AND ROUTED
 * TO THE CPU AS PRIORITY DATA. THIS IS REALLY THE
 * INQUIRY-IN-PROCESS GUIDANCE LIGHT. IF THE
 * INQUIRY IS ABORTED BY THE OPERATOR AT THE 2791
 * AREA STATION, GUIDANCE LIGHT 1 WILL BE TURNED ON
 * AUTOMATICALLY. ALL AREA STATIONS THAT USE
 * INQUIRY DISPLAY TRANSACTIONS MUST RESERVE
 * GUIDANCE LIGHT 1 FOR THE INQUIRY ABORT
 * SITUATION.

MANUAL TRLIST ROUTE=(DISK,LOG),TRID=3
 ASLIST M,NORM=27,LENGTH=(5,23)
 ASLIST M,NORM=26,LENGTH=(5,22)
 ASLIST M,NORM=25,LENGTH=(7,21),ENTRY=M

* THE MANUAL TRANSACTION CONSISTS OF 3 STEPS
 * AND WILL BE ROUTED TO THE 2715 DISK AS DEFERRED
 * DATA AND TO THE 2740. MANUAL DATA ENTRIES WILL
 * BE MADE FOR ALL 3 STEPS. FOR THE FIRST STEP,
 * GUIDANCE LIGHT 27 WILL BE TURNED ON WHEN THE
 * STEP IS ENTERED AND GUIDANCE LIGHT 23 WILL BE
 * TURNED ON IF THE DATA ENTRY LENGTH IS NOT 5.
 * FOR THE SECOND STEP, GUIDANCE LIGHT 26 WILL BE
 * TURNED ON WHEN THE STEP IS ENTERED AND GUIDANCE
 * LIGHT 22 WILL BE TURNED ON IF THE DATA ENTRY
 * LENGTH IS NOT 5. FOR THE THIRD STEP, GUIDANCE
 * LIGHT 25 WILL BE TURNED ON WHEN THE STEP IS
 * ENTERED AND GUIDANCE LIGHT 21 WILL BE TURNED ON
 * IF THE DATA LENGTH IS NOT 7. THE THIRD STEP IS
 * A MULTIPLE ENTRY STEP SO THAT 7 CHARACTERS CAN
 * BE ENTERED.

CARD TRLIST ROUTE=(LOG),TRID=4
 ASLIST C,NORM=17

CARD1 TRLIST ROUTE=(DISK,LOG),TRID=5
 ASLIST C,NORM=30,DIGIT=(2,1,10),LENGTH=(47,11)
 ASLIST M,NORM=18,LENGTH=(6,11)

* THE CARD1 TRANSACTION CONSISTS OF 2 STEPS AND
 * WILL BE ROUTED TO THE 2715 DISK AND TO THE

```

*           2740. THE FIRST STEP WILL BE A CARD ENTRY WITH
*           GUIDANCE LIGHT 30 TURNED ON WHEN THE STEP IS
*           ENTERED, IF THE CHARACTER IN POSITION 2 OF THE
*           DATA ENTRY IS NOT THE EBCDIC CHARACTER F1, THEN
*           GUIDANCE LIGHT 10 IS TURNED ON. IF THE LENGTH OF
*           THE CARD ENTRY IS NOT 47, THEN GUIDANCE LIGHT 11
*           IS TURNED ON. THE SECOND STEP WILL BE A MANUAL
*           ENTRY WITH GUIDANCE LIGHT 18 TURNED ON WHEN THE
*           STEP IS ENTERED. IF THE LENGTH IS NOT 6, THEN
*           GUIDANCE LIGHT 11 IS TURNED ON.
CARD2      TRLIST ROUTE=(CPU,LOG),TRID=6,DEM0D11=YES,INQDISP=YES
          ASLIST C,NORM=29,MODULUS=(2,15,5),LENGTH=(17,11)
          ASLIST B,NORM=20,LENGTH=(11,2)
          ASLIST M,NORM=15,INQDISP=7
EXP1      TRLIST ROUTE=(CPU,LOG),TRID=7,DEM0D10=YES,INQDISP=YES
          ASLIST B,NORM=16,MODULUS=(2,9,4),INQDISP=7
EXP2      TRLIST ROUTE=(CPU,LOG),TRID=8,DEM0D11=YES
          ASLIST B,NORM=16,MODULUS=(2,9,5)
*           THE EXP2 TRANSACTION CONSISTS OF 1 STEP AND
*           WILL BE ROUTED TO THE CPU AND TO THE 2740. THE
*           DATA ENTRY WILL BE A BADGE ENTERED AT THE AREA
*           STATION WITH GUIDANCE LIGHT 16 TURNED ON WHEN
*           THE TRANSACTION STEP IS ENTERED. A MODULUS 11
*           CHECK WILL BE PERFORMED ON POSITIONS 2 THROUGH
*           10 AND WILL BE CHECKED WITH THE SELF-CHECK
*           CHARACTER IN POSITION 11. IF THE MODULUS 11
*           CHECK FAILS, GUIDANCE LIGHT 5 WILL BE TURNED ON.
EXP3      TRLIST ROUTE=(CPU,LOG),TRID=9,DEM0D10=YES,INQDISP=YES
          ASLIST B,NORM=16,MODULUS=(2,7,4)
          ASLIST C,NORM=17,INQDISP=7,ENTRY=M
EXP4      TRLIST ROUTE=(LOG),TRID=10
          ASLIST B,NORM=16,DIGIT=(3,5,10),LENGTH=(11,11)
*           THE EXP4 TRANSACTION CONSISTS OF 1 STEP AND
*           WILL BE ROUTED TO THE 2740. THE STEP WILL BE A
*           BADGE ENTRY WITH GUIDANCE LIGHT 16 TURNED ON
*           WHEN THE STEP IS ENTERED. GUIDANCE LIGHT 10 WILL
*           BE TURNED ON BY THE 2715 IF THE CHARACTER IN
*           POSITION 3 IS NOT THE EBCDIC CHARACTER F5.
*           GUIDANCE LIGHT 11 WILL BE TURNED ON IF THE
*           LENGTH OF THE DATA ENTRY IS NOT 11.
EXP5      TRLIST ROUTE=(LOG),TRID=11
          ASLIST B,NORM=16
DEU1      TRLIST ROUTE=LOG,TRID=12
          DEULIST DIGIT=(2,1),DIGIT2=(3,1)
*           THE DEU1 TRANSACTION CONSISTS OF 1 STEP AND
*           WILL BE ROUTED TO THE 2740. THE DATA ENTRY WILL
*           BE MADE FROM A DATA ENTRY UNIT. AN ERROR WILL BE
*           INDICATED AT THE DEU IF POSITION 2 DOES NOT
*           CONTAIN THE EBCDIC CHARACTER F1 OR IF POSITION
*           3 DOES NOT CONTAIN THE EBCDIC CHARACTER F1.

```

```

DEU2      TRLIST ROUTE=(LOG),TRID=13,DEM0D10=YES
          DEULIST DIGIT2=(2,1),MODULUS=(3,10)
DEU3      TRLIST ROUTE=(LOG),TRID=14,DEM0D11=YES
          DEULIST DIGIT2=(2,5),MODULUS=(3,10),LENGTH=13
DEU4      TRLIST ROUTE=(LOG),TRID=15
          DEULIST DIGIT=(2,6),DIGIT2=(3,9) LENGTH=11
DEUEXP    TRLIST ROUTE=(LOG,NULL),DEM0D10=YES,TRID=16,TEXT=YES
          DEULIST MODULUS=(3,10),MSG=' THIS IS AN EXPANDED TRANSACTION'
*          THE DEUEXP TRANSACTION CONSISTS OF 1 STEP AND
*          WILL BE ROUTED TO THE 2740 AND TO THE PRINTER
*          ATTACHED TO THE AREA STATION THAT WILL BE
*          SPECIFIED BY THE OPERATOR IN THE FIRST DATA
*          ENTRY. THE DATA ENTRY WILL BE ENTERED FROM A DEU
*          AND A DEFINED MESSAGE (IMPLICIT TEXT) WILL BE
*          INCLUDED WITH THE TRANSACTION. A MODULUS 10
*          CHECK WILL BE PERFORMED ON POSITIONS 3 THROUGH
*          10 AND WILL BE COMPARED WITH THE CHECK
*          CHARACTER IN POSITION 11. IF THE MODULUS 10
*          CHECK FAILS, THEN THE RED ERBOR BUTT0N WILL BE
*          INDICATED AT THE DEU.
DEU6      TRLIST ROUTE=(LOG),TRID=17
          DEULIST
EXPALM    TRLIST ROUTE=(CPU,59),TEXT=YES,TRID=18
          DEULIST MSG=' ' VENI VIDI VICI AT TWO PRINTERS, I HOPE'
*          THE EXPALM TRANSACTION CONSISTS OF 1 STEP AND
*          WILL BE ROUTED TO THE CPU AND TO THE PRINTER
*          ON THE AREA STATION WHOSE ID IS 59. THE DATA
*          ENTRY WILL BE ENTERED FROM A DEU AND A DEFINED
*          MESSAGE WILL BE ROUTED AL0NG WITH THE
*          TRANSACTION.
EXPALMTX  TRLIST ROUTE=(CPU,59),TEXT=YES,TRID=19
          DEULIST MSG=' ' TYPE AT TWO PRINTER AND NO ALARM'
EXPTX    TRLIST ROUTE=42,TRID=20
          DEULIST
          CTRLIST DEVC0D=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
          CTROP=READ
RDIPSG   TRLIST ROUTE=LOG,TRID=21
          DEULIST
          CTRLIST DEVC0D=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
          CTROP=READ
*          THE RDIPSG TRANSACTION CONSISTS OF 2 STEPS AND
*          WILL BE ROUTED TO THE 2740. THE FIRST STEP IS A
*          DATA ENTRY FROM A DEU. THE SECOND STEP IS THE
*          PULSE COUNT DATA ENTRY. THIS STEP WILL CAUSE THE
*          SINGLE COUNTER WHOSE IMPLIED ADDRESS RESULTS
*          FROM THE CONVERSION OF THE DEVICE ADDRESS OF
*          THE DEU INITIATING THE REQUEST. THERE WILL BE
*          NO CHANGE IN THE PRESENT COUNT TEST CONDITION
*          OF THE COUNTER.

```

```

RDEPSGM      TRLIST  ROUTE=LOG,TRID=22
              DEULIST
              CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                  CTROP=READ
*              THE RDEPSGM TRANSACTION CONSISTS OF 2 STEPS
*              AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*              IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*              THE PULSE COUNT DATA ENTRY SET UP TO READ THE
*              SINGLE COUNTER WHOSE ADDRESS IS EXPLICITLY
*              SPECIFIED IN THE MANUAL DATA ENTRY. THERE WILL
*              BE NO CHANGE IN THE PRESENT COUNT TEST
*              CONDITION OF THE COUNTER.
RDEPSGB      TRLIST  ROUTE=LOG,TRID=23
              DEULIST
              CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                  CTROP=READ
RDEPGPM      TRLIST  ROUTE=LOG,TRID=24
              DEULIST
              CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,    X
                  CTROP=READ
RDEPGPB      TRLIST  ROUTE=LOG,TRID=25
              DEULIST
              CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,    X
                  CTROP=READ
*              THE RDEPGPB TRANSACTION CONSISTS OF 2 STEPS
*              AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*              IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*              THE PULSE COUNT DATA ENTRY SET UP TO READ THE
*              GROUP OF COUNTERS THAT WILL BE EXPLICITLY
*              SPECIFIED IN THE BADGE DATA ENTRY. THERE WILL
*              BE NO CHANGE IN THE PRESENT COUNT TEST
*              CONDITIONS OF ANY OF THE COUNTERS.
RDRSTIP      TRLIST  ROUTE=LOG,TRID=26
              DEULIST
              CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
                  CTROP=READRST
*              THE RDRSTIP TRANSACTION CONSISTS OF 2 STEPS
*              AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*              IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*              THE PULSE COUNT DATA ENTRY. THE SINGLE COUNTER,
*              WHOSE ADDRESS IS IMPLIED FROM THE CONVERSION
*              OF THE DEVICE ADDRESS OF THE DEU INITIATING
*              THE REQUEST, WILL BE READ AND THEN THAT COUNTER
*              WILL BE RESET. THERE WILL BE NO CHANGE IN THE
*              PRESENT COUNT TEST CONDITION OF THE COUNTER.

```

```

RDRSTEPM      TRLIST ROUTE=LOG,TRID=27
               DEULIST
               CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=READRST
*** FIRST SET OF EXPANSION TRANSACTIONS FOR 2796 TC81-TC89 ***
*****
RDSTIPM       TRLIST ROUTE=(LOG,42),TRID=28
               DEULIST
               CTRLIST DEVCOD=M,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=READSET
RDSTEPM       TRLIST ROUTE=(LOG,42),TRID=29
               DEULIST
               CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=READSET
*
*             THE RDSTEPM TRANSACTION CONSISTS OF 2 STEPS
*             AND WILL BE ROUTED TO THE 2740 AND TO THE
*             PRINTER ON THE AREA STATION WHOSE ID IS 42.
*             THE FIRST STEP IS A DATA ENTRY FROM A DEU.
*             THE SECOND STEP IS THE PULSE COUNT DATA ENTRY
*             SET UP TO READ THE SINGLE COUNTER, WHOSE ADDRESS
*             IS EXPLICITLY SPECIFIED IN THE MANUAL DATA
*             ENTRY, AND THEN TO SET THE COUNTER TO THE
*             EXPLICITLY SPECIFIED VALUE. THERE WILL BE NO
*             CHANGE IN THE PRESENT COUNT TEST CONDITION OF
*             THE COUNTER.
RDSTEPB       TRLIST ROUTE=(LOG,42),TRID=30
               DEULIST
               CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=READSET
RDSIDEPH      TRLIST ROUTE=(LOG,42),TRID=31
               DEULIST
               CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=RDRESID
RDSIDGPM      TRLIST ROUTE=(LOG,42),TRID=32
               DEULIST
               CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,   X
               CTROP=RDRESID
RDSIDGPB      TRLIST ROUTE=(LOG,42),TRID=33
               DEULIST
               CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,   X
               CTROP=RDRESID
WRIPB         TRLIST ROUTE=LOG,TRID=34
               DEULIST
               CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,   X
               CTROP=SET

```

```

WREPM      TRLIST ROUTE=LOG,TRID=35
           DEULIST
           CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=SET
*           THE WREPM TRANSACTION CONSISTS OF 2 STEPS AND
*           WILL BE ROUTED TO THE 2740. THE FIRST STEP IS
*           THE PULSE COUNT DATA ENTRY SET UP TO SET THE
*           SINGLE COUNTER, WHOSE ADDRESS IS EXPLICITLY
*           SPECIFIED IN THE MANUAL ENTRY, TO THE MANUAL
*           VALUE SPECIFIED AT THE DEU. THERE WILL BE NO
*           CHANGE IN THE PRESENT COUNT TEST CONDITION OF
*           THE COUNTER.
RDGPEPAA   TRLIST ROUTE=42,TRID=36
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,    X
             CTROP=READ
RDGPEPBB   TRLIST ROUTE=59,TRID=37
           DEULIST
           CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,    X
             CTROP=READ
RDSIEPAA   TRLIST ROUTE=42,TEXT=YES,TRID=38
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ,MSG='A SINGLE COUNTER SHOULD ACCOMPANY
             THIS MESSAGE'
RDMSIEP    TRLIST ROUTE=42,TEXT=YES,TRID=39
           DEULIST
           CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ,MSG='A SINGLE COUNTER SHOULD ACCOMPANY
             THIS MESSAGE'
RDSIIPAA   TRLIST ROUTE=42,TEXT=YES,TRID=40
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ,MSG='''THIS IS AN ALARM MESSAGE WITH
             A COUNTER'
RDSIIPAB   TRLIST ROUTE=59,TEXT=YES,TRID=41
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ,MSG-'''ALARM MESSAGE AT TWO AREA STATIONS
             AND A COUNTER VALUE AT ONE'
EXP        TRLIST ROUTE=LOG,TRID=42
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ
READ      TRLIST ROUTE=LOG,TRID=43
           DEULIST
           CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
             CTROP=READ

```

```

READST      TRLIST ROUTE=LOG,TRID=44
            DEULIST
            CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
            CTROP=READSET
READSID     TRLIST ROUTE=LOG,TRID=45
            DEULIST
            CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
            CTROP=RDRESID
ALARM       TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=46
            DEULIST MSG=' '
ALARMTX     TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=47
            DEULIST MSG=' ' VENI VIDI VICI '
TEXT        TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=48
            DEULIST MSG=' ' THE ALARM HAD BETTER NOT HAVE SOUNDED '
DISK        TRLIST ROUTE=DISK,TRID=49
            DEULIST
TESTO       TRLIST TRID=60,ROUTE=LOG
            GDULIST PARAMNO=01,NORGUID=1,DISPMSG=DG17
            GDULIST PARAMNO=02,NORGUID=1,DISPMSG=DG1
            GDULIST PARAMNO=33,NORGUID=(1,3,5),IDENT=4
*           THE TESTO TRANSACTION CONSISTS OF 3 STEPS AND
*           WILL BE ROUTED TO THE 2740. ALL 3 STEPS WILL
*           BE DATA ENTRIES FROM THE 2798. WHEN THE FIRST IS
*           ENTERED, GUIDANCE LIGHT 1 IS TURNED ON AT THE
*           OPERATOR GUIDANCE PANEL AND THE MESSAGE DEFINED
*           BY DISPGUID MACRO DG17 WILL BE DISPLAYED ON THE
*           2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL USE
*           PARAMETER LIST NUMBER 01 TO GET TO THE
*           PARAMETER LIST DEFINED BY PARMLIST MACRO, PAR1.
*           THIS PARAMETER LIST WILL BE USED IN CHECKING
*           THE FIRST DATA ENTRY. WHEN THE SECOND STEP IS
*           ENTERED, GUIDANCE LIGHT 1 IS TURNED ON AT THE
*           OPERATOR GUIDANCE PANEL AND THE MESSAGE DEFINED
*           BY DISPGUID MACRO DG1 WILL BE DISPLAYED ON THE
*           2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL USE
*           PARAMETER LIST NUMBER 02 TO GET TO THE
*           PARAMETER LIST DEFINED BY THE PARMLIST MACRO
*           PAR2. THIS PARAMETER LIST WILL BE USED IN
*           CHECKING THE SECOND DATA ENTRY. WHEN THE THIRD
*           STEP IS ENTERED, GUIDANCE LIGHTS 1,3,AND 5 WILL
*           BE TURNED ON AT THE OPERATOR GUIDANCE PANEL AND
*           THE MESSAGE DEFINED IN THE FIFTH IDENTIFIER
*           IN THE IDENTIFIER TABLE WILL BE DISPLAYED ON
*           THE 2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL
*           USE PARAMETER LIST NUMBER 33 TO GET TO THE
*           PARAMETER LIST DEFINED BY THE PARMLIST MACRO
*           PAR33. THIS PARAMETER LIST WILL BE USED IN
*           CHECKING THE THIRD DATA ENTRY.

```

```

TEST1      TRLIST TRID=61,ROUTE=LOG
           GDULIST PARAMNO=03,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=04,NORGUID=1,DISPMSG=DG3
TEST2      TRLIST TRID=62,ROUTE=LOG
           GDULIST PARAMNO=05,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=06,NORGUID=1,DISPMSG=DG3
           GDULIST PARAMNO=07,NORGUID=1,DISPMSG=DG4
           GDULIST PARAMNO=08,NORGUID=1,DISPMSG=DG5
           GDULIST PARAMNO=09,NORGUID=1,DISPMSG=DG6
TEST3      TRLIST TRID=63,ROUTE=LOG
           GDULIST PARAMNO=10,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=11,NORGUID=1,DISPMSG=DG3
           GDULIST PARAMNO=12,NORGUID=1,DISPMSG=DG4
           GDULIST PARAMNO=13,NORGUID=1,DISPMSG=DG5
           GDULIST PARAMNO=14,NORGUID=1,DISPMSG=DG6
TEST4      TRLIST TRID=64,ROUTE=LOG
           GDULIST PARAMNO=15,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=16,NORGUID=1,DISPMSG=DG3
           GDULIST PARAMNO=17,NORGUID=1,DISPMSG=DG4
TEST5      TRLIST TRID=65,ROUTE=LOG
           GDULIST PARAMNO=18,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=19,NORGUID=1,DISPMSG=DG3
           GDULIST PARAMNO=20,NORGUID=1,DISPMSG=DG4
           GDULIST PARAMNO=21,NORGUID=1,DISPMSG=DG5
TEST6      TRLIST TRID=66,ROUTE=LOG
           GDULIST PARAMNO=22,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=23,NORGUID=1,DISPMSG=DG3
TEST7      TRLIST TRID=67,ROUTE=LOG
           GDULIST PARAMNO=24,NORGUID=1,DISPMSG=DG11
           GDULIST PARAMNO=38,NORGUID=1,DISPMSG=DG3
TEST8      TRLIST TRID=68,ROUTE=LOG
           GDULIST PARAMNO=26,NORGUID=1,DISPMSG=DG1
           GDULIST PARAMNO=37,NORGUID=1,DISPMSG=DG7
ROUTE1    TRLIST TRID=70,ROUTE=(LOG,42),TEXT=YES
           GDULIST PARAMNO=28,NORGUID=(1,5),DISPMSG=DG9
           GDULIST PARAMNO=02,NORGUID=1,DISPMSG=DG8,ENTRY=M,      *
           MSG='**EMERGENCY'
*
*           THE ROUTE1 TRANSACTION CONSISTS OF 2 STEPS AND
*           WILL BE ROUTED TO THE 2740 AND TO THE PRINTER
*           ON THE AREA STATION WHOSE ID IS 42. BOTH STEPS
*           WILL BE DATA ENTRIES FROM THE 2798. WHEN THE
*           FIRST STEP IS ENTERED, GUIDANCE LIGHTS 1 AND 5
*           ARE TURNED ON AT THE OPERATOR GUIDANCE PANEL
*           AND THE MESSAGE DEFINED BY DISPGUID MACRO DG9
*           WILL BE DISPLAYED ON THE 2798 GUIDANCE DISPLAY
*           PANEL. THE 2715 WILL USE PARAMETER LIST NUMBER
*           28 TO GET TO THE PARAMETER LIST DEFINED BY THE
*           PARMLIST MACRO PAR28. THIS PARAMETER LIST WILL
*           BE USED BY THE 2715 IN CHECKING THE FIRST DATA

```



```

*          ENTRY. WHEN THE SECOND STEP IS ENTERED, GUIDANCE
*          LIGHT 1 IS TURNED ON AT THE OPERATOR GUIDANCE
*          PANEL AND THE MESSAGE DEFINED BY DISPGUID MACRO
*          DG8 WILL BE DISPLAYED ON THE 2798 GUIDANCE
*          DISPLAY PANEL. THE 2715 WILL USE PARAMETER LIST
*          NUMBER 02 TO GET TO THE PARAMETER LIST DEFINED
*          BY THE PARMLIST MACRO PAR2. THIS PARAMETER LIST
*          WILL BE USED BY THE 2715 IN CHECKING THE
*          SECOND DATA ENTRY. MULTIPLE ENTRIES CAN BE
*          ENTERED ON THE SECOND STEP. IMPLICIT TEXT WILL
*          BE INCLUDED WITH THE TRANSACTION WHEN IT IS
*          ROUTED.
ROUTE2      TRLIST TRID=71,ROUTE=(LOG,NULL)
            GDULIST PARAMNO=38,NORGUID=1,DISPMSG=DG29
            GDULIST PARAMNO=28,NORGUID=(1,5),DISPMSG=DG9
            GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG18
TESTJOB1    TRLIST TRID=72,ROUTE=LOG
            GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
            GDULIST PARAMNO=33,NORGUID=4
            GDULIST PARAMNO=31,NORGUID=5
TESTJOB2    TRLIST TRID=73,ROUTE=LOG
            GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
            GDULIST PARAMNO=32,NORGUID=3
            GDULIST PARAMNO=33,NORGUID=6
            GDULIST PARAMNO=33,NORGUID=(1,7),DISPMSG=DG11
            GDULIST PARAMNO=33,NORGUID=(1,8),DISPMSG=DG11
TESTJOB3    TRLIST TRID=74,ROUTE=LOG
            GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
            GDULIST PARAMNO=33,NORGUID=4
            GDULIST PARAMNO=34,NORGUID=5
CARDORD     TRLIST TRID=75,ROUTE=LOG
            GDULIST PARAMNO=35,NORGUID=(1,4),DISPMSG=DG14
            GDULIST PARAMNO=33,NORGUID=7
UALMAINT    TRLIST TRID=76,ROUTE=LOG
            GDULIST PARAMNO=33,NORGUID=(1,2),DISPMSG=DG10
            GDULIST PARAMNO=33,NORGUID=3,IDENT=4
            GDULIST PARAMNO=36,NORGUID=4
            GDULIST PARAMNO=33,NORGUID=(1,5),DISPMSG=DG15
            GDULIST PARAMNO=33,NORGUID=(1,5),DISPMSG=DG16
INV1        TRLIST TRID=79,ROUTE=LOG
            GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG19
INV2        TRLIST TRID=80,ROUTE=LOG
            GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG20
INV3        TRLIST TRID=81,ROUTE=LOG
            GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG21
INV4        TRLIST TRID=82,ROUTE=LOG
            GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG22
INV5        TRLIST TRID=83,ROUTE=LOG
            GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG23

```

STOCK TRLIST TRID=84,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4,8),DISPMSG=DG24
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 * IF THE INQUIRY IS ABORTED BY THE OPERATOR AT
 * THE 2798, GDU GUIDANCE LIGHT 16 WILL BE TURNED
 * ON AUTOMATICALLY AT THE 2798. ALL 2798'S THAT
 * USE INQUIRY DISPLAY TRANSACTIONS MUST RESERVE
 * GUIDANCE LIGHT 16 FOR THE INQUIRY ABORT
 * SITUATION.
 INPROC TRLIST TRID=85,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4,8),DISPMSG=DG25
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 LEADTIME TRLIST TRID=86,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 RATING TRLIST TRID=87,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 SUPPLIER TRLIST TRID=88,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 INTRANS TRLIST TRID=89,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 LINE TRLIST TRID=90,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 BIN TRLIST TRID=91,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 RAWMAT TRLIST TRID=92,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 ORDER TRLIST TRID=93,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 QUALCON TRLIST TRID=94,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 QUOTE TRLIST TRID=95,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 LASTPUR TRLIST TRID=96,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 ECONQTY TRLIST TRID=97,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 CREDIT TRLIST TRID=98,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=27,NORGUID=(1,2),DISPMSG=DG12
 GDULIST PARAMNO=25,NORGUID=1,DISPMSG=DG28
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 QUADQON TRLIST TRID=99,ROUTE=CPU,INQDISP=YES
 GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG37
 GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG38
 GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG39
 GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
 STEND

END
 /*

APPENDIX N: SAMPLE START-STOP AND BSC PROGRAMS

```

LCC OBJECT CODE  ADDR1 ADDR2  STMT  SOURCE STATEMENT  F3CSEP69  1C/C6/70
000000          1  EXAMPLE2 CSECT
2  *****
3  *
4  *          SAMPLE EXPANDED ID VERIFICATION PROGRAM
5  *
6  * THIS PROGRAM IS DESIGNED FOR A BINARY SYNCHRONOUS SWITCHED LINE
7  * OVER WHICH AN IBM 2770 OR 2790 CAN CALL THE CENTRAL COMPUTER.
8  * THE PROGRAM UTILIZES SOME OF THE EXPANDED ID VERIFICATION CAPABILI-
9  * TIES. THE ID OF THE 2790 IS RR; THE ID OF THE 2770 IS WW.
10 * IF THE USER WISHES TO EXECUTE THIS PROGRAM, HE MUST MODIFY IT
11 * TO USE THE ID'S OF HIS PARTICULAR TERMINALS.
12 *
13 * OPERATION:
14 *
15 * (1.) WHEN THE PROGRAM IS STARTED, A MESSAGE, 'SWTEST HAS BEGUN
16 * EXECUTION', WILL BE PRINTED ON THE CONSOLE.
17 *
18 * (2.) IF THE LINE CAN NOT BE OPENED, A MESSAGE, 'OPEN DID NOT
19 * COMPLETE SUCCESSFULLY', WILL BE PRINTED ON THE CONSOLE AND
20 * THE PROGRAM IS ABENDED.
21 *
22 * (3.) IF THE LINE IS OPENED SUCCESSFULLY, THE TERMINAL OPERATOR
23 * CAN THEN DIAL THE COMPUTER FROM EITHER TERMINAL AND
24 * SEND DATA. WHEN ALL OF THE DATA HAS BEEN READ FROM THE
25 * TERMINAL, THE PROGRAM DISCONNECTS THE LINE.
26 *
27 * (4.) THE TERMINAL OPERATOR CAN THE MANUALLY DIAL FROM EITHER
28 * TERMINAL AND RECEIVE SIX MESSAGES BEFORE THE LINE IS DIS-
29 * CONNECTED.
30 *
31 * (5.) THE PROGRAM THEN ISSUES TWO CHGNTRY MACROS TO CHANGE THE
32 * CONTROL BYTE VALUE FOR EACH AUTHORIZED ID. A SNAP DUMP IS
33 * TAKEN BEFORE AND AFTER THE CHGNTRY MACROS IN ORDER TO SHOW
34 * THE TERMINAL LIST CONTENTS BEFORE AND AFTER ISSUING THE
35 * CHGNTRY MACROS. (IN ORDER TO RECEIVE THE SNAP
36 * DUMP, THE USER MUST PROVIDE A DD CARD FOR HIS SNAP CCR DDNAME
37 * IN HIS JCL.)
38 *
39 * (6.) A MESSAGE, 'SWTEST HAS SUCCESSFULLY COMPLETED', WILL BE
40 * PRINTED ON THE CONSOLE.
41 *
42 * ACTFS -
43 *
44 * AFTER EACH READ AND WRITE OPERATION, THE PROGRAM CHECKS FOR
45 * NORMAL COMPLETION.
46 *
47 * IF THE OPERATION WAS A READ -
48 *
49 * (1.) WHICH COMPLETED NORMALLY WITH NO BITS IN DECFLAGS TURNED
50 * ON, THE PROGRAM CONTINUES WITH THE NEXT OPERATION.
51 * (2.) WHICH COMPLETED NORMALLY WITH A BIT ON IN DECFLAGS, THE
52 * PROGRAM TAKES A SNAP DUMP, THEN CHECKS FOR AN INVALID ID.
53 *
54 * (A.) IF AN INVALID ID WAS RECEIVED, A MESSAGE, 'AN INVALID
55 * ID WAS RECEIVED', IS PRINTED ON THE CONSOLE AND THE
56 * PROGRAM DISCONNECTS THE LINE.
57 *
58 * (B.) IF AN INVALID ID WAS NOT RECEIVED, THE PROGRAM
59 * DISCONNECTS THE LINE.
60 *
61 * (3.) WHICH ENDED WITH ERROR, THE PROGRAM TAKES A SNAP DUMP AND
62 * IF A DATA CHECK OCCURRED ON A READ CONTINUE OPERATION,
63 * PERFORMS A READ REPEAT. IF THE ERROR WAS NOT A DATA CHECK,
64 * AN ERROR MESSAGE, 'ERROR CANNOT BE HANDLED BY PROGRAM', IS
65 * PRINTED ON THE CONSOLE.
66 *
67 *
68 * IF THE OPERATION WAS A WRITE -
69 *
70 * (1.) WHICH COMPLETED NORMALLY WITH NO BITS TURNED ON IN
71 * DECFLAGS, THE PROGRAM CONTINUES WITH THE NEXT OPERATION.
72 *
73 * (2.) WHICH COMPLETED NORMALLY AND IS NOT A WRITE CONNECT, THE
74 * PROGRAM DISCONNECTS THE LINE.
75 *
76 * (3.) WHICH COMPLETED NORMALLY AND IS A WRITE CONNECT, THE WRITE
77 * CONNECT IS REISSUED IF A WACK OR IF NAK WAS RECEIVED. IF AN
78 * INVALID ID WAS RECEIVED, A MESSAGE, 'INVALID ID RECEIVED -
79 * NOTHING TRANSMITTED TO TERMINAL', IS PRINTED ON THE CONSOLE.
80 * OTHERWISE, THE PROGRAM DISCONNECTS THE LINE.
81 *
82 * (4.) WHICH COMPLETED WITH ERROR, THE PROGRAM TAKES A SNAP DUMP
83 * AND CHECKS FOR A TIMEOUT CONDITION.
84 *
85 * (A.) IF THE ERROR WAS A TIMEOUT, THE PROGRAM REINITIATES
86 * THE OPERATION.
87 *
88 * (B.) IF THE ERROR WAS NOT A TIMEOUT, AN ERROR MESSAGE,
89 * 'ERROR CANNOT BE HANDLED BY PROGRAM', IS PRINTED ON THE
90 * CONSOLE.
91 *
92 *****
93 PRINT ACCE:

```

```

          95          SAVE (14,12)          SAVF REGISTERS
000004 0500          95 ENTRY HALP BASEREG,0
000006 100          100 USING *,BASEREG          ESTABLISH ADDRESSABILITY
000008 101          101 USING THADCB,CCFRFG          ESTABLISH ADDRESSABILITY FOR DCB
000010 102          102 USING IFCDFCP,CECBFRFG          AND DECB
000016 5000 C7C6          007CC 104          ST SAVEREG,SAVE+4          STORE ADDRESS OF SAVE AREA
000018 4100 C7C2          007CC 105          LA SAVFREG,SAVE
000020 4160 C6C6          0060C 106          LA CCBFRFG,MYDCB
000022 4170 C73A          0074C 107          LA DFCBFRFG,MYDCB
          105          WTC 'SWTEST HAS BEGUN EXECUTION'
          115          OPEN (SNAPDCB,(OUTPUT))          OPEN THE SNAP DCB

```

```

LCC CPJECT CODE ADDR1 ADDR2 STMT SCLRCF STATEMENT F30SEP69 10/06/70
          12E          OPEN (MYDCB)          OPEN THE LINE DCB
000052 5110 6030          0003C 133          TM CCRFFLGS,X'10'          DID OPEN COMPLETE SUCCESSFULLY
000054 4710 C05A          000A0 134          RC REGIM          IF SC, ISSUE READ CONNECT
          136          WTC 'OPEN DID NOT COMPLETE SUCCESSFULLY'
00008E 47F0 C250          00296 146          R EXIT
          14E          PRINT GEN
          149 ERRBLCK LERB 1
          150+ERRBLCK DC OF'0' ALIGN TABLE AND ATTACH NAME
          152**          ACCUMULATORS
          153+          CC F'C' TRANSMISSIONS
          154+          CC H'C' DATA CHECKS
          155+          CC H'C' INTERVENTIONS
          156+          CC H'C' NCNTEXT TIMECUTS
          157**          CCOUNTERS
          158+          CC X'C' TRANSMISSIONS
          159+          CC X'C' DATA CHECKS
          160+          CC X'C' INTERVENTIONS
          161+          CC X'C' NCNTEXT TIMECUTS
          162**          TRRESHOLDS
          163+          CC YL1(255) TRANSMISSIONS
          164+          CC YL1(10) DATA CHECKS
          165+          CC YL1(5) INTERVENTIONS
          166+          CC YL1(5) NCNTEXT TIMECUTS
          167**          RESEPTVEC
          168+          DC XL2'C'
          17C          PRINT NCGEN
0000AC          172 BEGIN          ECU *
          174 RTC          READ MYDCB,TC,MYDCB,,,ANSRLIST,1,PF=E
0000CA 4590 C25A          002A0 187          BAL 9,TIC          CHECK SID CONDITION CODE
0000CE 47F0 C05A          000AC 188          B RTC          REISSUE READ CONNECT
          19C RTT          ECU *
          191          L AREG,DECENTRY          POINT TO TERMINAL LIST
          192          CLC '(2,AREG),=X'0C0000'          WAS VALID ID RECEIVED
          193          RE WTTD          IF NOT, DISCONNECT THE LINE
          195          READ MYDCB,TT,,,,1,PF=E
0000FA 455C C25A          002A0 204          PAL 9,TIC          CHECK SID CONDITION CODE
0000FE 47FC C0CC          000D2 205          R RTT          REISSUE READ CONTINUE

```

```

LCC CPJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT F30SEP69 10/06/70
000102 5850 70CC          0000C 207          L AREG,DECAPEA          GET ADDRESS OF MESSAGE AREA
000106 5537 5000          0000C 208          CLI 3(AREG),X'37'          HAS NOT BEEN RECEIVED
00010A 4780 C1CC          00112 209          RE WTTD          IF SC, DISCONNECT THE LINE
00010E 47FC C0CC          000D2 210          R RTT          IF NOT, READ MORE TEXT
          212 WTTD          WRITE MYDCB,TC,PF=E
000124 455C C25A          002A0 219          PAL 9,TIC          CHECK SID CONDITION CODE
00012P 47FC C1CC          00112 220          R WTTD          REISSUE WRITE DISCONNECT
          222          LA AREAREG,CUTMSG          POINT TO OUTPUT MESSAGE
          224 WTC          WRITE MYDCB,TC,MYDCB,(AREAREG),58,CIALLIST,1,PF=E
000166 459C C25A          002A0 240          PAL 9,TIC          CHECK SID CONDITION CODE
00016A 47FC C12A          0013C 241          R WTC          REISSUE WRITE CONNECT

```

00016E 5P50 7C14	00014	243	L	AREG,DECENTRY	POINT TO TERMINAL LIST
000172 05C2 50C8	00014	244	CLC	9(3,AREG),=X'E6E62D'	WAS IC RECEIVED FROM 2770-RR,END
000178 4770 C17C	00014	245	RNE	MCT2770	BRANCH IF ACT
00017C 0201 P0C1	00014	246	MVC	1(2,AREAREG),DC15P	IF SC, MOVE THE DEVICE CONTROL
		247 *			CHARACTER FOR THE PRINTER
		248 *			INTC THE OLPUT MESSAGE
000182		250	ACT2770	EGU *	
		252	WTT	WRITE MYDCFB,TT,,(AREAREG),27,,1,MF=E	
0001AC 4590 C29A	002AC	265	RAL	9,TIC	CHECK SIC CONDITION CODE
0001BC 47FC C17C	00182	266	R	WTT	REISSUE WRITE CONTINUE
000184 418B 001B	00018	268	LA	AREAREG,27(AREAREG)	POINT TO NEXT CLPUT MESSAGE
000188 58AC C8CA	0081C	269	L	CTREG,CCOUNTER	UPDATE COUNT OF
00018C 41AA 00C1	00001	270	LA	CTREG,1(CTREG)	MESSAGES SENT
000190 50A0 C*CA	0081C	271	ST	CTREG,CCOUNTER	SAVE THE NEW COUNT
000194 5906 C*CP	00812	272	CLI	CCOUNTER+3,X'06'	HAVE SIX MESSAGES BEEN SENT
000198 4770 C17C	00182	273	BLE	WTT	IF ACT, POINT TO NEXT MESSAGE
00019C 0703 C*CA	0081C	274	XC	CCOUNTER(4),CCOUNTER	RESET CCOUNTER TO ZERO
		276	WTD	WRITE MYDECB,TC,MF=E	
0001F4 4590 C29A	002AC	283	RAL	9,TIC	CHECK SIC CONDITION CODE
0001FA 47FC C1CC	001D2	284	R	WTC	REISSUE WRITE DISCONNECT
0001EC		286	SNAP	ECL *	

PAGE 5

LCC	CPJECT CODE	ACCR1	ADDR2	STMT	SCLRCE	STATEMENT	F30SEP69	IC/06/70
				288	SNAP	TC=2,MF=(F,SDUMP)	LOOK AT ANSWER LIST BEFORE CHGNTY	
				294	CHGNTY	ANSRLIST,SWLST,1,DISC	CHANGE CONTROL BYTE VALUE TO 1	
				305	CHGNTY	ANSRLIST,SWLST,2,DISC	CHANGE CONTROL BYTE VALUE TO 1	
				324	SNAP	ID=3,MF=(F,SDUMP)	LOOK AT ANSWER LIST AFTER CHGNTY	
				33C	WTC	'SWTEST HAS SUCCESSFULLY COMPLETED'		
0002PC				34C	CLCSE	EGU *		
				342	CLCSE	(MYDCB)	CLOSE THE LINE DCB	
				346	CLCSF	(SNAPDCB)	CLOSE THE SNAP DCB	
000294				356	EXIT	FCU *		
000296	58DC	C7C6	007CC	357	L	SAVEREG,SAVE+4	RESTORE REGISTERS	
				359	RETURN	(14,12)	RETURN CONTROL	
0002AC	12FF			363	TIC	LTR 15,15	EXCP ISSUED	
0002A2	4780	C2CE	00204	364	RZ	WAIT	ISSUE WAIT IF GOOD SIC	
				366	WTC	'SIC WAS NOT GOOD'		
				376	SNAP	ID=4,MF=(E,SDUMP)	LOOK AT DECFLAGS	
0002DC	47F0	C27A	00280	382	R	CLCSF		
0002E4				384	WAIT	ECL *		
				386	WAITP	1,ECB=(DECBREG)	WAIT FOR COMPLETION OF OPERATION	
0002FC	91C1	70C5	000C5	392	TM	DECTYPE+1,X'C1'	IS THIS A READ OPERATION	
0002E4	4780	C37A	00380	393	RZ	WPRTRN	IF ACT, GO TO WRITE ROUTINE	
0002EB				394	TESTCCCF	EGU *		
0002EB	917F	70C0	000C0	395	TM	DECFDECB,X'7F'	WAS ECB POSTED NORMALLY	
0002EC	47F0	C332	00338	396	RAC	RLFRTRN	IF ACT, CHECK ERRORS	
0002FC	9500	7018	0001E	397	CLI	DECFLAGS,X'0C'	ARE ALL FLAGS ZERO	
0002F4	4789	C0C4	00004	398	BE	4(C)	IF SC, CONTINUE NORMALLY	
				40C	SNAP	ID=5,MF=(E,SDUMP)		
000306	5510	7C18	0001E	406	CLI	DECFLAGS,X'10'	WAS AN INVALID IC RECEIVED	
00030A	4770	C10C	00112	407	RNE	WTTD	IF ACT, DISCONNECT LINE	
				409	WTC	'AN INVALID ID WAS RECEIVED'		
000334	47FC	C1CC	00112	415	R	WTTD		

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	F30SEP69	10/06/70
00032A				421	BUFFTN EQU *		
				423	SNAP IF=6,MF=(E,SDUMP)		LOOK AT DECFLAGS
00034A	5503 7005	00005		429	CLI DECTYPE+1,X'03'		IS THIS A READ CCNTINUE
00034A	4770 C456		00450	430	RNE FINISH		IF NCT, PRINT ERROR MESSAGE
00034F	510A 7010	00010		431	TM DECSFN5C,X'0A'		WAS ERROR A DATA CHECK
000352	4760 C456		00450	432	PAC FINISH		IF NCT, PRINT ERROR MESSAGE
000356	58A0 C8CA		00810	433	L CTREG,CCOUNTER		UPDATE COUNT OF
00035A	41AA 00C1		00001	434	LA CTREG,1(CTREG)		ERRORS
00035E	50A0 C8CA		00810	435	ST CTREG,CCOUNTER		SAVE NEW CCUNT
000362	5502 C8C0	00813		436	CLI CCOUNTER+3,X'02'		IS COUNT OF ERRORS TWO
000366	4780 C456		00450	437	BNE FINISH		IF SC, WRITE ERROR MESSAGE
				439	READ NYDECB,TP,MF=E		
000370	47FC C29A		C02A0	446	B TIC		CHECK SID CONDITION CCDE
000380	917F 7000	00000		448	WRTFTN TM DECSDECB,X'7F'		WAS ECB POSTED NORMALLY
00038A	47E0 C42C		00432	449	RNE WPFRR		IF NCT, CHECK ERROR
00038B	9500 7018	00018		450	CLI DECFLAGS,X'00'		ALL FLAGS ZERO
00038C	4785 C004		00004	451	BE 4(S)		IF SO, CCNTINUE NORMALLY
000390	9510 7005	00005		452	CLI DECTYPE+1,X'10'		IS OPERATION WRITE CCNNECT
000394	4770 C1CC		00102	453	RNE WTD		IF NCT, DISCONNECT THE LINE
000398	5500 7018	00018		454	CLI DECFLAGS,X'00'		WAS WACK RECEIVED
00039C	4780 C386		00380	455	BF RETURN		IF SC, CHECK FOR SECOND TIME
0003A0	5850 7014		00014	456	L AREG,DECENTRY		POINT TO TERMINAL LIST
0003A4	4155 C008		00008	457	LA AREG,8(AREG)		POINT TO READ-IN-AREA
0003A8	0502 5000	C818 00000	0081E	458	CLC 0(3,AREG),=X'D9093D'		WAS IC NAK RECEIVED FROM 2780
0003AE	4780 C386		00380	459	RE RETURN		IF SO, CHECK FOR SECOND TIME
0003B2	0502 5000	C818 00000	00821	460	CLC 0(3,AREG),=X'E6F63D'		WAS IC NAK RECEIVED FROM 2770
0003B8	4770 C3D4		003DA	461	BNE FINI		IF NOT, CHECK FOR INVALID ID
0003BC				462	RETURN EQU *		
0003BC	58A0 C8CA		00810	463	L CTREG,CCOUNTER		UPDATE COUNT OF
0003C0	41AA 00C1		00001	464	LA CTREG,1(CTREG)		TIMES RECEIVED
0003C4	50A0 C8CA		00810	465	ST CTREG,CCOUNTER		SAVE NEW CCUNT
0003C8	9502 C8C0		00P13	466	CLI CCOUNTER+3,X'02'		HAS THIS BEEN RECEIVED TWICE
0003CC	4770 C12A		00130	467	BNE WTC		IF NCT, REISSUE WRITE CCNNECT
0003C0	0703 C8CA	C8DA 00810	00810	468	XC CCOUNTER(4),CCOUNTER		RESET CCOUNTER TO ZERO
0003D6	47F0 C1CC		00102	469	B WTD		DISCONNECT THE LINE
0003DA				471	FINI EQU *		
				473	SNAP IF=7,MF=(E,SDUMP)		LOOK AT THE SENSE IN THE DECB
0003E8	9510 7018	00018		479	CLI DECFLAGS,X'10'		WAS AN INVALID ID RECEIVED
0003EC	4770 C1CC		00102	480	BNE WTD		IF NCT, DISCONNECT THE LINE
				482	WTC 'INVALID ID RECEIVED-NOTHING TRANSMITTED TO TERMINAL'		
00042E	47FC C1E6		C01EC	492	B SNAP		

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	F30SEP69	10/06/70
000432				494	WPFRR EQU *		
				496	SNAP IF=8,MF=(E,SDUMP)		
000440	5501 7010	00010		502	CLI DECSFN5C,X'01'		WAS ERROR TIME OUT
000444	4770 C456		00450	503	RNE FINISH		IF NCT, PRINT ERROR MESSAGE
000448	58A0 C8CA		00810	504	L CTREG,CCOUNTER		UPDATE COUNT OF
00044C	41AA 00C1		00001	505	LA CTREG,1(CTREG)		ERRORS
000450	50A0 C8CA		00810	506	ST CTREG,CCOUNTER		SAVE NEW CCUNT
000454	5502 C8C0	00813		507	CLI CCOUNTER+3,X'02'		IS ERROR CCUNT 2
000458	476A C000		00000	508	BL 0(0)		IF NCT, REISSUE OPERATION
				510	FINISH WTC 'ERROR CANNOT BE HANDLED BY PROGRAM'		
00048E	47E0 C27A		C0280	520	B CLPSE		

```

000400 0277C1
00040F F300C9E24CC9E240
0004A5 1526
0004A7 02
0004AE E300C9E240C9E240
0004C0 1526
0004C2 02
0004C3 E300C9E240C9E240
0004C9 1526
0004DC 02
0004DF E300C9E24CC9E240
0004FE 1526
0004FE 02
0004F9 E300C9E240C9E240
000511 1526
000513 02
000514 E300C9E240C9E240
000520 1526
522 PRINT GEN
523 OUTMSG DC X'0227C1'
524 DC C'THIS IS TEST MESSAGE 1'
525 DC X'1526'
526 DC X'02'
527 DC C'THIS IS TEST MESSAGE 2 '
528 DC X'1526'
529 DC X'02'
530 DC C'THIS IS TEST MESSAGE 3 '
531 DC X'1526'
532 DC X'02'
533 DC C'THIS IS TEST MESSAGE 4 '
534 DC X'1526'
535 DC X'02'
536 DC C'THIS IS TEST MESSAGE 5 '
537 DC X'1526'
538 DC X'02'
539 DC C'THIS IS TEST MESSAGE 6 '
540 DC X'1526'

```

```

542 *****
543 *
544 * IF THE USER WISHES TO EXCLUDE THIS PROGRAM, HE MUST CHECK HIS
545 * CONTROL MACROS TO HANDLE THE ID'S OF HIS PARTICULAR TERMINALS.
546 *
547 *****

```

```

000530
000530 FF
000531 000000
000534 C2
000535 C4
000536 C3
545 ANSRLIST DFTPLST SWLST,AN,4,C,2,107C,(C9C92C),(E6E62C)
550+ANSRLIST DS OF PUT CN FULL WORD BOUNDARY 000L
551+ DC XL1'FF' IDENTIFY AS SWLST 000L
552+ DC XL3'0' PTR TO MATCHING SEQUENCE 000L
553+ DC HL1'2' NUMBER OF LIST ENTRIES 000L
554+ DC HL1'4' ENTRY LENGTH 000L
555+ DC HL1'3' REAC-IN-AREA LENGTH 000L

```

LCC OBJECT CODE ADDR1 ADDR2 STMT SCLRCE STATEMENT F30SEP69 10/06/70

```

000537 CC 556+ CC HL1'0' NC. OF DIAL DIGITS 000L
000538 C0C0CC 557+ CC XL3'0' REAC-IN-AREA 000L
000539 C2 558+ CC HL1'2' IDCCUNT 000L
000520 1070 559+ DC XL2'107C' IDSENT 000L
000520 090920 560+ DC X'D9092C' AUTHORIZED SEQUENCE 000L
000541 C0 561+ CC XL1'00' CONTROL BYTE VALUE CF 0 000L
000542 E6E620 562+ DC X'E6E62C' AUTHORIZED SEQUENCE 000L
000545 CC 563+ DC XL1'00' CONTROL BYTE VALUE CF 0 000L

```

```

000546
000546 FF
000549 C0C0CC
00054C 02
00054C C5
00054E C4
00054F C0
000550 C0C0C0C0
000554 C1
000555 20
000556 0909107C
00055A C1
00055B E6E6107C
00055F C1
565 DIALLIST DFTPLST SWLST,MD,5,0,1,20,(D909107C,1),(E6E6107C,1)
566+DIALLIST DS OF PUT CN FULL WORD BOUNDARY 000L
567+ DC XL1'FF' IDENTIFY AS SWLST 000L
568+ DC XL3'0' PTR TO MATCHING SEQUENCE 000L
569+ DC HL1'2' NUMBER OF LIST ENTRIES 000L
570+ DC HL1'5' ENTRY LENGTH 000L
571+ DC HL1'4' REAC-IN-AREA LENGTH 000L
572+ DC HL1'0' NC. OF DIAL DIGITS 000L
573+ DC XL4'0' REAC-IN-AREA 000L
574+ DC HL1'1' IDCCUNT 000L
575+ DC XL1'20' IDSENT 000L
576+ DC X'D909107C' AUTHORIZED SEQUENCE 000L
577+ DC YL1(1) CONTROL BYTE 000L
578+ DC X'E6E6107C' AUTHORIZED SEQUENCE 000L
579+ DC YL1(1) CONTROL BYTE 000L

```

```

000560 C0C0C0C0C0000000 581 INAREA CC 10CF'0'
583 MYCCB DCB CSORG=CX,DEV=BS,MACRF=(R,W),CCNAME=B277C0C1,
LEP=ERPBLCK,EROPT=TC X

```

```

585+ DATA CONTROL BLCK
586+
000600 587+ CRG *-20 TO ELIMINATE UNUSED SPACE
000600 588+MYCCB DS OF CRIGIN CN WORD BOUNDRY
0006FC 589+ CRG *-20 TO CRIGIN GENERATION

```

```

591+ COMMON ACCESS METHOD INTERFACE
0006FC CC 593+ CC AL1(0) BUFAC
0006F1 C0C0C1 594+ CC AL3(1) BUFCA
0006F4 C0CC 595+ CC AL2(0) BUFL
0006F6 10CC 596+ DC RL2'00010CC00000000' DSGRG
0006FF C0C0C0C1 597+ CC A(1) IOBAC

```

```

599+ FOUNDATION EXTENSION
0006FC CC 601+ DC RL1'0000000' BTEK,BFALN,MIARCHY
0006FC 10 602+ DC RL1'00011CC0' BTAM EROPT CODE
0006FE FF 603+ DC AL1(255) BTAM BUFFER CCUNT
0006FF 00 604+ DC AL1(C)
000700 CC 605+ DC BL1'000C00CC' RECFM

```

LCC	PROJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	F30SEP69	10/06/70
000701	000000			606+	DC	AL3(0) EXLST		
				608**		FOUNDATION BLOCK		
000704	C2F2F7F7F0C4FCF1			610+	DC	CLR'R2770CC1' DCNAME		
000700	02			611+	DC	RL1'00000010' OFLGS		
000700	00			612+	DC	RL1'00000000' IFLG		
000700	2020			613+	DC	RL2'0010000000100000' MACR		
				615**		BTAM INTERFACE		
000710	00000000			617+	DC	A(FRRBLOCK) DCPLERR		
000714	00			618+	DC	RL1'00000000' MCCE		
000715	00			619+	DC	BL1'10000000' MAS,CCDE		
000716	0000000000000000			620+	DC	XL26'0' CONTROL CHARS		
000720	0000000000000000			621+	DC	4F'0' RESEVED		
				624	READ	MYDECR,TT,MYDCR,INAREA,40C,AASRLIST,1,MF=L		
000740				625+	DS	7F		
000740	00000000			626+MYDECR	FC	A(0) EVENT CONTROL BLOCK		
000744	00			627+	FC	RL1'0000'		
000745	03			628+	FC	AL1(3) TYPE FIELD		
000746	0190			629+	DC	AL2(400) LENGTH	00CB	
000746	00000600			630+	DC	A(MYCCB) DCB ADDRESS		
000740	00000560			631+	DC	A(INAREA) AREA ADDRESS	000P	
000750	00000000			632+	DC	A(0) ERROR INFO. FIELD ADDR		
000754	00000530			633+	DC	A(AASRLIST) TERMINAL LIST ADDRESS		
000750	0001			634+	DC	AL2(1) LINE NUMBER		
00075A	0000			635+	DC	AL2(2) REPCASE FIELD		
000750	00			636+	DC	AL1(0) TP-CP CODE		
000750	00			637+	DC	AL1(0) ERROR STATUS		
00075E	0000			638+	DC	AL2(0) CSW STATUS		
000760	00000000			639+	FC	AL4(0) CURRENT ADDR LIST PTR		
000764	00000000			640+	DC	AL4(0) CURRENT ADDR POLL PTR		
				642	SNAPCCP	DGP DSCRG=PS,RECFM=WBA,MACRF=W,BLKSIZE=1622,LRECL=125,DCNAME=SNAPSW,DEVO=PR		X
				644**		DATA CONTROL BLOCK		
				645**				
00075F				646+	CRG	*-16 TO ELIMINATE UNUSED SPACE		
00075B				647+SNAPCCP	DS	OF ORIGIN ON WORD BOUNDRY		
00076F				648+	CRG	**16 TO CRIGIN GENERATION		
				650**		PRINTER DEVICE INTERFACE		
00076F	0000			652+	DC	RL2'0000000000000000' PRISP,CEVT		
00076A	0000			653+	DC	H'0'		
				654**		COMMON ACCESS METHOD INTERFACE		

LCC	PROJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	F30SEP69	10/06/70
000760	00			657+	DC	AL1(0) BLFNC		
000760	000001			658+	DC	AL3(1) BUFCR		
000770	0000			659+	DC	AL2(0) BUFL		
000772	4000			660+	DC	RL2'0100000000000000' DSCRG		
000774	0000001			661+	FC	A(1) IOBAD		
				663**		FOUNDATION EXTENSION		
000770	00			665+	DC	RL1'00000000' BFTEK,BFALA,HIARCHY		
000775	000001			666+	DC	AL3(1) ECDAD		
000770	54			667+	DC	BL1'01010100' RECFM		
000770	000000			668+	DC	AL3(0) EXLST		
				670**		FOUNDATION BLOCK		
000780	E2050107E2E64040			672+	DC	CLR'SNAPSW' DCNAME		
000780	02			673+	DC	RL1'00000010' OFLGS		
000785	00			674+	DC	RL1'00000000' IFLG		
00078A	0020			675+	DC	RL2'0000000000000000' MACR		
				677**		ESAM-RPAM-CSAM INTERFACE		
000780	00			679+	DC	RL1'00000000' RER1		
000780	000001			680+	DC	AL3(1) CHECK, GEPR, PERP		
000790	00000001			681+	DC	A(1) SYNAD		
000794	0000			682+	DC	H'0' CIND1, CIND2		
000796	0660			683+	DC	AL2(1632) PLKSIZE		
000798	00000000			684+	DC	F'0' WCPC, WCFL, CFFSR, OFFSW		
000790	00000001			685+	DC	A(1) IOBA		
0007AC	00			686+	DC	AL1(0) ACP		
0007A1	000001			687+	DC	AL3(1) EDBR, ECRAD		


```

                                ASAP-EPAM INTERFACE
0007A4 00000001          691+      DC      A(1) EOBW
0007A8 0000          692+      CC      H'0' DIRCT
0007AA 0000          693+      DC      AL2(125) LRECL
0007AC 00000001          694+      CC      A(1) CNTRL, NOTE, PCINT

                                696 SELMF SNAP  CCP=SNAPDCB, ID=1, PCDATA=(AFCS),
                                STORAGE=(ANSRLIST, SDUMP), MF=L
0007BC          697+ SELMF  DS      OF
0007BD 01          698+      DC      AL1(1) IC NUMBER
0007BE 00          699+      CC      AL1(0)
0007BF 06          700+      DC      AL1(134) OPTICN FLAGS
0007C0 20          701+      CC      AL1(32) OPTICN FLAGS
0007C4 00000758          702+      CC      A(SNAPDCB) DCR ADDRESS
0007BE 00000000          703+      DC      A(C) TCB ADDRESS
0007BC 00000700          704+      DC      A(*+4) ADDRESS OF SNAP-SMCT LIST
0007CC 00000530          705+      CC      A(ANSRLIST) STARTING/ENDING ADDRESS
0007C4 00000780          706+      DC      A(SDUMP) STARTING/ENDING ADDRESS
0007C4          707+      CRG     *-4
0007C4 80          708+      CC      X'RO'
0007CE          709+      CRG     *+3

```

PAGE 11

```

LCC OBJECT CODE      ADDR1 ADDR2  STMT  SCLRC  STATEMENT                                F3CSEP69  IC/06/70
0007C8          711 SAVE      DS      18F'0'          SAVE AREA
000A1C 00000000          712 CCLNTER  CC      F'0'          CCUNTER
000814 1140          713 DC1SP    DC      X'1140'       DC1,SPACE
000005          714 AREG     ECU      5              REG USED TC PCINT TC CFTRMLST
000006          715 DCRREG   EQU      6              DCR REGISTER
000007          716 DCEPREG  EQU      7              DFCB REGISTER
00000A          717 CTREG    EQU      10             CCUNTER REGISTER
00000B          718 AREAREG  ECL     11             MESSAGE AREA REGISTER
00000C          719 BASEREG  EQU     12             BASE REGISTER
00000C          720 SAVEREG  ECL     13             SAVE AREA REGISTER

                                722          DCBC  DEVD=BS, DSCRG=BX

                                724+*          CCB SYMBOLIC DEFINITION FOR
                                725+*          BTAP LINE GROUP

000000          727+IHACCB  DSECT

                                729+*          BTAP LINE GROUP INTERFACE

000014          731+          CRG     I+ACCB+20
000014          732+CCBPLFNC DS      0AL1
000014          733+DCBPLFCB DS      4
000018          734+DCBPLFL  DS      AL2
00001A          735+CCBPCSRG DS      BL2
00001C          736+CCBICBAC DS      4
00002C          737+CCBFIARC DS      0PL1
00002C          738+CCBFFTEK DS      BL1
000021          739+CCBFRRFC DS      AL1
000022          740+CCBFLFCT DS      AL1
000023          741+          DS      AL1
000024          742+CCBFXLST DS      A
00001C          743+          CRG     I+ACCB+28
00001C          744+CCBEVTP  DS      BL1
000024          745+          CRG     I+ACCB+36
000024          746+CCBEICPX DS      BL1

                                749+*          FCUNDACTION BEFORE OPEN

000028          751+          CRG     I+ACCB+40
00002E          752+CCBCCNAM DS      CLP
00003C          753+CCBFCFLS DS      BL1
000031          754+CCBIFLC  DS      PL1
000032          755+CCBMACR  DS      BL2

                                757+*          FCUNDACTION AFTER OPEN

00002E          759+          CRG     I+ACCB+40
00002E          760+CCBETICT DS      BL2
00002A          761+CCBMACPF DS      PL2
00002C          762+CCBIFLGS DS      0PL1
00002C          763+CCBEEBAC DS      A

```

```

LCC OBJECT CODE ADDR1 ADDR2 STMT SCLPCE STATEMENT
000030 764+DCPRFAC DS 0A
000030 765+DCBWRITE DS 0A

000034 767** ETAM INTERFACE
000034 768** CRG IF+ACCB+52
000034 765+DCCELERR DS A

000038 771+DCBXMCCF CS RL1
000039 772+DCBXCCLF CS RL1
00003A 773+DCBXSRSV DS CL1
00003B 774+DCBXSAPT DS CL1
00003C 775+DCBXSSTX DS CL1
00003D 776+DCBXSSTX DS CL1
00003E 777+DCBXSSTX DS CL1
00003F 778+DCBXSSTX DS CL1
000040 779+DCBXSAMC DS CL2
000042 780+DCBXSAM1 DS CL2
000044 781+DCBXSAMC DS CL1
000045 782+DCBXSAMC DS CL1
000046 783+DCBXSSTX DS CL1
000047 784+DCBXSSTX DS CL1
000048 785+DCBXSSTX DS CL1
000049 786+DCBXSSTX DS CL3
00004C 787+DCBXSSTX DS CL2
00004E 788+DCBXSSTX DS CL2
000050 789+DCBXSSTX DS CL2
    
```

```

000000 791 IECTDECP
000000 792+IECTDECP DSECT DECB DUMMY SECTION
000000 793** -----
000000 794** +
000000 795** C + STANDARD EVENT CONTROL BLCK +
000000 796** +
000000 797** -----
000000 798** +
000000 799** + OPERATION TYPE + AREA LENGTH +
000000 800** +
000000 801** -----
000000 802** + CN-LINE +
000000 803** P + TERMINAL + ADDRESS OF DCB +
000000 804** + TEST +
000000 805** -----
000000 806** +
000000 807** 12 + RESERVED + ADDRESS OF AREA +
000000 808** +
000000 809** -----
000000 810** +
000000 811** 16 + SENSE BYTES + RESIDUAL CCLAT +
000000 812** + 1 6 2 +
000000 813** -----
000000 814** +
000000 815** 2C + COMMAND + ADDRESS OF TERMINAL LIST +
000000 816** + CODE +
000000 817** -----
    
```

```

LCC OBJECT CODE ADDR1 ADDR2 STMT SCLPCE STATEMENT
818** +
819** 24 + STATLS + RELATIVE+ + ADDRESS + VRC/LRC +
820** + FLAGS + NUMREP + RESPGNSE+ RESPONSE +
821** -----
822** +
823** 28 + TP-CP + ERROR + CSW STATLS +
824** + CODE + STATUS +
825** -----
826** +
827** 32 +RESERVED+ + ADDRESS OF CURRENT +
828** + + ADDRESSING ENTRY +
829** -----
830** +
831** 36 +RESERVED+ + ADDRESS OF CURRENT +
832** + + PCLLING ENTRY +
833** -----
834** +
835** 4C +RESERVED+ +RESERVED+ + WRITE AREA LENGTH+ +
836** + + +
837** -----
838** +
839** 44 +RESERVED+ + ADDRESS OF WRITE AREA +
840** + +
841** -----
    
```

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SCLRCE	STATEMENT	F30SEP65	1C/C6/70
C00000				843+DECSCECB	DS	1F STATUS FLAG + ADDRESS OF THE TC?		
C00004				845+RECTYF	DS	1F OPERATION TYPE		
C00006				847+DECLNGTH	DS	1F AREA LENGTH		
C00008				849+DECCALTT	DS	0CL1 RESERVED FOR ON-LINE TERMINAL TEST		
C00008				850+DECCCBAT	DS	1F ADDRESS OF DCB		
C0000C				852+DECAREA	DS	1F ADDRESS OF AREA		
C0001C				854+DECSSENSC	DS	1C 1ST SENSE BYTE		
000011				856+DECSSENS1	DS	1C 2ND SENSE BYTE		
C00012				858+DECCCLAT	DS	1F RESIDUAL CCUMT		
C00014				860+DECCMCCD	DS	0CL1 COMMAND CODE		
C00014				861+DECENTRY	DS	1F ADDRESS OF TERMINAL LIST		
000018				864+DECFLAGS	DS	1C STATUS FLAGS		
000019				866+DECRLA	DS	1C RELATIVE LINE NUMBER		
00001A				868+DECRFSPN	DS	1F RESPCASE FIELDS		
C0001C				870+DECTFCCD	DS	1C TP-CP CCDE		
00001C				872+DECEPRST	DS	1C ERROR STATUS		
00001E				874+DECCSWST	DS	1F CSM STATUS		
00002C				876+DECA CRT	DS	1F ADDRESS OF CURRENT ADDRESSING ENTRY		
000024				878+DECPCLPT	DS	1F ADDRESS OF CURRENT POLLING ENTRY		
00002E				880+	DS	2C RESERVED		
00002A				882+DECNLAG	DS	1F WRITE AREA LENGTH		
00002C				884+DECNAREA	DS	1F ADDRESS OF WRITE AREA		
				887		END		
C0001E	C0000C			888		=X'000000'		
C0001E	E6E62D			889		=X'E6E62D'		
00001E	C9C53C			890		=X'C9D93E'		
000021	E6E62D			891		=X'E6E62D'		

CRCSS-REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES	1C/C6/70
ANSRLIST	000C4	000530	00550	0180 C296 C304 0311 0319 0633 0705	
AREAREG	000C1	00000B	00718	0222 C232 C246 0258 C268 0268	
AREG	00001	000005	00714	0191 0192 C207 0208 0243 0244 0456 C457 0457 0458 0460	
BASEREG	000C1	000000	00715	0099 0100	
BEGIN	000C1	0000A0	00172	0134	
BLFRTA	000C1	000338	00421	0396	
CLCSE	000C1	000280	00340	0382 C520	
CCUNTER	000C4	000810	00712	0269 0271 C272 C274 C274 0423 0435 0436 0463 0465 0466 0468 0468 0504 0506	
CTREG	000C1	0000CA	00717	0507 0269 C270 C270 C271 C423 0434 C434 0435 0463 0464 0464 0465 0504 C505 0505	
DCBBSAK	000C2	000040	00775	0506	
DCBBSAK1	000C2	000042	00780		
DCBBSCL	000C1	000047	00784		
DCBBSENC	000C1	000044	00781		
DCBBSSECT	000C1	000048	00785		
DCBBSSETE	000C1	000046	00783		
DCBBSSETX	000C1	00003F	00778		
DCBBSNAK	000C1	000045	00782		
DCBBSCNL	000C2	00004C	00787		
DCBBSPSV	000C1	00003A	00773		
DCBBSPI	000C2	000050	00789		
DCBBSNAK	000C2	00004E	0078E		
DCBBSSTX	000C1	00003F	00776		
DCBBSV	000C3	000049	00786		

DCPPSTX CCCC1 C0003F C0777
 DCPPSTX CCCC1 C0003C C0775
 DCFFSWRT CCCC1 C0003P C0774
 DCERLFCP CCCC4 C00014 C0733
 DCERLFCF CCCC1 C00022 C074C
 DCERLFL CCCC2 C00018 C0734
 DCERLUNC CCCC1 C00014 C0732
 DCERDNAM CCCC8 C00028 C0752
 DCERCEAD CCCC4 C0002C C0763
 DCBDEVTP CCCC1 C0001C C0744
 DCBCSCRG CCCC2 C0001A C0735
 DCREICBX CCCC1 C00024 C0746
 DCBFRRCP CCCC1 C00021 C0739
 DCBFXLST CCCC4 C00024 C0742
 DCBHIARC CCCC1 C0002C C0737
 DCRIFLG CCCC1 C00031 C0754
 DCRIFLGS CCCC1 C0002C C0762
 DCRICBAD CCCC4 C0001C C0736
 DCBLERB CCCC4 C00034 C0765
 DCBMACR CCCC2 C00032 C0755
 DCBMACRF CCCC2 C0002A C0761
 DCRCFLGS CCCC1 C0002C C0753 0133
 DCBREAC CCCC4 C00030 C0764
 DCPREG CCCC1 C00006 C0715 01C1 01C6
 DCRTICT CCCC2 C00028 C076C
 DCRWRTF CCCC4 C00030 C0765
 DCBXCCCE CCCC1 C00039 C0772
 DCBXMFCE CCCC1 C00038 C0771
 DC1SP CCCC2 C00014 C0712 0246

CROSS-REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFCADRPT	CCCC4	C00020	C0876	
DFCAREA	CCCC4	C000CC	C0852	0207
DFCPREG	CCCC1	C00007	C0716	01C2 01C7 0387
DFCCMCCD	CCCC1	C00014	C086C	
DFCCCLNT	CCCC2	C00012	C085E	
DFCCSWST	CCCC2	C0001F	C0874	
DFCCCEAD	CCCC4	C000CP	C085C	
DFCENTRY	CCCC4	C00014	C0861	01C1 0243 0456
DFCERRST	CCCC1	C0001C	C0872	
DFCFLAGS	CCCC1	C0001P	C0864	0397 0406 0450 0454 0479
DFCLACTH	CCCC2	C00006	C0847	
DFCCNLTT	CCCC1	C0000P	C0849	
DFCPCLPT	CCCC4	C00024	C087E	
DFCFSPN	CCCC2	C0001A	C086E	
DECRLA	CCCC1	C00019	C0866	
DFCSCECB	CCCC4	C0000C	C0843	0395 0448
DFCSENSC	CCCC1	C00010	C0854	0431 05C2
DFCSENS1	CCCC1	C00011	C0856	
DFCTPCCD	CCCC1	C0001C	C087C	
DFCTYFE	CCCC2	C00004	C0845	0392 0429 0452
DFCWAREA	CCCC4	C0002C	C0884	
DFCFLAG	CCCC2	C0002A	C0862	
DIALLIST	CCCC4	C0004P	C0566	0233
ENRY	CCCC2	CCCC04	C0C99	
ERRRCLCK	CCCC4	C0008C	C015C	0617
EXAMPLE2	CCCC1	C0000C	C0C01	
EXIT	CCCC1	C00296	C0356	0146
FJAT	CCCC1	C003DA	C0471	0461
FINISH	CCCC4	C0045C	C0512	0430 0432 0437 05C3
IFCTDFCP	CCCC1	C000CC	C0792	01C2
IWADCE	CCCC1	C000CP	C0727	01C1 0721 0742 0745 0751 0755 076P
IHP00C2	CCCC1	C0003A	C0115	0112
IHP00C2A	CCCC2	C0003A	C0116	0111
IHP00C5	CCCC1	C00096	C0142	0139
IHP00C5A	CCCC2	C00096	C0143	013P
IHP00C3	CCCC1	C0027F	C0336	0333
IHP00C3A	CCCC2	C0027F	C0337	0332
IHP00C7	CCCC1	C002CC	C0372	0369
IHP00C7A	CCCC2	C002CC	C0373	0368
IHP00C1	CCCC1	C00332	C0415	0412
IHP00C1A	CCCC2	C00332	C0416	0411
IHP00C6	CCCC1	C0042B	C0488	04P5
IHP00C6A	CCCC2	C0042C	C0489	0484
IHP00C8	CCCC1	C00486	C0516	0513
IHP00C8A	CCCC2	C00486	C0517	0512
IHP00C8B	CCCC4	C0056C	C0581	0631
MYCCR	CCCC4	C0060C	C058P	01C6 017C 0178 0184 0230 0237 0346 0630
MYCEB	CCCC4	C0074C	C0626	01C7 0175 0196 0212 0225 0253 0277 044C
NOT277C	CCCC1	C00182	C025C	0245
OUTMSE	CCCC3	C0048C	C0523	0222
PFURN	CCCC1	C0038C	C0462	0455 0459
PTC	CCCC4	C000AC	C0175	018P
RTT	CCCC1	C000C2	C019C	02C5 0210
SAVE	CCCC4	C0070B	C0711	01C4 01C5 0357
SAVEREG	CCCC1	C0000C	C072C	01C4 01C5 0357

```

LCC  PROJECT CODE  ADDR1 ADDR2  STMT  SOURCE STATEMENT  F3CSEP69  10/06/70

000000 1 EXAMPLE1 CSECT
2 *****
3 *
4 *      SAMPLE IBM 2740/2760 PROGRAM
5 *
6 *      THIS PROGRAM EXERCISES THE BTAM ONLINE TEST LOGIC FOR A
7 *      NONSWITCHED POINT-TO-POINT LINE TO WHICH IS CONNECTED AN
8 *      IBM 2740 EQUIPPED WITH AN IBM 2760 OPTICAL IMAGE UNIT.
9 *
10 *     OPERATION - WHEN THE PROGRAM HAS BEEN STARTED, TWO MESSAGES
11 *     WILL BE TYPED ON THE SYSTEM CONSOLE:
12 *
13 *           IXTRT022 IS LOADED AT XXXXXXXX
14 *           IXTRT022 FOR NONSWITCHED 2760 HAS STARTED.
15 *
16 *     THE TERMINAL OPERATOR CAN NOW DO ONE OF THE FOLLOWING:
17 *
18 *     1.) PROBE THE 'LOAD' UTILITY RESPONSE POINT OF THE 2760.
19 *     'LOAD' SHOULD NOT BE PROBED WHEN THE FILMSTRIP IS
20 *     ALREADY LOADED.
21 *
22 *     2.) ENTER AN ON-LINE TERMINAL TEST REQUEST MESSAGE FOR ANY
23 *     2740- OR 2760-TYPE TEST.
24 *
25 *     3.) PROBE THE 'UNLOAD' UTILITY RESPONSE POINT. THIS WILL
26 *     CAUSE THE FILM TO BE RETRACTED AND THE CARTRIDGE
27 *     TO BE EJECTED. YOU MAY NOW BEGIN AGAIN.
28 *
29 *     4.) ENTER A MESSAGE FROM THE 2740 KEYBOARD REQUESTING A
30 *     CLOSDOWN. THE 6 CHARACTERS OF THE MESSAGE ARE
31 *     CLCSE# WHERE #=EOT CHARACTER (UPPER CASE)
32 *
33 *     THE PROGRAM IGNORES ANY OTHER RESPONSE. THE FOLLOWING
34 *     MESSAGE WILL BE PRINTED ON THE 2740 KEYBOARD.
35 *     'THE IMMEDIATELY PRECEDING MESSAGE WAS NOT
36 *     IN THE PROPER FORMAT.
37 *     YOU ARE EXPECTED TO PROBE UNLOAD OR LOAD,
38 *     OR ENTER AN ONLINE TEST REQUEST MESSAGE.'
39 *
40 *****
41      PRINT NCGEN

000043 43 WCRKREG FCL 0
000044 44 DCRREG EQU 9
000045 45 BASEREG ECL 12
000046 46 SAVEREG EQU 13          PCINTER TO SAVE AREA
000047 47 RETREG EQU 14          PAL RETURN REGISTER
000048 48 BRREG ECL 15          PAL BRANCH ADDRESS REGISTER
000049 49 START FCL *

000051 51 SAVE (14,12)          SAVE THE REGISTERS

000055 55 BALR BASEREG,0
000056 56 USING *,BASEREG          ESTABLISH ADDRESSABILITY
000057 57 USING IMADCB,DCRREG

```

```

LCC  PROJECT CODE  ADDR1 ADDR2  STMT  SOURCE STATEMENT  F30SEP69  10/06/70

000060 5000 C3F2          003FF 50 ST BASEREG,BASE          COMPUTE THE
000061 5000 C3F4 C3E6 C3F2 003EC 003FB 55 UNPK UNPACK(5),BASE(5)          LOCAL ADDRESS
000062 5000 C307 C3F6 C307 003EC 003CD 60 TR UNPACK(8),TRTABLE-240          FOR THE
000063 5000 C2C7 C034 C3E6 C0C3A 003EC 61 MVC WTOP+30(8),UNPACK          WTC MESSAGE

000062 62 WTCP WTC 'IXTRT022 IS LOADED AT
000063 73 WTC 'IXTRT022 FOR NONSWITCHED 2760 HAS STARTED'

000064 5000 C32F          00334 63 ST SAVEREG,SAVEAREA+4
000065 4100 C32A          0033F 64 LA SAVEREG,SAVEAREA
000066 4100 C5C2          005C8 65 LA DCRREG,DCR          BASE FOR DCB
000067 9200 C3E1          003F7 66 CPEN ECL *
000068 9200 C3E1          003F7 67 MVI ICCODE,0          INDICATE OPEN INITIATED

000069 95 OPEN DCB          OPEN THE DCB

000070 9110 5030          00C37 96 TM DCPFLGS,X'10'          DID OPEN COMPLETE SUCCESSFULLY
000071 4770 C0C0          00C06 97 BIZ OPFNCK
000072 95 WTC 'OPEN DID NOT COMPLETE SUCCESSFULLY'

```

0000CC	5800	C32F	00534	105		SAVEREG,SAVEAPEA+4	
				111		RETURN (14,12)	RESTORE THE REGISTERS
0000CF				115	CPENCF	ECU *	
0000CF	9200	C3FD	003FA	116		MVI EPPCRCT,0	ZERC ERROR CCUNT
				118		TRANSLATE DCB,SD40,ERRMSG,LENGTH	TRANSLATE ERROR MESSAGE
0000FE				131	REACTI	ECL *	
0000FE	9204	C3F1	003E7	132		MVI ICCODE,4	INDICATE READ TI
				134		READ READECB,TI,,INAREA,27,,1,MF=E	
00013C	12FF			148		LTR BRREG,BRREG	NORMAL INITIATION
000132	4770	C2C6	002CC	149		BNZ ICERROR	BRANCH IF NO
000136	45F0	C1FA	00200	150		BAL RETREG,WAIT	
00013A	C501	C372	C3DA	00378	003FC	CLC INAREA(2),PREC	IS MESSAGE FROM 2760
00014C	4770	C1AC		00182		BNE IGNORE	BRANCH IF NO
000144	C501	C376	C3CC	0037C	003E2	CLC INAREA+4(2),UNLOAD	IS REQUEST TO UNLCAD
00014A	4780	C21A		00220		BE OUT	BRANCH IF YES
00014E	C501	C376	C3DE	0037C	003F4	CLC INAREA+4(2),LOAD	IS REQUEST TO LOAD
000154	4770	C1AC		00182		BNE IGNORE	IGNORE MESSAGE IF IT IS NOT
				157	*		LOAD CR UNLCAD
000158	65FF	C3F2	003F8	158		CLI LCADDONE,X'FF'	IS FILM ALREADY LOADED
00015C	4780	C1AC		00182		BE	BRANCH IF YES
00016C	C202	C372	C403	00378	00405	MVC INAPFA(3),LOADMSG	MOVE FAA FOR LOAD
000164				161	WRITETCC	ECU *	
000164	9200	C3E1	003E7	162		MVI ICCODE,12	INDICATE WRITE TCC
				164		WRITE READECB,TCC,,INAPEA,27,LOADMSG,1,MF=E	
0001AC	12FF			18C		LTR BRREG,BRREG	NORMAL INITIATION

PAGE 3

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SCLRCE	STATEMENT	
0001A2	4770	C2C6	002CC	181	BNZ	ICERROR	BRANCH IF NO
0001A6	45F0	C1FA	00200	182	BAL	RETREG,WAIT	
0001AA	92FF	C3F2	003E8	183	MVI	LCADDONE,X'FF'	INDICATE FILM LOADED
0001AE	47F0	C134	0013A	184	B	CCMPARE	
0001B2				186	IGNORE	ECL *	
0001B2	C504	C372	C4C7	00378	0040D	CLC INAREA(5),CLOSE	IS REQUEST TO CLCSE LINE
0001B6	4780	C256		0025C		BE EAC	YES
0001B8	9210	C3F1	003E7	189	MVI	ICCODE,16	INDICATE WRITE TI
				191		WRITE READECB,TIR,,ERRMSG,LENGTH,,1,MF=E	
0001F2	12FF			206	LTR	BRREG,BRREG	NORMAL INITIATION
0001F4	4770	C2C6	002CC	207	BNZ	ICERROR	BRANCH IF NO
0001F8	45F0	C1FA	00200	208	BAL	RETREG,WAIT	
0001FC	47F0	C3F8	000FF	209	B	REACTI	
00020C				211	WAIT	ECU *	
00020C	50EC	C3C6	0030C	212	ST	RETREG,TEMP	
				214	WAIT	1,FCB=READECB	
00020E	957F	C5FA	006CC	219	CLI	READECB,X'7F'	IS COMPLETION SATISFACTORY
000212	4770	C2C6	002CC	220	BNE	ICERROR	BRANCH IF NO
000216	9200	C3E0	003E6	221	MVI	ERRCRCT,0	RESET ERROR CCUNT
00021A	58EC	C3C6	003DC	222	L	RETREG,TEMP	RESTORE RETURN REGISTER
00021E	C7FF			223	BP	RETREG	RETURN
00022C				225	CLT	ECU *	
00022C	9208	C3E1	003E7	226	MVI	ICCODE,8	INDICATE WRITE TIC
				228		WRITE READECB,TIO,,3,REWIND,,MF=E	
00024A	12FF			240	LTR	BRREG,BRREG	NORMAL INITIATION
00024C	4770	C2C6	002CC	241	BNZ	ICERROR	BRANCH IF NO
00025C	45E0	C1FA	00200	242	BAL	RETREG,WAIT	
000254	9200	C3E2	003F8	243	MVI	LCADDONE,X'00'	INDICATE FILM NO LONGER LOADED
000258	47F0	C3F8	000FF	244	B	READTI	
00025C				246	EAC	ECL *	
				248	WTC	'IXTBT022 ENDED BY TERMINAL OPERATOR REQUEST'	
				258		WRITE READECB,TIO,,3,REWIND,,MF=E	
00028P	12FF			270	LTR	BRREG,BRREG	NORMAL INITIATION
00028A	4770	C2F8	002FF	271	BNZ	PERM	BRANCH IF NO

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
				432**		+ + +
				433**	4	+ OPERATION TYPE + AREA LENGTH +
				434**		+ + +
				435**		+-----+ +
				436**		+ ON-LINE +
				437**	8	+ TERMINAL + ADDRESS OF DCR
				438**		+ TEST +
				439**		+-----+ +
				440**		+ + +
				441**	12	+ RESERVED + ADDRESS OF AREA
				442**		+ + +
				443**		+-----+ +
				444**		+ + +
				445**	16	+ SENSE BYTES + RESIDUAL COUNT +
				446**		+ 1 2 +
				447**		+-----+ +
				448**		+ + +
				449**	20	+ COMMAND + ADDRESS OF TERMINAL LIST
				450**		+ CODE +
				451**		+-----+ +
				452**		+ RELATIVE +
				453**	24	+ STATUS + LINE + ADDRESS + VPC/LRC +
				454**		+ FLAGS + NUMBER + RESPONSE + RESPONSE +
				455**		+-----+ +
				456**		+ + +
				457**	28	+ TP-CP + ERROR + CSW STATUS +
				458**		+ CODE + STATUS +
				459**		+-----+ +
				460**		+ + +
				461**	32	+ RESERVED + ADDRESS OF CURRENT
				462**		+ ADDRESSING ENTRY +
				463**		+-----+ +
				464**		+ + +
				465**	36	+ RESERVED + ADDRESS OF CURRENT
				466**		+ POLLING ENTRY +
				467**		+-----+ +
				468**		+ + +
				469**	40	+ RESERVED + RESERVED + WRITE AREA LENGTH +
				470**		+ + +
				471**		+-----+ +
				472**		+ + +
				473**	44	+ RESERVED + ADDRESS OF WRITE AREA
				474**		+ + +
				475**		+-----+ +

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
00000				477+DECSDFCB	DS	1F STATUS FLAG + ADDRESS OF THE TC
00004				479+DECTYPE	DS	1F OPERATION TYPE
00006				481+DECLAGTH	DS	1F AREA LENGTH
00008				483+DECCALTT	DS	0011 RESERVED FOR ON-LINE TERMINAL TEST
00009				484+DECCCBAC	DS	1F ADDRESS OF DCR
0000C				486+DECFAREA	DS	1F ADDRESS OF AREA
00010				488+DECSFNSC	DS	1C 1ST SENSE BYTE
00011				490+DECSFNS1	DS	1C 2ND SENSE BYTE
00012				492+DECCOUNT	DS	1F RESIDUAL COUNT
00014				494+DECCMCCD	DS	0011 COMMAND CODE
00014				495+DECENTFY	DS	1F ADDRESS OF TERMINAL LIST
0001E				498+DECFLAGS	DS	1C STATUS FLAGS
00019				500+DECRLN	DS	1C RELATIVE LINE NUMBER
0001A				502+DECRESFN	DS	1F RESPONSE FIELDS
0001C				504+DECTPCCD	DS	1C TP-CP CODE
0001E				506+DECERRST	DS	1C ERROR STATUS
0001F				508+DECSWST	DS	1F CSW STATUS
00020				510+DECACRPT	DS	1F ADDRESS OF CURRENT ADDRESSING ENTRY
00024				512+DECFCLPT	DS	1F ADDRESS OF CURRENT POLLING ENTRY
0002F				514+	DS	2C RESERVED
0002A				516+DECKLNC	DS	1F WRITE AREA LENGTH
0002C				518+DECKAREA	DS	1F ADDRESS OF WRITE AREA

521 CCRD
 522 *,*** I-ACCP NO VALID DSRG SPECIFIED-EXCP ASSUMED

524** DCB SYMLIC DEFINITION FOR
 525** BASIC EXCP

00000 527+IHACCB DSFCT

LCC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT FRCSE65 10/06/70

531** FUNDATION BEFORE OPEN

00002F 533+ CPG I-ACCP+40
 00002F 534+DCEDCNAM DS CLR
 000030 535+DCBIFLGS DS RL1
 000031 536+DCBIFLGS DS RL1
 000032 537+DCBMACF DS RL2

535** FUNDATION AFTER OPEN

000028 541+ CPG I-ACCP+40
 000028 542+DCRTICT DS RL2
 00002A 543+DCBMACPF DS RL2
 00002C 544+DCBIFLGS DS JPL1
 00002C 545+DCCEBAC DS A

000030 546 CACP J.P

550 END

CROSS-REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES 10/06/70

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BASE	00005	0003FB	00325	0058 0055
BASEPFG	00001	000000	00045	0055 0056 0058
BRANCHT	00004	0002EA	00288	0287
BRPFG	00001	00000F	0004F	0148 0148 0180 0180 0206 0206 0240 0240 0270 0270
CLUSE	00005	000400	00331	0187
CCPAPAF	00006	00013A	00151	0184
CCF	00004	000508	00374	0085 0053 0123 0413
CCRFDNAM	00008	000028	00534	
CCRFEPAD	00004	000020	00545	
CCBIFLGS	00001	000031	00536	
CCBIFLGS	00001	000020	00544	
CCBMACF	00002	000032	00537	
CCBMACPF	00002	00002A	00543	
CCBIFLGS	00001	000030	00535	0056
CCBIFLGS	00001	000009	00044	0057 0055
CCRTICT	00002	000028	00542	
CCCADRPT	00004	000020	00510	
CCCAFBA	00004	000000	00486	
CCCFMCC	00001	000014	00494	
CCCCCLNT	00002	000012	00492	
CCCCSWST	00002	00001E	00508	
CCCCCFAD	00004	000008	00484	
CCCENTRY	00004	000014	00495	
CCCFRRST	00001	000010	00506	
CCCFLAGS	00001	000018	00498	
CCFLACTH	00002	000006	00481	
CCCOALTT	00001	000008	00483	
CCCPCLPT	00004	000024	00512	
CCCFRSPN	00002	00001A	00502	
CCCRLN	00001	000019	00500	
CCCSDECB	00004	000000	00477	
CCCFSENSG	00001	000010	00488	
CCCFSPAS1	00001	000011	00490	
CCCFCTCD	00001	000010	00504	
CCCFCTYPE	00002	000004	00475	
CCCFWAREA	00004	000020	00518	
CCCFWLNK	00002	00002A	00516	
CCCFAC	00001	000250	00246	0188
CCCFMSC	00004	000414	00332	0125 0158 0342
CCCFRCRT	00001	0003E6	00215	0116 0221 0279 02F3 0255
CCCFXAMPLE1	00001	000000	00001	
CCCFCTCEB	00001	000000	00426	
CCCFCTSE40	00001	000400	00350	0124
CCCFACRE	00001	0001R2	00186	0152 0156 0155 0252
CCCFIACCP	00001	000000	00527	0057 0523 0541

```

IMB0CC2 00CC1 000046 0CC65 0066
IMB0CC2A 00CC2 000046 0C07C 0065
IMB0CC3 00CC1 000079 0CC75 0076
IMB0CC3A 00CC2 00007A 0C08C 0075
IMB0CC5 00CC1 0000CA 0C1G5 0102
IMB0CC5A 00CC2 0000CA 00106 0101
IMB0CC1E 00CC1 00028F 0C254 0251
IMB0CC18A 00CC2 000290 00255 0250
IMB0CC22 00CC1 00C322 0C3C2 0259
IMB0CC22A 00002 000322 00303 0258

```

CRCSS-REFERENCE

```

SYMBOL  LEN  VALUE  CFFA  REFERENCES
INAREA  00004 000378 00314 0140 C151 C152 C155 C16C 017C 0187
ICCODE  00001 0003E7 00321 0087 C132 C162 0189 0224 0284
ICERPCR 00004 0002CC 00275 0149 C1E1 C2C7 C220 C241
LENGTH  00001 000CC2 00342 0120 C156
LCAD    00002 0003E4 0031E 0155
LCADCCNE 00001 0003F8 00326 0158 C182 C243
LCADMSG 00003 0004C9 00344 016C C172
CPEN    00001 000088 00C86 0288
CPENCK  00001 000CC6 0C115 0057
CUT     00001 00022C 0C225 0154 C25C
PERP    00001 0002FE 0C254 0271 C2E1
PREC    00002 0003E0 0C316 0151
READCEB 00004 00060C 004C5 0135 C165 C192 0215 0216 0225 0259
READYI  00001 0000FE 0C131 0209 0244 C2P5
RETREG  00001 0000CE 0CC47 C15C C182 C2C8 C212 C222 C223 C242 C272
REWIAC  00003 0004D6 0C342 0234 0244
SAVEARE 00004 000330 00313 0083 C0E4 C1C5 C272 C3C6
SAVEREG 00001 0000C0 00046 0083 00E4 C1C5 0273 03C6
START   00001 0000C0 0CC45
TEMP    00004 0003DC 0C315 0212 0222
TRTABLE 00016 0003F0 0033C 006C
UNLOAD  00002 0003E2 0C317 0153
LNPACK  00004 0003EC 00328 0055 C06C CC61
WAIT    00001 000200 0C211 015C C182 C2C8 C242 0272
WRKRREG 00001 000008 0CC42 0282 C2E2 C2E3 C2E4 C2E4 C2E5 C286 C287
WRITETCO 00001 000166 00161 0291
WTCP    00004 00001C 0CC65 0061

```

```

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
*STATISTICS* SOURCE RECORDS (SYSIN) = 237 SOURCE RECORDS (SYSLIB) = 4790
*OPTIONS IN EFFECT* LIST, DECK, NOLCAB, NCREAT, XREF, NCTEST, ALGN, OS, LINECNT = 55
469 PRINTED LINES

```

This Page Intentionally Left Blank

APPENDIX O: LOCAL 3270 SAMPLE PROGRAM

The local 3270 sample program shows how BTAM support for the local 3270 display system works and demonstrates some of the capabilities of the display system. The sample program can operate from one to 255 local 3270 devices, at least one of which must be a 3277 display station with a keyboard. If more than one device is used, some or all of the remaining devices can be 3284 or 3286 printers (although the number of display stations should equal or exceed the number of printers). The sample program can work with model 1 or model 2 devices or both.

The local 3270 sample program is distributed as a member of SYS1.SAMPLIB named SAMP327L. This PDS member contains (in the form of 80-character card images) all the source statements for the sample program and most of the JCL needed to assemble, link-edit, and execute the program.

DEFINING THE LOCAL 3270 DISPLAY SYSTEM

The local 3270 display system used by the sample program is defined in two ways. The data definition (DD) statement DD3270 (and any other DD statements concatenated with it) specifies the devices to be used and assigns relative line numbers to them. The PARM parameter for the execution of the program describes the size and composition of the display system.

The DD statement DD3270 should be added to the JCL for the GO step (see the examples below). The UNIT parameter, which is the only required parameter, specifies the device to be used by device name (3277, 3284, or 3286), by device address, or by a UNITNAME defined during system generation. If more devices than one are to be used, DD statements for the additional devices should be concatenated with DD3270. Print output generated by a display station is directed to the printer specified by the DD statement immediately preceding the DD statement for the display station. (DD statements for printers should be separated by one or more DD statements for display stations.) if no printer is associated with a display station in this way, print output from that display station is directed to SYSPRINT.

Example 1: One 3270 device is used; it is attached to address 240:

```
.  
.  
.  
//GO.SYSABEND DD SYSOUT=A  
//GO.DD3270 DD UNIT=240  
/*
```

Example 2: Three 3270 devices are used, two 3277 display stations and one 3286 printer; they are specified by device name:

```
.  
.  
.  
//GO.SYSABEND DD SYSOUT=A  
//GO.DD3270 DD UNIT=3277 relative line number 1  
// DD UNIT=3286 relative number 2  
// DD UNIT=3277 relative line number 3  
/*
```

Print output from the 3277 display station on relative line number 1 is directed to SYSPRINT. Print output from the 3277 display station on relative line number 3 is directed to the printer on relative line number 2.

The PARM parameter of the EXEC statement is used to specify the number of devices to be used, the number of printers to be used, and the relative line numbers of the printers. The format of the EXEC statement with the PARM parameter is:

```
// EXEC ASMFCLG,GO.PARM='number of devices[,number of printers  
[,printer rln]...]
```

Defaults and limits for these parameters are:

Number of devices: default = 1; minimum = 1; maximum = 255

Number of printers: default = 0; minimum = 0; maximum = one less than number of devices

Printer rln: if number of printers is zero = 0; if number of printers is not zero, default = 1; maximum = number of devices

Parameters are separated by commas. Any invalid characters are treated as zeros. If a parameter is longer than three characters, the rightmost three characters are used, and any others are ignored.

Example 3: One-device system with a 3277 display station and no printer (as specified in example 1):

```
// EXEC ASMFCLG
```

The EXEC statement supplied with the sample program can be used, since all the required values are defaults.

Example 4: A three-device system with two 3277 display stations and a printer on relative line number 2 (as specified in example 2):

```
// EXEC ASMFCLG,GO.PARM='3,1,2'
```

Example 5: A four-device system with three 3277 display stations and a printer on relative line number 1:

```
// EXEC ASMFCLG,GO.PARM='4,1'
```

The default for the relative line number of the printer is used.

Example 6: A two-device system with two 3277 display stations and no printer:

```
// EXEC ASMFCLG,GO.PARM='2'
```

OPERATING THE SAMPLE PROGRAM

When the sample program is initialized, the initial format shown in Figure 119 or 120 is displayed on each display station. The subsequent operation of the program is controlled by the display station operator by means of the ENTER, CLEAR, PA1, and PA2 keys.

Pressing the PA2 key (except when the ending format is displayed) causes the ending format shown in Figure 125 or 126 to be displayed. Further input from the display station is inhibited. After the PA2 key has been pressed on each display station, the sample program is ter-

minated. (Any unstarted printer operations are lost when the program is terminated.)

Pressing the CLEAR key (except when the ending format is displayed) causes the control options format shown in Figure 123 or 124 to be displayed.

The result of pressing the ENTER key or PA1 key depends on the format being displayed:

```
ENTER DATA REQUESTED BELOW:

NAME: _
ADDR: _
CITY:
STATE:                ZIP:

ENTER KEY: ENTER DATA;
PA2 KEY: END PROGRAM;
CLEAR KEY: CONTROL OPTIONS.
```

Figure 119. Initial Format on Model 1 3277 Display Station

```
ENTER DATA REQUESTED BELOW:
NAME: _                ADDR:                ZIP:
CITY:                STATE:

ENTER KEY: ENTER DATA;                PA2 KEY: END PROGRAM;
CLEAR KEY: CONTROL OPTIONS.
```

Figure 120. Initial Format on Model 2 3277 Display Station

```
DATA GIVEN BELOW ENTERED:

NAME:
ADDR:
CITY:
STATE:                ZIP:

ENTER KEY: UPDATE DATA;
PA1 KEY: PRINT DATA;
PA2 KEY: END PROGRAM;
CLEAR KEY: CONTROL OPTIONS.
```

Figure 121. Verification Format on Model 1 3277 Display Station

DATA GIVEN BELOW ENTERED:

NAME:	ADDR:	
CITY:	STATE:	ZIP:

ENTER KEY: UPDATE DATA;	PA1 KEY: PRINT DATA;
PA2 KEY: END PROGRAM;	CLEAR KEY: CONTROL OPTIONS.

Figure 122. Verification Format on Model 2 3277 Display Station

XXYY3CUU
 _

ENTER KEY: RESUME AND CONTINUE;
 PA1 KEY: BEGIN NEW ENTRY;
 PA2 KEY: END PROGRAM;

TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST
 MESSAGE OVER SAMPLE FORMAT ABOVE:
 XX=TEST NO. (23-28) YY=REPEATS (01-99)
 CUU=ADDRESS OF THE TARGET DEVICE
 THEN HIT ERASE EOF AND THEN TEST REQ.
 USE CLEAR KEY TO RESUME AFTER TEST.

Figure 123. Control Options Format on Model 1 3277 Display Station

XXYY3CUU

ENTER KEY: RESUME AND CONTINUE; PA1 KEY: BEGIN NEW ENTRY;
 PA2 KEY: END PROGRAM;

TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESSAGE OVER SAMPLE FORMAT ABOVE:
 XX=TEST NO. (23-28) YY=REPEATS (01-99) CUU=ADDRESS OF TARGET DEVICE
 THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KEY TO RESUME AFTER TEST.

Figure 124. Control Options on Model 2 3277 Display Station

1. Initial format (Figure 119 or 120): This is the first format displayed when the program is initialized. It can be redisplayed as described under the control options format. The CLEAR, PA2, and ENTER key are valid for this format.

There are five unprotected fields on the screen; the field following "ZIP:" is numeric-only. Data may be entered into one or more of these fields. Then the ENTER key should be pressed to transmit the data from the display station to the program, which initializes an internal data area associated with the display station.

The data is displayed as part of the verification format after the ENTER key has been pressed.

2. Verification format (Figure 121 or 122): This format displays the data in the data area for the display station. The display station operator can verify, modify, or print the data. The CLEAR, PA2, ENTER and PA1 key are valid for this format.

The ENTER key is used to transmit any modifications to the data to the program, which updates the data area for the display station.

The PA1 key is used to have the data in the data area printed.

The verification format is redisplayed after the ENTER or PA1 key has been pressed.

3. Control options format (Figure 123 or 124): This format is displayed after the CLEAR key has been pressed. The CLEAR, PA2, ENTER, and PA1 keys are valid for this format.

Pressing the ENTER key causes the verification format to be displayed.

Pressing the PA1 key clears the data area for the display station and causes the initial format to be displayed.

Also, the display station operator can enter a request for a BTAM online test (OLT) pattern by following the directions on the format. The pattern may be sent to any display station or printer being operated by the sample program. To continue, press the RESET key, then press the CLEAR key after an OLT pattern is sent or received to redisplay the control options format.

4. Ending format (Figure 125 or 126): This format is displayed after the PA1 key has been pressed.

Further input from the display station is inhibited.

If a key that is not valid is pressed, input from the display station is inhibited by the display station hardware and ignored by the sample program. To continue, the display station operator should press the RESET key to manually enable the keyboard; he should then press a valid key for the format being displayed.

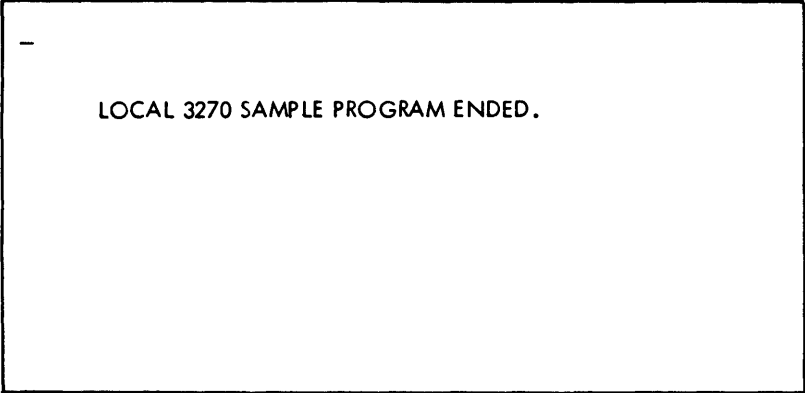


Figure 125. Ending Format on Model 1 3277 Display Station

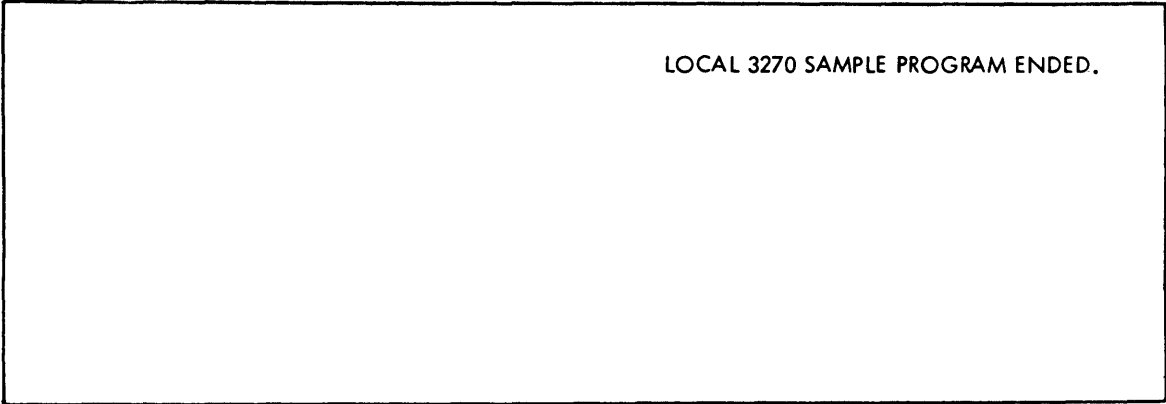


Figure 126. Ending Format on Model 3277 Display Station

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
000000				2	SAMP327L CSECT	
				3	* STATUS - CHANGE LEVEL 000	
				4	*	
				5	* ENTRY POINT - SAMP327L	
				6	*	
				7	* INPUT - REGISTERS 1, 13, AND 14 PROVIDE INPUT AS FOLLOWS	
				8	* 1 - ADDRESS OF PARM AREA POINTER	
				9	* 13- ADDRESS OF SAVE AREA	
				10	* 14- RETURN ADDRESS	
				11	*	
				12	* THE PARM AREA HAS THE FOLLOWING FORMAT	
				13	* +0 LENGTH FIELD (HALF WORD)	
				14	* +2 NO. OF DEVICES IN LINE GROUP (1-3 DIGITS, 1-255, 1=DEFAULT)	
				15	* NO. OF PRINTERS IN LINE GROUP (1-3 DIGITS, NO. DEVICES, 0=DEF)	
				16	* RLN'S OF PRINTERS (1-3 DIGITS, NO LARGER THAN NO. OF DEVICES,	
				17	* DEFAULT OF 1 IF NO. OF PRINTERS =1)	
				18	* ALL FIELDS SEPARATED BY COMMAS	
				19	*	
				20	* DDNAME FOR 3270 DEVICES = DD3270	
				21	* DD CARDS ARRANGED - PRINTER, DISPLAYS, PRINTER, DISPLAYS, ETC.	
				22	* THOSE FOR PRINTERS OMITTED IF NO PRINTERS	
				23	*	
				24	* SYSPRINT DD SYSOUT=A REQUIRED	
				25	*	
				26	* OUTPUT - DISPLAY OUTPUT, PRINTER OUTPUT	
				27	*	
		00000		28	ZEROREG EQU 0	
		00001		29	PARMREG EQU 1	PARAMETER REGISTER
		00002		30	EVENREG EQU 2	
		00003		31	ODDREG EQU 3	
		00004		32	LOOPREG EQU 4	
		00005		33	INDXREG EQU 5	
		00006		34	CNTREG EQU 6	
		00006		35	RLNREG EQU CNTREG	
		00007		36	LOOPREG2 EQU 7	
		00007		37	DECBREG EQU 7	
		00008		38	INDXREG2 EQU 8	
		00008		39	BUFREG EQU 8	
		00009		40	TABBASE EQU 9	BASE FOR CONTROL TABLE
		0000A		41	BASEREG2 EQU 10	
		0000B		42	BASEREG EQU 11	CSECT BASE REGISTER
		0000D		43	SAVEREG EQU 13	
		0000E		44	RETNREG EQU 14	
		0000F		45	EPREG EQU 15	
				46	*	
		00005		47	TYPE EQU 5	DISP TO TYPE FIELD IN DECB
		00012		48	COUNT EQU 18	DISP TO RESIDUAL COUNT IN DECB
		00019		49	RLN EQU 25	DISP TO RLN FIELD IN DECB
		00024		50	POLPT EQU 36	DISP TO POLPT FIELD IN DECB
				51	*	
				52	SAVE (14,12),T,*	SAVE REGISTER
000000	47F0 F00E		0000E	53+	B 14(0,15)	BRANCH AROUND ID
000004	08			54+	DC AL1(8)	
000005	E2C1D4D7F3F2F7D3			55+	DC CL8'SAMP327L'	IDENTIFIER
00000D	00					

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
00000E	90EC D00C	0000C		56+	STM 14,12,12(13)	SAVE REGISTERS
000012	05B0			57	BALR BASEREG,0	ESTABLISH BASE REG
			00014	58	USING *,BASEREG,BASEREG2	& ADDRESSABILITY
000014	41AB 0FFF	00FFF		59	LA BASEREG2,4095(BASEREG)	INITIALIZE SECOND BASE
000018	41AA 0001	00001		60	LA BASEREG2,1(BASEREG2)	
00001C	50D0 BASC	00A70		61	ST SAVEREG,SAVE+4	ESTABLISH
000020	4120 BA58	00A6C		62	LA EVENREG,SAVE	SAVE AREA
000024	502D 0008	00008		63	ST EVENREG,8(SAVEREG)	CHAIN
000028	18D2			64	LR SAVEREG,EVENREG	ESTABLISH NEW SAVE AREA
				65 *		
00002A	5831 0000	00000		66	L ODDREG,0(PARMREG)	GET ADDR OF PARM AREA
00002E	4843 0000	00000		67	LH LOOPREG,0(ODDREG)	GET LENGTH OF PARM AREA
000032	1244			68	LTR LOOPREG,LOOPREG	PARMS PASSED
000034	4780 B054	00068		69	BZ DEFAULTS	NO, TAKE DEFAULTS
000038	4153 0002	00002		70	LA INDXREG,2(ODDREG)	INITIALIZE POINTER TO PARM STRING
00003C	45E0 B918	0092C		71	BAL RETNREG,GETPARM	GET NUMBER OF DEVICES
000040	1211			72	LTR PARMREG,PARMREG	VALUE LESS THAN 1
000042	47D0 B03E	00052		73	BNP DFLTNO	YES, TREAT AS 1
000046	5910 BAA0	00AB4		74	C PARMREG,MAXRLN	VALUE GREATER THAN 255
00004A	4720 B03E	00052		75	BH DFLTNO	YES, TREAT AS 1
00004E	5010 BAA4	00AB8		76	ST PARMREG,NODVCS	SAVE NUMBER OF DEVICES
				77 *		
000052	1244			78	PARM2 LTR LOOPREG,LOOPREG	ANY MORE PARMS
			00052	79	DFLTNO EQU PARM2	DEFAULT NO OF DVCS = 1
000054	4780 B054	00068		80	BZ DFLTPTR	NO, DEFAULT NUMBER OF PRINTERS
				81 *		
000058	45E0 B918	0092C		82	BAL RETNREG,GETPARM	GET NUMBER OF PRINTERS
00005C	5910 BAA4	00AB8		83	C PARMREG,NODVCS	NUMBER OF PRINTERS LESS THAN
000060	47B0 B054	00068		84	BNL DFLTPTR	NUMBER OF DEVICES
000064	5010 BAA8	00ABC		85	ST PARMREG,NOPTRS	SAVE NUMBER OF PRINTERS
				86 *		
000068	5820 BAA4	00AB8		87	GETTAB L EVENREG,NODVCS	GET SIZE OF TERMINAL ENTRIES
				88 *		
			00068	89	DEFAULTS EQU GETTAB	
			00068	90	DFLTPTR EQU GETTAB	DEFAULT NO OF PTRS = 0
				91 *		
00006C	8B20 0003	00003		92	SLA EVENREG,3	
000070	5830 BAA8	00ABC		93	L ODDREG,NOPTRS	GET SIZE OF PRINTER ECB'S
000074	8B30 0002	00002		94	SLA ODDREG,2	
000078	4102 3170	00170		95	LA ZEROREG,368(EVENREG,ODDREG)	GET SIZE OF CONTROL TABLE
00007C	5000 BAAC	00AC0		96	ST ZEROREG,TABSIZE	SAVE SIZE
				97	GETMAIN R,LV=(0)	
000080	4510 B070	00084		98+	BAL 1,++4	INDICATE GETMAIN
000084	0A0A			99+	SVC 10	ISSUE GETMAIN SVC
000086	1891			100	LR TABBASE,PARMREG	GET ADDR OF CONTROL TABLE
000088	D201 9000 BAA6	00000 00ABA		101	MYC 0(2,TABBASE),NODVCS+2	SAVE NO. OF DEVICES
00008E	D701 9002 9002	00002 00002		102	XC 2(2,TABBASE),2(TABBASE)	CLEAR ACTIVE NO. OF DEVICES
000094	4112 9010	00010		103	LA PARMREG,16(EVENREG,TABBASE)	GET ADDR OF ECB LIST
000098	5019 0004	00004		104	ST PARMREG,4(TABBASE)	& SAVE IT
00009C	4111 3004	00004		105	LA PARMREG,4(PARMREG,ODDREG)	GET ADDR OF DECB
0000A0	5019 0008	00008		106	ST PARMREG,8(TABBASE)	& SAVE IT
0000A4	4111 0028	00028		107	LA PARMREG,40(PARMREG)	GET ADDR OF BUFFER
0000A8	5019 000C	0000C		108	ST PARMREG,12(TAB ASE)	& SAVE IT
				109 *		
0000AC	4112 315F	0015F		110	LA PARMREG,351(EVENREG,ODDREG)	GET LENGTH TO CLEAR

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
0000B0	9200 9010	00010		111	MVI 16(TABBASE),0	CLEAR FIRST BYTE
0000B4	4129 0010	00010		112	LA EVENREG,16(TABBASE)	INITIALIZE ADDRESS POINTER
0000B8	5910 BAB0	00AC4		113	CLRLOOP C PARMREG,F256	MOVE LENGTH OVER 256
0000BC	4740 B0C2	000D6		114	BL LASTMOVE	NO, DO LAST MOVE
0000C0	D2FF 2001	2000 00001 00000		115	MVC 1(256,EVENREG),0(EVENREG)	CLEAR 256 BYTES
0000C6	4122 0100	00100		116	LA EVENREG,256(EVENREG)	INCREMENT ADDRESS POINTER
0000CA	5B10 BAB0	00AC4		117	S PARMREG,F256	DECR MOVE COUNT
0000CE	4770 B0A4	000B8		118	BNZ CLRLOOP	REPEAT IF MORE TO DO
0000D2	47F0 B0D2	000E6		119	B GETPTRS	GET PRINTER RLNS
				120	*	
0000D6	0610			121	LASTMOVE BCTR PARMREG,0	DECR COUNT FOR EXECUTE
0000D8	4410 B0CC	000E0		122	EX PARMREG,CLRMOVE	CLEAR REST OF AREA
0000DC	47F0 B0D2	000E6		123	B GETPTRS	GET PRINTER RLNS
				124	*	
0000E0	D200 2001	2000 00001 00000		125	CLRMOVE MVC 1(0,EVENREG),0(EVENREG)	
				126	*	
				127	*	
0000E6	1877			128	GETPTRS SR LOOPREG2,LOOPREG2	INITIALIZE COUNTER
0000E8	1888			129	SR INDXREG2,INDXREG2	INITIALIZE PRINTER ECB INDEX
0000EA	5970 BAA8	00ABC		130	PTRLOOP C LOOPREG2,NOPTRS	ALL PTR RLNS OBTAINED
0000EE	47B0 B122	00136		131	BNL FININIT	YES, COMPLETE INITIALIZATION
0000F2	1244			132	ENDLIST LTR LOOPREG,LOOPREG	ANY MORE PARMS
0000F4	4780 B114	00128		133	BZ DFLTRLN	NO, TAKE DEFAULTS
0000F8	45E0 B918	0092C		134	BAL RETNREG,GETPARM	
0000FC	1211			135	LTR PARMREG,PARMREG	RLN VALID
0000FE	4780 B0DE	000F2		136	BZ ENDLIST	NO, IGNORE IT
000102	5910 BAA4	00AB8		137	C PARMREG,NODVCS	
000106	4720 B0DE	000F2		138	BH ENDLIST	
00010A	8B10 0003	00003		139	INITPTR SLA PARMREG,3	GET INDEX TO ENTRY
00010E	4119 1008	00008		140	LA PARMREG,8(TABBASE,PARMREG)	GET ADDR OF ENTRY FOR PTR
000112	D201 1000	BB60 00000 00B74		141	MVC 0(2,PARMREG),MIN1	FREE PTR FORMAT
000118	4081 0002	00002		142	STH INDXREG2,2(PARMREG)	INDEX TO ECB PTR
00011C	4188 0004	00004		143	LA INDXREG2,4(INDXREG2)	INCR TO NEXT ECB
000120	4177 0001	00001		144	LA LOOPREG2,1(LOOPREG2)	INCR PRINTER COUNTER
000124	47F0 B0D6	000EA		145	B PTRLOOP	
				146	*	
				147	*	
000128	1277			148	DFLTRLN LTR LOOPREG2,LOOPREG2	NO PTR RLNS PROCESSED
00012A	4770 B122	00136		149	BNZ FININIT	NO, COMPLETE INITIALIZATION
00012E	4110 0001	00001		150	LA PARMREG,1	YES, DEFAULT RLN = 1
000132	47F0 B0F6	0010A		151	B INITPTR	
				152	*	
				153	*	
000136	5810 BAA4	00AB8		154	FININIT L PARMREG,NODVCS	GET NUMBER OF DISPLAYS
00013A	1841			155	LR LOOPREG,PARMREG	SAVE LOOP COUNTER
00013C	5B10 BAA8	00ABC		156	S PARMREG,NOPTRS	
000140	5C00 BAB4	00AC8		157	M ZEROREG,DATASZ	GET SIZE OF DATA AREAS
000144	1801			158	LR ZEROREG,PARMREG	
000146	5000 BAB8	00ACC		159	ST ZEROREG,DATBLKSZ	SAVE DATA AREA SIZE
				160	GETMAIN R,LV=(0)	GET CORE FOR DATA AREAS
00014A	4510 B13A	0014E		161+	BAL 1,*+4	INDICATE GETMAIN
00014E	0A0A			162+	SVC 10	ISSUE GETMAIN SVC
000150	5010 BABC	00AD0		163	ST PARMREG,DATBLKAD	SAVE DATA AREA ADDR
000154	1851			164	LR INDXREG,PARMREG	GET ADDRESS OF AREA
000156	4189 0010	00010		165	LA INDXREG2,16(TABBASE)	GET ADDR OF ENTRY

XA01560
XA01560

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
00015A	1B66			166	SR CNTREG,CNTREG	PRINTER RLN
00015C	4170 0001	00001		167	LA LOOPREG2,1	RLN COUNTER
				168 *		
000160	D501 8000	BB60	00000	00B74	169 INITLOOP CLC 0(2,INDXREG2),MIN1	PRINTER ENTRY
000166	4780 B1A0	001B4		170	BE PTRINIT	YES, GET RLN
00016A	D781 5000	5000	00000	00000	171 XC 0(130,INDXREG),0(INDXREG)	CLEAR DATA AREA
000170	4068 0002	00002		172	STH CNTREG,2(INDXREG2)	STORE PRINTER RLN FOR THIS DISPLAY
000174	5058 0004	00004		173	ST INDXREG,4(INDXREG2)	STORE ADDR OF DATA AREA
000178	5A50 BAA4	00AC8		174	A INDXREG,DATASZ	INCR DATA AREA PTR
00017C	4188 0008	00008		175	INCRPTRS LA INDXREG2,8(INDXREG2)	INCR ENTRY PTR
000180	4177 0001	00001		176	LA LOOPREG2,1(LOOPREG2)	INCR RLN
000184	4640 B14C	00160		177	BCT LOOPREG,INITLOOP	BR IF MORE TO DO
000188	5819 0004	00004		178	L PARMREG,4(TABBASE)	GET ADDR OF ECB LIST
00018C	5850 BAA8	00ABC		179	L INDXREG,NOPTRS	GET INDEX PAST PRINTER ECB
000190	1255			180	LTR INDXREG,INDXREG	
000192	4780 B192	001A6		181	BZ DISPECB	
000196	4120 BAD8	00AEC		182	LA EVENREG,DUMMYECB	
00019A	5021 0000	00000		183	PTRECB ST EVENREG,0(PARMREG)	
00019E	4111 0004	00004		184	LA PARMREG,4(PARMREG)	
0001A2	4650 B186	0019A		185	BCT INDXREG,PTRECB	
0001A6	9280 1000	00000		186	DISPECB MVI 0(PARMREG),X'80'	
0001AA	D202 1001	9009	00001	00009	187 MVC 1(3,PARMREG),9(TABBASE)	MOVE IN ADDR OF ECB
0001B0	47F0 B1A8	001BC		188	B IOINIT	
0001B4	1867			189	PTRINIT LR CNTREG,LOOPREG2	GET RLN OF PRINTER
0001B6	47F0 B168	0017C		190	B INCRPTRS	
				191	IOINIT OPEN (SYSPRINT,OUTPUT,DD3270)	
0001BA	0700			192+	CNOP 0,4	
0001BC	4510 B1B4	001C8		193+	IOINIT BAL 1,*+12	ALIGN LIST TO FULLWORD
0001C0	0F			194+	DC AL1(15)	LOAD REG1 W/LIST ADDR.
0001C1	000AF0			195+	DC AL3(SYSPRINT)	OPTION BYTE
0001C4	80			196+	DC AL1(128)	DCB ADDRESS
0001C5	000B3C			197+	DC AL3(DD3270)	OPTION BYTE
0001C8	0A13			198+	SVC 19	DCB ADDRESS
				199	WTO 'LOCAL 3270 SAMPLE PROGRAM RUNNING'	ISSUE OPEN SVC
0001CA	0700			200+	CNOP 0,4	
0001CC	4510 B1E2	001F6		201+	BAL 1,IHB0005A	BRANCH AROUND MESSAGE
0001D0	0025			202+	DC AL2(37)	TEXT LENGTH
0001D2	0000			203+	DC B'0000000000000000'	MCS FLAGS
0001D4	D3D6C3C1D340F3F2			204+	DC C'LOCAL 3270 SAMPLE PROGRAM RUNNING'	
0001F6				205+	IHB0005A DS 0H	
0001F6	0A23			206+	SVC 35	
0001F8	5840 BAA4	00AB8		207	L LOOPREG,NODVCS	GET LOOP CTR
0001FC	5879 0008	00008		208	L DECBREG,8(TABBASE)	GET ADDR OF DECB
000200	5889 000C	0000C		209	L BUFREG,12(TABBASE)	GET ADDR OF BUFFER
000204	4159 0010	00010		210	LA INDXREG,16(TABBASE)	GET ADDR OF FIRST TERM ENTRY
000208	4160 0001	00001		211	LA RLNREG,1	INITIALIZE RLN REG
				212 *		
00020C	41F0 B6F6	0070A		213	IOLOOP LA EPREG,WRTFMT1	GET ADDR OF FIRST WRITE RTN
000210	D501 5000	BB60	00000	00B74	214 CLC 0(2,INDXREG),MIN1	PRINTER ENTRY
000216	4780 B21A	0022E		215	BE UPDCOUNT	YES, DON'T DO WRITE
00021A	1816			216	LR PARMREG,RLNREG	PUT RLN IN REG 1
00021C	05EP			217	BALR RETNREG,EPREG	WRITE FIRST MSG
00021E	12FF			218	LTR EPREG,EPREG	SUCCESSFUL
000220	4780 B21A	0022E		219	BZ UPDCOUNT	YES, TRY NEXT
000224	D701 5000	5000	00000	00000	220 XC 0(2,INDXREG),0(INDXREG)	MARK TERMINAL INACTIVE

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	
00022A	47F0 B226	0023A		221	B	IOLPCNT	
				222	*		
00022E	4829 0002	00002		223	UPCOUNT LH	EVENREG,2(TABBASE)	INCREMENT
000232	4122 0001	00001		224	LA	EVENREG,1(EVENREG)	COUNT OF ACTIVE
000236	4029 0002	00002		225	STH	EVENREG,2(TABBASE)	DEVICES
00023A	4166 0001	00001		226	IOLPCNT LA	RLNREG,1(RLNREG)	INCR RLN
00023E	4155 0008	00008		227	LA	INDXREG,8(INDXREG)	INCR TO NEXT ENTRY
000242	4640 B1F8	0020C		228	BCT	LOOPREG,IOLOOP	LOOP IF MORE TO DO
				229	*		
000246	4160 0001	00001		230	LA	CNTREG,1	INIT RLN
00024A	4159 0010	00010		231	LA	INDXREG,16(TABBASE)	POINTER TO FIRST ENTRY
00024E	5840 BAA4	00AB8		232	L	LOOPREG,NODVCS	LOOP LIMIT
000252	D501 5000	BB60 00000	00B74	233	FINDFRST CLC	0(2,INDXREG),MIN1	PRINTER
000258	4770 B258	0026C		234	BNE	DSPLY1	NO, SAVE RLN
00025C	4166 0001	00001		235	LA	CNTREG,1(CNTREG)	INCR RLN
000260	4155 0008	00008		236	LA	INDXREG,8(INDXREG)	INCR TO NEXT ENTRY
000264	4640 B23E	00252		237	BCT	LOOPREG,FINDFRST	
000268	4160 0001	00001		238	LA	CNTREG,1	
00026C	5060 BAC0	00AD4		239	DSPLY1 ST	CNTREG,READRLN	SAVE RLN FOR READS
				240	*		
000270	.5860 BAC0	00AD4		241	READ L	RLNREG,READRLN	GET READ RLN
000274	D703 7000	7000 00000	00000	242	XC	0(4,DECBREG),0(DECBREG)	
				243	READ	(DECBREG),TI,DD3270,(BUFREG),308,,(RLNREG),MF=E	
00027A	1817			244+	LR	1,DECBREG	LOAD DECB ADDRESS
00027C	9404 1004	00004		245+	NI	4(1),4	
000280	9201 1005	00005		246+	MVI	5(1),1	
000284	41E0 0134	00134		247+	LA	14,308(0,0)	
000288	40E0 1006	00006		248+	STH	14,6(0,1)	STORE LENGTH 000B
00028C	41E0 BB28	00B3C		249+	LA	14,DD3270	
000290	50E0 1008	00008		250+	ST	14,8(0,1)	STORE DCB ADDRESS
000294	5080 100C	0000C		251+	ST	BUFREG,12(0,1)	STORE AREA ADDR 000B
000298	4060 1018	00018		252+	STH	RLNREG,24(0,1)	STORE LINE NUMBER
00029C	58F0 BB58	00B6C		253+	L	15,DD3270+48	LOAD RDWRT ROUT ADDR
0002A0	05EF			254+	BALR	14,15	A38557
0002A2	12FF			255	LTR	EPREG,EPREG	OK
0002A4	4770 B90A	0091E		256	BNZ	ERRABEND	NO, ABEND
0002A8	5839 0004	00004		257	TWAIT L	ODDREG,4(TABBASE)	GET ADDR OF ECB LIST
				258	TWAIT	(EVENREG),ECBLIST=(ODDREG)	WAIT FOR OPERATION TO END
0002AC				259+	DS	0H	
				260+	IECA0008 EQU	*	
0002AC	4113 0000	00000		261+	LA	1,0(ODDREG)	000A
0002B0	1801			262+	LR	0,1	SAVE PARAMETER LIST ADDRESS.
0002B2	58F1 0000	00000		263+	IECB0008 L	15,0(1)	LOAD ECB ADDRESS.
0002B6	9140 F000	00000		264+	TM	0(15),X'40'	IS COMPLETE BIT ON?
0002BA	4770 B2CE	002E2		265+	BNZ	IECC0008	
0002BE	9180 1000	00000		266+	TM	0(1),X'80'	IF NOT, TEST FOR END OF LIST.
0002C2	4710 B2BA	002CE		267+	BO	IECD0008	
0002C6	4111 0004	00004		268+	LA	1,4(1)	STEP TO NEXT ENTRY IN LIST.
0002CA	47F0 B29E	002B2		269+	B	IECB0008	
0002CE	1810			270+	IECD0008 LR	1,0	RESTORE PARAMETER LIST REGISTER.
0002D0	4100 0001	00001		271+	LA	0,1	LOAD WAIT COUNT.
0002D4	4110 1000	00000		272+	LA	1,0(0,1)	CLEAR HIGH BYTE OF REG
0002D8	1800			273+	LR	0,0	LOAD PARAMETER REG 0
0002DA	1311			274+	LCR	1,1	INDICATE ECBLIST USED
0002DC	0A01			275+	SVC	1	LINK TO WAIT ROUTINE

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
0002DE	47F0 B298	002AC		276+	B IECA0008
0002E2	412F 0000	00000		277+IECC0008	LA (EVENREG),0(15) IF ON, LOAD SPECIFIED REGISTER WITH ADDRESS OF COMPLETED ECB
0002E6	18F1			278+*	
0002E8	1BF0			279+	LR 15,1
				280+	SR 15,0
				281 *	
0002EA	9501 2005	00005		282	CLI TYPE(EVENREG),X'01' OPERATION = READ INITIAL
0002EE	4770 B76A	0077E		283	BNE PRNTREND NO, PRINTER OPERATION
0002F2	9544 2000	00000		284	CLI 0(EVENREG),X'44' INTERCEPTED
0002F6	4780 B25C	00270		285	BE READ YES, RETRY READ
0002FA	957F 2000	00000		286	CLI 0(EVENREG),X'7F' SUCCESSFUL
0002FE	4770 B90A	0091E		287	BNE ERRABEND NO, END
000302	5812 0024	00024		288	L PARMREG,POLPT(EVENREG) GET RESPONDING RLN
000306	1831			289	LR ODDREG,PARMREG
000308	8B30 0003	00003		290	SLA ODDREG,3 GET INDEX TO TERM ENTRY
00030C	4869 3008	00008		291	LH CNTREG,8(TABBASE,ODDREG) GET FORMAT NO.
000310	8B60 0002	00002		292	SLA CNTREG,2 MULTIPLY BY 4
000314	47F6 B304	00318		293	B READRTN(CNTREG)
000318	47F0 B90A	0091E		294	B ERRABEND RLN = INACTIVE DEVICE
00031C	47F0 B318	0032C		295	B READFMT1 FORMAT 1 ON SCREEN
000320	47F0 B5C2	005D6		296	B READFMT2 FORMAT 2 ON SCREEN
000324	47F0 B6B4	006C8		297	B READFMT3 FORMAT 3 ON SCREEN
000328	47F0 B25C	00270		298	B READ FORMAT 4 ON SCREEN - IGNORE IT
				299 *	
				300 *	
00032C	956E 8000	00000		301	READFMT1 CLI 0(BUFREG),X'6E' INTERRUPT IS PA2 KEY
000330	4770 B412	00426		302	BNE CLR1 NO, TEST FOR CLEAR KEY
				303 *	
000334	1861			304	WRTFMT4 LR RLNREG,PARMREG SAVE RLN
		00334		305	ENDMSG EQU WRTFMT4 RETURN = REESTABLISH READ
000336	8B10 0003	00003		306	SLA PARMREG,3 CONVERT RLN TO TABLE INDEX
00033A	4120 0004	00004		307	LA EVENREG,4 FORMAT ID
00033E	4029 1008	00008		308	STH EVENREG,8(TABBASE,PARMREG) STORE ID IN TABLE
				309	CHGNTRY DD3270,ATTLST,(RLNREG),,SKIP DEACTIVATE TERMINAL
000342				310+	DS 0H
000342	4110 BB28	00B3C		311+	LA 1,DD3270
000346	58F1 002C	0002C		312+	L 15,44(1) GET DEB ADDR
00034A	58EF 0018	00018		313+	L 14,24(15) GET DCB ADDR FROM DEB
00034E	41EE 0000	00000		314+	LA 14,0(14) CLEAR HIGH-ORDER BYTE
000352	4111 0000	00000		315+	LA 1,0(1) CLEAR HIGH-ORDER BYTE
000356	191E			316+	CR 1,14 DCB-DEB LOOP COMPLETE
000358	4770 B36C	00380		317+	BNE IECA0010 NO, GIVE RETURN CODE = 8
00035C	181F			318+	LR 1,15 DEB ADDR TO REG 1
00035E	18F6			319+	LR 15,RLNREG GET RLN
000360	1BEE			320+	SR 14,14
000362	43E1 0010	00010		321+	IC 14,16(1) GET NUMBER OF EXTENTS
000366	19FE			322+	CR 15,14 IS RLN VALID
000368	4720 B36C	00380		323+	BH IECA0010 NO, GIVE RETURN CODE = 8
00036C	8BF0 0002	00002		324+	SLA 15,2 MULTIPLY RLN BY 4
000370	581F 101C	0001C		325+	L 1,28(15,1) GET ADDR OF UCB FROM DEB
000374	1B00			326+	SR 0,0
000376	41F0 0002	00002		327+	LA 15,2 FUNCTION IS SKIP
00037A	0A74			328+	SVC 116 INVOKE ESR
00037C	47F0 B370	00384		329+	B IECB0010 BRANCH TO EXIT
000380	41F0 0008	00008		330+IECA0010	LA 15,8 SET RETURN CODE OF 8

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
				00384	331+IECB0010 EQU *	
				332 *		FOR READ INITIAL
000384	D703 7000 7000	00000	00000	333	WRITE# XC 0(4,DECBREG),0(DECBREG) CLEAR ECB	
				334	WRITE (DECBREG),TS,DD3270,FORMAT#,FMT#SZ,,(RLNREG),MF=E	
00038A	1817			335+	LR 1,DECBREG	LOAD DECB ADDRESS
00038C	9404 1004	00004		336+	NI 4(1),4	
000390	920E 1005	00005		337+	MVI 5(1),14	
000394	47F0 B386	0039A		338+	B *+6	000B
000398	0024			339+	DC AL2(FMT#SZ)	000B
00039A	D201 1006 B384	00006	00398	340+	MVC 6(2,1),*-2	MOVE IN LENGTH 000B
0003A0	41E0 BB28	00B3C		341+	LA 14,DD3270	
0003A4	50E0 1008	00008		342+	ST 14,8(0,1)	STORE DCB ADDRESS
0003A8	41E0 BE9A	00EAE		343+	LA 14,FORMAT4	000B
0003AC	50E0 100C	0000C		344+	ST 14,12(0,1)	STORE AREA ADDR 000B
0003B0	4060 1018	00018		345+	STH RLNREG,24(0,1)	STORE LINE NUMBER
0003B4	58F0 BB58	00B6C		346+	L 15,DD3270+48	LOAD RDWRT ROUT ADDR
0003B8	05EF			347+	BALR 14,15	A38557
0003BA	12FF			348	LTR EPREG,EPREG	END IF RC
0003BC	4770 B90A	0091E		349	BNZ ERRABEND	NON-ZERO
				350	WAIT ECB=(DECBREG)	
0003C0	4110 7000	00000		351+	LA 1,0(0,DECBREG)	CLEAR HIGH BYTE OF REG
0003C4	4100 0001	00001		352+	LA 0,1(0,0)	COUNT OMITTED,1 USED
0003C8	0A01			353+	SVC 1	LINK TO WAIT ROUTINE
0003CA	957F 7000	00000		354	CLI 0(DECBREG),X'7F'	SATISFACTORY COMPLETION
0003CE	4780 B3CA	003DE		355	BE QUIESCE	UPDATE ACTIVE TERM COUNT
0003D2	9544 7000	00000		356	CLI 0(DECBREG),X'44'	INTERCEPT
0003D6	4780 B370	00384		357	BE WRITE#	YES, RETRY
0003DA	47F0 B90A	0091E		358	B ERRABEND	NO, END
				359 *		
				360 *		
0003DE	4829 0002	00002		361	QUIESCE LH EVENREG,2(TABBASE)	GET NO. OF ACTIVE TERMS
0003E2	4620 B40A	0041E		362	BCT EVENREG,QCNTSTR	DECR COUNT
				363	CLOSE (SYSPRINT,,DD3270)	IF ZERO, CLOSE DCB'S
0003E6	0700			364+	CNOP 0,4	ALIGN LIST TO FULLWORD
0003E8	4510 B3E0	003F4		365+	BAL 1,*+12	LOAD REG1 W/LIST ADDR
0003EC	00			366+	DC AL1(0)	OPTION BYTE
0003ED	000AF0			367+	DC AL3(SYSPRINT)	DCB ADDRESS
0003F0	80			368+	DC AL1(128)	OPTION BYTE
0003F1	000B3C			369+	DC AL3(DD3270)	DCB ADDRESS
0003F4	0A14			370+	SVC 20	ISSUE CLOSE SVC
0003F6	5810 BABC	00AD0		371	L PARMREG,DATBLKAD	LOCATION OF DATA AREAS
0003FA	5800 BAB8	00ACC		372	L ZEROREG,DATBLKSZ	SIZE OF DATA AREAS
				373	FREEMAIN R,LV=(0),A=(1)	FREE DATA AREAS
0003FE	4111 0000	00000		374+	LA 1,0(1,0)	CLEAR THE HIGH ORDER BYTE XM4571
000402	0A0A			375+	SVC 10	ISSUE FREEMAIN SVC P2504
000404	1819			376	LR PARMREG,TABBASE	LOCATION OF CONTROL TABLE
000406	5800 BAAC	00AC0		377	L ZEROREG,TABSIZE	SIZE OF CONTROL TABLE
				378	FREEMAIN R,LV=(0),A=(1)	FREE CONTROL TABLE
00040A	4111 0000	00000		379+	LA 1,0(1,0)	CLEAR THE HIGH ORDER BYTE XM4571
00040E	0A0A			380+	SVC 10	ISSUE FREEMAIN SVC P2504
000410	58D0 BA5C	00A70		381	L SAVEREG,SAVE+4	
				382	RETURN (14,12),T	RETURN TO CALLING PROGRAM
000414	98EC D00C	0000C		383+	LM 14,12,12(13)	RESTORE THE REGISTERS
000418	92FF D00C	0000C		384+	MVI 12(13),X'FF'	SET RETURN INDICATION
00041C	07FE			385+	BR 14	RETURN

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
00041E	4029 0002	00002		386 *	QCNTSTR	STH EVENREG,2(TABBASE) SAVE UPDATE ACTIVE TERM CNT
000422	47F0 B25C	00270		387	B	READ READ SET UP READ
				388		
				389 *		
				390 *		
000426	956D 8000	00000		391	CLR1	CLI 0(BUFREG),X'6D' CLEAR KEY
00042A	4770 B482	00496		392	BNE	ENT1 NO, TEST FOR ENTER KEY
				393 *		
00042E	1861			394	WRTFMT3	LR RLNREG,PARMREG SAVE RLN
		0042E		395	CNTRLMSG	EQU WRTFMT3
				396 *		
000430	8B10 0003	00003		397	SLA	PARMREG,3 WRITE CONTROL OPTIONS
000434	4120 0003	00003		398	LA	EVENREG,3 CONVERT RLN TO TABLE INDEX
000438	4029 1008	00008		399	STH	EVENREG,8(TABBASE,PARMREG) STORE ID IN TABLE
00043C	D703 7000 7000	00000 00000		400	WRITE3	XC 0(4,DECBREG),0(DECBREG) CLEAR ECB
				401	WRITE	(DECBREG),TS,DD3270,FORMAT3,FMT3SZ,,(RLNREG),MF=E
000442	1817			402+	LR	1,DECBREG LOAD DECB ADDRESS
000444	9404 1004	00004		403+	NI	4(1),4
000448	920E 1005	00005		404+	MVI	5(1),14
00044C	47F0 B43E	00452		405+	B	*+6
000450	0148			406+	DC	AL2(FMT3SZ) 000B
000452	D201 1006 B43C	00006 00450		407+	MVC	6(2,1),*-2 MOVE IN LENGTH 000B
000458	41E0 BB28	0CB3C		408+	LA	14,DD3270
00045C	50E0 1008	00008		409+	ST	14,8(0,1) STORE DCB ADDRESS
000460	41E0 BD52	00D66		410+	LA	14,FORMAT3 000B
000464	50E0 100C	0000C		411+	ST	14,12(0,1) STORE AREA ADDR 000B
000468	4060 1018	00018		412+	STH	RLNREG,24(0,1) STORE LINE NUMBER
00046C	58F0 BB58	00B6C		413+	L	15,DD3270+48 LOAD RDWRT ROUT ADDR
000470	05EF			414+	BALR	14,15 A38557
000472	12FF			415	LTR	EPREG,EPREG END IF RC
000474	4770 B90A	0091E		416	BNZ	ERRABEND NON-ZERO
				417	WAIT	ECB=(DECBREG) WAIT FOR COMPLETION
000478	4110 7000	00000		418+	LA	1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
00047C	4100 0001	00001		419+	LA	0,1(0,0) COUNT OMITTED,1 USED
000480	0A01			420+	SVC	1 LINK TO WAIT ROUTINE
000482	957F 7000	00000		421	CLI	0(DECBREG),X'7F' SATISFACTORY COMPLETION
000486	4780 B25C	00270		422	BE	READ YES, SET UP READ
00048A	9544 7000	00000		423	CLI	0(DECBREG),X'44' INTERCEPT
00048E	4780 B428	0043C		424	BE	WRITE3 YES, RETRY OPERATION
000492	47F0 B90A	0091E		425	B	ERRABEND NO, END
				426 *		
000496	957D 8000	00000		427	ENT1	CLI 0(BUFREG),X'7D' ENTER KEY
00049A	4770 B25C	00270		428	BNE	READ NO, REESTABLISH READ
00049E	5839 300C	0000C		429	L	ODDREG,12(TABBASE,ODDREG) GET ADDR OF DATA AREA
0004A2	4158 0003	00003		430	LA	INDXREG,3(BUFREG) INDEX PAST AID & CURSOR ADDR
0004A6	4140 0134	00134		431	LA	LOOPREG,308
0004AA	4B42 0012	00012		432	SH	LOOPREG,COUNT(EVENREG) GET NUMBER OF BYTES READ
0004AE	5B40 BAC8	00ADC		433	S	LOOPREG,THREE ADJUST FOR LENGTH OF AID & ADDR
0004B2	1244			434	LTR	LOOPREG,LOOPREG
0004B4	47D0 B530	00544		435	BNP	WRTRSPNS NOTHING ENTERED
				436 *		
0004B8	D502 5000 BB63	00000 00B77		437	CLC	0(3,INDXREG),FLD1 FIRST FIELD
0004BE	4770 B4BE	004D2		438	BNE	FLD2CHK NO, CHECK FOR SECOND
0004C2	D720 3000 3000	00000 00000		439	XC	0(33,ODDREG),0(ODDREG) CLEAR FIELD IN DATA AREA
0004C8	45E0 B998	009AC		440	BAL	RETNREG,GETFIELD FIND FIELD DELIMITERS

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
0004CC	1244			441	LTR LOOPREG,LOOPREG	END OF DATA
0004CZ	47D0 B530	00544		442	BNP WRTRSPNS	YES, WRITE RESPONSE
				443 *		
0004D2	4133 0022	00022		444	FLD2CHK LA ODDREG,34(ODDREG)	INCR TO NEXT FIELD IN DATA AREA
0004D6	D502 5000 BB66	00000	00B7A	445	CLC 0(3,INDXREG),FLD2	SECOND FIELD
0004DC	4770 B4DC	004F0		446	BNE FLD3CHK	NO, CHECK FOR THIRD FIELD
0004E0	D720 3000 3000	00000	00000	447	XC 0(33,ODDREG),0(ODDREG)	CLEAR FIELD IN DATA AREA
0004E6	45E0 B998	009AC		448	BAL RETNREG,GETFIELD	GET FIELD
0004EA	1244			449	LTR LOOPREG,LOOPREG	END OF DATA
0004EC	47D0 B530	00544		450	BNP WRTRSPNS	YES, WRITE RESPONSE
				451 *		
0004F0	4133 0022	00022		452	FLD3CHK LA ODDREG,34(ODDREG)	INCR TO THIRD FIELD IN DATA AREA
0004F4	D502 5000 BD69	00000	00B7D	453	CLC 0(3,INDXREG),FLD3	THIRD FIELD
0004FA	4770 B4FA	0050E		454	BNE FLD4CHK	NO, CHECK FOR FOURTH FIELD
0004FE	D720 3000 3000	00000	00000	455	XC 0(33,ODDREG),0(ODDREG)	CLEAR FIELD
000504	45E0 B998	009AC		456	BAL RETNREG,GETFIELD	GET FIELD
000508	1244			457	LTR LOOPREG,LOOPREG	END OF DATA
00050A	47D0 B530	00544		458	BNP WRTRSPNS	YES, WRITE RESPONSE
				459 *		
00050E	4133 0022	00022		460	FLD4CHK LA ODDREG,34(ODDREG)	INCR TO FOURTH FIELD
000512	D502 5000 BD6C	00000	00B80	461	CLC 0(3,INDXREG),FLD4	FOURTH FIELD
000518	4770 B518	0052C		462	BNE FLD5CHK	NO, CHECK FOR LAST FIELD
00051C	D714 3000 3000	00000	00000	463	XC 0(21,ODDREG),0(ODDREG)	CLEAR FIELD
000522	45E0 B998	009AC		464	BAL RETNREG,GETFIELD	GET FIELD
000526	1244			465	LTR LOOPREG,LOOPREG	END OF DATA
000528	47D0 B530	00544		466	BNP WRTRSPNS	YES, WRITE RESPONSE
				467 *		
00052C	4133 0016	00016		468	FLD5CHK LA ODDREG,22(ODDREG)	INCR TO LAST FIELD
000530	D502 5000 BD6F	00000	00B83	469	CLC 0(3,INDXREG),FLD5	FIFTH FIELD
000536	4770 B530	00544		470	BNE WRTRSPNS	NO, WRITE RESPONSE
00053A	D704 3000 3000	00000	00000	471	XC 0(5,ODDREG),0(ODDREG)	CLEAR FIELD
000540	45E0 B998	009AC		472	BAL RETNREG,GETFIELD	GET FIELD
				473 *		
				474 *		
000544	1861			475	WRTFMT2 LR RLNREG,PARMREG	SAVE RLN
		00544		476	WRTRSPNS EQU WRTFMT2	RETURN = REESTABLISH READ
				477 *		WRITE FORMAT 2
000546	D2FF 8000 BC20	00000	00C34	478	MVC 0(256,BUFREG),FORMAT2	MOVE BASE MESSAGE
00054C	D231 8100 BD20	00100	00D34	479	MVC 256(FMT2SZ-256,BUFREG),FORMAT2+256	INTO BUFFER
000552	8B10 0003	00003		480	SLA PARMREG,3	CONVERT RLN TO TABLE INDEX
000556	4120 0002	00002		481	LA EVENREG,2	GET FORMAT ID
00055A	4029 1008	00008		482	STH EVENREG,8(TABBASE,PARMREG)	6 STORE IN TABLE ENTRY
00055E	5839 100C	0000C		483	L ODDREG,12(TABBASE,PARMREG)	GET DATA AREA ADDR
000562	D220 8028 3000	00028	00000	484	MVC FMT2FLD1(33,BUFREG),0(ODDREG)	MOVE DATA
000568	D220 8052 3022	00052	00022	485	MVC FMT2FLD2(33,BUFREG),34(ODDREG)	FROM
00056E	D220 807C 3044	0007C	00044	486	MVC FMT2FLD3(33,BUFREG),68(ODDREG)	DATA AREA
000574	D214 80A7 3066	000A7	00066	487	MVC FMT2FLD4(21,BUFREG),102(ODDREG)	TO
00057A	D204 80C4 307C	000C4	0007C	488	MVC FMT2FLD5(5,BUFREG),124(ODDREG)	BUFFER
000580	D703 7000 7000	00000	00000	489	WRITE2 XC 0(4,DECBREG),0(DECBREG)	CLEAR ECB
				490	WRITE (DECBREG),TS,DD3270,(BUFREG),FMT2SZ,,(RLNREG),MF=E	LOAD DECB ADDRESS
000586	1817			491+	LR 1,DECBREG	
000588	9404 1004	00004		492+	NI 4(1),4	
00058C	920E 1005	00005		493+	MVI 5(1),14	
000590	47F0 B582	00596		494+	B *+6	000B
000594	0132			495+	DC AL2(FMT2SZ)	000B

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
000596	D201 1006 B580	00006	00594	496+	MVC 6(2,1),*-2	MOVE IN LENGTH 000B
00059C	41E0 BB28	00B3C		497+	LA 14,DD3270	
0005A0	50E0 1008	00000		498+	ST 14,8(0,1)	STORE DCB ADDRESS
0005A4	5080 100C	0000C		499+	ST BUFREG,12(0,1)	STORE AREA ADDR 000B
0005A8	4060 1018	00018		500+	STH RLNREG,24(0,1)	STORE LINE NUMBER
0005AC	58F0 BB58	00B6C		501+	L 15,DD3270+48	LOAD RDWRT ROUT ADDR
0005B0	05EF			502+	BALR 14,15	A38557
0005B2	12FF			503	LTR EPREG,EPREG	END IF RC
0005B4	4770 B90A	0091E		504	BNZ ERRABEND	NON-ZERO
				505	WAIT ECB=(DECBREG)	WAIT FOR OPERATION TO END
0005B8	4110 7000	00000		506+	LA 1,0(0,DECBREG)	CLEAR HIGH BYTE OF REG
0005BC	4100 0001	00001		507+	LA 0,1(0,0)	COUNT OMITTED,1 USED
0005C0	0A01			508+	SVC 1	LINK TO WAIT ROUTINE
0005C2	957F 7000	00000		509	CLI 0(DECBREG),X'7F'	COMPLETION SATISFACTORY
0005C6	4780 B25C	00270		510	BE READ	YES, SET UP READ
0005CA	9544 7000	00000		511	CLI 0(DECBREG),X'44'	INTERCEPT
0005CE	4780 B56C	00580		512	BE WRITE2	YES, RETRY
0005D2	47F0 B90A	0091E		513	B ERRABEND	NO, END
				514 *		
0005D6	956C 8000	00000		515	READPMT2 CLI 0(BUFREG),X'6C'	PA1 KEY
0005DA	4770 B318	0032C		516	BNE READPMT1	NO, GO TO READ FORMAT 1 LOGIC
0005DE	1821			517	LR EVENREG,PARMREG	GET RLN OF DISPLAY
0005E0	8B20 0003	00003		518	SLA EVENREG,3	GET INDEX TO ENTRY
0005E4	4832 900A	0000A		519	LH ODDREG,10(EVENREG,TABBASE)	GET RLN OF ASSIGNED PTR
0005E8	1233			520	LTR ODDREG,ODDREG	PRINTER ASSIGNED
0005EA	4780 B67E	00692		521	BZ SYSOUT	NO, USE SYSOUT
0005EE	1863			522	LR RLNREG,ODDREG	SAVE RLN
0005F0	8B30 0003	00003		523	SLA ODDREG,3	GET INDEX TO ENTRY OF PRINTER
0005F4	4100 00B8	000B8		524	LA ZEROREG,184	
				525	GETMAIN R,LV=(0)	GET CORE FOR PRINTER REQUEST BLOCK
0005F8	4510 B5E8	005FC		526+	BAL 1,*+4	INDICATE GETMAIN
0005FC	0A0A			527+	SVC 10	ISSUE GETMAIN SVC
0005FE	D7B7 1000 1000	00000	00000	528	XC 0(184,PARMREG),0(PARMREG)	CLEAR IT
000604	4141 002C	0002C		529	LA LOOPREG,44(PARMREG)	GET ADDR OF PRINT BUFFER
000608	5852 900C	0000C		530	L INDXREG,12(EVENREG,TABBASE)	GET ADDR OF DATA AREA
00060C	92C8 4000	00000		531	MVI 0(LOOPREG),X'CB'	STORE WCC IN BUFFER
000610	D280 4001 5000	00001	00000	532	MVC 1(129,LOOPREG),0(INDXREG)	MOVE IN DATA
000616	9215 4022	00022		533	MVI 34(LOOPREG),X'15'	NEW LINE
00061A	9215 4044	00044		534	MVI 68(LOOPREG),X'15'	NEW LINE
00061E	9215 4066	00066		535	MVI 102(LOOPREG),X'15'	NEW LINE
000622	9240 407C	0007C		536	MVI 124(LOOPREG),C'	BLANK
000626	9219 4082	00082		537	MVI 130(LOOPREG),X'19'	END OF MESSAGE
				538 *		
00062A	4809 3008	00008		539	LH ZEROREG,8(TABBASE,ODDREG)	GET CURRENT PRINTER FORMAT
00062E	4900 BB60	00B74		540	CH ZEROREG,MIN1	PRINTER BUSY
000632	4780 B638	0064C		541	BE STRTPRTR	NO, START OPERATION
000636	5859 300C	0000C		542	L INDXREG,12(TABBASE,ODDREG)	GET POINTER TO LAST REQ
00063A	5015 0000	00000		543	ST PARMREG,0(INDXREG)	STORE POINTER
00063E	5019 300C	0000C		544	ST PARMREG,12(TABBASE,ODDREG)	UPDATE POINTER TO CURR REQ
000642	1812			545	RSTRKYBD LR PARMREG,EVENREG	RECOVER RLN OF DISPLAY
000644	8A10 0003	00003		546	SRA PARMREG,3	FROM TABLE INDEX
000648	47F0 B530	00544		547	B WRTRSPNS	GO TO UNLOCK KEYBOARD
				548 *		
00064C	5019 300C	0000C		549	STRTPRTR ST PARMREG,12(TABBASE,ODDREG)	STORE POINTER TO REQ
000650	4111 0004	00004		550	LA PARMREG,4(PARMREG)	GET ADDRESS OF DECB

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	
000654	5859 0004	00004		551	L	INDXREG,4(TABBASE) GET ADDRESS OF ECB LIST	
000658	4A59 300A	0000A		552	AH	INDXREG,10(TABBASE,ODDREG) ADD INDEX TO PRTR'S PTR	
00065C	5015 0000	00000		553	ST	PARMREG,0(INDXREG) STORE ECB ADDR IN LIST	
000660	0600			554	BCTR	ZEROREG,0 INDICATE	
000662	4009 3008	00008		555	STH	ZEROREG,8(TABBASE,ODDREG) PRINTER BUSY	
				556	WRITE	(PARMREG),TS,DD3270,(LOOPREG),131,,(RLNREG),MF-E	
000666	1811			557+	LR	1,PARMREG LOAD DECB ADDRESS	
000668	9404 1004	00004		558+	NI	4(1),4	
00066C	920E 1005	00005		559+	MVI	5(1),14	
000670	41E0 0083	00083		560+	LA	14,131(0,0)	000B
000674	40E0 1006	00006		561+	STH	14,6(0,1) STORE LENGTH	000B
000678	41E0 BB28	00B3C		562+	LA	14,DD3270	
00067C	50E0 1008	00008		563+	ST	14,8(0,1) STORE DCB ADDRESS	
000680	5040 100C	0000C		564+	ST	LOOPREG,12(0,1) STORE AREA ADDR	000B
000684	4060 1018	00018		565+	STH	RLNREG,24(0,1) STORE LINE NUMBER	
000688	58F0 BB58	00B6C		566+	L	15,DD3270+48 LOAD RDWRT ROUT ADDR	
00068C	05EF			567+	BALR	14,15	A38557
00068E	47F0 B62E	00642		568	B	RSTRKYBD	
				569 *			
000692	5859 200C	0000C		570	SYSOUT L	INDXREG,12(TABBASE,EVENREG) GET ADDR OF DATA AREA	
000696	D281 B9D4	5000 009E8	00000	571	MVC	SYSOUTBF(130),0(INDXREG) MOVE DATA TO PRINT BUFFER	
00069C	926B B9F5	00A09		572	MVI	SYSOUTBF+33,C', ' MOVE IN	
0006A0	926B BA17	00A2B		573	MVI	SYSOUTBF+67,C', ' FIELD DELIMITERS	
0006A4	926B BA39	00A4D		574	MVI	SYSOUTBF+101,C', ' FOR LINE	
0006A8	9240 BA4F	00A63		575	MVI	SYSOUTBF+123,C', ' TO BE	
0006AC	924B BA55	00A69		576	MVI	SYSOUTBF+129,C', ' PRINTED	
0006B0	DC81 B9D4	BE8E 009E8	00ED2	577	TR	SYSOUTBF(130),FLDXLATE TRANSLATE TO VALIDATE DATA	
				578	PUT	SYSPRINT,SYSOUTBF WRITE TO SYSOUT	
0006B6	4110 BADC	00AF0		579+	LA	1,SYSPRINT LOAD PARAMETER REG 1	
0006BA	4100 B9D4	009E8		580+	LA	0,SYSOUTBF LOAD PARAMETER REG 0	
0006BE	58F0 1030	00030		581+	L	15,48(0,1) LOAD PUT ROUTINE ADDR	
0006C2	05EF			582+	BALR	14,15 LINK TO PUT ROUTINE	
0006C4	47F0 B62E	00642		583	B	RSTRKYBD	
				584 *			
0006C8	95F0 8000	00000		585	READFMT3 CLI	0(BUFREG),X'F0' RFT KEY	3XA11698
0006CC	4780 B25C	00270		586	BE	READ YES,REISSUE READ	3XA11698
0006D0	D502 8000	B906 00000	0091A	587	CLC	0(3,BUFREG),SOHRFT RFT HEADER	3XA11698
0006D6	4780 B25C	00270		588	BE	READ YES, REISSUE READ	3XA11698
0006DA	957D 8000	00000		589	CLI	0(BUFREG),X'7D' ENTER KEY	3XA11698
0006DE	4780 B530	00544		590	BE	WRTRSPNS YES, WRITE RESPONSE	
0006E2	956E 8000	00000		591	CLI	0(BUFREG),X'6E' PA2 KEY	
0006E6	4780 B320	00334		592	BE	ENDMSG YES, END TERMINAL	
0006EA	956C 8000	00000		593	CLI	0(BUFREG),X'6C' PA1 KEY	
0006EE	4770 B41A	0042E		594	BNE	CNTRLMMSG NO, WRITE CONTROL OPTIONS MSG	
				595 *			
0006F2	5829 300C	0000C		596	L	EVENREG,12(TABBASE,ODDREG) GET ADDR OF DATA AREA	
0006F6	D781 2000	2000 00000	00000	597	XC	0(130,EVENREG),0(EVENREG) CLEAR IT	
0006FC	45E0 B6F6	0070A		598	BAL	RETNRREG,WRTFMT1 WRITE OUT FIRST FORMAT	
000700	12FF			599	LTR	EPREG,EPREG WRITE SUCCESSFUL	
000702	4770 B90A	0091E		600	BNZ	ERRABEND NO, END	
000706	47F0 B25C	00270		601	B	READ	
				602 *			
00070A	183E			603	WRTFMT1 LR	ODDREG,RETNRREG SAVE RETURN ADDRESS	
00070C	1861			604	LR	RLNREG,PARMREG SAVE RLN	
00070E	D703 7000	7000 00000	00000	605	WRITE1 XC	0(4,DECBREG),0(DECBREG) CLEAR ECB	

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
				606	WRITE (DECBREG),TS,DD3270,FORMAT1,FMT1SZ,,(RLNREG),MF=E	
000714	1817			607+	LR 1,DECBREG	LOAD DECB ADDRESS
000716	9404 1004	00004		608+	NI 4(1),4	
00071A	920E 1005	00005		609+	MVI 5(1),14	
00071E	47F0 B710	00724		610+	B **6	000B
000722	00AE			611+	DC AL2(FMT1SZ)	000B
000724	D201 1006 B70E	00006	00722	612+	MVC 6(2,1),*-2	MOVE IN LENGTH 000B
00072A	41E0 BB28	00B3C		613+	LA 14,DD3270	
00072E	50E0 1008	00008		614+	ST 14,8(0,1)	STORE DCB ADDRESS
000732	41E0 BB72	00B86		615+	LA 14,FORMAT1	000B
000736	50E0 100C	0000C		616+	ST 14,12(0,1)	STORE AREA ADDR 000B
00073A	4060 1018	00018		617+	STH RLNREG,24(0,1)	STORE LINE NUMBER
00073E	58F0 BB58	00B6C		618+	L 15,DD3270+48	LOAD RDWRT ROUT ADDR
000742	05EF			619+	BALR 14,15	A38557
000744	12FF			620	LTR EPREG,EPREG	END IF RC
000746	4770 B90A	0091E		621	BNZ ERRABEND	NON-ZERO
				622	WAIT ECB=(DECBREG)	WAIT FOR OPERATION TO END
00074A	4110 7000	00000		623+	LA 1,0(0,DECBREG)	CLEAR HIGH BYTE OF REG
00074E	4100 0001	00001		624+	LA 0,1(0,0)	COUNT OMITTED,1 USED
000752	0A01			625+	SVC 1	LINK TO WAIT ROUTINE
000754	957F 7000	00000		626	CLI 0(DECBREG),X'7F'	SATISFACTORY COMPLETION
000758	4780 B756	0076A		627	BE FMT1CD	YES, STORE FORMAT CODE IN ENTRY
00075C	9544 7000	00000		628	CLI 0(DECBREG),X'44'	INTERCEPT
000760	4780 B6FA	0070E		629	BE WRITE1	YES, RETRY OPERATION
000764	41F0 B750	00764		630	LA EPREG,*	SET NON-ZERO RETURN CODE
000768	07F3			631	BR ODDREG	AND RETURN TO CALLER
				632 *		
00076A	4120 0001	00001		633	FMT1CD LA EVENREG,1	GET FORMAT ID
00076E	8B60 0003	00003		634	SLA RLNREG,3	CONVERT RLN TO TABLE INDEX
000772	4029 6008	00008		635	STH EVENREG,8(TABBASE,RLNREG)	STORE ID IN TABLE
000776	8A60 0003	00003		636	SRA RLNREG,3	RESTORE RLN REGISTER TO RLN
00077A	1BFF			637	SR EPREG,EPREG	SET RETURN CODE =0
00077C	07F3			638	BR ODDREG	RETURN TO CALLER
				639 *		
00077E	1B33			640	PRNTREND SR ODDREG,ODDREG	
000780	4332 0019	00019		641	IC ODDREG,RLN(EVENREG)	GET RLN OF ENDING PRINTER
000784	1863			642	LR RLNREG,ODDREG	SAVE RLN
000786	8B30 0003	00003		643	SLA ODDREG,3	CONVERT RLN TO TABLE INDEX
00078A	5B20 BAD0	00AE4		644	S EVENREG,FOUR	GET BEGIN OF PRINTER RB
00078E	4133 9008	00008		645	LA ODDREG,8(ODDREG,TABBASE)	GET ADDR OF TABLE ENTRY
000792	5859 0004	00004		646	L INDXREG,4(TABBASE)	GET ADDR OF ECB LIST
000796	4A53 0002	00002		647	AH INDXREG,2(ODDREG)	GET ADDR OF PRINTER'S ECB
00079A	957F 2004	00004		648	CLI 4(EVENREG),X'7F'	GOOD END
00079E	4780 B79A	007AE		649	BE PRNTRNXT	YES, CLEAN UP
0007A2	9544 2004	00004		650	CLI 4(EVENREG),X'44'	INTERCEPT
0007A6	4780 B8FC	00910		651	BE WRITEPI	YES, RESTART OPERATION
0007AA	47F0 B90A	0091E		652	B ERRABEND	NO, END
				653 *		
0007AE	D203 3004 2000	00004	00000	654	PRNTRNXT MVC 4(4,ODDREG),0(EVENREG)	UPDATE LINK POINTER
0007B4	1812			655	LR PARMREG,EVENREG	ADDRESS OF PRINTER RB
0007B6	4100 00B8	000B8		656	LA ZEROREG,184	SIZE OF RB
				657	FREEMAIN R,LV=(0),A=(1)	FREE BLOCK
0007BA	4111 0000	00000		658+	LA 1,0(1,0)	CLEAR THE HIGH ORDER BYTE XM4571
0007BE	07FA			659+	SVC 10	ISSUE FREEMAIN SVC P2504
0007C0	5823 0004	00004		660	L EVENREG,4(ODDREG)	GET ADDR OF NEXT RB

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
0007C4	1222			661	LTR EVENREG,EVENREG	END OF CHAIN
0007C6	4770 B7C8	007DC		662	BNZ WRITEP	NO, START NEXT
0007CA	4120 BAD8	00AEC		663	LA EVENREG,DUMMYECB	
0007CE	5025 0000	00000		664	ST EVENREG,0(INDXREG)	XA01560
0007D2	D201 3000 BB60	00000	00B74	665	MVC 0(2,ODDREG),MIN1	MARK PRINTER AVAILABLE
0007D8	47F0 B294	002A8		666	B TWAIT	WAIT FOR NEXT COMPLETION
				667 *		
				668 WRITEP	RESETPL (DECBREG),ATTENT	RESET READ INITIAL
0007DC				669+WRITEP	DS 0H	
0007DC	1817			670+	LR 1,DECBREG	
0007DE	1B00			671+	SR 0,0	
0007E0	1BFF			672+	SR 15,15	
0007E2	4301 0019	00019		673+	IC 0,25(1)	GET RELATIVE LINE NUMBER FROM DECB.
0007E6	58E1 0008	00008		674+	L 14,8(1)	GET DCB ADDRESS FROM DECB.
0007EA	9110 E030	00030		675+	TM 48(14),X'10'	HAS DCB BEEN OPENED?
0007EE	4710 B7E6	007FA		676+	BO IECA0033	IF SO, CONTINUE.
0007F2	41F0 000C	0000C		677+IECJ0033	LA 15,12	IF NOT, SET RETURN CODE A28622
0007F6	47F0 B89E	008B2		678+	B IECB0033	AND EXIT.
		007FA		679+IECA0033	EQU *	
0007FA	58FE 002C	0002C		680+	L 15,44(14)	GET ADDR OF DEB
0007FE	41FF 0000	00000		681+	LA 15,0(15)	CLEAR HIGH-ORDER BYTE
000802	8B00 0002	00002		682+	SLA 0,2	MULTIPLY RLN BY 4
000806	1AF0			683+	AR 15,0	USE RLN*4 AS INDEX
000808	58FF 001C	0001C		684+	L 15,28(15)	TO DEBUCBAD-4 AND GET UCB ADDR
00080C	9510 F012	00012		685+	CLI 18(15),X'10'	DEVICE CLASS = GRAPHICS
000810	4770 B890	008A4		686+	BNE IECA0033	NO, DEVICE IS NOT ANR
000814	9501 1005	00005		687+	CLI 5(1),X'01'	IS OPERATION READ INITIAL
000818	4770 B888	0089C		688+	BNE IECX0033	NO, GIVE RC=10 AND EXIT
00081C	9140 1000	00000		689+	TM 0(1),X'40'	IS THE OPERATION COMPLETE
000820	4710 B880	00894		690+	BO IECQ0033	YES, GIVE RC=4 AND EXIT
000824	1801			691+	LR 0,1	DECB ADDR TO REG 0
000826	181F			692+	LR 1,15	UCB ADDR TO REG 1
000828	41F0 0004	00004		693+	LA 15,4	ROUTING CODE OF 4 IN REG 15
00082C	0A74			694+	SVC 116	INVOKE ESR TO DO RESET
				695+*	TO	TWO LINES OF CODE DELETED BY APAR OX12278
00082E	89F0 0018	00018		696+	SLL 15,24	MAKE SURE THAT ONLY... RH 2XA06368
000832	88F0 0018	00018		697+	SRL 15,24	RC IS IN REG 15 RH 2XA06368
000836	4110 0004	00004		698+	LA 1,4	ESTABLISH COMPARAND
00083A	19F1			699+	CR 15,1	IS RC 0 OR 4
00083C	4720 B89E	008B2		700+	BH IECB0033	RETURN WITH RC 2XA12278
000840	40F0 B898	008AC		701+	STH 15,IEC0033	SAVE RETURN CODE
				702+*		
000844	581E 001C	0001C		703+	L 1,28(14)	GET IOB BASE FROM DCB
000848	4111 0058	00058		704+	LA 1,88(1)	GET ADDR OF FIRST IOB
00084C	58EE 002C	0002C		705+	L 14,44(14)	GET DEB ADDR FROM DCB
000850	1B00			706+	SR 0,0	
000852	430E 0010	00010		707+	IC 0,16(14)	GET NO. OF EXTENTS FROM DEB
				708+*		AND USE AS LOOP COUNTER
000856	41EE 0020	00020		709+	LA 14,32(14)	GET ADDR OF FIRST DEB UCB PTR
				710+*		
00085A	9601 1001	00001		711+IECT0033	OI 1(1),X'01'	SET RESETPL ISSUED FLAG
00085E	9101 101C	0001C		712+	TM 28(1),X'01'	RPT IN PROGRESS
000862	4710 B876	0088A		713+	BO IECW0033	YES, GIVE RC=0 2XA12280
000866	58FE 0000	00000		714+	L 15,0(14)	GET ADDR OF UCB
00086A	9120 F006	00006		715+	TM 6(15),X'20'	OUTSTANDING I/O OPERATION

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
00086E	4710 B862	00876		716+	BO IECU0033	YES, KEEP IOB BUSY
000872	94BF 101C	0001C		717+	NI 28(1),X'BF'	TURN OFF IOB BUSY FLAG
				718+*		
000876	4111 0058	00058		719+IECU0033	LA 1,88(1)	STEP TO NEXT IOB
00087A	41EE 0004	00004		720+	LA 14,4(14)	STEP TO NEXT UCB PTR
00087E	4600 B846	0085A		721+	BCT 0,IECT0033	LOOP IF NOT FINISHED
000882	48F0 B898	008AC		722+	LH 15,IECK0033	RESTORE RETURN CODE
000886	47F0 B89E	008B2		723+	B IECB0033	EXIT WITH RC
				724+*	SEVEN LINES OF CODE DELETED BY APAR OX12278	0XA12278
				725+*		
00088A	D701 B898 B898	008AC 008AC		726+IECX0033	XC IECK0033,IECK0033	SET RC TO ZERO
000890	47F0 B862	00876		727+	B IECU0033	
				728+*		
000894	41F0 0004	00004		729+IECQ0033	LA 15,4	SET RETURN CODE OF 4
000898	47F0 B89E	008B2		730+	B IECB0033	AND EXIT
				731+*		
00089C	41F0 0010	00010		732+IECX0033	LA 15,16	SET RETURN CODE OF X'10'
0008A0	47F0 B89E	008B2		733+	B IECB0033	AND EXIT
				734+*		
0008A4	41F0 0008	00008		735+IECO0033	LA 15,8	INVALID DEVICE, SET RC = 8
0008A8	47F0 B89E	008B2		736+	B IECB0033	AND EXIT
0008AC	0000			737+IECK0033	DC H'0' SAVE AREA FOR WTTA	000G
0008AE	0080			738+IECL0033	DC X'0080'	FLAG FOR SVC 33 RH 0XA06387
0008B0	1BFF			739+IECE0033	SR 15,15	XA01629
		008B2		740+IECB0033	EQU *	
0008B2	12FF			741	LTR EPREG,EPREG	RC = 0
0008B4	4780 B8B6	008CA		742	BZ WRITEPS	YES, PROCEED
0008B8	59F0 BAD0	00AE4		743	C EPREG,FOUR	RC = 4
0008BC	4770 B90A	0091E		744	BNE ERRABEND	NO, END
				745	WAIT ECB=(DECBREG)	WAIT FOR READ TO COMPLETE
0008C0	4110 7000	00000		746+	LA 1,0(0,DECBREG)	CLEAR HIGH BYTE OF REG
0008C4	4100 0001	00001		747+	LA 0,1(0,0)	COUNT OMITTED,1 USED
0008C8	0A01			748+	SVC 1	LINK TO WAIT ROUTINE
				749 *		
0008CA	4122 0004	00004		750 WRITEPS	LA EVENREG,4(EVENREG)	GET ADDR OF ECB
0008CE	5025 0000	00000		751	ST EVENREG,0(INDXREG)	STORE ADDR OF ECB IN ECB LIST
0008D2	4132 0028	00028		752	LA ODDREG,40(EVENREG)	GET ADDR OF PRINT BUFFER
				753	WRITE (EVENREG),TS,DD3270,(ODDREG),131,,(RLNREG),MF=E	
0008D6	1812			754+	LR 1,EVENREG	LOAD DECB ADDRESS
0008D8	9404 1004	00004		755+	NI 4(1),4	
0008DC	920E 1005	00005		756+	MVI 5(1),14	
0008E0	41E0 0083	00083		757+	LA 14,131(0,0)	000B
0008E4	40E0 1006	00006		758+	STH 14,6(0,1)	STORE LENGTH 000B
0008E8	41E0 BB28	00B3C		759+	LA 14,DD3270	
0008EC	50E0 1008	00008		760+	ST 14,8(0,1)	STORE DCB ADDRESS
0008F0	5030 100C	0000C		761+	ST ODDREG,12(0,1)	STORE AREA ADDR 000B
0008F4	4060 1018	00018		762+	STH RLNREG,24(0,1)	STORE LINE NUMBER
0008F8	58F0 BB58	00B6C		763+	L 15,DD3270+48	LOAD RDWRT ROUT ADDR
0008FC	05EF			764+	BALR 14,15	A38557
0008FE	12FF			765	LTR EPREG,EPREG	END IF RC
000900	4770 B90A	0091E		766	BNZ ERRABEND	NON-ZERO
000904	9548 7000	00000		767	CLI 0(DECBREG),X'48'	READ INITIAL ENDED BY RESETPL
000908	4780 B25C	00270		768	BE READ	YES, REISSUE READ
00090C	47F0 B294	002A8		769	B TWAIT	NO, CHECK STATUS
				770 *		

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
000910	D703 2004 2004	00004	00004	771	WRITEPI XC	4(4,EVENREG),4(EVENREG) CLEAR ECB
000916	47F0 B7C8	007DC		772	B WRITEP	RESTART WRITE
				773	*	
				774	*	
00091A	016C61			775	SOHRPT DC	X'016C61' RFT HEADER 2XA11698
				776	ERRABEND ABEND	(15),DUMP
00091E	181F			777	+ERRABEND DS	OH
000920	4100 0080	00080		778+	LR	1,15 LOAD PARAMETER REG 1
000924	8900 0018	00018		779+	LA	0,128(0,0) PICK UP DUMP/STEP CODE XM4571
000928	1610			780+	SLL	0,24(0) SHIFT TO HIGH BYTE
00092A	0A0D			781+	OR	1,0 OR IN WITH COMPCODE
00092C	1B22			782+	SVC	13 LINK TO ABEND ROUTINE
00092E	1B66			783	GETPARM SR	EVENREG,EVENREG CLEAR REGISTER
000930	95F0 5000	00000		784	SR	CNTREG,CNTREG
000934	4740 B96A	0097E		785	GETCHAR CLI	0(INDXREG),C'0' CHARACTER A NUMBER
000938	95F9 5000	00000		786	BL	DELIM NO, CHECK FOR DELIMITER
00093C	4720 B986	0099A		787	CLI	0(INDXREG),C'9' MAYBE
000940	8920 0008	00008		788	BH	INVLDCHR NO, INVALID CHARACTER
000944	4325 0000	00000		789	SLL	EVENREG,8 CLEAR LOW CHARACTER
000948	4166 0001	00001		790	IC	EVENREG,0(INDXREG) GET CHARACTER
00094C	4155 0001	00001		791	PARMLOOP LA	CNTREG,1(CNTREG) INCR LENGTH
000950	4640 B91C	00930		792	LA	INDXREG,1(INDXREG) INCR STRING POINTER
000954	5960 BAC8	00ADC		793	BCT	LOOPREG,GETCHAR GET NEXT CHARACTER, IF ANY LEFT
000958	47D0 B950	00964		794	HAVEPARM C	CNTREG,THREE CHECK PARM LENGTH
00095C	5420 BACC	00AE0		795	BNH	CNVRT BR IF VALID
000960	5860 BAC8	00ADC		796	N	EVENREG,SIXFS REDUCE TO THREE BYTES
000964	5020 BAC4	00AD8		797	L	CNTREG,THREE SET COUNT TO THREE
000968	4130 0004	00004		798	CNVRT ST	EVENREG,RAWPARM STORE RAW PARM VALUE
00096C	1B36			799	LA	ODDREG,4 LENGTH OF RAWPARM
00096E	4133 BAC4	00AD8		800	SR	ODDREG,CNTREG SUBT LENGTH OF PARM &
000972	0660			801	LA	ODDREG,RAWPARM(ODDREG) GET ADDR OF FIRST CHARACTER
000974	4460 B992	009A6		802	BCTR	CNTREG,0 DECR COUNT FOR EXECUTE
000978	4F10 B9CC	009E0		803	EX	CNTREG,PACK PACK PARM
00097C	07FE			804	CVB	PARMREG,PCKDPARM & CONVERT TO BINARY
				805	BR	RETNREG RETURN TO CALLER
				806	*	
00097E	956B 5000	00000		807	DELIM CLI	0(INDXREG),C',' COMMA
000982	4770 B986	0099A		808	BNE	INVLDCHR NO, TREAT AS INVALID
000986	1266			809	LTR	CNTREG,CNTREG PARM OMITTED
000988	4780 B982	00996		810	BZ	ZEROPARM YES, SET IT TO ZERO
00098C	4155 0001	00001		811	LA	INDXREG,1(INDXREG) INCR STRING POINTER
000990	0640			812	BCTR	LOOPREG,0 DECR LOOP COUNTER
000992	47F0 B940	00954		813	B	HAVEPARM BR TO CONVERT PARM
				814	*	
000996	1B11			815	ZEROPARM SR	PARMREG,PARMREG SET PARM = 0
000998	07FE			816	BR	RETNREG RETURN
00099A	8920 0008	00008		817	INVLDCHR SLL	EVENREG,8 CLEAR LOW CHARACTER
00099E	4320 BB62	00B76		818	IC	EVENREG,C0 SUBSTITUTE C'0' FOR INVALID CHAR
0099A2	47F0 B934	00948		819	B	PARMLOOP BR TO ADJUST COUNTERS & POINTERS
				820	*	
0009A6	F270 B9CC 3000	009E0 00000		821	PACK PACK	PCKDPARM(8),0(0,ODDREG)
				822	*	
0009AC	4125 0003	00003		823	GETFIELD LA	EVENREG,3(INDXREG) INCR TO FIRST DATA BYTE OF FIELD
0009B0	5B40 BAC8	00ADC		824	S	LOOPREG,THREE CORRECT LENGTH FOR SBA SEQUENCE
0009B4	1B66			825	SR	CNTREG,CNTREG SET COUNT = ZERO

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
0009B6	9511 2000	00000		826	FNDLOOP	CLI 0(EVENREG),X'11'
0009BA	4780 B9B6	009CA		827	BE	MVFLD
0009BE	4166 0001	00001		828	LA	CNTREG,1(CNTREG)
0009C2	4122 0001	00001		829	LA	EVENREG,1(EVENREG)
0009C6	4640 B9A2	009B6		830	BCT	LOOPREG,FNDLOOP
0009CA	0660			831	MVFLD	BCTR CNTREG,0
0009CC	4460 B9C0	009D4		832	EX	CNTREG,FLDMOVE
0009D0	1852			833	LR	INDXREG,EVENREG
0009D2	07FE			834	BR	RETNREG
				835	*	
0009D4	D200 3000 5003 00000 00003			836	FLDMOVE	MVC 0(0,ODDREG),3(INDXREG) MOVE FIELD FROM BUFFER TO DATA AREA
				837	*	
0009E0				838	DS	0D
0009E0	F0F0F0F0F0F0F0F0			839	PCKDPARM	DC 8C'0'
0009E8				840	SYSOUTBF	DC 0D
0009E8				841	DS	CL130
000A6C				842	SAVE	DS 18F
000AB4	000000FF			843	MAXRLN	DC F'255'
000AB8	00000001			844	NODVCS	DC F'1'
000ABC	00000000			845	NOPTRS	DC F'0'
000AC0				846	TABSIZE	DS F
000AC4	00000100			847	F256	DC F'256'
000AC8	00000082			848	DATASZ	DC F'130'
000ACC				849	DATBLKSZ	DS F
000AD0				850	DATBLKAD	DS F
000AD4	00000001			851	READRLN	DC F'1'
000AD8				852	RAWFARM	DS 1F
000ADC	00000003			853	THREE	DC F'3'
000AE0	00FFFFFF			854	SIXFS	DC XL4'00FFFFFF'
000AE4	00000004			855	FOUR	DC F'4'
000AE8	00000000			856	ZEROWRD	DC F'0'
000AEC	00000000			857	DUMMYECB	DC F'0'
				858	SYSPRINT	DCB DDNAME=SYSPRINT,DSORG=PS,DEVD=DA,MACRF=(PM),RECFM=F,LREC=L=130,BLKSIZE=130
				860+*		DATA CONTROL BLOCK
				861+*		
000AF0				862+*	SYSPRINT	DC 0F'0' ORIGIN ON WORD BOUNDARY
				864+*		DIRECT ACCESS DEVICE INTERFACE
000AF0	0000000000000000			866+	DC	BL16'0' PDAD,DVTBL
000B00	00000000			867+	DC	A(0) KEYLE,DEVT,TRBAL
				869+*		COMMON ACCESS METHOD INTERFACE
000B04	00			871+	DC	AL1(0) BUFNO
000B05	000001			872+	DC	AL3(1) BUFCE
000B08	0000			873+	DC	AL2(0) BUFL
000B0A	4000			874+	DC	BL2'0100000000000000'
				+		DSORG
000B0C	00000001			875+	DC	A(1) IOBAD
				877+*		FOUNDATION EXTENSION

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
000B10	00			879+	DC BL1'00000000'	BPTEK,BFLN,HIARCHY
000B11	000001			880+	DC AL3(1)	EODAD
000B14	80			881+	DC BL1'10000000'	*
				+		
000B15	000000			882+	DC AL3(0)	RECFM EXLST
				884+*		FOUNDATION BLOCK
000B18	E2E8E2D7D9C9D5E3			886+	DC CL8'SYSPRINT'	DDNAME
000B20	02			887+	DC BL1'00000010'	OPLGS
000B21	00			888+	DC BL1'00000000'	IFLG
000B22	0050			889+	DC BL2'0000000001010000'	*
				+		
				+		MACR
				891+*		BSAM-BPAM-QSAM INTERFACE
000B24	00			893+	DC BL1'00000000'	*
				+		RER1
000B25	000001			894+	DC AL3(1)	CHECK, GERR, FERR
000B28	00000001			895+	DC A(1)	SYNAD
000B2C	0000			896+	DC H'0'	CIND1, CIND2
000B2E	0082			897+	DC AL2(130)	BLKSIZE
000B30	00000000			898+	DC F'0'	WCPO, WCPL, OFFSR, OFFSW
000B34	00000001			899+	DC A(1)	IOBA
000B38	00			900+	DC AL1(0)	NCP
000B39	000001			901+	DC AL3(1)	EOBR, ECBAD
				903+*		QSAM INTERFACE
000B3C	00000001			905+	DC A(1)	RECAD
000B40	0000			906+	DC H'0'	QSW5
000B42	0082			907+	DC AL2(130)	LRECL
000B44	00			908+	DC BL1'00000000'	EROPT
000B45	000001			909+	DC AL3(1)	CNTRL
000B48	00000000			910+	DC F'0'	PRECL
000B4C	00000001			911+	DC A(1)	EOB
				912	DD3270	DCB DDNAME=DD3270,DSORG=CX,MACRF=(R,W),EROPT=T
				914+*		DATA CONTROL BLOCK
				915+*		
000B50		00B3C		916+	ORG *-20	TO ELIMINATE UNUSED SPACE
000B3C			00B3C	917+DD3270	DS 0F'0'	ORIGIN ON WORD BOUNDARY
000B3C		00B50		918+	ORG *+20	TO ORIGIN GENERATION
				920+*		COMMON ACCESS METHOD INTERFACE
000B50	00			922+	DC AL1(0)	BUFNO
000B51	000001			923+	DC AL3(1)	BUFCB
000B54	0000			924+	DC AL2(0)	BUFL
000B56	1000			925+	DC BL2'0001000000000000'	*
				+		DSORG
000B58	00000001			926+	DC A(1)	IOBAD

LOCAL 3270 SAMPLE PROGRAM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 16.45 08/09/76
				926+*	FOUNDATION EXTENSION	
000B5C	00			930+	DC BL1'00000000'	BFTEK,BFLN,HIARCHY
000B5D	10			931+	DC BL1'00010000'	BTAM EROPT CODE
000B5E	FF			932+	DC AL1(255)	BTAM BUFFER COUNT
000B5F	00			933+	DC AL1(0)	
000B60	00			934+	DC BL1'00000000'	
				+		RECFM
000B61	000000			935+	DC AL3(0)	EXLST
				937+*	FOUNDATION BLOCK	
000B64	C4C4F3F2F7F04040			939+	DC CL8'DD3270'	DDNAME
000B6C	00			940+	DC BL1'00000010'	OPLGS
000B6D	02			941+	DC BL1'00000000'	IFLG
000B6E	2020			942+	DC BL2'0010000000100000'	
				+		
				+		MACR
				944+*	BTAM INTERFACE	
000B70	00000000			946+	DC A(0)	LERB
000B74	FFFF			947	MIN1 DC H'-1'	FREE PRINTER FORMAT ID
000B76	F0			948	C0 DC C'0'	
000B77	11C1D6			949	FLD1 DC XL3'11C1D6'	SBA SEQ OF FIRST FIELD (SBA = 86)
000B7A	11C17E			950	FLD2 DC XL3'11C17E'	SBA SEQ OF SECOND FIELD(SBA =126)
000B7D	11C2E6			951	FLD3 DC XL3'11C2E6'	SBA SEQ OF THIRD FIELD (SBA =166)
000B80	11C34F			952	FLD4 DC XL3'11C34F'	SBA SEQ OF FOURTH FIELD (SBA =207)
000B83	11C36A			953	FLD5 DC XL3'11C36A'	SBA SEQ OF FIFTH FIELD (SBA =234)
000B86	C7114040			954	FORMAT1 DC X'C7114040'	WCC, SBA = 0
000B8A	C5D5E3C5D940C4C			955	DC C'ENTER DATA REQUESTED BELOW:'	
000BA5	11C150			956	DC X'11C150'	SBA = 80
000BA8	D5C1D4C57A			957	DC C'NAME:'	
000BAD	1D401311C1F71D60			958	DC X'1D401311C1F71D60'	SF = UNPROT, IC, SBA=119, SF=PROT
000BB5	C1C4C4D97A			959	DC C'ADDR:'	
000BBA	1D4011C25F1D60			960	DC X'1D4011C25F1D60'	SF = UNPROT, SBA = 159, SF = PROT
000BC1	C3C9E3E87A			961	DC C'CITY:'	
000BC6	1D4011C3C71D60			962	DC X'1D4011C3C71D60'	SF = UNPROT, SBA=199, SF=PROT
000BCD	E2E3C1E3C57A			963	DC C'STATE:'	
000BD3	1D4011C3E41D60			964	DC X'1D4011C3E41D60'	SF = UNPROT, SBA=228, SF=PROT
000BDA	E9C9D77A			965	DC C'ZIP:'	
000BDE	1D5011C36F1D601			966	DC X'1D5011C36F1D6011C540'	SF=UNPROT, SBA=239, SF=PROT,
				967	*	SBA= 320
000BE8	C5D5E3C5D940D2C5			968	DC C'ENTER KEY: ENTER DATA:'	
000BFE	11C5E8			969	DC X'11C5E8'	SBA = 360
000C01	D7C1F240D2C5E87A			970	DC C'PA2 KEY: END PROGRAM:'	
000C16	11C650			971	DC X'11C650'	SBA = 400
000C19	C3D3C5C1D940D2C5			972	DC C'CLEAR KEY: CONTROL OPTIONS.'	
		000AE		973	FMT1SZ EQU *-FORMAT1	
				974	*	
000C34	C7114040			975	FORMAT2 DC X'C7114040'	WCC, SBA = 0
000C38	C4C1E3C140C7C9E5			976	DC C'DATA GIVEN BELOW ENTERED:'	
000C51	11C150			977	DC X'11C150'	SBA = 80
000C54	D5C1D4C57A			978	DC C'NAME:'	
000C59	1D4013			979	DC X'1D4013'	SF=UNPROT, IC

LOCAL 3270 SAMPLE PROGRAM

ASM 0200 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
				00028	980	FMT2FLD1 EQU *-FORMAT2
000C5C	0000000000000000			981	DC	XL33'0' DATA FIELD 1
000C7D	1D60			982	DC	X'1D60' SF = PROT
000C7F	C1C4C4D97A			983	DC	C'ADDR:'
000C84	1D40			984	DC	X'1D40' SF= UNPROT
				00052	985	FMT2FLD2 EQU *-FORMAT2
000C86	0000000000000000			986	DC	XL33'0' DATA FIELD 2
000CA7	1D60			987	DC	X'1D60' SF = PROT
000CA9	C3C9E3E87A			988	DC	C'CITY:'
000CAE	1D40			989	DC	X'1D40' SF = UNPROT
				0007C	990	FMT2FLD3 EQU *-FORMAT2
000CB0	0000000000000000			991	DC	XL33'0' DATA FIELD 3
000CD1	1D60			992	DC	X'1D60' SF = PROT
000CD3	E2E3C1E3C57A			993	DC	C'STATE:'
000CD9	1D40			994	DC	X'1D40' SF = UNPROT
				000A7	995	FMT2FLD4 EQU *-FORMAT2
000CDB	0000000000000000			996	DC	XL21'0'
000CF0	1D60			997	DC	X'1D60' SF = PROT
000CF2	E9C9D77A			998	DC	C'ZIP:'
000CF6	1D50			999	DC	X'1D50' SF = UNPROT, NUM ONLY
				000C4	1000	FMT2FLD5 EQU *-FORMAT2
000CF8	0000000000			1001	DC	XL5'0'
000CFD	1D6011C540			1002	DC	X'1D6011C540' SF = PROT, SBA = 320
000D02	C5D5E3C5D940D2C5			1003	DC	C'ENTER KEY: UPDATE DATA;'
000D19	11C5E8			1004	DC	X'11C5E8' SBA = 360
000D1C	D7C1F140D2C5E87A			1005	DC	C'PA1 KEY: PRINT DATA;'
000D30	11C650			1006	DC	X'11C650' SBA = 400
000D33	D7C1F240D2C5E87A			1007	DC	C'PA2 KEY: END PROGRAM;'
000D48	11C6F8			1008	DC	X'11C6F8' SBA = 440
000D48	C3D3C5C1D940D2C5			1009	DC	C'CLEAR KEY: CONTROL OPTIONS.'
				00132	1010	FMT2SZ EQU *-FORMAT2
				1011	*	
000D66	C711404013			1012	FORMAT3 DC	X'C711404013' WCC, SBA = 0, IC
000D6B	E7E7E8E8F3C3E4E4			1013	DC	C'XXYY3CUU'
000D73	11C150			1014	DC	X'11C150' SBA = 80
000D76	C5D5E3C5D940D2C5			1015	DC	C'ENTER KEY: RESUME AND CONTINUE;'
000D95	11C1F8			1016	DC	X'11C1F8' SBA = 120
000D98	D7C1F140D2C5E87A			1017	DC	C'PA1 KEY: BEGIN NEW ENTRY;'
000DB1	11C260			1018	DC	X'11C260' SBA = 160
000DB4	D7C1F240D2C5E87A			1019	DC	C'PA2 KEY: END PROGRAM;'
000DC9	11C3F0			1020	DC	X'11C3F0' SBA = 240
000DCC	E3D640D9C5D8E4C5			1021	DC	C'TC REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESSAGE *
000DD4	E2E340C2E3C1D440					OVER SAMPLE FORMAT ABOVE: '
000E1D	E7E77EE3C5E2E340			1022	DC	C'XX=TEST NO. (23-28) YY=REPEATS (01-99)'
000E44	40C3E4E47EC1C4C4			1023	DC	C'CUU=ADDRESS OF TARGET DEVICE'
000E61	11C6D1			1024	DC	X'11C6D1' SBA = 401
000E64	E3C8C5D540C8C9E3			1025	DC	C'THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KEY T*
000E6C	40C5D9C1E2C540C5					O RESUME AFTER TEST.'
				00148	1026	FMT3SZ EQU *-FORMAT3
				1027	*	
000EAE	C711C17D			1028	FORMAT4 DC	X'C711C17D' WCC, SBA = 125
000EB2	D3D6C3C1D340F3F2			1029	DC	C'LOCAL 3270 SAMPLE PROGRAM ENDED.'
				00024	1030	FMT4SZ EQU *-FORMAT4
				1031	*	
				1032	*	

LOCAL 3270 SAMPLE PROGRAM

ASM 020C 16.45 08/09/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
				1033 *	0123456789ABCDEF
000ED2	4040404040404040			1034	FLDXLATE DC C' 0
000EE2	4040404040404040			1035	DC C' 1
000EF2	4040404040404040			1036	DC C' 2
000F02	4040404040404040			1037	DC C' 3
000F12	4040404040404040			1038	DC C' [. (+1' 4
000F22	5040404040404040			1039	DC C' && 1\$*)]' 5
000F32	6061404040404040			1040	DC C' -/ ,% 2?' 6
000F42	4040404040404040			1041	DC C' :#2T'=" 7
000F52	40C1C2C3C4C5C6C7			1042	DC C' ABCDEFGHI- 8
000F62	40D1D2D3D4D5D6D7			1043	DC C' JKLMNOPQR 9
000F72	4040E2E3E4E5E6E7			1044	DC C' STUVWXYZ A
000F82	4040404040404040			1045	DC C' B
000F92	40C1C2C3C4C5C6C7			1046	DC C' ABCDEFGHI C
000FA2	40D1D2D3D4D5D6D7			1047	DC C' JKLMNOPQR D
000FB2	4040E2E3E4E5E6E7			1048	DC C' STUVWXYZ E
000FC2	F0F1F2F3F4F5F6F7			1049	DC C' 0123456789 F
				1050 *	0123456789ABCDEF
				1051	END

RELOCATION DICTIONARY

ASM 0200 16.45 08/09/76

POS.ID	REL.ID	FLAGS	ADDRESS
0001	0001	08	0001C1
0001	0001	08	0001C5
0001	0001	08	0003ED
0001	0001	08	0003F1

CROSS-REFERENCE

ASM 0200 16.45 08/09/76

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BASEREG	00001	0000000B	00042	00057 00058 00059
BASEREG2	00001	0000000A	00041	00058 00059 00060 00060
BUFREG	00001	00000008	00039	00209 00251 00301 00391 00427 00430 00478 00479 00484 00485 00486 00487 00488 004
				00585 00587 00589 00591 00593
CLRLOOP	00004	000000B8	00113	00118
CLRMOVE	00006	000000E0	00125	00122
CLR1	00004	00000426	00391	00302
CNTREG	00001	00000006	00034	00035 00166 00166 00172 00189 00230 00235 00235 00238 00239 00291 00292 00293 007
				00791 00791 00794 00797 00200 00802 00803 00809 00809 00825 00825 00828 00828 008
CNTRLMSG	00002	0000042E	00395	00594
CNVRT	00004	00000964	00798	00795
COUNT	00001	00000012	00048	00432
C0	00001	00000B76	00948	00818
DATASZ	00004	00000AC8	00248	00157 00174
DATBLKAD	00004	00000AD0	00850	00163 00371
DATBLKSZ	00004	00000ACC	00849	00159 00372
DD3270	00004	00000B3C	00917	00197 00249 00253 00311 00341 00346 00369 00408 00413 00497 00501 00562 00566 006
				00759 00763
DECBREG	00001	00000007	00037	00208 00242 00242 00244 00333 00333 00335 00351 00354 00356 00400 00400 00402 004
				00423 00489 00489 00491 00506 00509 00511 00605 00605 00607 00623 00626 00628 006
				00767
DEFAULTS	00004	00000068	00089	00069
DELIM	00004	0000097E	00807	00786
DFLTNO	00002	00000052	00079	00073 00075
DFLTPTN	00004	00000068	00090	00080 00084
DFLTRLN	00002	00000128	00148	00133
DISPECB	00004	000001A6	00186	00181
DSPLY1	00004	0000026C	00239	00234
DUMMYECB	00004	00000AEC	00857	00182 00663
ENDLIST	00002	000000F2	00132	00136 00138
ENDMSG	00002	00000334	00305	00592
ENT1	00004	00000496	00427	00392
EPREG	00001	0000000F	00045	00213 00217 00218 00218 00255 00255 00348 00348 00415 00415 00503 00503 00599 005
				00620 00630 00637 00637 00741 00741 00743 00765 00765
ERRABEND	00002	0000091E	00777	00256 00287 00294 00349 00358 00416 00425 00504 00513 00600 00621 00652 00744 007
EVENREG	00001	00000002	00030	00062 00063 00064 00087 00092 00095 00103 00110 00112 00115 00115 00116 00116 001
				00182 00183 00223 00224 00224 00225 00277 00282 00284 00286 00288 00307 00308 003
				00307 00398 00399 00432 00481 00482 00517 00518 00519 00530 00545 00570 00596 005
				00633 00635 00641 00644 00648 00650 00654 00655 00660 00661 00661 00663 00664 007
				00751 00752 00754 00771 00771 00783 00783 00789 00790 00796 00798 00817 00818 008
				00829 00829 00833
FINDFRST	00006	00000252	00233	00237
FININIT	00004	00000136	00154	00131 00149
FLDMOVE	00006	000009D4	00836	00832
FLDXLATE	00016	00000ED2	01034	00577
FLD1	00003	00000B77	00949	00437
FLD2	00003	00000B7A	00950	00445
FLD2CHK	00004	000004D2	00444	00438
FLD3	00003	00000B7D	00951	00453
FLD3CHK	00004	000004F0	00452	00446
FLD4	00003	00000B80	00952	00461
FLD4CHK	00004	0000050E	00460	00454
FLD5	00003	00000B83	00953	00469
FLD5CHK	00004	0000052C	00468	00462
FMT1CD	00004	0000076A	00633	00627

CROSS-REFERENCE

ASM 0200 16.45 08/09/76

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FMT1SZ	00001	000000AE	00973	00611
FMT2FLD1	00001	00000028	00980	00484
FMT2FLD2	00001	00000052	00985	00485
FMT2FLD3	00001	0000007C	00990	00486
FMT2FLD4	00001	000000A7	00995	00487
FMT2FLD5	00001	000000C4	01000	00488
FMT2SZ	00001	00000132	01010	00479 00495
FMT3SZ	00001	00000148	01026	00406
FMT4SZ	00001	00000024	01030	00339
FNDLOOP	00004	000009B6	00826	00830
FORMAT1	00004	00000B86	00954	00615 00973
FORMAT2	00004	00000C34	00975	00478 00479 00980 00985 00990 00995 01000 01010
FORMAT3	00005	00000D66	01012	00410 01026
FORMAT4	00004	00000EAE	01028	00343 01030
FOUR	00004	00000AE4	00855	00644 00743
F256	00004	00000AC4	00847	00113 00117
GETCHAR	00004	00000930	00785	00793
GETFIELD	00004	000009AC	00823	00440 00448 00456 00464 00472
GETPARM	00002	0000092C	00783	00071 00082 00134
GETPTRS	00002	000000E6	00128	00119 00123
GETTAB	00004	00000068	00087	00089 00090
HAVEPARM	00004	00000954	00794	00813
IECA0008	00001	000002AC	00260	00276
IECA0010	00004	00000380	00330	00317 00323
IECA0033	00001	000007FA	00679	00676
IECB0008	00004	000002B2	00263	00269
IECB0010	00001	00000384	00331	00329
IECB0033	00001	000008B2	00740	00672 00700 00723 00730 00733 00736
IECC0008	00004	000002E2	00277	00265
IECD0008	00002	000002CE	00270	00267
IECK0033	00002	000008AC	00737	00701 00722 00726 00726
IECO0033	00004	000008A4	00735	00686
IECQ0033	00004	00000894	00729	00690
IECT0033	00004	0000085A	00711	00721
IECU0033	00004	00000876	00719	00716 00727
IECW0033	00006	0000088A	00726	00713
IECX0033	00004	0000089C	00732	00688
IHB0005A	00002	000001F6	00205	00201
INCRPTRS	00004	0000017C	00175	00190
INDXREG	00001	00000005	00633	00070 00164 00171 00171 00171 00173 00174 00174 00179 00180 00180 00185 00210 00214 00220 002 00227 00231 00233 00236 00236 00430 00437 00445 00453 00461 00469 00530 00532 005 00551 00552 00553 00570 00571 00646 00647 00664 00751 00785 00787 00790 00792 007
INDXREG2	00001	00000008	00038	00129 00129 00142 00143 00143 00165 00169 00172 00173 00175 00175
INITLOOP	00006	00000160	00169	00177
INITPTR	00004	0000010A	00139	00151
INVLDCR	00004	0000099A	00817	00788 00808
IOINIT	00004	000001BC	00193	00188
IOLOOP	00004	0000020C	00213	00228
IOLPCNT	00004	0000023A	00226	00221
LASTMOVE	00002	000000D6	00121	00114
LOOPREG	00001	00000004	00032	00067 00068 00062 00078 00078 00132 00132 00155 00177 00207 00228 00232 00237 004 00433 00434 00434 00441 00441 00449 00449 00457 00457 00465 00465 00529 00531 005 00534 00535 00536 00537 00564 00793 00812 00824 00830 00128 00128 00130 00144 00144 00148 00148 00167 00176 00176 00189

CROSS-REFERENCE

ASM 0200 16.45 08/09/76

SYMBOL	LEN	VALUE	DEFN	REFERENCES
MAXRLN	00004	00000AB4	00843	00074
MIN1	00002	00000B74	00947	00141 00169 00214 00233 00540 00665
MVFLD	00002	000009CA	00831	00827
MODVCS	00004	00000AB8	00844	00076 00083 00087 00101 00137 00154 00207 00232
HOPTRS	00004	00000AEC	00845	00085 00093 00130 00156 00179
ODDREG	00001	00000003	00031	00066 00067 00070 00093 00094 00095 00105 00110 00257 00261 00289 00290 00291 0040439 00439 00444 00444 00447 00447 00452 00452 00455 00455 00460 00460 00463 0040468 00471 00471 00483 00484 00485 00486 00487 00488 00519 00520 00520 00522 00500542 00544 00549 00552 00555 00596 00603 00631 00638 00640 00640 00641 00642 00600645 00647 00654 00660 00665 00752 00761 00799 00800 00801 00801 00821 00836
PACK	00006	000009A6	00821	00803
PARMLoop	00004	00000948	00791	00819
PARMREG	00001	00000001	00029	00066 00072 00072 00074 00076 00083 00085 00100 00103 00104 00105 00105 00106 0010108 00110 00113 00117 00121 00122 00135 00135 00137 00139 00140 00140 00141 0010154 00155 00156 00158 00163 00164 00178 00183 00184 00184 00186 00187 00216 00200304 00306 00308 00371 00376 00394 00397 00399 00475 00480 00482 00483 00517 00500529 00543 00544 00545 00546 00549 00550 00550 00553 00557 00604 00655 00804 008
PARM2	00002	00000052	00078	00079
PCKDFARM	00001	000009E0	00839	00804 00821
POLPT	00001	00000024	00050	00288
PRNTREND	00002	0000077E	00640	00283
PRNTRNXT	00006	000007AE	00654	00649
PTRECB	00004	0000019A	00183	00185
PTRINIT	00002	000001B4	00189	00170
PTRLOOP	00004	000000EA	00130	00145
QCNTSTR	00004	0000041E	00387	00362
QUIESCE	00004	000003DE	00361	00355
RAWPARM	00004	00000ADC	00852	00798 00801
READ	00004	00000270	00241	00285 00298 00388 00422 00428 00510 00586 00588 00601 00768
READFMT1	00004	0000032C	00301	00295 00516
READFMT2	00004	000005D6	00515	00296
READFMT3	00004	000006C8	00585	00297
READRLN	00004	00000AD4	00851	00239 00241
READRTN	00004	00000318	00294	00293
RETNRG	00001	0000000E	00044	00071 00082 00134 00217 00440 00448 00456 00464 00472 00598 00603 00805 00816 008
RLN	00001	00000019	00049	00641
RLNREG	00001	00000006	00035	00211 00216 00226 00226 00241 00252 00304 00319 00345 00394 00412 00475 00500 0050604 00617 00634 00635 00636 00642 00762
RSTRKYBD	00002	00000642	00545	00568 00583
SAVE	00004	00000A6C	00842	00061 00062 00381
SAVEREG	00001	0000000D	00043	00061 00063 00064 00381
SIXFS	00004	00000AE0	00854	00796
SOHRFT	00003	0000091A	00775	00587
STRTPRTR	00004	0000064C	00549	00541
SYSOUT	00004	00000692	00570	00521
SYSOUTBF	00008	000009E8	00840	00571 00572 00573 00574 00575 00576 00577 00580
SYSPRINT	00004	00000AF0	00862	00195 00367 00579
TABBASE	00001	00000009	00040	00100 00101 00102 00102 00103 00104 00106 00108 00111 00112 00140 00165 00178 0010209 00210 00223 00225 00231 00257 00291 00308 00361 00376 00387 00399 00429 00400519 00530 00539 00542 00544 00549 00551 00552 00555 00570 00596 00635 00645 006
TABSIZE	00004	00000AC0	00846	00096 00377
THREE	00004	00000ADC	00853	00433 00794 00797 00824
TWAIT	00004	000002A8	00257	00666 00769
TYPE	00001	00000005	00047	00282
UPCOUNT	00004	0000022E	00223	00215 00219

CROSS-REFERENCE

ASM 0200 16.45 08/09/76

SYMBOL	LEN	VALUE	DEFN	REFERENCES
WRITEP	00002	000007DC	00669	00662 00772
WRITEPI	00006	00000910	00771	00651
WRITEPS	00004	000008CA	00750	00742
WRITE1	00006	0000070E	00605	00629
WRITE2	00006	00000580	00489	00512
WRITE3	00006	0000043C	00400	00424
WRITE4	00006	00000384	00333	00357
WRTFMT1	00002	0000070A	00603	00213 00590
WRTFMT2	00002	00000544	00475	00476
WRTFMT3	00002	0000042E	00394	00395
WRTFMT4	00002	00000334	00304	00305
WRTRSPNS	00002	00000544	00476	00435 00442 00450 00450 00466 00470 00547 00590
ZEROPARM	00002	00000996	00815	00810
ZEROREG	00001	00000000	00028	00095 00096 00157 00158 00159 00372 00377 00524 00539 00540 00554 00555 00656

ASSEMBLER DIAGNOSTICS AND STATISTICS

ASM 0200 16.45 08/09/76

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

HIGHEST SEVERITY WAS 0

OPTIONS FOR THIS ASSEMBLY

ALIGN, ALOGIC, BUFSIZE(STD), DECK, ESD, FLAG(0), LINECOUNT(55), LIST, NOMCALL, YFLAG
NOMLOGIC, NONUMBER, NOOBJECT, NORENT, RLD, NOSTMT, NOLIBMAC, NOTERMINAL, NOTEST, XREF(SHORT)
SYSPARM()

WORK FILE BUFFER SIZE = 7294

TOTAL RECORDS READ FROM SYSTEM INPUT 702

TOTAL RECORDS READ FROM SYSTEM LIBRARY 7851

TOTAL RECORDS PUNCHED 76

TOTAL RECORDS PRINTED 1313

APPENDIX P: REMOTE 3270 SAMPLE PROGRAM

The remote 3270 sample program, which exercises the remote 3270 display system, is distributed as a member of SYS1.SAMPLIB named SAMP327R. This PDS member contains (in the form of 80-character card images) all the source statements for the sample program and all the JCL needed to assemble, link-edit, and execute the program:

```
//SAMP327R JOB BTAM,MSGLEVEL=1
// EXEC ASMFCLG
//ASM.SYSIN DD *
.
.
.
Source Code
.
.
.
/*
//GO.DD3270 DD UNIT=address of BSC line
//GO.SYSABEND DD SYSOUT=A
/*
```

Before assembling the program, supply the address of the BSC line in the 270X control unit as the UNIT parameter in the //GO.DD3270 DD card. Place the card in the card reader, and perform the assemble, link-edit, and go procedure. The messages and instructions that appear on the screen of the display station are self-explanatory.

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT

ASM 0200 11.07 06/07/76

000000

1 SAMP327R CSECT

```
3 * THIS IS A SAMPLE PROGRAM FOR A REMOTE 3270 INFORMATION DISPLAY
4 * SYSTEM. PLEASE NOTE THAT THIS PROGRAM HAS BEEN WRITTEN FOR TWO
5 * 3270 DEVICES. BOTH OF WHICH MUST BE 3277 DISPLAYS.
6 *
7 * THIS PROGRAM IS SET UP FOR AN ASSEMBLE, LINKEDIT AND GO .
8 * THE DD3270 DD CARD WILL HAVE TO BE CHANGED TO REFLECT THE
9 * ADDRESS OF THE BSC LINE IN THE TRANSMISSION CONTROL UNIT .
10 *
11 * IT MAY BE NECESSARY TO ALTER THIS SAMPLE PROGRAM SO THAT IT
12 * MAY FUNCTION WITH MORE THAN TWO DEVICES. TO DO SO, THE FOLLOWING
13 * CARDS MUST BE CHANGED IN THE SOURCE DECK:
14 * 1. DFTRMLST MACROS (SELDSPLY)
15 * (SPECPOL)
16 *
17 * THIS SAMPLE PROGRAM IS RESTRICTED TO A MAXIMUM OF 32 DEVICES.
18 * ALL OF WHICH MUST BE ATTACHED TO ONE REMOTE 3271 CONTROL UNIT.
19 * ***** APARS FIXED *****
20 * OX12969
21 * OX12970
22 * OX13692
23 * *****
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT

ASM 0200 11.07 06/07/76

			25 *	REGISTER EQUATES	
00000			26 REGZERO	EQU 0	REGISTER 0
00002			27 REG2	EQU 2	WORK REGISTER
00003			28 WORKREG	EQU 3	WORK REGISTER
00004			29 MSGADDR	EQU 4	ADDR OF OUTPUT MESSAGE
00005			30 MSGLEN	EQU 5	LENGTH OF OUTPUT MESSAGE
00006			31 SELREG	EQU 6	SPEC POLL ADDR OF 3270 DISPLAY
00008			32 FMTREG	EQU 8	FORMAT IDENTIFIER
00009			33 DSPYABRG	EQU 9	ADDRESS OF DISPLAY TABLE
0000A			34 LNKREG	EQU 10	LINKAGE REGISTER
0000B			35 BASEREG	EQU 11	FIRST BASE REG
0000C			36 BASEREG2	EQU 12	SECOND BASE REG
0000D			37 PRNTAB	EQU 13	PRINTER RLM TABLE ADDRESS
0000F			38 RTNCDRG	EQU 15	RETURN CODE REGISTER
			40 *	EQUATES	
00000			41 ZERO	EQU 0	LENGTH OF 0
00001			42 ONE	EQU 1	LENGTH OF 1
00002			43 TWO	EQU 2	LENGTH OF 2
00003			44 THREE	EQU 3	LENGTH OF 3
00004			45 FOUR	EQU 4	FORMAT 1 IDENTIFIER
00005			46 FIVE	EQU 5	LENGTH OF TERMINAL LIST ENTRIES
00006			47 SIX	EQU 6	LENGTH OF 6
0000A			48 TP06	EQU 6	TP CODE OF 6
00007			49 SEVEN	EQU 7	LENGTH OF 7
00009			50 EIGHT	EQU 8	FORMAT 2 IDENTIFIER
00011			51 TP11	EQU X'11'	TP CODE OF 11
00010			52 SIXTEEN	EQU 16	LENGTH OF 16
00020			53 TP20	EQU X'20'	
0001A			54 TWENTY4	EQU 24	LENGTH OF 24
0001C			55 TWENTY8	EQU 28	LENGTH OF 28
00001			56 TIMEOUT	EQU X'01'	DECB TIME OUT FLAG
00010			57 SENSE	EQU X'10'	SENSE/STATUS RECEIVED FLAG
00040			58 ECTRSPTX	EQU X'40'	ENT RESPONSE TO TEXT
00040			59 EOTRCVD	EQU X'40'	EOT RECEIVED FLAG
00041			60 FORTY1	EQU X'41'	I/O ERROR COMP CODE
0006C			61 PA1	EQU X'6C'	ATTENTION ID FOR PA1 KEY
0006D			62 CLEAR	EQU X'6D'	ATTENTION ID FOR CLEAR KEY
0006E			63 PA2	EQU X'6E'	ATTENTION ID FOR PA2 (CNCL) KEY
0007D			64 ENTER	EQU X'7D'	ATTENTION ID FOR ENTER KEY
0007F			65 SEVENF	EQU X'7F'	NORMAL COMPLETION CODE
00080			66 LAST	EQU X'80'	SIGNIFIES END OF POLLING LIST

```

000000 000000 000000 000000 000000 69 SAVE (14,12)
000000 000000 000000 000000 69+ OS 0M
000000 000000 000000 000000 70+ STM 14,12,17(11) SAVE REGISTERS
000000 000000 000000 000000 71 RALR BASEREG,0 ESTABLISH ADDRESSABILITY
000000 000000 000000 000000 72 USING %,BASEREG,BASEREG2 INITIALIZE SECOND BASE
000000 000000 000000 000000 73 LR BASEREG2,BASEREG
000000 000000 000000 000000 74 AH BASEREG2,%0000
000000 000000 000000 000000 75 ST 13,SAVE+4
000000 000000 000000 000000 76 LA 13,SAVE
000000 000000 000000 000000 77 B BEGIN BRANCH AROUND PATCH 0X113692
000000 000000 000000 000000 78 OC CL%15AMP,327R' EYECATCHER 0X113692
000000 000000 000000 000000 79 OC XL2'4347' DATE LAST CHANGED 0X113692
000000 000000 000000 000000 80 OC CL8'6SYSDATE' DATE LAST ASSEMBLY 0X113692
000000 000000 000000 000000 81+ OC CL8'06/07/76' DATE LAST ASSEMBLY 0X113692
000000 000000 000000 000000 82 OC S(*) BASE/DISPLACEMENT OF PATCH 0X113692
000000 000000 000000 000000 83 PATCH OC 32X'00' PATCH 0X113692
000000 000000 000000 000000 84 BEGIN OS 0M 0X113692
000000 000000 000000 000000 85 * OPEN THE LINE GROUP
000000 000000 000000 000000 86 OPEN (DCB) OPEN THE DCB
000000 000000 000000 000000 87+ CNOP 0,4 ALIGN LIST TO FULLWORD
000000 000000 000000 000000 88+ BAL 1,*,* LOAD REGI W/LIST ADDR.
000000 000000 000000 000000 89+ DC AL1(128) OPTION BYTE
000000 000000 000000 000000 90+ DC AL3(DCBR) DCB ADDRESS
000000 000000 000000 000000 91+ SVC 19 ISSUE OPEN SVC
000000 000000 000000 000000 92 TM DC9R+49,X'10' TEST FOR SUCCESSFUL OPEN
000000 000000 000000 000000 93 BO START YES, GO TO START
000000 000000 000000 000000 94 WTO 'OPEN FAILURE'
000000 000000 000000 000000 95+ CNOP 0,4
000000 000000 000000 000000 96+ BAL 1,IMP0007A BRANCH AROUND MESSAGE
000000 000000 000000 000000 97+ DC AL2(16) TEXT LENGTH
000000 000000 000000 000000 98+ DC B'CC00000000000000' MCS FLAGS
000000 000000 000000 000000 99+ DC C'OPEN FAILURE'
000000 000000 000000 000000 100+IMB0007A OS 0M
000000 000000 000000 000000 101+ SVC 75
000000 000000 000000 000000 102 R ABNORMAL
000000 000000 000000 000000 103 START LA DSPTABRG,DSPTAB ADDRESS THE DISPLAY TABLE
000000 000000 000000 000000 104 LA SELRFG,SELDSPLY GET SELECTION ADDRESS OF FIRST 3270 DISPLAY
105 *

```

```

000082 000082 000082 000082 107 INITIAL EQU *
000082 000082 000082 000082 108 LA MSGADDR,FORMAT0 ADDR OF FORMAT0 MESSAGE
000082 000082 000082 000082 109 LA MSGLEN,FMTOZ2 LENGTH OF MESSAGE
000082 000082 000082 000082 110 BAL LNKREG,WRITETI GO WRITE FORMAT 0
000082 000082 000082 000082 111 BAL LNKREG,RETCODE CHECK RETURN CODE
000082 000082 000082 000082 112 BAL LNKREG,WAITD WAIT FOR COMPLETION
000082 000082 000082 000082 113 LA SELREG,FIVE(SELREG) ADDR NEXT SPECIFIC POLL ENTRY
000082 000082 000082 000082 114 TM ZERO(SELREG),LAST END OF SELECTION LIST
000082 000082 000082 000082 115 BO READ YES, GO ISSUE A READ
000082 000082 000082 000082 116 LA SELREG,ONE(SELREG) ADDR OF NEXT ENTRY
000082 000082 000082 000082 117 B INITIAL NO. WRITE TO REMAINING DISPLAYS

```


LOC	OBJECT	CNOE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0200 11.07 06/07/76
			000AA	119	READ	EQU	*	
0000AA	4140	B772	00778	120		LA	MSGADDR,INAREA ADDR OF INPUT AREA	
0000AE	D7FE	B772 B772	00778	0077A	121	XC	INAREA(255),INAREA CLEAR INPUT	
0000RA	D72A	B871 B871	00877	00877	122	XC	INAREA+255(43),INAREA+255 AREA	BXA12970
0000RA	4540	B632	00638	123		BAL	LNKREG,READY! GO READ A DISPLAY	
0000RE	4540	B212	00218	124		BAL	LNKREG,RETCODE CHECK RETURN CODE	
0000C2	4540	B262	00268	125		BAL	LNKREG,WAITD WAIT FOR COMPLETION	
0000C6	0701	B767 B774	0076D	0077A	126	MVC	CUDVSAVE(TWO),INAREA+TWO SAVE CU,OV	
0000CC	9102	BFF6	00FFC	127		TM	DECBD+TWENTY4,X'02' STATUS MSG RECEIVED	
0000D0	4710	B2FE	00304	128		BO	SSCHECK	
0000D4	4160	B71F	00724	129		LA	SELREG,SPECPOL ADDR OO SPEC POLL TABLE	
0000D8	1922			130		SR	REG2,REG2 CLEAR REGISTER 2	
			0000A	131	CHKIT	EQU	*	
0000DA	D501	B001 B774	00001	0077A	132	CLC	ONE(TWO,SELREG),INAREA+TWO CHECK FOR CU,OV	
0000E0	4780	B0F6	000FC	133		BE	FNCSEL YES. GET SELECTION ADDR	
0000E4	4122	B001	00001	134		LA	REG2,ONE(REG2) ADD ONE TO INDEX	
0000E8	4166	B005	00005	135		LA	SELREG,FIVE(SELREG) POINT TO END OF ENTRY	BXA13692
0000EC	9180	B000	00000	136		TM	ZERO(SELREG),LAST END OF LIST	
0000F0	4710	B2FE	00304	137		BO	ABNORMAL	
0000F4	4166	B001	00001	138		LA	SELREG,ONE(SELREG) POINT TO NEXT ENTRY	BXA13692
0000F8	47F0	B0D4	000DA	139		B	CHKIT NO KEEP CHECKING	
			000FC	140	FNCSEL	EQU	*	
0000FC	4220	B766	0076C	141		STC	REG2,INDEX SAVE INDEX BYTE	
000100	1932			142		LR	WORKREG,REG2 GET INDEX INTO ODD REGISTER	
000102	5C20	B76E	00774	143		M	REG2,SIXL MULTIPLY INDEX BY 6	
000106	1923			144		LR	REG2,WORKREG RE-ESTABLISH INDEX REG	
000109	4160	B712	00718	145		LA	SELREG,SELDSPY GET SELECTION ADDR	
00010C	1A67			146		AR	SELREG,REG2 ADDR OUTPUT ENTRY IN TABLE	
00010E	5060	B76A	00770	147		ST	SELREG,SELSAVE SAVE SELECTION ADDR	
000112	1988			148		SR	FMTREG,FMTREG CLEAR FORMAT REG	
000114	4190	B742	00748	149		LA	DSPTABRG,DSPTAB ADDR OF DISPLAY TABLE	
000118	4320	B766	0076C	150		IC	REG2,INDEX	BXA12969
00011C	4380	B0C0	00000	151		IC	FMTREG,ZERO(DSPTABRG,REG2) GET FORMAT ID	
000120	47FA	B11E	00124	152		B	FORMATR(FMTREG)	
			00124	154	FORMATR	EQU	*	
000124	47F0	B12A	00100	155		B	FMT0 FORMAT 0 ON SCREEN	
000128	47F0	B14A	00150	156		B	FMT1 FORMAT 1 ON SCREEN	
00012C	47F0	B1D6	001DC	157		B	FMT2 FORMAT 2 ON SCREEN	
			00130	159	FMT0	EQU	*	
				160	*		VERIFY THE NAME AND SOCIAL SECURITY NUMBER. ASSUMING THAT THEY	
				161	*		ARE VALID, WE SHALL CONTINUE PROCESSING.	
			00130	162	FMT01	EQU	*	
000130	4180	B004	00004	163		LA	FMTREG,FOUR GET FORMAT 1 ID	
000134	4280	B000	00000	164		STC	FMTREG,ZERO(DSPTABRG,REG2) STORE IN DISPLAY TABLE	
000138	4140	B38D	003C3	165		LA	MSGADDR,FORMAT1 ADDR OF FORMAT1 MESSAGE	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0200 11.07 06/07/76
00017C	4150	00DA		166	LA	MSGLEN,FMT1S2	LENGTH OF MESSAGE
000140	45A0	0060A		167	BAL	LNKREG,WRITETI	GO WRITE FORMAT 1
000144	45A0	0021A		168	BAL	LNKREG,RETCODE	CHECK RETURN CODE
000148	45A0	00268		169	BAL	LNKREG,WAITD	WAIT FOR COMPLETION
00014C	47F0	000AA		170	E	READ	GO READ ANOTHER DISPLAY
		00150		172	FMT1	EQU *	
000150	9570	0077C		173	CLI	INAREA+FOUR,ENTER	ENTER KEY INTERRUPT
000154	47A0	00174		174	BE	ENTERINT	YES, GO UPDATE RECORDS
000158	956C	0077C		175	CLI	INAREA+FOUR,PA1	PA1 KEY INTERRUPT
00015C	4780	0019C		176	BE	PA1INT	YES, GO MAKE HARD COPY
000160	956F	0077C		177	CLI	INAREA+FOUR,PA2	PA2 OR CNCL KEY INTERRUPT
000164	4780	0019C		178	BE	PA2INT	YES, GO DEACTIVATE TERMINAL
000168	956D	0077C		179	CLI	INAREA+FOUR,CLEAR	CLEAR KEY INTERRUPT
00016C	4780	0019C		180	BE	CLEARINT	YES, GO WRITE FORMAT 2
000170	47F0	000AA		181	B	READ	IGNORE THE INTERRUPT AND GO READ
		00174		183	ENTERINT	EQU *	
				184	* CREATE A NEW OR UPDATE AN EXISTING ENTRY IN YOUR PERMANENT		
				185	* DATA SET.		
000174	4140	00601		186	LA	MSGADDR,FRALUNP	ADDR OF MESSAGE
000178	4150	00074		187	LA	MSGLEN,FRALUNPL	LENGTH OF MESSAGE
00017C	45A0	0060A		188	BAL	LNKREG,WRITETI	GO ERASE ALL UNPROTECTED DATA
000180	45A0	0021A		189	BAL	LNKREG,RETCODE	CHECK RETURN CODE
000184	45A0	00268		190	BAL	LNKREG,WAITD	WAIT FOR COMPLETION
000188	47F0	000AA		191	B	READ	GO READ ANOTHER DISPLAY
		0019C		192	PA1INT	EQU *	
00019C	47F0	00700		193	B	NOPRINT	NO PRINTER DEFINED
		00190		194	PA2INT	EQU *	
				195	* DETERMINE IF ANY DATA WAS ENTERED. IF SO, CREATE A NEW OR UPDATE		
				196	* AN EXISTING ENTRY IN YOUR PERMANENT DATA SET. NOW DEACTIVATE THE		
				197	* TERMINAL.		
		00190		198	PA2INT1	EQU *	
000190	4140	005C1		199	LA	MSGADDR,CLOSEMG	ADDR OF CLOSE MSG
000194	4150	0003C		200	LA	MSGLEN,CLOSEMGL	LENGTH OF MSG
000198	4160	00718		201	LA	SELREG,SELDSPLY	ADDR OF SELECTION TABLE
		0019C		202	CNCEMRE	EQU *	
00019C	45A0	00608		203	BAL	LNKREG,WRITETI	GO WRITE ENDING MSG
0001A0	45A0	0021A		204	BAL	LNKREG,RETCODE	CHECK RETURN CODE
0001A4	45A0	00268		205	BAL	LNKREG,WAITD	WAIT FOR COMPLETION
0001A8	4166	00005		206	LA	SELREG,FIVE(SELREG)	POINT TO INDICATOR BYTE
0001AC	91A0	00000		207	TM	ZERO(SELREG),LAST	END OF SELECTION LIST
0001B0	4710	002F0		208	BO	CLOSE	YES, TERMINATE PROGRAM
0001B4	4166	00001		209	LA	SELREG,ONE(SELREG)	POINT TO NEXT ADDRESS
0001B8	47F0	0019C		210	B	CNCEMRE	NO, WRITE ANOTHER MESSAGE
		0018C		211	CLEARINT	EQU *	
00018C	4140	0049D		212	LA	MSGADDR,FORMAT2	ADDR OF FORMAT 2 MSG

LOC	OBJECT CODE	ADDR1	ADDR2	SYMT	SOURCE STATEMENT	
						ASM 0200 11.07 06/07/76
0001C0	4150 00F7	000F7		213	LA MSGLEN,FMT2SZ	LENGTH OF MSG
0001C4	45A0 B602	0060A		214	BAL LNKREG,WRITETI	GO WRITE FORMAT 2
0001C8	45A0 B212	0021B		215	BAL LNKREG,RETCODE	CHECK RETURN CODE
0001CC	45A0 B262	0026B		216	BAL LNKREG,WAITD	WAIT FOR COMPLETION
0001D0	4180 0008	00008		217	LA FMTREG,EIGHT	GET FORMAT 2 ID
0001D4	4280 2000	00000		218	STC FMTREG,ZERO(DSPYTABG,REG2)	STORE IN DISPLAY TABLE
0001D8	47F0 B0A4	000AA		219	B READ	GO READ ANOTHER DISPLAY
			001DC	220	FMT2	
				221	EQU *	
0001DC	9570 B776	0077C		221	CLI INAREA+FOUR,ENTER	ENTER KEY INTERRUPT
0001E0	47B0 B12A	00130		222	BE FMT01	YES. GO WRITE FORMAT 1
0001E4	956C 9776	0077C		223	CLI INAREA+FOUR,PA1	PA1 KEY INTERRUPT
0001E8	47B0 B186	0019C		224	BE PA1INT	YES. GO MAKE HARD COPY
0001EC	956E A776	0077C		225	CLI INAREA+FOUR,PA2	PA2 OR CNCL KEY INTERRUPT
0001F0	47B0 B18A	00190		226	BE PA2INT1	YES. GO DEACTIVATE TERMINAL
0001F4	956D B776	0077C		227	CLI INAREA+FOUR,CLFAR	CLEAR KEY INTERRUPT
0001F8	47B0 B186	0019C		228	BE CLEARINT	GO WRITE FORMAT 2
0001FC	47F0 B0A4	000AA		229	B READ	GO READ ANOTHER DISPLAY
			00200	230	NOFFINT	
				231	EQU *	
000200	4140 95AF	0059A		231	LA MSGADDR,NOPTR	ADDR OF NO PRINTER MSG
000204	4150 002D	0002D		232	LA MSGLEN,NOPTRL	LENGTH OF MSG
000208	45AJ B602	0060A		233	BAL LNKREG,WRITETI	GO WRITE MSG
00020C	45A0 B212	0021B		234	BAL LNKREG,RETCODE	CHECK RETURN CODE
000210	45A0 B262	0026B		235	BAL LNKREG,WAITD	WAIT FOR COMPLETION
000214	47F0 B0A4	000AA		236	B READ	GO READ ANOTHER DISPLAY

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT

ASM 0209 11.07 06/07/76

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000218	47FF B216	0021C	00218	238	RETCODE	EQU *
				239	B	RTNCDTAB(RTNCDORG) BRANCH TO CORRESPONDING ENVRV
		0021C		240	RTNCDTAB	EQU *
00021C	47F0 B23E	00244		241	B	RTNCD0 I/O SUCCESSFULLY INITIATED
000220	47F0 B240	00246		242	B	RTNCD4 DTFT BUSY
000224	47F0 B246	0024C		243	B	RTNCD8 INVALID RLN
000228	47F0 B246	0024C		244	B	RTNCD8 INVALID TYPE CODE
00022C	47F0 B24A	00250		245	B	RTNCD10 ALL SKIP BITS ON
000230	47F0 B24E	0025A		246	B	RTNCD14 LINE ERROR AT OPEN
000234	47F0 B252	0025A		247	B	RTNCD18 NO BUFFERS
000238	47F0 B256	0025C		248	B	RTNCD1C NO BUFFER POOL
00023C	47F0 B25A	00260		249	B	RTNCD20 NO BUFFER MANAGEMENT
000240	47F0 B25E	00264		250	B	RTNCD24 BSC USAGE COUNT EXCEEDED
		00244		251	RTNCD0	EQU *
000244	07FA			252	BR	LNKREG RETURN
		00246		253	RTNCD4	FOU *
000246	58A0 B73E	00744		254	S	LNKREG.EIGHTS SUBTRACT 8 FROM RETURN ADDR
00024A	07FA			255	RR	LNKREG TO RETRY THE OPERATION
		0024C		256	RTACDA	FOU *
		0024C		257	RTACDC	FOU *
00024C	47F0 B2FE	00304		258	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR
		00250		259	RTNCD10	EQU *
000250	47F0 B2EA	002F0		260	B	CLCSF ALL TERMINALS. TERMINATE
		00254		261	RTNCD14	EQU *
000254	47F0 B2FE	00304		262	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR
		00258		263	RTNCD18	EQU *
000258	47F0 B2FE	00304		264	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR
		0025C		265	RTNCD1C	EQU *
00025C	47F0 B2FE	00304		266	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR
		00260		267	RTNCD20	EQU *
000260	47F0 B2FE	00304		268	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR
		00264		269	RTNCD24	FOU *
000264	47F0 B2FE	00304		270	B	ABNORMAL THIS CONDITION SHOULD NOT OCCUR

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0200 11.07 06/07/76
			00268	272	WAITD EQU *	
				273	WAIT ECB=CECBD	
00026A	4110 BFDE	00FE4		274+	LA 1,DECB0	LOAD PARAMETER REG 1
00026C	4100 0001	00001		275+	LA 0,1(0.0)	COUNT OMITTED,1 USED
000270	0A01			276+	SVC 1	LINK TO WAIT ROUTINE
			00272	277	CHK7FCC EQU *	
000272	957F BFDE	00FE4		278	CLI CECBC,SEVENF	NORMAL COMPLETION CODE
000276	4770 9276	0027C		279	BNE CHK41CC	NO, KEEP CHECKING
00027A	07FA			280	BR LNKREG	RETURN
			0027C	281	CHK41CC EQU *	
00027C	9541 BFDE	00FE4		282	CLI DFCB0,FOURTY1	I/O ERROR COMP CODE
000280	4770 92FE	00304		283	BNE ABNORMAL	INVALID COMP CODE
000284	9506 BFFA	01000		284	CLI DECB0+TWENTY8,TP06	TP CODE OF 6
000288	4770 929A	0029A		285	BNE CKTP20	NO, CHECK OTHER TP CODES
00028C	9101 BFEE	00FF4		286	TM DECB0+SIXTN,TIMEOUT	DID DEVICE TIME OUT
000290	47E0 B2FE	00304		287	BNO ABNORMAL	NO, TERMINATE
000294	58A0 9762	0076A		288	S LNKREG,TWELVE	SUBTRACT 12 FROM RETURN ADDR
000298	07FA			289	BR LNKREG	TO RETRY THE OPERATION
			0029A	290	CKTP20 EQU *	
00029A	9520 BFFA	01000		291	CLI DECB0+TWENTY8,TP20	TP CODE OF 20
00029E	4770 92A8	002AE		292	BNE CKTP11	NO, CHECK OTHERS
			002A2	293	CKRECT EQU *	
0002A2	9140 BFF6	00FFC		294	TM DECB0+TWENTY4,FOTRSPTX	EOT RESPONSE TO TEXT
0002A6	47E0 92FE	00304		295	BNO ABNORMAL	NO, TERMINATE
0002AA	47F0 90A4	000AA		296	B READ	YES, GO GET SENSE STATUS MESSAGE
			002AF	297	CKTP11 EQU *	
0002AE	9511 BFFA	01000		298	CLI CECBC+TWENTY8,TP11	TP CODE OF 11
0002B2	4770 92FE	00304		299	BNE ABNORMAL	NO, TERMINATE
0002B6	47F0 B29C	002A2		300	B CKRECT	YES, CHECK ECT RESPONSE TO TEXT
			002BA	301	CKTP20 EQU *	
0002BA	9520 BFFA	01000		302	CLI DFCB0+TWENTY8,TP20	TP CODE OF 20
0002BE	4770 B2FE	00304		303	BNE ABNORMAL	NO, TERMINATE
				304	WRITE DECB0,TR,DCBR,INAREA,.,.,1,MF=E	
0002C2	4110 BFDE	00FE4		305+	LA 1,DECB0	LOAD DECB ADDRESS
0002C6	9405 1004	00004		306+	NI 4(1),5	
0002CA	920A 1005	00005		307+	MVI 5(1),10	
0002CE	41E0 BF7A	00F80		308+	LA 14,DCBR	
0002D2	50F0 1009	00009		309+	ST 14,8(0,1)	STORE DCB ADDRESS
0002D6	41E0 B772	0077A		310+	LA 14,INAREA	
0002DA	50E0 100C	0000C		311+	ST 14,12(0,1)	STORE AREA ADDR 0009
0002DE	41E0 0001	00001		312+	LA 14,1(0,0)	STORE AREA ADDR 0C08
0002E2	40F0 1018	00018		313+	STM 14,24(0,1)	STORE LINE NUMBER
0002E6	58F0 BFAA	00F80		314+	L 15,DCBR+48	LOAD RDWRT ROUT ADDR
0002EA	05EF			315+	BALR 14,15	A38557
0002EC	47F0 B26C	00272		316	B CHK7FCC	CHECK COMPLETION CODE

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
			002F0	318	CLCSE EQU *	
				319	CLOSE (DCBR) CLCSE THE DCB	
0002F0				320+	CNOP 0.4	ALIGN LIST TO FULLWORD
0002F0	4510 B2F2	002F9		321+	BAL 1.0+9	LOAD REG1 W/LIST ADDR
0002F4	R0			322+	DC AL1(128)	OPTION BYTE
0002F5	0C0F80			323+	DC AL3(DCBR)	DCB ADDRESS
0002F8	0A14			324+	SVC 20	ISSUE CLOSE SVC
0002FA	58D0 BF46	00F4C		325	L 13.SAVE+4	
				326	RETURN (14.12)	
0002FE	98EC D00C	0000C		327+	LM 14.12.12(13)	RESTORE THE REGISTERS
000302	07FE			328+	BR 14	RETURN

00304 330 SSCHECK EQU *
 331 * INVESTIGATE THE SENSE/STATUS BYTES SENT BY THE REMOTE DEVICE.
 332 * IF RECOVERY IS POSSIBLE, ATTEMPT TO DO SO. WE SHALL ASSUME THAT
 333 * THE ERROR IS UNRECOVERABLE AND TERMINATE.

			00304	335	ABNORMAL EQU *	
				336	ABEND 1.DUMP TAKE A DUMP	
000304				337+	DS 0M	
000304				338+	CNOP 0.4	
000304	47F0 B306	0030C		339+	R 0+8 BRANCH AROUND CONSTANT	
000308	80			340+	DC AL1(128)	DUMP/STEP CODE
000309	070001			341+	DC AL3(1)	COMPLETION CODE
00030C	5810 B302	00308		342+	L 1.0-4	LOAD CODES INTO REG 1
000310	0A00			343+	SVC 13	LINK TO ABEND ROUTINE
000312	58D0 BF46	00F4C		344	L 13.SAVE+4	
				345	RETURN (14.12)	
000316	98EC D00C	0000C		346+	LM 14.12.12(13)	RESTORE THE REGISTERS
00031A	07FE			347+	BR 14	RETURN

LDC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT

ASM 0200 11.07 06/07/76

```

00031C 0227F5          0031C 349 FORMAT0 EQU 0
00031F C71DC811C150 350 DC X'0227F5' STX,ESC,E/W
000325 C7D6D6C440D4D6D9 351 DC X'C71DC811C150' WCC, SF = PROT, SBA = 80
000332 1D6011C15F 352 DC C'GOOD MORNING.'
000337 E3C9C9E240C2C5C7 353 DC X'1D6011C15F' SF = PROT, SBA = 94
000355 D6C840E3C8C54040 354 DC C'THIS BEGINS THE DEMONSTRATION '
000365 1DC8 355 DC C'OF THE OS/STAN '
000367 F3F2F7F040 356 DC X'1DC8' SF = PROT
00036C 1D60 357 DC C'3270 '
00036E D9C5D4D6E3C540F2 358 DC X'1D60' SF = PROT
000384 1D6011C3F0 359 DC C'REMOTE SAMPLE PROGRAM.'
000389 C9D5E3C5D940E3C9 360 DC X'1D6011C3F0' SF = PROT, SBA = 240
000390 1D6011C408 361 DC C'ENTER THE FOLLOWING:'
0003A2 D5C1D4C57A 362 DC X'1D6011C408' SF = PROT, SBA = 290
0003A7 1D4013 363 DC C'NAME:'
0003AA 11C5401D60 364 DC X'1D4013' SF = UNPROT, IC
0003AF E2D6C340E2C5C340 365 DC X'11C5401D60' SBA = 320,SF = PROT
0003B8 1D40 366 DC C'SOC SEC NUM:'
0003B0 11C5E91D60 367 DC X'1D40' SF = UNPROT
0003C2 03 368 DC X'11C5E91D60' SBA = 360,SF = PROT
                                369 DC X'03' ETX
                                000A7 37C FMT05Z EQU 0--FORMAT0

0003C3 0227F5          003C3 372 FORMAT1 EQU 0
0003C6 C71D60114040 373 DC X'0227F5' STX,ESC,E/W
0003CC C9D5E3C5D940C4C1 374 DC X'C71D60114040' WCC, SF = PROT, SBA = 0
0003F7 11C150 375 DC C'ENTER DATA REQUESTED BELOW:'
0003EA 05C1D4C57A 376 DC C'11C150' SBA = 80
0003EF 1D401311C1F81D60 377 DC C'NAME:'
                                378 DC X'1D401311C1F81D60' SF = UNPROT, IC, SBA = 120,
                                379 0 SF = PROT
0003F7 C1C4C4D97A 380 DC C'ADDR:'
0003FC 1D4011C2601D60 381 DC X'1D4011C2601D60' SF = UNPROT, SBA = 160, SF = PRCT
000403 C3C9E3E87A 382 DC C'CITY:'
000408 1D4011C3C81D60 383 DC X'1D4011C3C81D60' SF = UNPROT, SBA = 200, SF = PRCT
00040F E2F7C1E3C57A 384 DC C'STATE:'
000415 1D4011C3E41D60 385 DC X'1D4011C3E41D60' SF = UNPROT, SBA = 228, SF = PRCT
00041C E9C9D77A 386 DC C'ZIP:'
000420 1D5011C3F01D6011 387 DC X'1D5011C3F01D6011' SF = UNPROT, SBA = 240,
                                388 0 SF = PROT, SBA = 280
00042A C9D5E3C5D94002C5 389 DC C'ENTER KEY: ENTER DATA:'
000440 11C540 390 DC X'11C540' SBA = 320
000443 D7C1F14002C5E7A 391 DC C'PA1 KEY: PRINT DATA:'
000457 11C5E8 392 DC X'11C5E8' SBA = 360
00045A D7C1F24040C3D5C3 393 DC C'PA2 (CNCL) KEY: DEACTIVATE TERMINAL:'
00047E 11C650 394 DC X'11C650' SBA = 400
000481 C3D3C5C1D940D2C5 395 DC C'CLEAR KEY: CONTROL OPTIONS:'
00049C 03 396 DC X'03' ETX
                                000DA 397 FMT15Z EQU 0--FORMAT1

```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
				0049D	399	FORMAT2 EQU *
000490	0227F5			400	DC	X'0227F5' STX.ESC.F/W
0004A0	C711404013			401	DC	X'C711404013' WCC. SBA = 0. IC
0004A5	E7E7E8E8F4C3C3C4			402	DC	C'XYYVACDD'
0004AE	1140E8			403	DC	X'1140E8' SBA = 40
0004B1	E3D640D9C5D9F4C5			404	DC	C'TO REQUEST STAM DLT -- ENTER REQUEST FOR TEST MESS'
0004E3	C1C7C540D6E9C5D9			405	DC	C'AGE OVER SAMPLE FORMAT ABOVE: XX=TEST NO. (23-28)'
000515	4040E8E87ED9C5D7			406	DC	C' YY=REPEATS (01-99) CCOD=ADDRESS OF TARGET DEVICE'
000547	E3C8C5D540C8C9E1			407	DC	C'THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KE'
000579	E840E3D640D9C5F2			408	DC	C'Y TO RESUME AFTER TEST.'
000590	11C3C8			409	DC	X'11C3C8' SBA = 200
000593	03			410	DC	X'03' ETX
				000F7	411	FMT252 EQU *-FORMAT2
				00594	413	NOPTW EQU *
000594	0227F1			414	DC	X'0227F1' STX.ESC.WRITE
000597	C61DC811C6F8			415	DC	X'C61DC811C6F8' WCC. SF = PROT, SBA = 440
00059D	D5D640D7D9C9D5F3			416	DC	C'NO PRINTER DEFINED FOR THIS PROGRAM'
0005C0	03			417	DC	X'03' ETX
				0002D	418	NOPTWL EQU *-NOPTW
				005C1	420	CLCSENG EQU *
0005C1	0227F5			421	DC	X'0227F5' STX.ESC.E/W
0005C4	C71140401DCA			422	DC	X'C71140401DCA' WCC. SBA = 0. SF = PROT
0005CA	4040404040E3C8C5			423	DC	C' THE REMOTE 3270 SAMPLE PROGRAM HAS CONCLUDED.'
0005FC	03			424	DC	X'03' ETX
				0007C	425	CLCSENGL EQU *-CLOSENG
				005FD	428	READBUF EQU *
0005FD	0227F203			429	DC	X'0227F203' STX.ESC.RD BUF.ETX
				00004	430	READRUF EQU *-READBUF
				00601	432	ERALUNP EQU *
000601	02276F03			433	DC	X'02276F03' STX.ESC.EAU.ETX

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
				00004	434	ERALUNPL EQU *-ERALUNP


```

000608
      436 *      READ AND WRITE MACROS
      437       OS      OF
00608      438 WRITEYI EQU *
      439       WRITE DECBD.YI.DCBR.(MSGADDR).(MSGLEN).(SELREG).1.NF=E
000608 4110 BFDE      00FE4      440+      LA      1.DECBD      LOAD DCB ADDRESS
00060C 9200 1004      00004      441+      MVI     4(1).0      XA01648
000610 9202 1005      00005      442+      MVI     5(1).2
000614 4050 1006      00006      443+      STM     MSGLEN.6(0.1)      STORE LENGTH      0C08
000618 41E0 BF7A      00F80      444+      LA      14.DCBR
00061C 50E0 1008      00008      445+      ST      14.8(0.1)      STORE DCB ADDRESS
000620 5040 100C      0000C      446+      ST     MSGADDR.12(0.1)      STORE AREA ADDR  0C08
000624 5060 1014      00014      447+      ST     SELREG.20(0.1)      STORE TERM. LIST ADDR.
000628 41E0 0001      00001      448+      LA      14.1(0.0)
00062C 40E0 1018      00018      449+      STM     14.24(0.1)      STORE LINE NUMBER
000630 58F0 BF4A      00F80      450+      L       15.DCSR+48      LOAD RDWRT RQUT ADDR
000634 05FF      451+      BALR   14.15      A38557
000636 07FA      452      BR     LNKREG
    
```

```

00638      454 REACTI EQU *
      455       READ DECBD.YI.DCBR.INAREA.300.PCLDSPLY.1.NF=E
000638 4110 BFDE      00FE4      456+      LA      1.DECBD      LOAD DCB ADDRESS
00063C 9200 1074      00004      457+      MVI     4(1).0      XA01648
000640 9201 1005      00005      458+      MVI     5(1).1
000644 41E0 012C      0012C      459+      LA      14.30C(0.0)
000648 40F0 1306      00006      460+      STM     14.6(0.1)      STORE LENGTH      0C08
00064C 41E0 BF7A      00F80      461+      LA      14.DCBR
000650 50E0 100A      00008      462+      ST      14.8(0.1)      STORE DCB ADDRESS
000654 41E0 9772      0077A      463+      LA      14.INAREA      0C08
000658 50E0 100C      0000C      464+      ST     14.12(0.1)      STORE AREA ADDR  0C08
00065C 41E0 972D      00733      465+      LA      14.PCLDSPLY
000660 57E0 1014      00014      466+      ST     14.20(0.1)      STORE TERM. LIST ADDR.
000664 41E0 0001      00001      467+      LA      14.1(0.0)
000668 40F0 1018      00018      468+      STM     14.24(0.1)      STORE LINE NUMBER
00066C 58F0 9FAA      00F80      469+      L       15.DCSR+48      LOAD RDWRT RQUT ADDR
000670 05FF      470+      BALR   14.15      A38557
000672 07FA      471      BR     LNKREG
    
```

LCC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
				00674	473 REACTRV EQU *	
				474	READ CECBD,TRV,DCBR,INAREA,256,.1,MF=F	
000474	4110 BFDE	00FE4		475+	LA 1,CECBD	LOAD DECR ADDRESS
000478	9404 1004	00004		476+	NI 4(1),4	
00067C	9219 1005	00005		477+	MVI 5(1),27	
0006A0	41E0 0100	00100		478+	LA 14,256(0,0)	000B
000684	40E0 1006	00006		479+	STH 14,6(0,1)	STORE LENGTH 000B
000688	41E0 9F7A	00FA0		480+	LA 14,DCBR	
00068C	50E0 1028	00078		481+	ST 14,2(0,1)	STORE DECR ADDRESS
000690	41E0 8772	00778		482+	LA 14,INAREA	000B
000694	50E0 100C	0007C		483+	ST 14,12(0,1)	STORE AREA ADDR 000B
000698	41E0 0001	00001		484+	LA 14,1(0,0)	
00069C	40E0 1018	00018		485+	STH 14,24(0,1)	STORE LINE NUMBER
0006A0	50F0 BFAA	00FB0		486+	L 15,DCBR+48	LOAD ROWRT ROUT ADDR
0006A4	05EF			487+	BALR 14,15	A38557
0006A6	07FA			488	BR LNKREG	RETURN

				006A8	450 WRITETIV EQU *	
				491	WRITE DECBD,TIV,DCBR,(INAREA,(MSGADDR)),(256,(MSGLEN)),	
					{SELREG},1,MF=F	
0006A8	4110 BFDE	00FE4		492+	LA 1,CECBD	LOAD DECR ADDRESS
0006AC	9200 1004	00004		493+	MVI 4(1),0	XA01648
0006B0	9200 1005	00005		494+	MVI 5(1),13	
0006B4	41E0 0100	00100		495+	LA 14,256(0,0)	000B
0006B8	40E0 1006	00006		496+	STH 14,6(0,1)	STORE LENGTH 000B
0006BC	4050 102A	0002A		497+	STH MSGLEN,42(0,1)	STORE WLFNGTH 000B
0006C0	41E0 8F7A	00FB0		498+	LA 14,DCBR	
0006C4	50E0 1008	00008		499+	ST 14,8(0,1)	STORE DECR ADDRESS
0006C8	41E0 9772	00778		500+	LA 14,INAREA	000B
0006CC	50F0 100C	0007C		501+	ST 14,12(0,1)	STORE AREA ADDR 000B
0006D0	5040 102C	0007C		502+	ST MSGADDR,44(0,1)	STORE WAREA ADDR 000B
0006D4	5060 1014	00014		503+	ST SELREG,20(0,1)	STORE TERM. LIST ADDR.
0006D8	41E0 0001	00001		504+	LA 14,1(0,0)	
0006DC	40F0 1018	00018		505+	STH 14,24(0,1)	STORE LINE NUMBER
0006E0	50F0 BFAA	00FB0		506+	L 15,DCBR+48	LOAD ROWRT ROUT ADDR
0006E4	05EF			507+	BALR 14,15	A38557
0006E6	07FA			508	BR LNKREG	RETURN

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT

ASM 0200 11.07 06/07/76

```

006E8 510 REACTT EQU *
006E9 511 READ CECBD,TT,DCBR,(MSGADDR).256..1.MF=E
006EA 512+ LA 1,DCBD LOAD DCB ADDRESS
006EB 513+ NI 4(1),4
006EC 514+ MVI 5(1),3
006ED 515+ LA 14,226(0,0)
006EE 516+ STM 14,6(0,1) STORE LENGTH 000B
006EF 517+ LA 14,DCBR 0C0B
006F0 518+ ST 14,6(0,1) STORE DCB ADDRESS
006F1 519+ ST MSGADDR.12(0,1) STORE AREA ADDR 0C0B
006F2 520+ LA 14,1(0,0)
006F3 521+ STM 14,24(0,1) STORE LINE NUMBER
006F4 522+ L 15,DCBR+48 LOAD ROWRT ROUT ADDR
006F5 523+ BALR 14,15 438557
006F6 524 RR LNKREG RETURN

```

```

526 * DISPLAY SELECTION ADDRESSES
527 * THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS:
528 * 1. 0TH CU, 0TH DEV (60604040)
529 * 2. 0TH CU, 1ST DEV (6060C1C1)
530 SELDSPLY OFTRMLST OPRLST.(6060404020.6060C1C120)
000710 531+SELDSPLY DS 0X A28630
000718 606040402D 532+ DC X'6060404020' TERMINAL LIST ENTRY A46E21
00071D 01 533+ DC AL(1) PROCEDURE FLAGS
00071E 6060C1C12D 534+ DC X'6060C1C120' TERMINAL LIST ENTRY A46E21
000723 82 535+ DC AL(130) PROCEDURE FLAGS

```

```

537 * DISPLAY SPECIFIC POLLING ADDRESSES
538 * THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS:
539 * 1. 0TH CU, 0TH DEV (40404040)
540 * 2. 0TH CU, 1ST DEV (4040C1C1)
541 SPECPLC OFTRMLST OPRLST.(4040404020.4040C1C120)
000724 542+SPECPLC DS 0X A28E30
000724 404040402D 543+ DC X'4040404020' TERMINAL LIST ENTRY A46E21
000729 01 544+ DC AL(1) PROCEDURE FLAGS
00072A 4040C1C12D 545+ DC X'4040C1C120' TERMINAL LIST ENTRY A46E21
00072F A2 546+ DC AL(170) PROCEDURE FLAGS

```

LCC OBJECT CODE ADDR1 ADDR2 SYMT SOURCE STATEMENT

ASM 0200 11.07 06/07/76

```

54E * DISPLAY GENERAL POLLING ADDRESS
54F PCLOSPLY OFTRMLST AUTOWLST.(40407F7F20.37373737)
000730 0292 550+ DC 2YL(2) TOTAL AND ACTIVE ENTRY CCUN000A
000732 00 551+ DC VL1(X'20'06+X'10'*1) PACKED WIDTH AND WRAP BIT 000A
000733 552+PCLOSPLY DS 0X 000A
000734 40407F7F20 553+ DC XL5'40407F7F20' A57567
000738 01 554+ DC VL1(1) 000A
000739 3737 173737 555+ DC XLE'3737373737' A57567
00073E 02 556+ DC VL1(2) 000A
00073F FE 557+ DC X'FE' END OF LIST CHARACTER 000A
000740 0010 558+ DC VL2(2*6+4) OFFSET 000A

```

```

561 * CONSTANTS
000742 1000 562 MAC96 DC M'4096'
000744 0000000A 563 EIGHTH DC F'8' CONSTANT OF 8
000748 0000000000000000 564 DSFTAB DC XL32'0' DISPLAY TABLE
000768 0000000C 565 TWELVE DC F'12' LENGTH OF 12
00076C 00 566 INCEX DC X'00' INDEX BYTE SAVE AREA
00076D 0000 567 CUCVSAVE DC X'0000' CU.DV SAVEAREA
0007AF 00
000770 00000000 568 SELSAVE DC F'0'
000774 00000004 569 SIX DC F'6'
000778 0000000000000000 570 INAREA DC 500F'0' INPUT AREA
007D0 571 INAREAL EQU *-INAREA
000F4F 0000000000000000 572 SAVE DC 18F'C'
000F90 573 DS 0F
000F90 90 574 OFCEADDR DC X'80'
000F91 000FE4 575 DC AL3(DECBD)
576 DCER DCB DSNRG=CX,MACRF=(R,W),DONAME=NO,3P70,F,ROPT=Y,DEV0=RS,
MODE=(,A,A),CODE=EBCDIC

```

```

578+* DATA CONTROL BLOCK
579+*
000F94 00F80 580+ ORG 0-20 TO ELIMINATE UNUSED SPACE
000F80 581+DCER DS 0F'0' ORIGIN ON WORD BOUNDARY
000F80 00F94 582+ ORG 0+20 TO ORIGIN GENERATION

```

```

584+* COMMON ACCESS METHOD INTERFACE
000F94 00 586+ DC AL1(0) BUFP0
000F95 000001 587+ DC AL3(1) BUFC9
000F98 0000 588+ DC AL2(0) BUFL
000F9A 1000 589+ DC BL2'0001000000000000'
000F9C 00000001 590+ DC A(1) I0BAD DSNRG

```

592+* FOUNDATION EXTENSION

000FA0 00	594+	DC	BL1'00000000'	RFTEK,BFLN,MIARCHY
000FA1 10	595+	DC	BL1'00010000'	BTAM EROPT CODE
000FA2 FF	596+	DC	AL1(255)	BTAM BUFFER COUNT
000FA3 00	597+	DC	AL1(0)	
000FA4 00	598+	DC	BL1'00000000'	
				RECFN
000FA5 000000	599+	DC	AL3(0)	EXLST

601+* FOUNDATION BLOCK

000FAB C4C4F3F2F7F0A0A0	607+	DC	CL8'003270'	ODNAME
000FB0 02	604+	DC	BL1'00000010'	OFLGS
000FB1 00	605+	DC	BL1'00000000'	IFLG
000FB2 2020	606+	DC	BL2'0010000000100000'	
				MACR

608+* BTAM INTERFACE

000FBA 00000000	610+	DC	A(0)	LEBB
000FBB 00	611+	DC	BL1'00000000'	MODE
000FB9 00	612+	DC	BL1'10000000'	MAS.CODE
000FBA 0000000000000000	613+	DC	XL28'0'	CONTROL CHARS
000FB4 0000000000000000	614+	DC	4F'0'	RESERVED

000FE4	616	READ	DCRD,PI,DCRR,MF=L	
000FE4 00000000	617+	DS	OF	
000FFB 00	618+DFCED	DC	A(0)	EVENT CONTROL BLOCK
000FFB 00	619+	DC	BL1'000'	
000FF9 01	620+	DC	AL1(1)	TYPE FIELD
000FFA 0000	621+	DC	AL2(0)	LENGTH
000FFC 00000000	622+	DC	A(0CDR)	OCB ADDRESS
000FF0 00000000	623+	DC	A(0)	AREA ADDRESS
000FF4 00000000	624+	DC	A(0)	ERRR INFO. FIELD ADDR
000FFA 00000000	625+	DC	A(0)	TERMINAL LIST ADDRESS
000FFC 0000	626+	DC	AL2(0)	LINE NUMBR
000FFE 0000	627+	DC	AL2(0)	RESPONSE FIELD
001000 00	628+	DC	AL1(0)	TP-OP CODE
001001 00	629+	DC	AL1(0)	ERROR STATUS
001002 0000	630+	DC	AL2(0)	CSW STATUS
001004 00000000	631+	DC	AL4(0)	CURRENT ADDR LIST PTR
001008 00000000	632+	DC	AL4(0)	CURRENT ADDR POLL PTR
	633	END		

RELOCATION DICTIONARY

ASM 0200 11.07 06/07/76

POS.ID	REL.ID	FLAGS	ADDRESS
0001	0001	08	000051
0001	0001	09	0002F5
0001	0001	0A	000F01
0001	0001	0C	000FEC

CROSS-REFERENCE

ASM 0200 11.07 06/07/76

SYMBOL	LEN	VALUE	DEPN	REFERENCES
ABNORMAL	00001	00000304	00335	00102 00137 00258 00262 00264 00266 00268 00270 00283 00287 00295 00299 00303
MASEREG	00001	00000008	00035	00071 00072 00073
MASEREG2	00001	0000000C	00036	00072 00073 00074
BEGIN	00002	0000004C	00084	00077
CHKEDT	00001	000002A2	00293	00300
CHKIT	00001	000000DA	00131	00139
CHK41CC	00001	0000027C	00281	00279
CHK7PCC	00001	00000272	00277	00316
CKTP11	00001	000002AE	00297	00292
CKTP20	00001	0000029A	00290	00285
CLEAR	00001	00000060	00062	00179 00227
CLEARINT	00001	0000019C	00211	0019C 00228
CLOSE	00001	000002F0	00318	00208 0026C
CLOSEMG	00001	000005C1	00420	00199 00425
CLOSEMGL	00001	0000007C	00425	00200
CLOSVAVE	00002	00000760	00567	0012A
ACBR	00004	00000F40	00581	0009C 00092 00308 00314 00323 00444 00480 00461 00469 00480 00496 00498 00506 00517 00522 00622
DECRO	00004	00000FE4	0061A	00127 00274 00278 00282 00284 00286 00291 00294 00299 00307 00309 00440 00456 00475 00482 00512 00575
DSPTAB	30037	00000748	00564	001C3 00149
DSPTABRG	00001	00000009	00033	00193 00149 00151 00164 00219
EIGHT	00001	00000008	00050	00217
EIGHTR	00004	00007744	00563	00254
ENTER	00001	00000070	00064	00173 00221
ENTERINT	00001	00000174	00183	00174
EOTRSPTX	00001	00000740	00058	00294
ERALUNP	00001	00000601	00432	00146 00474
ERALUNPL	00001	00000004	00474	00187
FIVE	00001	00000005	00046	00113 00135 00206
FMTREG	00001	00000008	00032	00148 00148 00152 00152 00164 00217 00219
FMT0	00001	00000130	00159	00155
FMT0S2	00001	00000047	00170	001C9
FMT01	00001	00000130	00162	00222
FMT1	00001	00000150	00172	00156
FMT1S2	00001	00000004	00307	00166
FMT2	00001	0000010C	00220	00157
FMT2S2	00001	000000F7	00411	00213
FNDSEL	00001	000000FC	00140	00133
FORMATBR	00001	00000124	00154	00152
FORMAT0	00001	0000031C	00349	00108 00170
FORMAT1	00001	000007C3	00372	00165 00397
FORMAT2	00001	00000490	00399	00212 00411
FOUR	00001	00000004	00045	00163 00173 00175 00177 00179 00221 00223 00224 00227
FOURTY1	00001	00000041	00060	00282
H4096	00002	00000742	00562	00074
INR0003A	00002	00000074	00100	00096
INAREA	00004	00000778	00570	0012C 00121 00121 00122 00122 00126 00132 00173 00175 00177 00179 00221 00223 00225 00227 00310 00463 00482 00500 00571
INDEX	00001	0000074C	00566	00141 00150
INITIAL	00001	000000R2	00107	00117
LAST	00001	000000R0	00066	00114 00136 00207
LNKREG	00001	0000000A	00034	00110 00111 00112 00123 00124 00125 00147 00168 00169 00199 00199 00199 00203 00204 00205 00214 00215 00216 00233 00234 00235 00252 00254 00255 00280 00280 00280 00452 00471 00488 00508 00524

CROSS-REFERENCE

SYMBOL	LEN	VALUE	DEPN	REFERENCES	ASM 0200 11.07 06/07/76
MSGADDR	00001	00000004	00029	00108 C0120 00165 00186 00199 00212 00231 00446 00502 00519	
MSGLEN	00001	00000005	00030	00109 C0166 C0187 00200 00213 00232 00443 00497	
NOPRINT	00001	00000200	00230	00193	
NOPTX	00001	00000594	00413	00221 C0418	
NOPTRL	00001	00000020	00418	00232	
ONCEMORE	00001	0000019C	00202	00210	
ONE	00001	00000001	00042	00116 C0132 C0134 00138 00209	
PA1	00001	0000006C	00061	00175 C0223	
PA1INT	00001	0000019C	00192	00176 C0224	
PA2	00001	0000006E	00063	00177 00225	
PA2INT	00001	00000190	00194	00178	
PA2INT1	00001	00000190	00194	00226	
POLDSPLY	00001	00000733	00552	00465	
READ	00001	000000AA	00119	00115 00170 C0181 00191 00219 00229 00236 00296	
READBUF	00001	000005FD	00424	00430	
READI	00001	00000638	00454	00123	
REG2	00001	00000002	00027	00130 C0130 00134 00134 00141 00142 00143 00144 00146 00150 00151 00164 00218	
RETCODE	00001	00000218	00234	00111 C0124 C0168 00189 00204 00215 00234	
RTNCD0	00001	0000024C	00257	00244	
RTNCD0B	00001	0000000F	00038	00239	
RTNCD0TA	00001	0000021C	00240	00239	
RTNCD00	00001	00000244	00251	00241	
RTNCD01C	00001	0000025C	00265	00248	
RTNCD010	00001	00000250	00259	00245	
RTNCD014	00001	00000254	00261	00246	
RTNCD019	00001	00000259	00263	00247	
RTNCD020	00001	00000260	00267	00249	
RTNCD024	00001	00000264	00269	00250	
RTNCD04	00001	00000246	00253	00242	
RTNCD0R	00001	0000024C	00256	00243	
SAVE	00004	00000F49	00572	00075 00076 C0325 00344	
SFLDSPLY	00001	0000071A	00531	00104 C0145 C0201	
SELREF	00001	0000007A	00031	00104 00113 C0113 00114 00115 00116 C0129 00132 00135 00135 00136 00138 00138 00145 00146 00147 C0201 C0206 C0206 00207 00209 00209 00447 00503	
SELSAVE	00004	00000770	00564	00147	
SEVENF	00001	0000007F	00065	00278	
SIXL	00004	000000774	00569	00143	
SIXTN	00001	00000010	00052	00286	
SPECPOL	00001	000000724	00547	00125	
SSCHECK	00001	000000304	00330	00128	
START	00004	0000007A	00103	00093	
TIMEDUT	00001	00000001	00056	00286	
TP06	00001	00000006	00048	00284	
TP11	00001	00000011	00051	00290	
TP20	00001	00000020	00053	00291 C0302	
TWELVE	00004	000000769	00565	00288	
TWENTY4	00001	00000018	00054	00127 C0294	
TWENTY8	00001	0000001C	00055	00284 00291 C0298 00302	
TWO	00001	00000002	00043	00126 C0126 C0132 00132	
WAITD	00001	00000268	00272	00112 C0125 00169 00190 00205 00216 00235	
WORKRFG	00001	00000003	00028	00142 C0144	
WRITETI	00001	00000608	00438	00110 C0167 00188 00203 00214 00233	
ZERO	00001	00000000	00041	00114 00136 C0151 00164 C0207 00218	

ASSEMBLER DIAGNOSTICS AND STATISTICS

ASM 0200 11.07 06/07/76

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
 HIGHEST SEVERITY WAS 0
 OPTIONS FOR THIS ASSEMBLY
 ALIGN, ALOGIC, BUFSIZE(STD), DECK, ESD, FLAG(0), LINECOUNT(55), LIST, NOMCALL, YFLAG
 NOMLOGIC, NOMNUMBER, NOMOBJECT, NORENT, PLC, NOSTMT, NOLIBMAC, NOTERMIAL, NOTEST, XREF(SHORT)
 SYSPARM(1)
 WORK FILE BUFFER SIZE = 7294
 TOTAL RECORDS READ FROM SYSTEM INPUT 448
 TOTAL RECORDS READ FROM SYSTEM LIBRARY 7077
 TOTAL RECORDS PUNCHED 77
 TOTAL RECORDS PRINTED 431

Using an IBM 3872 or 3875 Modem with the Switched Network Backup feature, an application program can communicate with one or more terminals on a failing nonswitched line by having the system operator dial and connect specified terminals on a switched line.

The application programmer must do these things to use Switched Network Backup:

- Define the switched backup lines as nonswitched lines in a DCB macro instruction. (The 3872 or 3875 has the ability to convert nonswitched commands into switched operation.)
- Write code to determine that a line is out of operation or has excessive errors. (For example, the LERB macro instruction can be used to maintain error counts, and a Switched Network Backup routine can be branched to when a specified threshold has been exceeded.)
- Write a Switched Network Backup routine, which will allow the system operator to set up the modem and make the dial connection, open the switched backup line, and communicate with the designated terminal. If more than one terminal is to be connected, separate connection must be made and separate program-to-terminal communication must be conducted with each terminal.

Here is a possible procedure for using Switched Network Backup:

1. On determining line failure or excessive errors, ensure no more activity on that line.
2. The program should write a message to the system console requesting the operator to follow a designated Switched Network Backup procedure, specifying the failing line and the

dial number for the terminal to be connected on the backup line.

3. The system operator, following the procedure provided for him, should make required changes to the 3872 or 3875 Modem, dial the designated number, tell the terminal operator to make required changes to his 3872 or 3875 Modem, and notify the program from the console when assured that proper connection exists.
4. The application can now communicate with the connected terminal. READ and WRITE macro instructions should refer to the DCB defined for the backup line. Since the 3872 or 3875 allows communication as though the terminal were still connected on a nonswitched line, the same READ and WRITE macro instruction types applicable to nonswitched operation are used for backup operation. For multipoint WRITE macro instructions, the same addressing list can be used for selection that was used to write to the terminal prior to backup operation. For multipoint READ macro instructions, a polling terminal list of the AUTOLST format must be defined with a single entry, containing the polling characters for the connected terminal. (This will prevent recurrent timeouts from no response if devices other than the one connected were to be polled.)
5. When communication is complete between the program and the terminal (determined either by the terminal operator hanging up or by the program determining that no further communication is required), the program should notify the system operator to return the 3872 or 3875 Modem to normal operation.

For additional information, see IBM 3872 Modem User's Guide, GA27-3058, or IBM 3875 Modem User's Guide, GA33-0001.

This Page Intentionally Left Blank

APPENDIX R: BTAM STORAGE ESTIMATES FOR VS2

In VS2, BTAM modules occupy approximately 1900 bytes of the following storage in the nucleus as follows:

- IGC058 310 bytes
- IECTATEN 570 bytes
- IECTSVC 1000 bytes

The virtual storage requirement for retrieving or storing a data set with BTAM is estimated by adding the following requirements:

- coding-space area (BTAM module and macro expansion size)
- control-block area by line groups (Device I/O module size)
- control-information area
- control-block area by lines
- channel-program area by lines
- code translation table area

The coding-space estimate (Figure 125) includes the BTAM code required to support the READ, WRITE, REQBUF, and RELBUF macro instructions, and dynamic buffer allocation. This code is sharable across line groups and is not duplicated for multiple data control blocks open at the same time.

Description	Remote Requirement	Local 3270 Requirement
Primary requirement:		
• without buffer management	11000	5000
• with buffer pool support (REQBUF and RELBUF)	11500	5500
• with dynamic buffering	14000	(Not applicable)
Optional requirement:		
• online test	3200	750
• if ONLTST macro is used	500	(Not applicable)
• line-error print (LERPRT)	400	(Not applicable)
• line open (LOPEN)	600	600
• translate (TRANSLATE)	180	(Not applicable)
• change entry for polling list or local 3270 (CHGNTRY)	380	90
• if RESETPL macro is used	800	800
with ATTENT specified	(not applicable)	250
with POLLING specified	600	(not applicable)
with ANSRING specified	330	(not applicable)
• World Trade Telegraph terminals	1200	(Not applicable)
• change entry for expanded ID verification (CHGNTRY)	50	(Not applicable)
• edit routine TPEDIT, IECTEDIT	2048	(Not applicable)

Figure 125. BTAM Coding-Space Estimate

Select the appropriate entry from Figure 126 for each type of terminal to be supported under BTAM.

Terminal Device	Virtual Storage Requirement (bytes)
IBM 1030 Data Collection System	224
IBM 1030 Data Communication System with Autopoll	224
IBM 1050 Data Communication System	240
IBM 1050 Data Communication System on a switched network	312
IBM 1050 Data Communication System with Autopoll	208
IBM 1060 Data Communication System	192
IBM 1060 Data Communication System with Autopoll	200
IBM 2260 Display Unit attached as a remote terminal with a 2701 adapter	296
IBM 2740 Communication Terminal	152
IBM 2740 Communication Terminal with checking	272
IBM 2740 Communication Terminal with checking and OIU (Optical Image)	264
IBM 2740 Communication Terminal with station control	152
IBM 2740 Communication Terminal with station control and checking	216
IBM 2740 Communication Terminal on a switched network	176
IBM 2740 Communication Terminal with checking on a switched network	272

Figure 126 (Part 1 of 2). BTAM Control-Information Area Estimate by Device Type

Terminal Device	Virtual Storage Requirement (bytes)
IBM 2740 Communication Terminal with transmit control on switched network	192
IBM 2740 Communication Terminal with checking and OIU on a switched network	336
IBM 2740 Communication Terminal with transmit control and checking on a switched network	272
IBM 2740 Communication Terminal with station control, checking, and Autopoll	216
IBM 2740 Communication Terminal with station control and Autopoll	152
IBM 2741 Communication Terminal	112
IBM 2741 Communication Terminal on a switched network	144
IBM 3270 Display Station Local	120
IBM BSC Terminal on a nonswitched, point-to-point network	288
IBM BSC Terminal on switched network	424
IBM BSC Terminal on a nonswitched multipoint network	320
AT&T Model 33/35 TWX stations	176
AT&T 83B3 Selective Calling Stations	152
Western Union Plan 115A Outstation	144
World Trade Telegraph Terminals	160

Figure 126 (Part 2 of 2). BTAM Control-Information Area Estimate by Device Type

Select the appropriate entry from Figure 127 for each line group.

Control Block	Remote Virtual Storage Requirement (bytes)	Local 3270 Virtual Storage Requirement (bytes)
Data control block	56	56
• with binary synchronous communications	84	
Data extent block	120+(4 per line)	56+(4 per line)
Interruption request block	124	250
Interruption queue element	24	48

Figure 127. BTAM Control Block Area for Each Line Group

Select the appropriate entries from Figure 128 for each line and add to obtain the total control block area requirement for each line.

Control Block	Remote Virtual Storage Requirement (bytes)	Local 3270 Virtual Storage Requirement (bytes)
Data event control block	40	40
• with binary synchronous communications	48	
Input/output block with a minimum of 4 CCWs		
• with binary synchronous communications	96	
• with start/stop communications	120	
• with local 3270 communications		88*
Unit control block	20	44
Line error block (LERB macro instruction)	20	(Not applicable)

*Only has space for 3 CCW's.

Figure 128. BTAM Control Block Space for Each Line

Select entries from Figure 129 for each line according to its device.

Terminal Device	Virtual Storage Requirement (bytes)
IBM 1030 Data Collection System	64
IBM 1030 Data Collection System (P)	88
IBM 1050 Data Communications System	64
IBM 1050 Data Communications System (P)	80
IBM 1050 Data Communications System (A,D)	88
IBM 1060 Data Communications System	56
IBM 1060 Data Communications System (P)	80
IBM 2740 Communications Terminal	40
IBM 2740 Communications Terminal (C)	48
IBM 2740 Communications Terminal (C,O)	64
IBM 2740 Communications Terminal (A)	48
IBM 2740 Communications Terminal (D)	56
IBM 2740 Communications Terminal (A,C)	48
IBM 2740 Communications Terminal (D,C)	64
IBM 2740 Communications Terminal (D,C,O)	64
IBM 2740 Communications Terminal (A,D,T)	64
IBM 2740 Communications Terminal (A,D,T,C)	64
IBM 2740 Communications Terminal (S)	56
IBM 2740 Communications Terminal (S,P)	88
IBM 2740 Communications Terminal (S,C)	64
IBM 2740 Communications Terminal (S,C,P)	88
IBM 2741 Communications Terminal	48
IBM 2741 Communications Terminal (A)	48
IBM 3277 Display Station (Local)	24
IBM BSC Terminal on a nonswitched point-to-point network	72
IBM BSC Terminal on a switched network	80
IBM BSC Terminal on a nonswitched multipoint network	88
IBM 2260 Display Unit (R)	64
AT&T 83B3 Selective Calling Stations	48
AT&T Model 33/35 Teletypewriter Exchange Terminal using the eight-bit Data Interchange Code (A)	56
AT&T Model 33/35 Teletypewriter Exchange Terminal using the eight-bit Data Interchange Code (D)	56
Western Union Plan 115A Outstations	40
World Trade Telegraph Terminals	40
Where:	
A = Automatic answering	
C = Checking	
D = Dialing (automatic calling)	
P = Autopoll	
R = Remote attachment with an IBM 2701 Type III Adapter	
S = Station control	
T = Transmit control	
O = IBM 2760 Optical Image Unit	

Figure 129. BTAM Channel Program Space Estimate by Device per Line

Figure 130 contains the storage requirement for code-translation tables (AMSTRTAB)_per device.

Description	Virtual Storage Requirement (bytes)
Input Translation (transmission code to EBCDIC)	256
Output Translation (EBCDIC to transmission code)	256

Figure 130. Storage Requirement for Code-Translation Tables for BTAM

BTAM EXAMPLE

This example shows how to estimate the dynamic storage required by a telecommunications application with Auto Poll and buffer pool support but without dynamic buffering or binary synchronous communications.

Assume a VS2 configuration of:

One line with three IBM 1050 Data Communication System Terminals and one line with two of these same terminals.

Basic system information:

- one line group
start-stop error recovery procedures
- translation
- one DECB per line

```
BTAM coding-space area estimate (11500+180+380).... 12060
Control-information area by device type..... 208
Control-block area estimate for one line group..... 332
Control-block area estimate for two lines..... 360
Channel-program space for two lines..... 160
Translation tables for input and output (256x2).... 512
```

Total 13,632 bytes

ABEND	ABnormal END
ACK	ACKnowledgment
ADDR(S)(ING)	ADDRess(ES),(ING)
ADJ	ADJust
AID	AttentIOn IDentifier
ALT	ALternate
ANSWR	ANSWeR
APPEND	APPENDage
ASCII	AmerIcan national Standard Interchange
ATTN	ATTentIOn
AUTH	AUTHorize/AUTHority
AUTO	AUTOMatic
AUTOPOLL	AUTOMatic POLLing of a multi-point line.
AVAIL	AVAILable
BCT	Branch on Count
BISYNC	BIinary SYNChronous communication
BLK	BLock
BR	BRanch
BRK	BReaK
BSC	See BISYNC
BTAM	BasIc Telecommunications Access Method
BTBL	BRanch TABLe
BUFR	BUFFer
BUSSOUT	Outbound signals or power from processing unit
CALC	CALCulate
CC	CondItIOn Code
CCC	ChanneL Control Check
CCH	ChanneL Check Handler
CCW	ChanneL Control Word
CCWCMD	CCW CoMmanD code
CCWREG	Register used to address the CCW
CC3	CondItIOn Code of 3
CDC	ChanneL Data Check
CE	ChanneL End
CEA	ChanneL End Appendage
CHAN	CHANnel
CHAR(S)	CHARacter(S)
CHE/ABE	ChanneL End or ABnormal End appendage
CHGNTRY	CHAnGe eNTRY
CHK	CHeck
CLR	CLear
CMD(S)	CoMmanD(S)
CNT(S)	CoUNt(S)
COM CH	CoMmanD CHaining
COMP	CoMPLete
COND	CoNDItIOn
CONN	CoNNection
CONT	CoNTInuous, CoNTInue
CPA	ChanneL Program Area
CPS	ChAracters Per Second
CPU	CentraL Processing Unit
CR	Carriage Return or credit symbol
CSECT	Control SECTIon
CSW	ChanneL Status Word

CTRL	Control
CUE	Control Unit End
CVT	Communication Vector Table
DBL WD	DOUBLEWORD
DC	Data Check or Define Constant
DCB	Data Control Block
DE	Device End
DEB	Data Extent Block
DEC	DECimal
DECB	Data Event Control Block
DECR	DECREment
DEL	the DElete character
DEV	DEvice
DIAG	DIAGnosis
DLE	Data Link Escape
D/TNNNN	Device Type with NNNN relating to the device number
DUP	DUPLICATE
DYN(BUF)	DYNAMIC BUffering
EAU	Erase All Unprotected
ECB	Event Control Block
ENQ	ENquiry
EOA	End Of Address
EOL	End Of Line
EQM	End Of Message
EOR	End Of Record
EOT	End-Of-Transmission
EP	Entry Point
ERP	Error Recovery Proecedures
ERR	ERRor
ERRCT	ERRor Count
ESC	ESCAPE
ESR	Extended Svc Router
ESTAE	Extended Specify Task Abnormal Exit macro instruction
ETB	End of Text Block
ETX	End of Text
EWA	ERP (Error Recovery Procedure) Work Area
EWASTUP	ERP Status UPdate
EWR	Erase WRite
EXCP	EXecute Channel Program
EXT	EXTent OR EXTernal
EXTKEY	EXTernal KEY
FEAT	FEATure
FLD	FiELD
FLG	FLaG
GRAF	GRAPhics
GS	Group Separator
GT	Greater Than
HIO	Halt Input/Output
IAM	Identification of terminal
ICC	Interface Control Check
ID	IDentification
INCR	INCREment
IND	INDicate
INFO	INFORMAtion

INIT	INITialize
INST	INSTRUction
INTERV	INTERVention
INTRNL	INTeRNAL
I/O	Input or Output
IOB	Input/Output Block
IOS	Input/Output Supervisor
IOSB	Input/Output Supervisor Block
IPT	INpuT
IQE	Interruption Queue Element
IR	Intervention Required
IRB	Interruption Request Block
ITB	Intermediate Text Block
LD	Load OR Lost Data
LERB	Line Error Recording Block
LVL	LeVeL
LZF	Left Zero Fill
L3270	Local 3270
MAINT	MAINTenance
MASTERUCB	MASTER Unit Control Block
MAX	MAXimum
MDR	Miscellaneous Data Record
MOD	MODule
MPT	MultiPoint
MSG	MeSSaGe
NAK	Negative AcKnowledgment
NOP	No OPeration
NORM	NORMal
NSI	Next Sequential Instruction
OSR	OutBoard Recording
OIU	OptiocalImage Unit
OLT	OnLine Test
OLTEP	OnLine Test Executive Program
ONLTST	This is a macro instruction
OP	OPeration
OPER	OPERand
ORIG	ORIGin
PARMLIST	PARaMeter LIST
PCI	Program Control Interrupt
PPL	Purge Parameter List
PREV	PREVious
PROT	PROTection
PSW	Program Status Word
PT	Point
PTOP	1130/2250 GRAPHICS TERMINAL
PTR	PoinTeR
R	Real
RB	Request Block
RC	Return Code
RCV	ReCeive
RD	ReaD
RDY	ReaDY
READINDX	READ INDeX
READYQ	READY Queue

REC	RECOrd or RECeive
RECFM	RECOrd ForMat
REG	REGister
REL	RELease
REQD	REQuireD
RES	RESident
RESETPL	RESET Polling List or RESET Line
RESP	RESPOse
RFT	RequeSt For Test
RLN	Relative Line Number
RQE	RequeSt Queue Element
RTN	RouTiNe
RTRY	ReTRY
R/W	Read or Write
RVI	ReVerse Interrupt
SAD	Set Address
SBA	Set Buffer Address
SDWA	System Diagnostic WorkArea
SEL	SElect
SEQ	SEQuence
SIG	SIGNal
SIO	Start Input/Output
SIOCC	Start Input/Output Condition Code
SLI	Supprss Length Indicator
SM	Status Modifier
SNO	Should Not Occur
SNS	SeNSE
SOH	Start Of Header
SOH%N	Header format for system/non-user data message with N relating to the type of message.
SOM	Start Of Message
SOR	Start Of Record
SPEC	SPECify or SPECial
S/S	Start-Stop
STAT	STATus
STRAM	Synchronous Transmit and Receive Access Method
STX	Start of TeXt
SVC	SuperVisor Call
SW	SWitched
SWLST	Operand and form of terminal list
SYS	SYStem
T.O	TimeOut
TBL	TabLe
TCB	Task Control Block
TCCW	IOS(Input/Output Supervisor) Translation Control block
TCU	Transmission Control Unit
TERM	TERMination
TP-OP CODE	TeleProcessing Operation CODE
TRANS	SEE X**** AREA OF LIST
TRANSRTN	TRANSLate RouTiNe
TRANSRTV	TRANSLate Real address To Virtual address
TRANSCODE	6 BIT CODE
UC	Unit Check
UCB	Unit Control Block

UEX	Unit EXception
US	Unit Specify
W/	With
WACK	WAIt before transmit
WCC	Write Control Character
WRT	WRiTe
WRU	Who aRe toU
WTG	Where To Go
WTO	Write To Operator
WTOR	Write To Operator with Reply
XCTL	Transfer ConTrol
XFER	Transfer
XLATE	TransLATE
XMISSION	TransMISSION
XTNTS	eXTeNTS
XX	Online test parameter (Test number to be run)
YY	Online test parameter (Number of times to be run)

This Page Intentionally Left Blank

Access Method: any data management facility available to the user for transferring data between main storage and an input/output device.

Addressing: the means whereby the central processing unit (CPU) selects the unit to which it is going to send a message.

Audio Response Unit: a control unit, such as the IBM 7770 or IBM 7772, that is able to deliver an audio response to a digital inquiry.

Basic Access Method: any access method in which each input/output statement causes a corresponding machine input/output operation to occur.

Basic Telecommunications Access Method: a basic access method for communication with terminals.

Baud: a unit of signalling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one bit per second in a train of binary signals, or one 3-bit value per second in a train of signals each of which can assume one of eight different states.

Baudot code: a code for the transmission of data in which five equal-length bits represent one character.

Binary Synchronous Communications: a general purpose data link control procedure permitting a variety of types of devices, acting as station, to communicate with one another over a communication line using a standardized set of control characters for synchronous transmission of binary-coded data.

Block: a portion of a message terminated by an End-of-Block character or by an End-of-Text character.

BSC Intermix: the capability of different BSC devices to communicate over the same nonswitched multipoint communications line, or over the same problem program-defined, switched point-to-point communications line, to a control or central station.

Buffer (program input/output): a portion of main storage into which data is read or from which it is written, concurrent with asynchronous CPU activity, which is used to compensate for a difference in rate of flow of data, or time of data events, when transmitting data from one device to another.

Buffer Pool: main storage area per line group reserved for buffering.

Calling: a switch exchange service which enables a telephone user to select another subscriber, that is, dialing.

Central Station: the controlling station (usually a CPU) in a centralized, switched point-to-point data communications system. Message traffic is always between the central station and one of the remote stations.

Centralized Communication System: a system in which message traffic is always between the controlling station and one of the remote stations.

Chaining: a system of storing records in which each record belongs to a list or group of records and has a linking field for tracing the chain.

Command Control Block: refer to IBM System/360 DOS System Control and System Services Programs, GC24-5036.

Communication Line: the medium (wire, carrier channel, radio channel, etc.) over which signals may be sent.

Contention (start-stop): a condition on a multipoint communication channel when two or more locations try to transmit at the same time.

Contention (BSC): the condition on a point-to-point communication when the CPU and the remote device bid for control of the line simultaneously.

Control Character: a character whose occurrence in a particular context initiates, modifies, or stops control action: for example, a character that controls transmission of data over communication networks. The control character is recognized by the control units of the terminals as well as by the telecommunications control units.

Control Mode: the state that all terminals on a line must be in to allow line discipline, line control, or terminal selection to occur. When all terminals on a line are in control mode, characters on the line are viewed as control characters performing line discipline, that is, polling or addressing.

Control Station: the station (usually a CPU) in a multipoint nonswitched data communication system that controls message traffic by means of polling and selection.

Cursor: an automatically inserted, visual, display position marked that denotes the display position on the 2260 Display Unit screen that the next character entered will occupy and/or marks the end of a message.

Data Adapter Unit (2701): a control unit that houses transmission adapters. Each transmission adapter contains the circuitry and the logic for the control and logical connection of a terminal. One transmission adapter is required per line (see Telecommunication Control Unit).

Data Collection System: a system mainly used for gathering data.

Data Communication System: a system mainly used for the transmission of information.

Data Link: the communications lines, modems, and communication controls of all stations connected to the line, used in the transmission of information between two or more stations.

Data Set: a device which performs the modulation/demodulation and control functions necessary to provide compatibility between business machines and communications facilities.

Digitally Coded Voice (DCV): binary information necessary to the IBM 7772 Audio Response Unit to synthesize the sound, word, or sentence so coded, when received from the CPU through a WRITE command. The digital equivalent of all the sounds, words, or sentences used by a given application will usually be stored on a direct access device. An average of 300 bytes per second of audio output is necessary.

End of Block: a control character used to partition a message into blocks and initiate a control function.

End of Inquiry: any character recognized by the IBM 7770 or IBM 7772 as a signal that the inquiry is complete and the end of read command status should be generated and sent to the CPU. The character itself will be sent as the last character of the input message.

End of Transmission: a control character (or character string) that specifies the end of a message and causes a reset of all stations.

Event (in BTAM): the satisfying of a WAIT condition, an event may be one of the following:

1. A successful completion of a READ, WRITE, CONTROL, or RESETPL operation;
2. A PCI (7772 only);
3. An unsuccessful operation, as follows:
 - a. polling: when the end of an OPEN list is reached, polling stops;
 - b. addressing: when the addressed device is not ready, no message is sent;
 - c. multi-addressing: when one of the addressed devices is not ready, the message is not sent to the ready devices;
 - d. an error condition.

Inquiry: an input message calling for an answer.

Leased Line: a data path reserved for the exclusive use of one customer.

Line: see Communication Line.

Line Connection: a physical connection of terminals attached to a switched network that must be established before data transmission can take place.

Line Control Characters: a set of special characters which are not part of the text. A line control character is recognized by a telecommunications control unit, which performs the function associated with that character.

Local Device: a device directly attached to a channel by means of a control unit (see Remote Device).

Message: an arbitrary amount of information whose beginning and end are defined or implied.

Message Segment: a portion of a message that is contained in a single buffer.

Modulation: the periodic variation of amplitude, frequency, or phase of a carrier wave created by the data flow.

Multicomponent Addressing: the selection by the CPU of several receiving devices of the same terminal and the sending of a message that is recorded by all selected devices.

Multidrop Line: see Multipoint Line.

Multipoint Line: a line with one or more terminals attached to it (see Point-to-Point).

Multiterminal Addressing: The selection by the CPU of several receiving devices pertaining to several terminals on one line and the sending of a message which is recorded by all selected devices.

Network: a series of points interconnected by communication channels (see Switched Network).

Noncentralized Communication System: a system in which a terminal may send a message directly to another terminal without involving the control station in a message-switching operation.

Nonswitched Line: a multipoint or point-to-point configuration in which connection is not established through a switched network.

Point-to-Point: a line configuration in which the line permits exchange of information between two stations only. For example, on a switched network, once connection is established the configuration is point-to-point (see Multipoint Line).

Polling: the process of inviting stations within a data link to transmit messages. The stations are invited one at a time in an orderly fashion. The basic function of polling is to prevent contention by ensuring that only one station transmits at a time.

Polling Characters: characters used to establish contact with a terminal.

Private Line: a data path reserved exclusively for one user.

Record: a whole message or a portion of a message, received or sent by a single read or write command. A record will be terminated by an EOB character, an ETX character, or an EOT character.

Remote Device: a device that is attached to a telecommunication control unit by means of a communication network.

Remote Station: a station, other than the central station, on a centralized, point-to-point switched network which can communicate only with the central station. A remote station can be selected by the central station, or can call the central station if it has a message to send.

Start-Stop Transmission: a synchronous transmission in which each group of code elements corresponding to a character signal is preceded by a start signal which serves to prepare the receiving mechanism for the reception and registration of a character, and is followed by a stop signal which serves to bring the receiving

mechanism to rest in preparation for the reception of the next character.

Station: an aggregate of equipment and controls attached to any one of the several ends of a communication line.

Switched Lines: lines whose connection must be established prior to the start of data transmission.

Switched Network: a configuration in which the connection is established between the calling party and the called party prior to the start of data transmission and is broken at the end of the data transmission.

Telecommunications: pertaining to the transmission of signals over long distances, such as by telegraph, radio, or television.

Telecommunications Control Unit: a device used to transmit or adapt messages, coming from a remote device, for the CPU.

Teleprocessing: a term associated with IBM telecommunication systems expressing data transmission between a computer and remote devices.

Terminal: any device capable of sending and/or receiving information over a communication channel (see Station).

Terminal Component: an input or output device that is part of a terminal.

Text Mode: message transfer state. The message transfer state exists on a data link during the transfer of a message or messages from sender to receiver and the replies required to ensure their correct transfer (see Control Mode).

Transmission Control Unit (2702, 2703): a control unit that houses terminal controls. Each terminal control contains the circuitry and logic for the control and logical connection of all terminals of the same type within the system (see Data Adapter Unit).

Tributary Station: a station, other than a control station, on a centralized multipoint communication system which can communicate only with the control station, and only when polled or selected by the control station. On a noncentralized multipoint communication system, a tributary station that has been granted use of the line by the control station can select another tributary station as the receiver.

Unit: in teleprocessing, a physical I/O unit, characterized by a unique physical address. For example, a line is a unit; the IBM 2260 Local is a unit.

This Page Intentionally Left Blank

In this index, the word "macro" is equivalent to "macro instruction."

- abbreviations ABBR1
- Abend codes C18-C20
- access method, defined 1
- accumulators, error threshold 241
- acknowledgments (see responses)
- ACK-0, ACK-1 sequences 20
- Acronyms ABBR1
- ACSMETH operand (DATAMGT macro) D1
- action operand (CHGNTRY macro) 23
- ACTIVATE operand (CHGNTRY macro) 23
 - for IBM 3270 (local) 225
- activating teleprocessing system 34-37
 - for IBM 3270 (local) 225
 - for IBM 3270 (remote) 217
- activating terminal list entries 21
- AD operand (DFTRMLST macro) 131
- ADAPTER operand (IODEVICE macro) D1 D2
- ADDRESS operand
 - of IODEVICE macro D1 D2
- addressability, establishing for DCBs and DECBS 34
- addresses, display line (IBM 2260 and 2265) 64
- addressing
 - definition of 38
 - double 111
 - BSC stations required to use 111
 - for IBM 3270 (remote)
 - control unit addresses 210
 - device addresses 211
- addressing characters 5
- addressing lists
 - (see also under individual terminal on line types in Start-Stop and BSC Read and Write Operations chapters)
 - defining 23
 - DFTRMLST macro, defining with A1
 - formats of 23, A1-A8
- all-skip-bits-on return code (of READ and WRITE macros) 49
- alternate source for DCB, parameters 13
- AN operand (DFTRMLST macro) 130
- ANDGUID operand (PARMLIST macro) 174
- ANSRING operand (RESETPL macro) 51
- answerback
 - for IBM 2760 85
 - generally 6
- answering
 - definition of 39
 - for BSC stations 127-131
 - lists 21
- appearance, line 4
- area operand
 - DFTRMLST macro 22
 - TRANSLATE macro 32
- AREA operand (ONLTST macro) 293
- area station address (IBM 2715) 165
- AS macro (IBM 2715)
 - generally 165
 - ID operands table 166
- ASCTR macro (IBM 2715) 168
- ASGROUP operand (AS macro) 166
- ASLIST macro (IBM 2715) 178
- ASLOG operand (ASCTR macro) 169
- ASMRRTAB macro 31
- assembly error messages C1
- AT&T 83B3 100
- ATTENT operand (RESETPL macro for local IBM 3270) 227
- attention interruptions 226
- ATTLST operand (CHGNTRY macro for local IBM 3270) 226
- attributes of communications lines 1-4
- authsequence operand (DFTRMLST macro) 130-132
- Auto Poll feature
 - polling list 21
 - index byte, use of 50
 - reset function 50
 - restriction on use of WRITE TTV macro 45
 - use of CHGNTRY macro with 23
 - specifying in IODEVICE macro D3
 - start-stop index byte 107
 - start-stop, programming considerations 107-108
 - start-stop, terminal list for 107
- AUTOLST operand
 - CHGNTRY macro 23
 - DFTRMLST macro 21
- automatic answering
 - generally 4
 - specifying at system generation D3
- automatic dialing
 - generally 4
 - specifying at system generation D3
- automatic EOM mode (IBM 2760) 84
- automatic polling (see Auto Poll feature)
- AUTOWLST operand
 - CHGNTRY macro 23
 - DFTRMLST macro 21
- A1,A2 characters (IBM 2760) 81,87,90
- BFTEK operand (DCB macro) 14
- bidding for use of line 38
- blocks, message, use of 39
- BSC program, sample N1
- BSC1ST operand (DFTRMLST macro) 21
- BSC1,BSC2,BSC3 values in IODEVICE macro 109
- BUFCB operand (DCB macro) 14
- BUFFER operand (TPEDIT macro) L2
- bufferaddr operand (RELBUF macro) 30
- buffering, dynamic
 - effect of stopping CPU 28
 - in read operations 27
 - in write operations 28
 - lost messages 28
 - specifying 27
- buffering, programmer 27
- buffering, techniques of 6

buffers
 control block 14
 defining 25
 extra-buffer condition 28
 IBM 3270 (local) 225
 IBM 3270 (remote) 205
 length, specifying 14,25
 link field 29
 management of 29
 maximizing utilization of 7
 quantity, specifying 14
 releasing 30
 requesting 29
 restriction on use of CLOSE macro 36
 unused condition 29
buffer-unavailable condition
 generally 29
 return code for READ and WRITE
 macros 49
BUFL operand (DCB macro) 14
BUFNO operand (DCB macro) 14
BUILD macro 25,26
busy-line return code for READ and WRITE
 macros 49

calling
 definition of 38
 for BSC stations 127-132
 lists 21-22
central computer 1,8
central processing unit (CPU) 1
centrally controlled (multipoint)
 system 38
channel commands 8,39
channel program 8,39
 (see also sections on individual types
 of terminals (start-stop) or lines
 (BSC) in the Read and Write Operations
 chapters)
CHGNTY macro 23
 for IBM 3270 (local) 226-227
 for IBM 3270 (remote) 205
 restriction on use with certain types of
 terminal lists 23
cid operand (LERPRT macro) 243
CKAND operand (PARMLIST macro) 174
CKLENGTH operand (PARMLIST macro) 172
CKMOD10 operand (PARMLIST macro) 173
CKMOD11 operand (PARMLIST macro) 172
CKMONKY operand (PARMLIST macro) 172
CKNONUM operand (PARMLIST macro) 174
CKNUM operand (PARMLIST macro) 175
CKOR operand (PARMLIST macro) 173
CKRANGE operand (PARMLIST macro) 172
CLEAR operand (LERPRT macro) 243
CLOSE macro 36
cntrlseq operand (DFTRMLST macro) 22
code, transmission
 charts H1 I1 K1
 Dual Code feature 16
 specifying at system generation D3
 for BSC stations 109, E1
 for IBM 3270 (local) E1
 specifying for BSC 16
CODE operand (DCB macro) 16
code translation 31-33
 IBM 3270 (local) 225
 IBM 3270 (remote) 205-206
code translation tables 32-33
 assembling 31
 modifying 31
 names of 32
 World Trade telegraph terminals 32
coding format, macro instruction 8
commands, channel 8
 (see also sections on individual types
 of terminals (start-stop) or lines
 (BSC) in the Read and Write Operations
 chapters)
communications common carrier, definition
 of 1
communications line group
 characteristics of 12
 characters (see line control characters)
 closing 34
 definition of 11
 establishing 11-13
 examples of 12
 opening 34
 redefining using DD statements 13
 requirements 11
communications lines
 address of 5,11-13
 appearance of 4
 binary synchronous (BSC), defined 4
 closing (deactivating) 36-37
 configurations 2
 control over 4-7
 data representation on 4
 dial connections 4
 disabling 39
 enabling 39
 establishing communication on 5
 grouping local IBM 3270 devices 224-225
 identifying local IBM 3270 devices 224
 multipoint 4
 nonswitched 1
 opening (activating) 35-37
 point-to-point 4
 resetting 38,41,50
 specifying number of 11-13
 specifying type of 11
 start-stop, definition of 4
 switched 6
 types 1
 compatibility, BSC remote station 109
completion codes, ONLTST macro 294
component, definition of 1
CONFIGUR macro (IBM 2715) 163
configurations, line and station, supported
 by BTAM J1-J2
console messages, online test 298
contention system, definition of 5,38
continuation lines, BTAM macros 8
control byte, SWLST form of terminal list
 changing value of 23
 generally 110, 130 132
control characters, end-to-end
 definition of 8
 examples of 8,40
control characters, line (data link)
 coding in USASCII or Transcode 109
 examples of 40
 functions 38
 placing in output area 40

scanning input area for 40
 specifying 5
 use of 40
 where to find information on 41
 control codes, IBM 50 L2
 control mode 38
 control station 1,7,39
 control transactions (IBM 2715) 190
 control unit address 12
 "control unit not operational" message D10
 controlvalue operand (DFTRMLST macro) 130,132
 conversational operations
 defining DECB for 48
 generally 40
 conversion for IBM 3270 232
 CORE operand (CONFIGUR macro) 163
 count opearand
 REQBUF macro 29
 WAIT macro 52
 counter control byte 191
 counter readout (see IBM 2715 scheduled counter readout)
 counter testing (see IBM 2715 counter testing)
 CPU (central processing unit) 1
 CPU operand
 ASCTR macro 169
 TRLIST macro 177
 CTINIT operand (CTRGROUP macro) 170
 CTRADR operand (CTRLIST macro) 182
 CTRGROUP macro (IBM 2715) 169
 CTRLIST macro (IBM 2715) 182
 ctrno operand (CTRGROUP macro) 170
 CTROP operand (CTRLIST macro) 183
 CTRRD operand (CTRLIST macro) 183
 CTRSCHED macro (IBM 2715) 170
 cttest operand (CTRGROUP macro) 170
 CTEST operand (CTRLIST macro) 183
 cumulative counters (line error recording) 242

 data control block (DCB) (for line group)
 activating and deactivating 34-37
 contents of 20
 defining 11-14
 establishing addressability for 34
 format of 19
 reference to by DSECT 34
 data control block (DCB) (for local IBM 3270 device group) 224-225
 data, definition of 8
 data definition (DD) statements
 examples 12
 generally 11-13
 data event control block (DECB)
 changing contents of 42
 contents of B1-B10
 defining 41
 establishing addressability for 34
 for conversational operations 48
 for leading graphics operations 43
 format of B9
 parameters in 41
 purpose of 41
 reference to by DSECT 34

 data link control characters
 coding in USASCII or Transcode format 109
 examples of 40
 functions 38
 placing in output area 40
 scanning input area for 40
 specifying 5
 use of 40
 where to find information on 41
 data link control (see also communications lines, line control)
 data representation on communications line 4
 data tone
 defining 129
 recommended character sequence and length for 134
 specifying 22
 dataack operand (LERB macro) 242
 DATAMGT macro D1
 DCB (see data control block (DCB))
 DCB macro (for line groups) 14-18
 dcb operand
 CLOSE macro 37
 OPEN macro 37
 DCB operand (ONLTST macro) 293
 dcbaddr operand
 of CHGNTY macro for IBM 3270 (local) 226
 READ macro 43
 RELBUF macro 30
 REQBUF macro 29
 TRANSLATE macro 31
 WRITE macro 43
 DCBBUFL field 28
 DCBD macro 34
 DCBOFLGS field 34
 DD (data definition) statements 12
 DDNAME operand (DCB macro) 14
 deactivating teleprocessing system
 generally 34-37
 IBM 3270 (local) 225
 IBM 3270 (remote) 217
 DECAREA field 27
 DECB (see data event control block)
 DECB operand (ONLTST macro) 293
 decbaddr operand
 READ macro 42
 RESETPL macro 51
 RESETPL macro (local 3270) 227
 WRITE macro 42
 DECCMCO field 245,B-3
 DECCOUNT field 28,B3
 DECCSWST field 41,B7
 DECENTRY field 300
 DECERRST field 41,243,B-7
 DECFLAGS field 41,243,B4
 DECPOLPT field 50,B8
 DECRESPT field 41,B5
 DECSDECB field 243,B1
 DECSSENS0 field 41,243,B5
 DECTPCOD field 245,B6
 DECTYPE field 245,B4
 defining teleprocessing system 12
 defining terminal lists (see DFTRMLST macro)
 DEMOD10 operand (TRLIST macro) 177

DEMOD11 operand (TRLIST macro) 177
 DEU read counter capability (IBM 2715) 156
 DEU set counter capability (IBM 2715) 156
 DEUGROUP operand (AS macro) 166
 DEULIST macro (IBM 2715) 180
 DEVD operand
 DCB macro 15
 DCBD macro 34
 DEVCOD operand (CTRLIST macro) 182
 device operand (ASLIST macro) 178
 DFTRMLST macro, limit on number of entries 22
 DFTRMLST macro, operands for
 AT&T 83B3 100
 Auto Poll start-stop terminals 107
 IBM 1030 53
 IBM 1050 (nonswitched) 55
 IBM 1050 (switched) 57
 IBM 1060 60
 IBM 2260, 2265 62
 IBM 2740
 with Dial-up feature 71
 with Dial-up and Checking features 72
 with Dial-up and Checking features and IBM 2760 92
 with Dial-up and Transmit Control features 74
 with Dial-up, Transmit Control, and Checking features 76
 with Station Control feature 78
 with Station Control and Checking features 79
 IBM 3270 (remote) 202-205
 nonswitched multipoint stations 118
 switched point-to-point stations 127-132
 World Trade Telegraph terminals 105
 WU TWX 101
 WU 115A 103
 dialchars operand (DFTRMLST macro) 22,131
 dialcount operand (DFTRMLST macro) 22,131
 dialing
 definition of 4
 specifying digits 22
 DIALST operand
 CHGNTRY macro 23
 DFTRMLST macro 21
 digit checking (IBM 2715) 172
 DIGIT operand
 ASLIST macro 179
 DEULIST macro 180
 DIGIT2 operand (DEULIST macro) 181
 direction of transmission 8
 DISC operand (CHGNTRY macro) 23
 DISK operand
 ASCTR macro 169
 TRLIST macro 177
 DISPGUID macro (IBM 2715) 175
 display line addresses (IBM 2260, 2265) 64
 DISPMSG operand
 DISPGUID macro 176
 GDULIST macro 181
 distinction between teleprocessing and conventional data processing systems 4
 DLE ETB sequence 40
 DLE ETX sequence 40
 DLE STX sequence 40
 double addressing
 BSC station requirement for 111
 generally 111
 DSECTs, usefulness of in user program 34
 DSORG operand (DCB macro) 14
 Dual Code feature (IBM 2701)
 generally D5
 specifying 16, D5
 Dual Communications Interface feature (IBM 2701) 16, D5
 dummy transaction group (IBM 2715) 168
 duplex transmission 2
 dynamic buffering
 effect of stopping CPU 28
 in Read operations 27
 in Write operations 28
 lost messages 28
 specifying 27
 EBCDIC (extended binary-coded-decimal interchange code)
 generally 16
 transmission code, BSC stations using 109
 transparency feature (IBM 2770) 194
 ECB (event control block) 43,52
 ECB operand (WAIT macro) 52
 ECBLIST operand
 TWAIT macro 52
 WAIT macro 52
 Edit routine (IBM 50) L1-L8
 EDIT operand (TPEDIT macro) (IBM 50) L2
 EIB (error information byte) mode 110
 Enable command, effect of RESETPL macro on 51
 enabling of switched line, defined 42
 ENDERR operand (CONFIGUR macro) 164
 end-of-address (EOA) character 38
 end-of-block (EOB) character 39
 end-of-transmission (EOT) character 38
 end-of-transmission-block (ETB) character 39
 end-to-end control characters
 definition of 8
 examples of 8
 entry operand
 READ macro 48
 WRITE macro 48
 ENTRY operand
 ASLIST macro 179
 GDULIST macro 182
 ONLTST macro 279
 entrylength operand (DFTRMLST macro) 130,132,A5
 EOA (end-of-address) character 38
 EOB (end-of-block) character 39
 EOM operand (DCB macro) 18
 EOT (end-of-transmission) character 38
 EOT operand (DCB macro) 18
 EROPT operand (DCB macro) 14
 ERP (see error recovery procedures (ERP))
 ERROPT operand (TPEDIT macro) L3
 error codes (IBM 2715) 188
 error conditions
 IBM 2760 85
 IBM 3270 (local) 213
 IBM 3270 (remote) 213

indicators in DECB B1
 irrecoverable errors 240,241
 non-text errors 241
 number of retries 240
 permanent errors 240,241
 Read text errors 241
 sense byte analysis
 BSC ERP 256
 start-stop ERP 246
 user program 244
 should-not-occur errors 246
 specifying ERP options 241
 status analysis
 BSC ERP 256
 start-stop ERP 246
 user program 244
 suggested retry options for Read and
 Write operations 277-285
 temporary errors 240
 undefined errors 241
 use of Outboard Recorder (OBR) 240
 use of Statistical Data Recorder
 (SDR) 240
 user program analysis procedure 243-245
 write text errors 241
 error description word (IBM 50) L4-L5
 error information byte (EIB) mode 110
 error information placed in DECB 241
 error messages, BTAM C1-C20
 assembly errors C1-C7
 error status messages
 IBM 2715 C13-C14
 IBM 2770 C12-C13
 IBM 3270 (remote) C15-C18
 I/O errors C7-C10
 remote BSC station C11-C17
 TCU inoperative C10
 error recording 241-243
 error recovery procedures (ERP) 240-276
 appropriate action when ERP fails to
 recover 240-276
 basic functions 242,245,256,257
 buffer-unavailable condition 240
 completion codes 244
 detection and analysis 243-245
 ERP actions
 BSC 256-268
 IBM 3270 (local) 256,275-276
 start-stop 246-256
 error status flags (DECERRST
 field) 243, B7
 error status messages
 IBM 2715 C13-C14
 IBM 2770 C12-C13
 IBM 3270 (remote) C15-C18
 error threshold counters
 default count 242
 printing contents of 242-243
 errors, transmission 5
 irrecoverable 240
 permanent 240
 read text 241
 temporary 240
 undefined 246
 write text 241
 (see also error recovery procedures
 (ERP)) 240-276
 errors occurring during transmission of
 test messages 241
 ETB (end-of-transmission-block) 39
 event control block (ECB) 43,52
 exceptional conditions 43,222
 IBM 3270 (remote) 211
 indicated in DECFLAGS 243
 execute format
 CLOSE macro 37
 OPEN macro 37
 READ macro 49
 WRITE macro 49
 exit list (DCB macro) 14
 EXLST operand (DCB macro) 14
 expanded ID verification 130-131
 explicit counter addressing (IBM 2715) 156
 EXTALRM operand (ASCTR macro) 169
 external alarm 192-193

 faaseq operand (DFTRMLST macro) 22
 F-character (IBM 2760) 81
 FEATURE operand (IODEVICE macro)
 generally D2
 IBM 3270 (local) D3-D4
 filmstrip positioning (IBM 2760) 81
 formats, message (see message formats)
 frame change sequence (IBM 2760) 82
 FUNCERR operand (CONFIGUR macro) 163

 GDU keyboard character conversion 165
 GDU operand TRLIST macro 177
 CDUAS macro (IBM 2715) 167
 GDUNUMB operand (GDUAS macro) 167
 GDULIST macro (IBM 2715) 181
 GDUTRANS macro (IBM 2715) 170
 GETID operand (CONFIGUR macro) 165
 GETMAIN macro 26
 GETPOOL macro 25
 guidance lights (IBM 2715)
 ASLIST macro operand values for 179
 generally 164

 half-duplex transmission 2
 Halt I/O function 51
 HIGHCTR operand (ASCTR macro) 169
 HIGUID operand (PARMLIST macro) 173

 IAM operand (DCB macro) 18
 IBM System/7 145
 IBM System/370 and System/3
 double addressing 111
 models supported as remote stations
 under BTAM J1-J2
 programming considerations 137-139
 transmission over nonswitched multipoint
 line (Model 20 only) 143-145
 transmission over nonswitched 143-144
 point-to-point line 143
 transmission over switched
 point-to-point line 144
 (see also BSC Read and Write Operations
 chapter)
 IBM 50 L1-L8
 control codes L2
 data-editing L1
 programming considerations L8

IBM 1030 53-54
 IBM 1035 (IBM 2715) 184
 IBM 1050 (nonswitched) 55-56
 IBM 1050 (switched) 57-59
 IBM 1053 (IBM 2715) 186
 IBM 1060 60-61
 IBM 1130
 double addressing recommended 144
 transmission over nonswitched multipoint
 line 143-144
 transmission over nonswitched
 point-to-point line 143
 (see also BSC Read and Write Operations
 chapter)
 IBM 1800
 double addressing required 146
 transmission over nonswitched multipoint
 line 143-144
 transmission over nonswitched
 point-to-point line 143
 transmission over switched
 point-to-point line 144
 (see also BSC Read and Write Operations
 chapter)
 IBM 2213 (IBM 2770) 196-197
 IBM 2260-2848
 display line addresses 64
 generally 62-65
 IBM 2265-2845
 display line addresses 64
 generally 62-65
 IBM 2265 Model 2 (IBM 2770) 197
 IBM 2715
 control transactions 190
 counter addressing 157
 counter testing 154
 data entry 183-184
 error status messages C-13, C-14
 external alarm 192-193
 macro storage site estimates 164
 message formats 185-189
 message header
 System/370 to 2715 185
 2715 to System/370 185
 programming notes 183-185
 pulse count
 counter testing 155
 generally 154
 scheduled counter readout 156
 set counter 156
 transaction header 186
 transmission code 154
 user tables 157-161
 loading 161
 sample assembly 381
 size 157-161
 table-generating macros 165-183
 AS macro 165
 ASCTR macro 168
 ASLIST macro 178
 CONFIGUR macro 163
 CTRGROUP macro 169
 CTRLIST macro 182
 CTRSCHED macro 170
 DEULIST macro 180
 DISPGUID macro 175
 GDUAS macro 167
 GDULIST macro 179
 GDUTRANS macro 170
 PARAMNUM macro 171
 PARMLIST macro 171
 STEND macro 183
 TGROUPO macro 168
 TRANSLAT macro 176
 TRLIST macro 176
 IBM 2740 (General)
 buffer overflow 66
 Buffered Receive feature 66
 buffering restrictions 66
 features supported by BTAM 66
 IBM 2715, use with 193
 Model 2 66
 multiple-block messages 66
 responses 66
 TCU timeout 66
 IBM 2740
 Basic 68
 Checking 69-70
 Checking and IBM 2760 89-91
 Dial-up 71
 Dial-up and Checking 72-73
 Dial-up, Checking and IBM 2760 92-95
 Dial-up and Transmit Control 74-75
 Dial-up, Transmit Control, and
 Checking 76-77
 Model 2 66
 Station Control 78
 Station Control and Checking 79-80
 IBM 2741
 channel commands 96
 designing a message control routine 98
 general information 96
 interrupt facility 99
 program considerations 98
 IBM 2760
 automatic EOM mode 81
 A1, A2 characters 81
 error detection and recovery 85
 filmstrip positioning 81
 function (F) character codes 81
 image index counter 82
 initializing images 87
 manual EOM mode 82
 manual frame advance 82
 message formats 81-84
 modes of operation 81
 multiple message blocks 87
 online testing 85
 programming considerations 86
 signals to operator 84
 startup procedure 87
 utility response points 82
 V and H coordinates 82-83
 IBM 2770
 configuration 194
 device control 196
 EBCDIC Transparency feature 194
 error status messages 318-319
 field-control operation 197-198
 message formats 198
 nontransparent transmission 198
 polling and selection 194
 protected data 197
 record formats 198
 terminal function control 196
 transmission codes 194

- transmission delays 195
- vertical forms control 196
- IBM 2772 (see IBM 2770) 194
- IBM 2780
 - addressing sequences
 - examples 148
 - generally 147
 - component selection 153-155
 - examples 148
 - generally 148
 - end-to-end control characters 148
 - escape sequences 150
 - format records 151
 - horizontal format control 150
 - internal errors 153
 - Mode switch 149
 - MODE operand of DCB macro 147
 - multiple record transmission 151
 - multipoint polling/addressing 148
 - polling sequences
 - examples 148
 - generally 147
 - programming considerations 153
 - records, number and length of 152
 - responses to abnormal conditions 153
 - transmission length, maximum 153
 - transmission over nonswitched multipoint line 147
 - transmission over nonswitched point-to-point line 147 149
 - transmission over switched point-to-point line 147 150
 - use of NAK, EOT, and STX ENQ characters 153
 - vertical forms control 150
 - (see also BSC Read and Write Operations chapter)
- IBM 2790 (see IBM 2715)
- IBM 2972 200
- IBM 2980, character set and transmission code 367
- IBM 3270 Display System
 - capabilities supported 201
 - conversion to 232
 - devices supported 201
 - local 3270 display system 223-232
 - ACTIVATE operand 226
 - activating the system 225
 - ATTENT operand 227
 - attention interruptions 226
 - ATTLST operand 226
 - buffer management 225
 - character generator options 329
 - CHGNTRY macro 226-227
 - code translation 225
 - dcbaddr operand 226
 - deactivating the system 225
 - decbaddr operand 227
 - device addresses D-1
 - device grouping 224-225
 - device identifying 224
 - entry operand 228
 - erase and write buffer 229
 - erase unprotected fields 229
 - error recovery procedures 230
 - FEATURE operand D-2-D4
 - formats, sample 228-229
 - inoutarea operand 228
 - inoutlength operand 229
 - interruptions, attention 226
 - line group and relative line numbers 224-225
 - listposition operand 226
 - MODEL operand D-1, D-2
 - online testing 230
 - options D-3, D-4
 - read buffer 229
 - read buffer from position 229
 - read initial operations 226
 - read modified fields 228
 - read modified fields for operator input 226-227
 - read modified fields from position 228-229
 - read operations 227-229
 - READ TB 229
 - READ TBP 229
 - READ TI 228
 - READ TM 228
 - READ TMP 228
 - RESETPL macro 227
 - retry options 230
 - return codes for RESETPL 227
 - rln operand 228
 - sample program D-1
 - SKIP operand 226
 - write buffer 229
 - write operations 229
 - WRITE TI 229
 - WRITE TS 229
 - WRITE TSA 229
 - WRITE TSF 230
 - WRITE TUS 230
 - macro instruction sequences (switched 3275) 218
 - online testing 217
 - remote 3270 display system 201
 - activating the system 206
 - buffer management 205
 - CHGNTRY macro 205
 - code translation 205-208
 - connecting to 3275 with dial 217
 - control unit addresses for
 - polling 202
 - selection 202
 - copy 215
 - deactivating the system 207
 - device addresses 205
 - DFTRMLST
 - nonswitched 202
 - switched 214
 - Dial feature 214-223
 - entry operand
 - nonswitched 206
 - switched 219
 - erase and write buffer
 - nonswitched 212
 - switched 222
 - erase unprotected fields
 - nonswitched 212
 - switched 222
 - error conditions
 - nonswitched 212
 - switched 223

error recovery procedures
 nonswitched 213
 switched 223
error status message
 nonswitched C7-C8
 switched C11-C13
exceptional conditions
 nonswitched 213
 switched 223
ID verification 214
inoutarea operand
 nonswitched 209
 switched 219
input (see read operations)
I/O interface code 206
line control
 nonswitched 206
 switched 217
line groups
 nonswitched 201
 switched 214
message transmission
 nonswitched 206-213
 switched 217-223
online testing 214
operator considerations 222
outarea operand
 nonswitched 210
 switched 220
output (see write operations)
read buffer
 nonswitched 211
 switched 221
read buffer from position
 nonswitched 211
 switched 221
read modified fields
 nonswitched 210, 211
 switched 220
read modified fields for operator
 input
 nonswitched 206-208
 switched 219-220
 read modified fields from position
 nonswitched 211
 switched 220
 read operations
 nonswitched 206-212
 switched 218-222
READ TI
 nonswitched 206
 switched 219
READ TT
 nonswitched 206
 switched 219
retry options
 nonswitched 213
 switched 223
sample program P1
 switched 214
terminal lists
 nonswitched 202
 switched 214
write buffer
 nonswitched 212
 switched 222
write operations
 nonswitched 212-214
 switched 218,222

WRITE TC 222
WRITE TI
 nonswitched 212
 switched 222
WRITE TIR 213
WRITE TIV
 nonswitched 210
 switched 221
WRITE TR
 nonswitched 213
 switched 222
WRITE TT
 nonswitched 212
 switched 222
WRITE TTR 212
WRITE TTV
 nonswitched 211
 switched 221
storage estimates
 system generation D1-D4
 test message 297-298
IBM 3650 233
IBM 3660 234
IBM 3735 235
IBM 3741 Model 2 236
IBM 3747 236
IBM 3750 237
IBM 3780 238
IBM 3977 Model 2 modem 36
IBM 5275 239
ID ACK-0 sequence 23
ID ENQ sequence 23
ID operand
 AS macro 165
 ASCTR macro 169
 CTRGROUP macro 170
 GDUAS macro 167
ID verification
 expanded 110
 regular 109-110
 specifying 127-132
 switched 3275 214
idcount operand (DFTRMLST macro) 130, 131
IDCOUNT operand (CONFIGUR macro) 165
IDENT operand
 GDULIST macro 181
 PARMLIST macro 175
IDLST operand (DFTRMLST macro) 21
IECTDECB macro 34
image index counter (IBM 2760) 82
implicit counter addressing (IBM 2715) 156-157
inarea operand
 READ macro 43
 WRITE macro 43
index byte, Auto Poll 50
initialization, user program 34-35
initializing images (IBM 2760) 87
inlength operand
 READ macro 48
 WRITE macro 48
inoutarea operand
 READ macro 43
 IBM 3270 (local) 228-229
 IBM 3270 (remote) 206-209 219-221
 WRITE macro 43
 IBM 3270 (local) 229
 IBM 3270 (remote) 212-213 222

inoutlength operand
 READ macro 43
 WRITE macro 43
 IBM 3270 (local) 229
 input, definition of 8
 INQDISP operand
 ASLIST macro 179
 CONFIGUR macro 165
 TRLIST macro 177
 intermediate block checking function 110
 interruptions
 attention 226
 overflow 191
 power 192
 intreq operand (LERB macro) 242
 invalid optype return code for READ and
 WRITE macros 49
 invalid rln return code for READ and WRITE
 macros 49
 IUS (interchange unit separator)
 character 110
 I1,I2 characters (IBM 2760) 82

keyboard options, IBM 3270 D3-D4
 keyword operands, rules for coding 9

leading graphics
 defining DECB for 43
 ignored by IBM 1800,2715,2770 42
 use of READ TPL macro with EIB mode 110
 length checking (IBM 2715) 180
 length operand
 DFTRMLST macro 22
 TRANSLATE macro 32
 LENGTH operand
 ASLIST macro 178
 DEULIST macro 180
 ONLTST macro 293
 LERB macro 241-242
 LERB operand (DCB macro) 14
 LERPRT macro 242-243
 limit on number of entries in terminal
 list 21
 line (see communications line)
 line address 12
 line and station configurations supported
 in BTAM J1-J2
 line configuration, specifying BSC 11, D1
 line control 4,38
 line control characters
 BSC read and write 112
 coding in USASCII or Transcode
 format 109
 examples of 40
 functions 38
 placing in output area 40
 scanning input area for 40
 specifying 5
 use of 40,112
 where to find information on 41
 line-error-during-Open return code for READ
 and WRITE macros 49
 line error recording block 241,242
 line group (see communications lines) 241-242
 line number, relative (rln) 13

line termination, switched 6
 link field, buffer 29
 list format
 CLOSE macro 36
 OPEN macro 36
 READ macro 43
 WRITE macro 43
 list type, of DFTRMLST macro 21
 listaddr operand (CHGNTRY macro) 23
 listposition operand (CHGNTRY macro) 23
 for IBM 3270 (local) 214
 listtype operand, of CHGNTRY macro 23
 loading user tables (IBM 2715) 161
 local terminal, definition of 8
 LOG operand
 ASCTR macro 169
 TRLIST 177
 LOPEN macro
 generally 36
 restriction on use 36
 LOWGUIDE operand (PARMLIST macro) 173

machine requirements for BTAM J2
 MACRF operand (DCB macro) 14
 macro instructions
 coding format 9-10, Index 16
 index to main discussions Index 15
 manual EOM mode (IBM 2760) 84
 manual fram advance (IBM 2760) 82
 MD operand (DFTRMLST macro) 131
 message
 definition of 8
 determining source of 50
 message blocks, use of 39
 message control routine example (BSC) 113
 message formats
 BSC 111
 console (online tests) 298
 IBM 2715 185
 IBM 2760 81-84
 IBM 2770 198-199
 IBM 3270 (local) 227-229
 IBM 3270 (remote) 206- 211 219-222
 online test 280-285
 message transmission 294-299
 MF operand
 CLOSE macro 37
 OPEN macro 37
 READ macro 49
 WRITE macro 49
 MINLN operand (TPEDIT macro) L1
 MODE operand (DCB macro) 15
 MODEL operand
 (IODEVICE macro for local IBM 3270) D1
 MODULUS operand
 ASLIST macro 179
 DEULIST macro 180
 MON operand (DCB macro) 15
 MONDLY operand (DCB macro) 15
 MONERR operand (CONFIGUR macro) 164
 monitor key (IBM 2715) 184
 MSG operand
 ASLIST macro 180
 CTRLIST macro 183
 DEULIST macro 191
 GDULIST macro 192

multiple console support facility, use by
 ERP 241, 242-243

multiple Waits 51

NEXTAS operand (ASCTR macro) 169

nlines operand (LERB macro) 242

non-EIB mode 110

nonswitched line, definition of 2

nontransparent mode 40

nontto operand (LERB macro) 242

NORGUID operand (GDULIST macro) 181

NORM operand (ASLIST macro) 178

NULL operand (TRLIST macro) 177

numchars operand (CHGNTRY macro) 23

numcnsent operand (DFTRMLST macro) 22

numrec operand (DFTRMLST macro) 22

numsent operand (DFTRMLST macro) 22

OBRCNT operand (IODEVICE macro) D4

OLTEP for IBM 3270 230

online testing, 3270 binary synchronous
 BTAM responses to RFT messages from
 remote stations 299
 console messages 298
 how requested 290, 299
 how specified 290, 299
 IBM 3270 (local) 298-299, 301
 IBM 3270 (remote) 301
 minimum length requirement for Read
 Initial macro 47
 ONLTST macro 292-294
 requirements for entering test
 mode 290, 299
 RFT (request-for-test) message
 formats 294, 299
 summary of options 300-301
 test type codes 295-298
 use of IBM 2770 Job switch 291
 use of IBM 2780 Mode switch 291
 use of WACK sequence 292

online testing, start-stop
 how requested 286
 how specified 286
 restrictions 289-290
 test request message formats 286-287
 test type codes 287-289

ONLTST macro 292-294

Open flag of DCB 20

OPEN macro 34-36

OPENLST operand
 CHGNTRY macro 23
 DFTRMLST macro 21

open-type polling list (see polling, lists)

operands, rules for coding 9

operator's console, defined 2

optical image unit (see IBM 2760)

optype operand
 READ macro 42
 WRITE macro 42

ORGUID operand (PARMLIST macro) 174

outarea operand
 READ macro 43
 WRITE macro 43

Outboard Recorder (OBR) 240-241

outlength operand

READ macro 48

WRITE macro 48

output, definition of 8

overflow interruption (IBM 2715) 191

PARAMNO operand (GDULIST macro) 181

PARAMNUM macro (IBM 2715) 171

PARMLIST macro (IBM 2715) 171

PARMLST operand (PARAMNUM macro) 171

PC operand (CONFIGUR macro) 163

PLN operand (PARAMNUM macro) 171

point-to-point, definition of 2

polling
 control unit addresses 12
 control unit addresses, 3270 202
 definition of 5, 38
 device addresses, 3270 203
 double addressing 110-111
 for IBM 3270 (remote) 202
 termination of (see RESETPL macro)

polling characters 5-6

polling lists
 Auto Poll 21
 defining 21, 118
 formats of 21-22, A1-A8
 limit on number of entries 22
 open-type, format A1, A4
 restart address maintained in DECB 50
 wraparound type, format A4, A5
 (see also DFTRMLST macro)

POLLING operand (RESETPL macro) 51

positional operands, rules for coding 9

POST operand (CHGNTRY macro) 23

Prepare to Read 140, 201, 202, 230

problem program initialization 34

program-controlled interrupt (PCI) used in
 dynamic buffering 28

programmer buffering (see buffering,
 programmer)

pulse count feature (see IBM 2715)

READ Connect (TC) macro 128, 133

READ Connect with Tone (TCW) macro 134

READ macro
 operation type options 44-47
 summary of options for IBM 3270
 (local) 47
 summary of options for switched BSC
 lines 46-47, 128

| Read, Prepare to 140, 201, 202, 230

Read operations
 analyzing results of 40
 canceling 50
 extra buffer condition 27-28
 IBM 3270 (local) 226-228
 IBM 3270 (nonswitched) 206-212
 IBM 3270 (switched) 219-222
 read skip operations 52
 READ Skip macro 52
 (see also READ macro)

Read text error 245, 14

READ Initial (TI) macro
 IBM 3270 (local) 227
 IBM 3270 (nonswitched) 206
 IBM 3270 (switched) 219
 start-stop Auto Poll operations 107
 length restriction when BSC online test
 used 48

RECFM operand (TPEDIT macro) L3
regular ID verification 110
relative line number (rln) 13
RELBUF macro
 generally 30
 using to release extra buffer 30

This Page Intentionally Left Blank

remote computers, programming considerations for 143
remote station
 compatibility (BSC) 109
 definition of 8
 restrictions for connecting to lines in same line group 11
 specifying telephone number of 22
remote terminal, definition of 8
REPLACE operand (TPEDIT macro) L1
REQBUF macro 29
reset operations 41,50
RESETPL macro
 for IBM 3270 (local) 227
 used to change polling list while polling is in progress 50
resetting the line 38,50
response points (IBM 2760) 82
responses
 addressing 38
 ENQ (BSC) 112
 EOT (BSC) 113
 error-checking 39
 generally 6
 message text (BSC) 112
 polling 38
 selection 38
retry options
 IBM 3270 (local) 230 285
 IBM 3270 (nonswitched) 283-284
 IBM 3270 (switched) 280-282
return codes
 CHGNTRY macro 23
 local 3270 227
 LOPEN macro 36
 ONLTST macro 294
 READ macro 49
 RELBUF macro 30
 REQBUF macro 29
 RESETPL macro 51
 local 3270 227
 restrictions on use of WAIT or TWAIT macro after nonzero return code 50
 special IOB 269-273
 TPEDIT macro L4-L5
 WRITE macro 49
returnreq operand
 REQBUF macro 29
 TWAIT macro 51
RFT (request for test) messages 294-295
ridseq operand (DFTRMLST macro) 22
rln (relative line number) 13
rln operand
 LERPRT macro 243
 READ macro 48
 IBM 3270 (local) 228
 WRITE macro 48
RLN operand (ONLTST macro) 293
RNGETST operand (PARMLIST macro) 173
ROUTE operand
 ASCTR macro 169
 TRLIST macro 177
rules for coding BTAM macros G1
RVI (reverse interrupt) sequence 20

sample program for IBM 3270
 local O1-O32
 remote P1-P21
sched operand (CTRSCHED macro) 170

selection, station 38
SELTRAN operand (ASLIST macro) 179
sense/status analysis
 BTAM 245 258
 user program 244
set counter (see IBM 2715)
SETADDR operand (IODEVICE macro) D4
skip bit in terminal list 23
SKIP operand (CHGNTRY macro) 23
 IBM 3270 (local) 226
source of message, determining 50
sro operand (CTRGROUP macro) 170
SROENAB operand (CTRGROUP macro) 170
SSALST operand
 CHGNTRY macro 23
 DFTRMLST macro 21
SSAWLST operand
 CHGNTRY macro 23
 DFTRMLST macro 21
standard format
 CLOSE macro 34
 OPEN macro 34
start-stop Auco Poll operations 107
start-of-text (STX) character 38
start-stop line, definition of 4
station
 definition of 8
 restrictions for connecting to lines in same line group 11
 specifying type of 12
Statistical Data Recorder (SDR) 240
STEND macro (IBM 2715) 183
Storage Estimates (VS2) Q2
STORID operand (CONFIGUR macro) 165
STX (start-of-text) character 38
SUPPRES operand (DISPGUID macro) 176
switched line
 definition of 2
 disabling 39
 enabling 39
 termination 6
 use of ID verification 22,110
Switched Network Backup Q1
switched point-to-point operation (BSC) 127
SWLST form of terminal list 22
SWLST operand
 CHGNTRY macro 23
 DFTRMLST macro 130, 131
system generation
 generally 12, D1
 IBM 3270 230, D1

table definition block (IBM 2715) 158
table-generating macros (IBM 2715) (see IBM 2715, user tables)
tablename operand
 ASMTRTAB macro 31
 TRNSLATE macro 32
TCn operand (TGROUP macro) 170
TCU (see transmission control unit)
TCU operand (IODEVICE macro) D1
telecommunications subroutine library C5
teleprocessing (TP) system, definition of 1
terminal, definition of 1
terminal lists
 addressing 22

answering 127
 calling 127
 changing entries in (see CHGNTRY macro)
 defining 21,127
 for expanded ID verification 127
 formats A1-A8
 modifying 22
 polling (see polling, polling lists)
 restrictions on changing 23
 specifying entries in READ and WRITE
 macros 48
 userdata field in SWLST form 129
 terminal tests (see online testing)
 terminology used in this publication 8
 test request message formats 294-295,299
 test, definition of 8
 TEST operand
 ONLTST macro 293
 TRLIST macro 177
 TGROUP macro (IBM 2715) 168
 tidseq operand (DFTRMLST macro) 22
 timer exit, restriction on use of LOPEN
 macro within 36
 TPEDIT macro (IBM 50) L1-L8
 TP-Op codes B5-B7
 transaction (IBM 2715)
 control 198-200
 formats 198-199
 header 186
 identifier 177
 list, defining (see TRLIST macro)
 transaction code
 defined 154
 in TGROUP macro 162
 TRANSCH operand (TRANSLAT macro) 176
 transcode
 appearance in main storage 109
 used by IBM 2780 109
 TRANSL operand (PARMLIST macro) 175
 TRANSLAT macro (IBM 2715) 176
 translation tables, code (see code
 translation tables)
 transmct operand (LERB macro) 242
 transmission, direction of 8
 transmission code
 BSC stations 109
 generally 4, G13 I1
 transmission control unit (TCU)
 access lines 1,4
 conditioning line adapters of 35
 generally 1
 malfunctioning line adapters 36
 specifying type of adapter D2
 specifying type of TCU 12, D1
 transmission control unit inoperative
 message C10-C11
 transparent mode
 generally 40
 placing DLE STX sequence in output
 area 111
 TRANTXT operand (TRANSLAT macro) 176
 TRCODE operand (GDUTRANS macro) 171
 TRID operand (TRLIST macro) 177
 TRLIST macro (IBM 2715) 176
 TRLIST operand (GDUTRANS macro) 171
 TRNSLATE macro 31
 TRSLRCTW macro (World Trade telegraph) F1
 TRSLRCT3 macro (World Trade telegraph) F1

TRSLSCTW macro (World Trade
 telegraph) F1-F2
 TRSLSCT3 macro (World Trade
 telegraph) F1-F2
 TWAIT macro 52-53
 TWX,WU 101-102

 unavailable buffer condition
 generally 240
 return code for READ and WRITE
 macros 243
 UNIT operand
 IOCONTRL macro E1
 IODEVICE macro D1
 UNITNAME macro 12
 UNIT parameter (DD statement) 12
 UNITNAME macro 12
 unused buffer condition 28
 US (unit separator) character 110
 usage count too large return code for READ
 and WRITE macros 49
 USASCII code
 appearance in main storage 109
 BSC stations using 109
 user area, SWLST form of terminal list 22
 user program analysis (BSC) 243
 user program initialization 34-35
 user tables (see IBM 2715, user tables)
 userdata operand (DFTRMLST macro) 131,132
 userlength operand (DFTRMLST
 macro) 130,131
 utility response points (IBM 2760) 82

 V and H coordinates (IBM 2760) 82,83
 VERCHK operand (TPEDIT macro) L3

 WACK sequence 20
 WAIT macro
 generally 51
 restriction on use after nonzero return
 code 51
 World Trade telegraph terminals
 generally 104
 translation tables 104
 wraparound-type polling list (see polling,
 list)
 WRAPLST operand
 CHGNTRY macro 23
 DFTRMLST macro 21
 WRITE Connect (TC) macro 128
 WRITE macro
 operation type options 43-46
 summary of options for IBM 3270
 (local) 48
 summary of options for switched BSC
 lines 46-47,128
 write operations 46-47,128
 analyzing result of 40
 IBM 3270 (local) 229
 IBM 3270 (nonswitched) 212-213
 WTLIST operand (DFTRMLST macro) 22
 WTTALST operand (DFTRMLST macro) 22
 WU 115A 103

X operand (ONLTST macro) 293
xx operand (DFTRMLST macro) 22

Y operand (ONLTST macro) 293
yy operand (DFTRMLST macro) 22

INDEX OF BTAM AND SYSTEM MACRO INSTRUCTIONS

The index lists only the main discussions of the macro instructions.

AS	163	LERB	242
ASCTR	166	LERPRT	243
ASLIST	176	LOPEN	35
ASMTRTAB	30	ONLTST	285
BUILD	24	OPEN	34
CHGNTRY	21	PARAMNUM	169
CHGNTRY for Local 3270	224	PARMLIST	169
CLOSE	35	READ	41
CONFIGUR	161	RELBUF	29
CTRGROUP	167	REQBUF	28
CTRLIST	182	RESETPL	49
CTRSCHED	168	RESETPL for Local 3270	225
DATAMGT	C13	STEND	181
DCB	13	TGROUP	165
DCBD	33	TPEDIT	L1
DEULIST	180	TRANSLAT	174
DFTRMLST	20	TRLIST	174
DISPGUID	173	TRNSLATE	30
GDUAS	165	TRSLRCTW	F1
GDULIST	179	TRSLRCT3	F1
GDUTRANS	168	TRSLSCTW	F1
GETPOOL	24	TRSLSCT3	F2
IECTDECB	33	TWAIT	50
IODEVICE	12	UNITNAME	12
		WAIT	50
		WRITE	41

Order No. GC27-6980-4

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.

How did you use this publication?

- | | |
|--------------------------------------------------------------|-------------------------------------------------|
| <input type="checkbox"/> As an introduction | <input type="checkbox"/> As a text (student) |
| <input type="checkbox"/> As a reference manual | <input type="checkbox"/> As a text (instructor) |
| <input type="checkbox"/> For another purpose (explain) _____ | |

Please comment on the general usefulness of the book; suggest additions, deletions, and clarifications; list specific errors and omissions (give page numbers):

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Please include your name and address in the space below if you wish a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

ADDITIONAL COMMENTS:

Cut or Fold Along Line

Please Do Not Staple

Fold and Tape

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E02
P.O. Box 12195
Research Triangle Park
North Carolina 27709



Fold and Tape

Fold and Tape



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

GC27-6980-4

OS/VIS BTAM (File No. S370-30) Printed in U.S.A. GC27-6980-4

