BTAM

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

This publication is organized in the following manner. There are six main sections and a number of appendixes. The first four main sections contain general information concerning DOS BTAM and should be of interest to all readers. The remaining two main sections contain more detailed descriptions and information concerning channel programs for specific devices. The reader need only read the sections on the device or devices with which he is concerned. The appendixes contain formats, code charts, macro operands, and sample programs. They are to be used as a reference to supplement the main sections of the publication.

- The Introduction explains concepts of teleprocessing in general, and of BTAM in particular. Many terms which are used throughout the publication are defined in this section.

- The Summary of BTAM Components and Macro Instructions section gives an overview of the routines and macro instructions needed for a teleprocessing system.

- The BTAM Macro Instruction Format Descriptions section gives detailed information on operands for all BTAM macro instructions. A coding format illustration is provided for each macro described.

- The Programming Information section gives information on coding of the BTAM problem program. This section includes: assembly considerations; an explanation of how to define line groups, terminal lists, and DECBs; a discussion of error recovery procedures; on-line terminal test information; and an explanation of how to use intermixed Binary Synchronous Communications (BSC) terminals.

- The Start-Stop Device Dependent Considerations section lists all start-stop devices supported by BTAM and gives greater detail of the use of the macro instructions for each specific device and network configuration.

- The BSC Device Dependent Considerations section lists all BSC devices supported by BTAM, gives general information necessary for some devices, and explains applicable macro instructions by line configuration.

Prerequisites for Using this Publication

The prerequisites for a thorough understanding of this publication are a basic knowledge of the operations of the devices and a basic knowledge of System/370 machine concepts. The prerequisite publications are:

IBM System/370 Principles of Operation, GA22-7000

DOS Data Management Concepts, GC24-3427

DOS Supervisor and Input/Output Macros, GC24-5037

DOS Assembler Specifications, GC24-3414

DOS System Control and System Services Programs, GC24-5036

DOS Version 4 System Generation, GC33-5008

DOS Operating Guide, GC24-5022

The following publication is a prerequisite introduction to the data link (or line) control procedures for persons who intend to use the DOS BTAM support for BSC.

- General Information - Binary Synchronous Communications, GA27-3004

Refer to IBM SRL Bibliography - Supplement Teleprocessing, GA24-3089, for other related publications, including manuals concerning specific devices and material on binary synchronous communication.

SUMMARY OF CHANGES FROM GC30-5001

This edition reflects additional capabilities for, and modifications to, the Basic Telecommunications Access Method for Release 27 of the Disk Operating System. The changes in this edition provide information on:

- Extensions to the 2790 Data Communication System, including the 2798 Guidance Display Unit
- The 3270 Information Display System
- The 3735 Programmable Buffered Terminal
- BTAM support for Recovery Management Support Recording (RMSR)

In addition, these changes have been made to make the book more usable:

- New sections have been created for locally-attached devices and for the 7770 Audio Control Unit.
- Specific device designations have been added at the bottom of each page in the Device-Dependent Considerations sections.

INTRODUCTION 1
Machine Requirements 1
Concepts and Terminology 1
Telecommunications Networks 2
Line Control 3
Sharing the Processing Time 6
Multitasking 7
Communication Between the Problem Program, BTAM, and the Supervisor 7

SUMMARY OF BTAM COMPONENTS AND MACRO INSTRUCTIONS 9

BTAM MACRO INSTRUCTION FORMAT DESCRIPTIONS 12
BTMOD (BTAM Module) Macro Instruction 13
BTRD (Record on Demand) Macro Instruction 16
CHGNTRY (Change Terminal Entry) Macro Instruction 16
CLOSE Macro Instruction 18
CONTROL Macro Instruction 19
DFTRMLST (Define Terminal List) Macro Instruction 21
DTFBT (Define the File for BTAM) Macro Instruction 22
LERB (Line Error Block) Macro Instruction 27
LERPRT (Line Error Print) Macro Instruction 28
LOPEN Macro Instruction 29
ONLTST (On-Line Test) Macro Instruction 29
OPEN Macro Instruction 31
READ Macro Instruction 31
RELBUF (Release Buffer) Macro Instruction 33
REQBUF (Request Buffer) Macro Instruction 33
RESETPL Macro Instruction 34
RMSRTAB (Recovery Management Support Recording Table) Macro Instruction 34
SDRTAB (Statistical Data Recorder Table) Macro Instruction . . . 35
TRNSLATE (Translate) Macro Instruction 35
TWAIT Macro Instruction 35
WAIT Macro Instruction 37
WRITE Macro Instruction 38
Programming Notes 40
World Trade Telegraph Terminal Translation Tables 42
TRSRCTW and TRSRCT3 Macro Instructions 42
TRSSCTW and TRSSCT3 Macro Instructions 42

ASSEMBLY CONSIDERATIONS 45
Use of the SEPASMB Operand 45
Principles of BTAM Register Usage 45
Define the Line Groups 46
Define the Terminal Lists 46
Define the DECBs 47
Creation of a DECB (Data Event Control Block) 47
Posting in the DECB 48
Buffering 52
Buffer Pool 52
Buffer Format 52
Buffers for Read Operations 53
Buffers for Write Operations 53
Buffer Pool Updating 54
Posted Error Information 54
Serviceability Facilities 59
Start-Stop Error Recovery Procedures 59
Error Detection and Error Actions 60
Start-Stop Error Recovery Suggestions 64
Error Counts 65
BSC Error Recovery Procedures 65

Standard ERP for the 7770 Audio Response Unit 68
BTAM Program Cancellations 68
Start-Stop Online Terminal Test 69
Requesting of Tests 69
Terminal Test Restrictions 73
OnLine Terminal Test for Binary Synchronous Communication 73
Terminal-Initiated RFT 76
Program-Initiated RFT 76
Request-for-Test (RFT) Message Formats 76
Test Messages 77
Messages to the Console 79
BSC Intermixed Terminals 80

LOCAL DEVICE-DEPENDENT CONSIDERATIONS 81
Supported Local Devices 81
Local 2260-2848 81
General Information 81
Terminal to CPU 82
CPU to Terminal 82
Lock Option 83
IBM 2790 Data Communications System 83
IBM 3270 Information Display System (Local) 83
In General 83
Using This Section With the 3270 Component Description 84
Local 3270 Configuration 84
Differences Between a Local and Remote 3270 84
Sample Local 3270 Application Program 85
Character Sets 85
Defining a Local 3270 85
Controlling a Local 3270 86
Functions of READ and WRITE Macro Instructions 86
Using the WAIT Macro Instruction 86
How Attention Interruptions Are Handled 86
Read Operations 87
Read Modified Fields After Operator Action (READ Initial) 87
Skipping or Activating A Terminal: CHGNTRY 88
Terminating A Read Initial: RESETPL 88
Read Modified Fields (TM) 88
Read Modified Fields from Position (TMP) 89
Read Buffer (TB) 89
Read Buffer from Position (TBP) 90
Write Operations 90
Write Initial (TI) 91
Write Erase (TS) 91
Write Unprotected Erase (TUS) 91
Six-Bit Structured Data 92
Buffering 92
Error Recovery 92
Error Recording 92
Online Testing: RFT 92
Minimum Input Area Sizes for RFT 94
How to Request An Online Test from a Local 3270 94
Online Testing: Concurrent Use of OLTEP 95
Preparing the Program for Concurrent Use of OLTEP 95
How to Set Up and Run the BT3270SC Program 95
How to Respond to the BT3270SC Program 95
What to Do If the BTAM Application Ends During OLTEP 96
Assigning Devices During OLTEP 96
Examples of BT3270SC Processing 96
2260 to 3270 Conversion 97

AUDIO RESPONSE UNIT DEVICE-DEPENDENT CONSIDERATIONS 98
Supported Devices 98
IBM 7770 Audio Response Unit 98
General Information 98
Programming Information 98
Terminal to CPU 99
CPU to Terminal 99

CONTROL Macro Instructions 100

START-STOP DEVICE DEPENDENT CONSIDERATIONS 101
Supported Start-Stop Devices 101
Western Union Plan 115A Terminals 101
Terminal to CPU 101
CPU to Terminal 102
Terminal to Terminal 102
WU 83B3 Selectric Calling Stations 103
Terminal to CPU 103
CPU to Terminal 103
Terminal to Terminal 104
WU Model 33/35 TWX Terminals 105
Terminal to CPU 105
CPU to Terminal 106
IBM 1030 Terminals 108
Terminal to CPU 108
CPU to Terminal 109
IBM 1050 Terminals on Nonswitched Networks 111
Terminal to CPU 111
CPU to Terminal 112
Programming Notes 113
IBM 1050 Terminals on Switched Networks 115
Terminal to CPU 115
CPU to Terminal 117
Programming Notes 119
IBM 1060 Terminals 121
Terminal to CPU 121
CPU to Terminal 122
IBM 2740 Communication Terminals 124
2740 Basic 124
Terminal to CPU 124
CPU to Terminal 125
2740 witn Checking 125
Terminal to CPU 125
CPU to Terminal 126
2740 with Checking and OIU 127
Terminal to CPU 127
CPU to Terminal 129
2740 With Dial 131
Terminal to CPU 131
CPU to Terminal 132
2740 With Dial and Checking 133
Terminal to CPU 133
CPU to Terminal 134
2740 with Dial, Checking, and OIU 135
Terminal to CPU 135
CPU to Terminal 137
2740 With Dial and Transmit Control 141
Terminal to CPU 141
CPU to Terminal 142
2740 With Dial, Transmit Control, and Checking 143
Terminal to CPU 143
CPU to Terminal 145
2740 With Station Control 146
Terminal to CPU 146
CPU to Terminal 148
2740 With Station Control and Checking 148
Terminal to CPU 148
CPU to Terminal 150
IBM 2260-2848 Display Complex and IBM 2265-2845 Display Complex . . .152
Terminal to CPU 152
CPU to Terminal 154
World Trade Telegraph Terminals .157
General Information 157
Message Format 157
Contention Resolution 157
Channel Programs 158

Terminal to CPU 158
CPU to Terminal 159
World Trade Telex Terminals .160
General Information 160
Channel Programs 160
Terminal to CPU 160
CPU to Terminal 160
WRITE Break (TB) 160

BSC DEVICE-DEPENDENT CONSIDERATIONS 161
System/370 to System/370 162
Contention 162
Dial 162
ID Verification Procedures .163
Expanded ID Verification 165
Terminal Lists 168
IBM 2780 Data Transmission Terminal 173
Programming Notes 176
Component Selection 177
Point-to-Point Contention 177
Point-to-Point Dial 178
Multipoint Communication 180
IBM 1130 Computing System 182
Point-to-Point Communication 182
Multipoint Communication 182
IBM System/3 185
Point-to-Point Communication 185
Multipoint Communication 185
Programming Considerations .185
IBM System/3 Data Formats .185
Conversational Mode 185
Error Retry Count 185
Leading Graphics 185
System/360 Model 20 185
Point-to-Point Communication 185
Multipoint Communication 186
IBM 2972 General Banking Terminal System 188
Multipoint Communication 188
IBM 2790 Data Communications System 189
2715 Pulse Count Feature 189
Count Testing 191
Scheduled Counter Readout 191
DEU Set Counter Capability .192
DEU Read Counter Capability .192
2715 Tables 193
Loading the Tables 197
Macro Instructions 197
Macro Descriptions 198
CONFIGUR (Configuration) Macro Instruction 199
AS (Area Station) Macro Instruction 201
GDUAS (Guidance Display Unit Area Station) Macro Instruction . . .202
TGROUP (Transaction Group) Macro Instruction 203
ASCTR (Area Station Counter) Macro Instruction 203
CTRGROUP (Counter Group) Macro Instruction 205
CTRSCHED (Counter Schedule) Macro Instruction 206
GDUTRANS (Guidance Display Unit Transaction) Macro Instruction .206
PARAMNUM (Parameter List Number) Macro Instruction 207
PARMLIST (Parameter List) Macro Instruction 207
DISPGUID (Display Guidance) Macro Instruction 212
TRANSLAT (Translate Table) Macro Instruction 212
TRLIST (Transaction List) Macro Instruction 213
ASLIST (Area Station List) Macro Instruction 214
DEULIST (Data Entry Unit List) Macro Instruction 216
GDULIST (Guidance Display Unit List) Macro Instruction 217
CTRLIST (Counter List) Macro Instruction 218
STEND (Statement End) Macro Instruction 219
Programming Notes 220
Message Format 221

Message Header -- System/370 to 2715 221
Message Header -- 2715 to System/370 222
Transaction Header 222
Data with 2798 Transactions .225
Data with Counter Control 225
Pulse Count Transactions 227
Overflow Interrupt 227
Power Interrupt 227
External Alarm Contact Feature 227
Terminal Error Status Message (IBM 2715) 229
IBM 2770 Data Communications System 231
Transmission Codes 231
Programming -- General Information 231
Record Length 231
Blank Cards 231
Error Status Messages 231
End-to-End Control Characters .233
Buffer Description 234
IBM 50 Magnetic Data Inscriber (MDI) -- EDIT Function 234
General Characteristics 234
TPEDIT Macro Description 235
Input to the TPEDIT Macro 237
Return Codes From TPEDIT Macro 239
Error Record Identification .239
Sample Error Records 240
Program Considerations 240
End-of-Cartridge Code 241
Point-to-Point Communication .241
Nonswitched Network 242
Multipoint Communication 242
IBM 3270 Information Display System (Remote) 243
In General 243
Using this Section with Other Publications 243
Sample Remote 3270 Application Program 243
Remote 3270 Configuration 243
Programming Capabilities 243
Defining a Remote 3270 245
Defining BTAM: BTMOD 245
Defining a Line Group: DTFBT 245
Defining a Terminal List: DFTRMLST 246
Writing the DFTRMLST Macro Instruction 246
Reading and Writing with the Remote 3270 247
Read Operations 247
Read Modified Fields after Operator Action 248
Skipping or Activating a Terminal Entry: CHGNTRY 251
Terminating A Poll: RESETPL .252
Read Modified Fields 252
Read Modified Fields from Position 253
Read Buffer 255
Read Buffer from Position 256
Write Operations 257
Write 257
Erase and Write 258
Erase Unprotected Fields 259
Copy 260
Completion Analysis and Error Recovery 261
Buffering 261
Six-Bit Structured Data 261
Code Translation 261
Online Testing: (RFT) 262
How to Request an Online Test (RFT) from a Remote 3270 267
Online Testing: OLTEP 268
2260 to 3270 Conversion 268
IBM 3735 Programmable Buffered Terminal 268
BSC Channel Programs 269
Point-to-Point Contention 269
Point-to-Point Dial 273
Multipoint 282

Double Addressing 282

APPENDIX A: FORMAT OF TERMINAL LISTS 288
Start-Stop Lists 288
BSC Lists 293
Format of DIALST 293
Format of IDLST 293
Format of Auto Poll Terminal List 294
Format of SWLIST 294

APPENDIX B: HEXADECIMAL EQUIVALENTS FOR POLLING AND ADDRESSING CODES 297

APPENDIX C: BTAM MACRO OPERANDS-ALLOWABLE VALUES 298

APPENDIX D: DOS BTAM START-STOP SAMPLE PROGRAM 304

APPENDIX E: DOS BTAM BINARY SYNCHRONOUS SAMPLE PROGRAM 307

APPENDIX F: CODE CHARTS FOR BINARY SYNCHRONOUS COMMUNICATION 311

APPENDIX G: CODE CHARTS FOR IBM 2972 MODELS 8 AND 11 AND IBM 2780
MODELS 1, 2, AND 4 313

APPENDIX H: RMSR (RECOVERY MANAGEMENT SUPPORT RECORDING) IN BTAM . .318
What RMSR Is 318
How BTAM Provides RMSR 318
How to Have RMSR for a BTAM Application 318
Format of the RMSR Table 319
Examples of Using the RMSRTAB Macro Instruction 319

APPENDIX I: WORLD TRADE TELEGRAPH CODES ITA2 AND ZSC3 320

APPENDIX J: REMOTE 3270 POLLING AND SELECTION ADDRESSES 321

APPENDIX K: 3270 I/O INTERFACE CODE FOR SIX-BIT STRUCTURED DATA . . .323

APPENDIX L: SAMPLE LOCAL AND REMOTE 3270 APPLICATION PROGRAMS 324

APPENDIX M: 2715 USER-TABLE MACRO INSTRUCTIONS 340

APPENDIX N: SAMPLE 2715 TABLE LOAD MACRO ASSEMBLY 344

ACRONYMS 362

GLOSSARY 363

INDEX 366

Figure 1. Configuration of a Communication System (Part 1 of 2) . . 4
Figure 2. DOS BTAM general functions 8
Figure 3. Device-Dependent Restrictions for BTAM Macro Instructions 11
Figure 4. Format of BTMOD Macro Instruction 13
Figure 5. Format of CHGNTRY Macro Instruction 17
Figure 6. Format of the CONTROL Macro Instruction 19
Figure 7. Format of the DFTRMLST Macro Instruction 21
Figure 8. Format of the DTFBT Macro Instruction 22
Figure 9. Line Error Block 27
Figure 10. Format of READ Macro Instruction 31
Figure 11. Code Translation Tables Provided by BTAM 36
Figure 12. Format of the WRITE Macro Instruction 38
Figure 13. Contents of the Entry Field and Polling Pointer 41
Figure 14. Contents of the Entry Field and Address Pointer 41
Figure 15. READ/WRITE/CONTROL Options (Part 1 of 2) -- Audio and local devices 43
Figure 15. READ/WRITE/CONTROL Options -- Start-Stop Devices 44
Figure 16. Data Event Control Block 49
Figure 17. Buffer Pool and Buffer Formats 53
Figure 18. Buffer Pool Updating 55
Figure 19. Error Messages 61
Figure 20. Status Analysis (Start-Stop) 62
Figure 21. Sense Byte Analysis (Start-Stop) 62
Figure 22. Lost Data 62
Figure 23. Unit Exception (Start-Stop) 62
Figure 24. Incorrect Length (Start-Stop) 62
Figure 25. Time Out (Start-Stop) 63
Figure 26. Bus Out Check (Start-Stop) 63
Figure 27. Data Check (Start-Stop) 63
Figure 28. DECB Status Bytes 65
Figure 29. DECB Sense Byte 65
Figure 30. Guide to Error Recovery Actions Taken by BTAM (Binary Synchronous) 66
Figure 31. Summary of BSC Online Tests (except for the 3270) 74
Figure 32. Summary of Online Tests for the Local and Remote 3270 . . 75
Figure 33. Remote Station Types Supported for BSC1, BSC2, and BSC3. 80
Figure 34. Local 2260 Configuration 83
Figure 35. Locally Attached 3270 Display System 84
Figure 36. Local 3270 Optype Codes 86
Figure 37. Completion Code and Sense Byte Information for the Local 3270 93
Figure 38. Error Recovery Actions for the Local 3270 94
Figure 39. READ TI and READ TT Macro instructions 114
Figure 40. READ TI and WRITE TT Macro Instructions 114
Figure 41. Line Control Characters for Messages 114
Figure 42. Retry Options for Completion Codes on WRITE Initial in CPU-to-CPU Contention 163
Figure 43. Error Completion Codes and Retry Options for READ Initial (answering) in CPU-to-CPU Dial 164
Figure 44. Error Completion Codes and Retry Options for READ Initial (calling) in CPU-to-CPU Dial 165
Figure 45. Error Completion Codes and Retry Options for Initial-type WRITEs in CPU-to-CPU Dial 166
Figure 46. READ/WRITE/CONTROL Options for CPU-to-CPU Communication 167
Figure 47. Example of a Format Record 174
Figure 48. Block Number and Block Length Considerations 175
Figure 49. Examples of Component Selection 178
Figure 50. READ/WRITE/CONTROL Options for 2780 179
Figure 51. Polling/Addressing Operations for 2780 180
Figure 52. READ/WRITE/CONTROL Options for 1130 183

Figure 53. READ/WRITE/CONTROL Options for System/360 Model 20 and System/3 187
Figure 54. READ/WRITE/CONTROL Options for 2972 188
Figure 55. READ/WRITE/CONTROL Options for 2715 190
Figure 56. Format for User Transmission of Table Object Cards to the 2715 197
Figure 57. 2715 Macro storage Size Estimates 200
Figure 58. AS and GDUAS Macro ID Parameter Decimal and Hexadecimal Equivalents 202
Figure 59. Examples of Defining Transaction Group Tables for Data Entry Units 204
Figure 60. 2798 GDU Guidance Panels 211
Figure 61. 2798 GDU Keyboard Character Conversion 211
Figure 62. Matrix of ASLIST Operand Values for Guidance Lights by Position on the Area Station 216
Figure 63. Message Formats 222
Figure 64. System/370 to 2715 Message Control Byte Usage 222
Figure 65. Transmission Control Byte Usage (Part 1 of 2) 223
Figure 66. 2790 Pulse Counter Control Operation 228
Figure 67. READ/WRITE/CONTROL Options for IBM 2770 232
Figure 68. IBM 2213 Escape Sequence 233
Figure 69. IBM 2265 Valid Line Addresses 234
Figure 70. Format of Error Description Word for the TPEDIT Macro . .236
Figure 71. IBM 50 Magnetic Data Inscriber Control Codes 238
Figure 72. Sample Error Records 241
Figure 73. Remotely Attached 3270 Display System 244
Figure 74. Remote 3270 Read and Write Functions 248
Figure 75. Remote 3270 Optype Codes 249
Figure 76. Suggested Actions Following Completion of Remote 3270 Read Operations 262
Figure 77. Suggested Actions Following Completion of Remote 3270 Write Operations 263
Figure 78. Analyzing a Remote 3270 Status/Sense Message 264
Figure 79. Suggested Actions Based on Remote Status/Sense Messaages 266
Figure 80. Example of 3270 Message Translation from ASCII to EBCDIC 268
Figure 81. Example of 3270 Message Translation from EBCDIC to ASCII 268
Figure 82. Parameter List for Conversational WRITEs 271
Figure 83. Single Versus Double Addressing 282
Figure 84. Format of Parameter List 286
Figure 85. Format of Control Byte 288
Figure 86. Open Polling or Addressing List (OPENLST): Format and Examples 289
Figure 87. Wraparound Polling List (WRAPLST): Format and Example .290
Figure 88. 1050 Terminal List (DIALST): Format and Examples 291
Figure 89. TWX-Terminal List (IDLST): Format and Example 291
Figure 90. 2740SC Terminal List (SSAWLST): Format and Example . . .292
Figure 91. WTTA Terminal List (WTTALST): Format and Examples . . .292
Figure 92. Format of Auto Poll Terminal List (AUTOLST,AUTOWLST) . .294
Figure 93. Format of Calling and Answering Lists of the SWLST Form .295
Figure 94. RMSR Table 319
Figure 95. Control Unit Addresses for Polling and Identification . .321
Figure 96. Control Unit Addresses for Selection 321
Figure 97. Device Addresses for Polling, Selection, and Identification 322
Figure 98. I/O Interface Code for Six-Bit Structured Data 323

The Basic Telecommunications Access Method (BTAM) controls transmission and reception of messages over telecommunications lines in response to READ, WRITE and CONTROL macro instructions issued in the user's problem program. The primary functions of BTAM are channel program generation during program execution and, at the user's option, buffer management.

The facilities of BTAM are made available through the macro-generation capabilities of the System/370 Disk Operating System (DOS) assembler. The executable BTAM routines and certain information about the terminals and lines to be used are included in the Disk Operating System during system generation and initial program loading (IPL). During assembly of a problem program, macro instructions coded by the user are expanded into:

1. The BTAM logic module.
2. Inline instructions, including linkages to the executable routines in the BTAM logic module.
3. Tabular information defining the lines, terminals, and options to be used.

During execution, the problem program communicates with BTAM through various BTAM macro instructions and tables in order to request such teleprocessing services as preparing communication lines for use, performing data transmission, or providing buffer management facilities. Preparing the lines for subsequent data transmission is accomplished by the execution of an OPEN macro instruction. OPEN will generate and execute channel programs, where required, in order to initialize the associated control unit and data set. Optionally, OPEN will also organize and format a buffer pool.

After OPEN, a message may be sent or received by executing a READ or WRITE macro instruction, which causes linkage to the BTAM READ/WRITE/CONTROL routine. This routine builds a channel program to perform the requested operation and passes a request for execution to the DOS supervisor, which schedules the channel program execution. Control passes back to the user at this point, and execution of the channel program is performed asynchronously with respect to problem program execution.

An important feature of BTAM is repeated restarting of channel programs in response to line conditions. This allows a single READ macro instruction to successively poll (or invite to send) a number of terminals on a line and to continue this polling process (without interfering with the problem program) until a message has been read from one of the polled terminals. A single WRITE can signal a number of terminals to prepare to receive and then send them the message.

BTAM buffer management maintains buffer pools, providing buffers from the pool to the user as he requests them, and returning buffers to the pool when he releases them. Buffer management is performed synchronously with respect to the execution of the problem program.

Machine Requirements

BTAM requires a system with at least 32K bytes of main storage. The only additions to the minimum requirements of the Disk Operating System are the following:

- The IBM 1052 Printer-Keyboard is mandatory.
- All telecommunications devices except the local 2260, local 3270, and the 2701 with Synchronous Data Adapter - Type II (leased lines only) must be attached to the System/370 via the multiplexer channel.
- The 2701 SDA-II (leased line) may be attached to appropriate models of the System/370 via the selector channel provided the channel is not shared with any burst mode device, including the device on which SYSRES is kept. Since Auto Polling is used with this device, only one line can be operational while polling.
- A BTAM application program may operate locally attached 2260s and 3270s in burst mode on the multiplexer channel if no other device type is being run by the program.

CONCEPTS AND TERMINOLOGY

This section describes line and terminal configurations and message control considerations. In addition, it defines some terms used in this publication.

In general, a telecommunications system consists of a number of input, output, or input/output devices, usually in geographically-dispersed locations, connected by one or more communication lines. A telecommunications system operating under an IBM access method, such as BTAM, and the Disk Operating System (DOS), is more specifically defined as a network of terminals connected to a central computer by one or more communication lines. In addition, BTAM, used with DOS, supports CPU-to-CPU binary synchronous communication in which neither end of the communication line (switched or nonswitched) can be reasonably described as a terminal as that term has been defined in the telecommunications literature. In this publication the meaning of telecommunications system is broadened to include both remote and local terminals connected to a central computer and CPU-to-CPU communication.

A terminal is the unit or units of equipment that accepts keyed or punched data as input for sending to the computer and/or produces printed, punched, or visually-displayed data or audio messages as output received from the computer. All messages from one terminal to another pass through the computer. In addition, the computer itself may receive from the terminals and originate messages for the terminals.

A terminal consists of a terminal control unit and one or more input/output devices. Each such device is called a component. Each input device and each output device is considered a separate component, regardless of whether they are physically combined. For example, an IBM 1050 is referred to as a terminal; its constituent devices or components include the 1053 Printer, 1054 Paper-Tape Reader, keyboard section of the 1052 Printer-Keyboard, printer section of the 1052 Printer-Keyboard, etc.

Terminals in a telecommunications system operated under BTAM control are usually separated from the computer by a distance sufficient to require common-carrier facilities and transmission techniques to accomplish communication with the computer. The system may, however, include local terminals that are located on the same premises as the computer and attached directly to the channel. The important distinction, however, is the way in which the cable or lines are attached to the computer. Terminals connected through a 2701, 2702, or 2703 telecommunications control unit are all classified as remote. Units that are directly connected to a System/370 channel are classified as local.

In communications terminology, the terms communication line, data link, data path, circuit, and channel are somewhat synonymous. All of these terms are used to represent the medium by which the physical components of a system are connected. In this publication the term communication line (or line) is used, and, except as otherwise qualified, refers to any medium, whether it be a telegraph circuit, a telephone circuit, a privately owned line, etc.

Each remote terminal is connected to a communication line by either a data set or a modem (modulator/demodulator), depending on the kind of communication line and the kind of terminal involved. The precise functions vary, but the overall purpose is the same: to provide an interface between terminal and line. This publication uses the term data set to represent any of these units (not to be confused with a program data set, which is an Operating System term).

The applications programmer generally need not concern himself in any way with these data sets, because their presence generally exerts no influence on programming. They are mentioned in the interest of providing a complete, accurate picture of the line and terminal configuration.

The term station means the aggregate of equipment and controls attached to any one of the several possible ends of a communication line. Put another way, a station is a terminal (including the terminal components) plus the equipment by which the terminal is attached to the line. Even more generally, the meaning of station may be broadened to include either CPU in a CPU-to-CPU communication system.

Telecommunications Networks

A telecommunications system may include a switched network, a nonswitched network, or a combination of the two.

A nonswitched network consists of one or more private lines that connect the computer to one or more remote units. The computer and the remote units are physically connected; that is, the circuits making up the communication lines are continuously established for predetermined time periods during which data transmission may proceed between the computer and the remote stations. In this type of system, the computer can, under certain conditions, send messages to more than one remote device on the same line at the same time. These lines comprising a nonswitched network are known as private or nonswitched lines. Private lines are furnished by a common carrier on a contract basis, between specified locations for a continuous period or regularly recurring periods at stated hours, for the exclusive use of one customer.

A switched network consists of a number of remote stations with which the computer can communicate. The computer and the several remote devices are each connected by access lines to the common-carrier exchanges serving their respective locations. A continuous data path is established between computer and remote device only for the period of time during which transmission is to take place. The connection is established by dialing the telephone number of the unit (either terminal or CPU) at the other end. In this type of system, communication is established between the computer and only one unit at a time on each line. In this case, line refers to a discrete data path between the transmission control unit and the common-carrier exchange.

In this publication, the term switched network refers to any network in which the data link between computer and remote device must be established by dialing for data transmission to occur. The term nonswitched network refers to a network in which the communication lines linking computer and remote units are continuously established, thus requiring no dialing.

Figure 1 shows a teleprocessing system with both a switched and a nonswitched network.

Line Control

A programmer using BTAM must be aware of the line-control conventions for the particular type (or types) of terminal and line configurations for which he is programming. This section deals with general considerations about line control. Specific information about the types of terminals and line configurations is given in later sections of this publication.

In some respects, the functions that must be performed by the programmer using BTAM are parallel with those performed in telecommunications systems that are not computer-oriented. In any telecommunications system, contact between terminals must be established before a message is sent. In some systems, terminals wishing to send a message contend with one another for use of the line. The first terminal to initiate contact on a line that is not currently in use controls the line and prevents its use by other terminals until it has concluded its message transmission. A system operated in this manner is a contention system.

In other noncomputer-oriented systems, one of the terminals is specified as the control station. This terminal initiates all contacts for all other terminals on the line, using a procedure known as polling. Polling is a flexible, systematic, centrally-controlled method permitting terminals on a multiterminal line to transmit without contending for use of the line. The control station contacts the other terminals periodically and invites them to send any messages they have ready. In addition, the control station itself may elect to send. A system operated in this manner is a polling system.

Polling is accomplished by sending one or more selection characters on the line. In some cases, two characters are used: the first selects the terminal, and the second selects the specific component of that terminal.

In centralized operation, the terminal identified by these characters then sends a response to the control station: a message if it has a message to send, a negative response if it does not. The control station may poll a number of terminals and components, in turn, until one is found that has a message ready. Similarly, when the control station terminal has a message to send, it transmits one or more addressing or call-directing characters on the line. As in polling, two characters are often used: the first selects the terminal, and the second selects the component. The terminal identified by these characters returns a response. It returns a positive response if it is able to accept the message, or a negative response if it is not.

In noncentralized operation, a terminal that is polled by the control station may send a message directly to another terminal without involving the control station in a message-switching operation.

A telecommunications system operated under BTAM is usually a polling system in which the role of control station is played by the computer. Moreover, it is a centralized system; that is, terminals send their messages not to other terminals, but to the computer. The computer then relays the messages to the appropriate destination terminals. Polling is not used, however, for the local 2260, the local 3270, the audio response units, the 2740 without station control, or the World Trade telegraph terminals.

The polling and addressing functions are performed in both switched and nonswitched systems, with minor variations.

In a switched network, the line connection must be completed between computer and remote device before message transmission can proceed. The connection may be established by either the computer or remote unit. When the computer wishes to establish the connection, it dials the telephone

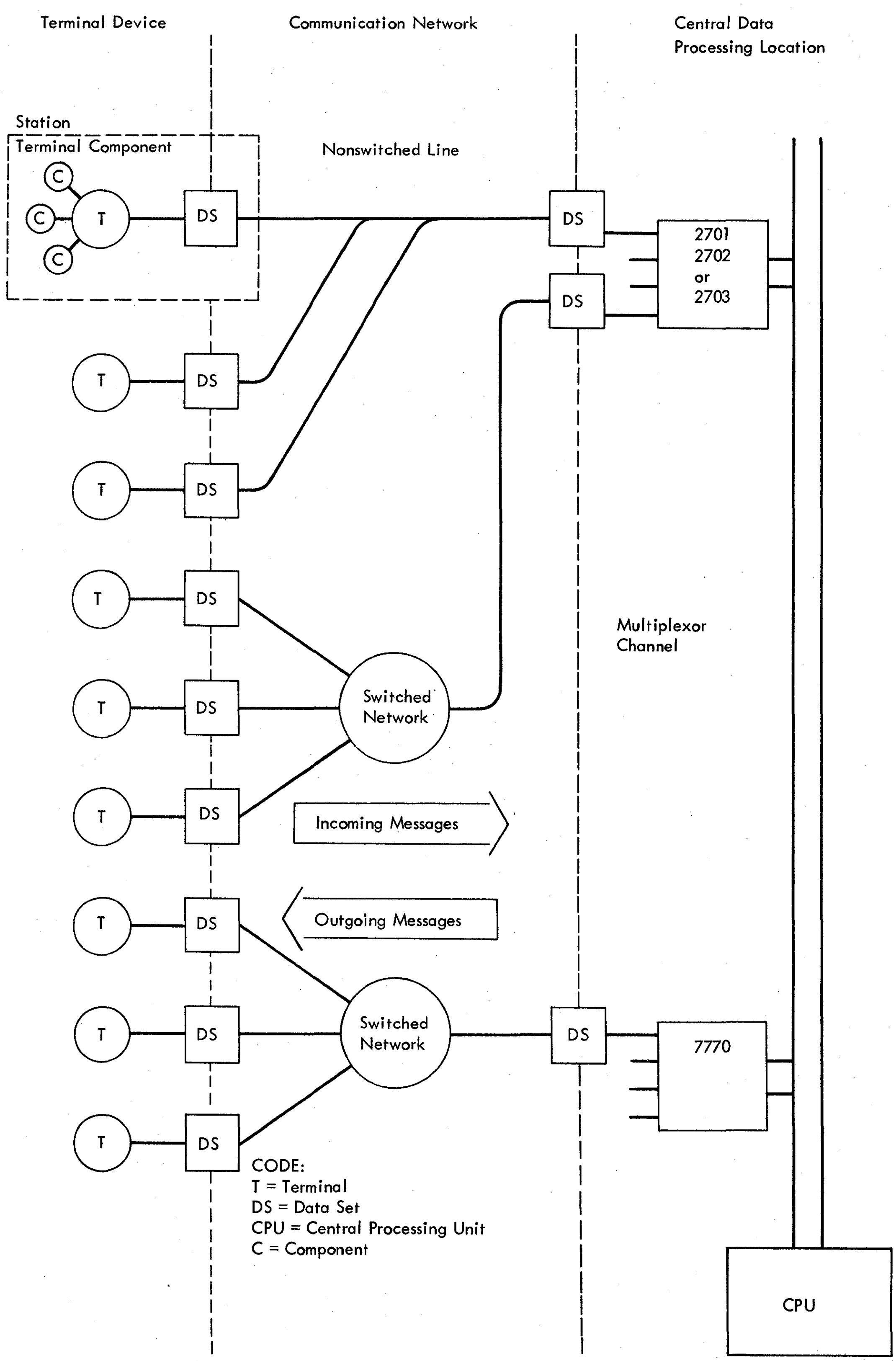


Figure 1. Configuration of a Communication System (Part 1 of 2) Start-Stop System

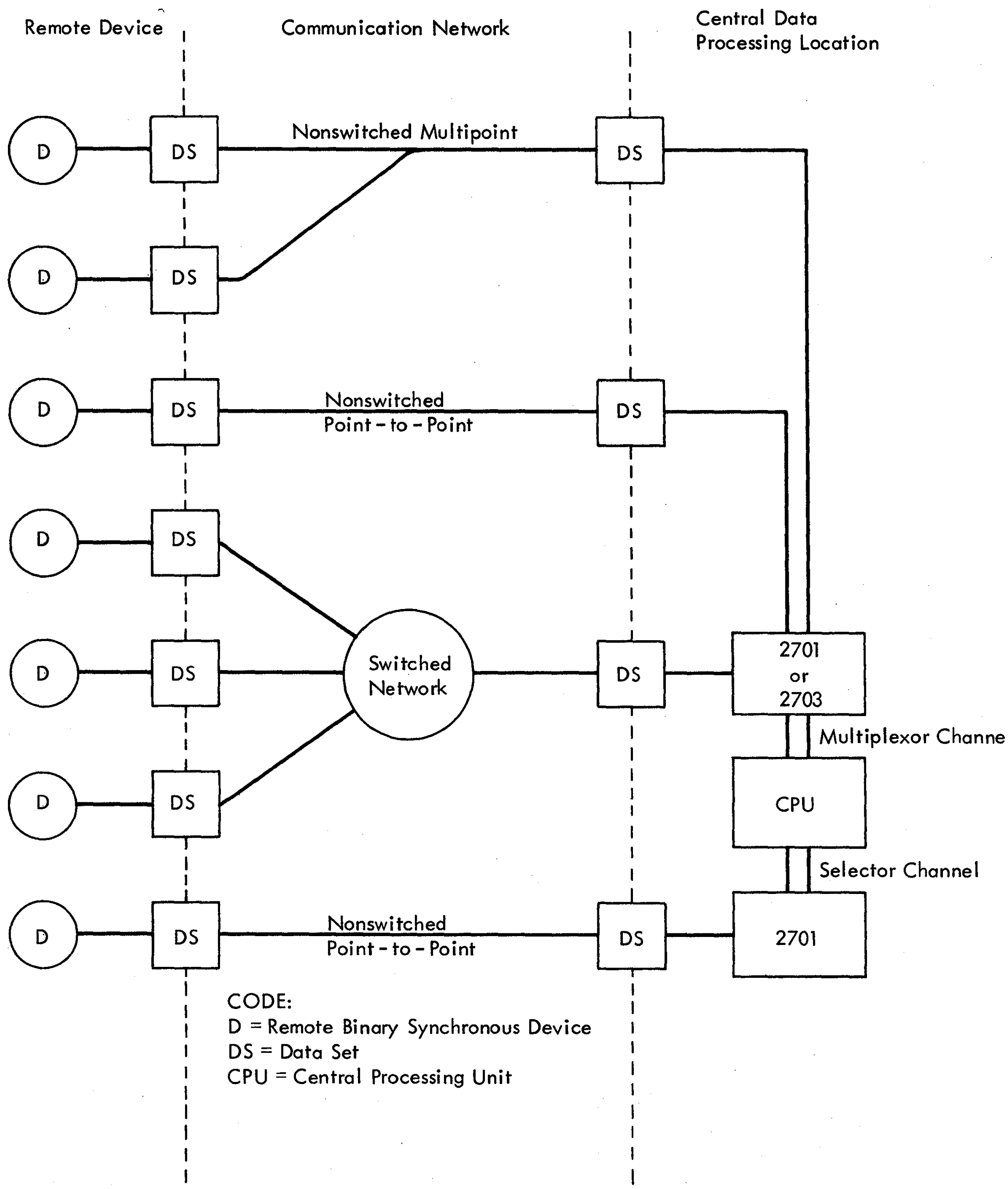


Figure 1. Configuration of a Communication System (Part 2 of 2) Binary Synchronous System

number of the remote unit. The connection becomes established when the unit responds. The functions performed by the computer in this case are known as <u>calling</u>. Polling, addressing, or ID verification, if needed, may then take place.

Ordinarily, the computer calls a terminal only when it wishes to address the terminal (to send it a message), rather than to poll it (to solicit messages). When a terminal wishes to establish the connection, the person operating the terminal dials the computer's telephone number (or one of its several numbers). The connection is established when the computer responds. The function performed by the computer in this case is known as <u>answering</u>. Polling or addressing may then take place.

Ordinarily, a terminal calls the computer only when it wishes to be polled for a message it has ready for the computer or another terminal. Note that regardless of which party -- computer or terminal -- establishes the line connection, message flow from the terminal to the computer is achieved by reading from the terminal, and message flow from the computer to a terminal is achieved by writing to the terminal, except for those devices where polling/ addressing is not used.

Although terminals can call the computer at any time, the computer, to fulfill its function as control station, must be able to accept or reject incoming calls. Therefore, the computer performs a function known as enabling the line. Enabling is the process of conditioning the telecommunications control unit (TCU) and the data set to accept incoming calls on a line. The user's program determines which lines are, at a given moment, to be enabled, and which are not. If a terminal calls in on a line that is currently enabled, and that is not in contact with another terminal, the line connection is complete and message transmission (which may be preceded by polling or addressing) can occur. If a terminal calls in on a line that is not currently enabled, or that is enabled but is occupied with another terminal, the calling terminal receives a busy signal, and contact is not established. The terminal must wait and call again later.

Calling and answering functions are needed in CPU-to-CPU communication when line connection is over a switched network. Although polling and addressing are not used, the programmer may elect to employ BTAM-supported ID verification procedures, which provide greater security protection and line assurance. The CPU-to-CPU Dial section discusses these procedures in detail.

In a nonswitched network, the line connections between computer and terminals are continuously established; hence, the calling and answering functions are not required. Only the computer can initiate contact with remote terminals, except in contention operation involving the 2740. As in switched networks, the computer polls terminals to solicit messages from them and addresses terminals to send messages to them.

Such functions as polling, addressing, calling, answering, and ID verification, while performed by BTAM, require that the programmer supply the appropriate characters to be used and issue the BTAM macro instructions performing these functions. There are, however, other functions automatically provided by BTAM and about which the programmer generally need not concern himself. For example, when the problem program sends a message via BTAM to a remote device, the remote device generally responds with one or more control characters, signifying negative or positive acknowledgment of the message (the negative or positive acknowledgment indicating that the message was received with or without I/O error). Such control characters are analyzed by BTAM and further action may be initiated by BTAM. In the case in which a negative acknowledgment is received, for example, BTAM will automatically initiate a retransmission when the Error Recovery Procedures are specified by the user.

There are other functions with which the programmer does need to concern himself. For example, the Start-of-Text (STX) control character is generally required as the first character sent to the remote device. Later sections of this publication discuss various device-dependent control character sequences in greater detail (see the Device Dependent Considerations sections).

In binary synchronous communication these line control procedures are standardized and are generally independent of the remote device being used, insofar as the basic functions are concerned. For example, alternating acknowledgments are implemented in response to messages sent and/or received, regardless of the remote device used. The alternating acknowledgments ACK-0 and ACK-1 (even and odd) are used, as the name suggests, to acknowledge (positively) alternate messages. For instance, if a response from the remote device to a message sent by the CPU is ACK-0 (even), the (positive) response to the next message sent by the CPU will be ACK-1 (odd). This affords the programmer greater message assurance: protection against duplication or loss of message blocks.

The use of alternating acknowledgments and other data link control characters is shown in the Device Dependent Considerations sections of this publication.

Polling, addressing, calling, and answering are not used by the World Trade telegraph terminals. These terminals are always ready to receive or to send a message. For further details, refer to the Device Dependent Considerations sections of this publication.

Sharing the Processing Time

The processing time in a multiprogramming environment may be shared by two types of problem programs: background and foreground. Background programs are initiated by job control from the batched-job input stream. Foreground programs are initiated by the operator from the console-typewriter or by job control from the batched-job input stream.

Background and foreground programs are initiated and terminated asynchronously with each other. Neither is aware of the status or existence of the other.

A telecommunications program may be run either as a foreground or as a background program. Normally, it is run as a foreground-one program to have the highest priority of any program being executed at a particular time. Normally, a background program is run along with the telecommunications program.

DOS is capable of concurrently operating one background program and one or two foreground programs. Priority for CPU processing is controlled by the supervisor, with foreground programs having priority over background programs. When an interruption occurs, the supervisor gains control, the interruption is processed, and the supervisor gives control to the highest priority program that is in a ready state. Control is taken away from the highest priority program when that program must wait for the completion of a specific event before further processing can be performed. Control is taken away from a lower priority program when an event for which a higher priority program has been waiting has been completed, or when that program must wait for the completion of a specific event before further processing can be performed. When all programs in the system are simultaneously waiting (i.e., no program can process), the system is placed in the wait state, enabled for interruptions. When an interruption occurs signifying the completion of the event for which the program was waiting, that program becomes active and competes with other programs for CPU processing time.

Multitasking

With multitasking it is possible to perform multiprogramming within any one or all of the partitions: background, foreground-one, and foreground-two. For multiprogramming users, multitasking extends the capabilities of the DOS Supervisor to execute twelve programs rather than three.

A telecommunications program may be run as any of the system's twelve or fewer tasks.

There should be one BTAM Logic Module and one set of tables defining the lines, terminals, and options per system task.

Figure 2 shows the general functions provided for a problem program operating under DOS/BTAM. Issuance of a READ/WRITE/CONTROL starts at numeral one (1); interruption processing starts at numeral nine (9).

Communication Between the Problem Program, BTAM, and the Supervisor

1. The problem program communicates requests for I/O operations and other services to BTAM through the BTAM macro instructions. The information needed by BTAM to perform these functions is provided directly by the macro operands and by tables pointed to via these operands. For example, for I/O operations, the DTFBT (a table containing information defining the line group) and the DECB (a block containing information needed to generate a channel program) are required by BTAM.

2. BTAM requests input/output operations from the supervisor, as do other DOS access methods. See DOS Supervisor and I/O Macros, GC24-5037, for information on these macros. An Execute Channel Program macro instruction (EXCP) specifies a Command Control Block (CCB) as an operand. The CCB contains the address of the channel program to be executed. Under BTAM the CCB is assembled as a part of a Line Control Block (LCB). The LCB contains other fields in which BTAM maintains control information for the line as well as an area in which the channel program is generated. Each line in a line group is associated with a separate LCB. The DTFBT, in turn, incorporates all the LCB's for the line group as well as other fields containing control information for the line group as a whole. Because the LCB is used by BTAM logic, it must not be used or modified by the problem program.

   Note: BTAM does not support DEVICE=IGN (Ignore).

3. I/O interruptions are asynchronous with the problem program. BTAM posts each line activity completion in the Data Event Control Block (DECB) associated with that line. When further processing is contingent on the termination of I/O operations, the user releases control by a WAIT macro instruction. When that WAIT is satisfied, control returns to the problem program at the instruction following the WAIT.

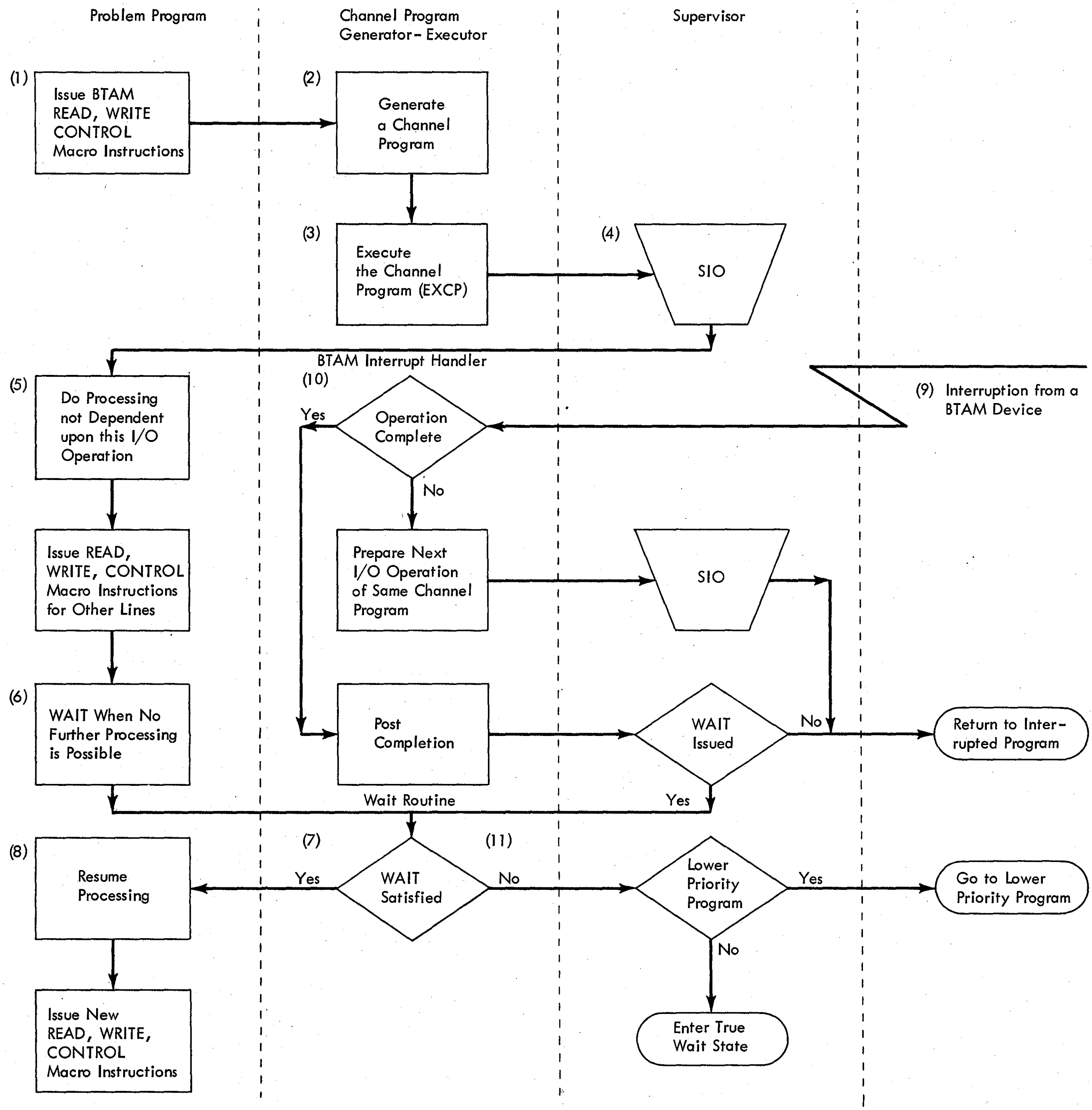


Figure 2. DOS BTAM general functions

The user assembles the appropriate BTAM routines according to the telecommunications requirements of his problem program. The BTAM module assembled includes routines for:

- READ, WRITE, and CONTROL.
- Multiple wait and RESETPL.
- I/O interruption processing.
- REQBUF and RELBUF (optional).
- Buffer Logic for READ and WRITE (optional).
- Error recovery procedures (optional except for binary synchronous communication, Audio Response units, and World Trade telegraph terminals).
- Error Counts (optional).
- Code Translation (optional).
- On-line terminal testing (optional).
- Local 2260 support (optional).
- Local 3270 support (optional).
- 7770 support (optional).
- Switched network support (optional).
- Binary Synchronous Communication logic (optional).
- Start-stop Auto Poll support (optional).
- BSC multipoint logic (optional).
- World Trade telegraph terminal support (optional).
- DECB Extension (optional).
- Recovery Management Support Recording (RMSR) (optional)

The macro instructions provide the user with the means of control over BTAM and can be broken down into two categories:

1. Declarative macro instructions are used to generate lists containing terminal and line information. They are also used to create or load tables. They are assembly-time macro instructions.

   DTFBT
   is the user's means of defining a line group and selecting any optional features of BTAM that are to be utilized for that line group.

   DFTRMLST
   creates terminal lists that contain the characters necessary to perform polling, addressing, dialing, automatic answering or ID checking.

   LERB
   generates and initializes a table for accumulation of error counts.

   RMSRTAB
   generates and initializes 20-byte tables for the accumulation of RMSR counts.

   SDRTAB
   replaced by RMSRTAB but still supported; generates an RMSR table.

   READ
   may be used to generate a DECB at assembly time.

   WRITE
   may be used to generate a DECB at assembly time.

   CONTROL
   may be used to generate a DECB at assembly time.

2. Imperative macro instructions are used to perform a specific function at execution time.

   BTMOD
   permits the user to select the options to be included in the BTAM module.

   LOPEN
   is used to condition a line on which an error occurred during OPEN, and, when necessary, to initiate line delays.

   OPEN
   prepares communication line groups for use, and, when necessary, OPEN generates and executes required channel programs. Optionally, it may also organize a buffer pool.

CLOSE
: removes communication line groups from use. Active channel programs are halted, if necessary.

READ
: updates the DECB, generates linkage to the BTAM READ/WRITE/CONTROL routine, and causes BTAM to perform a specified function (such as polling terminals or reading a message into an input buffer area).

WRITE
: updates the DECB and causes BTAM to perform a specified function (such as addressing a terminal and writing a message to it from an output area).

CONTROL
: causes BTAM to execute a control function (for example, enabling or disabling a line).

CHGNTRY
: deactivates or activates a polling or addressing entry in a terminal list without redefining the list. CHGNTRY may be used to deactivate an LCB for a local 2260 or local 3270.

RESETPL
: causes a read operation to terminate by interrupting polling. On a switched-connection line, it causes the halting of an outstanding Enable command. It is also used to terminate a Prepare command. RESETPL frees a DTFBT after a READ for a local 2260 or local 3270.

WAIT
: causes release of control of the CPU until a user-specified number of events have been completed.

TWAIT
: relinquishes control of the CPU until one of a number of events has been completed. Optionally, the terminal test facility may be implemented by this macro instruction.

LERPRT
: provides the ability to print the current values of the number of transmissions and error count accumulators.

BTRD
: generates linkage to the BTAM BTRD routine, and causes BTAM to log all RMSR counters to disk. The counters in main storage are reset after logging is complete.

TPEDIT
: is used to specify the type of editing to be done on the input received from the IBM 50 Magnetic Data Inscriber (MDI) attachment to the 2772. This macro is described in the section on the IBM 2770 Data Communications Terminal.

TRNSLATE
: causes the translation of message blocks from one code to another.

ONLTST
: is used to generate linkage to the binary synchronous on-line test logic in BTMOD.

REQBUF
: provides one or more buffers from a buffer pool if they are available.

RELBUF
: returns one or more buffers to a buffer pool.

Figure 3 shows which macro instructions do <u>not</u> apply to the various BTAM-supported devices. For this reason, the user must not use these macro instructions for the indicated devices.

Device	Non-Applicable Macro Instructions
1030 1050 (Nonswitched) 1050 (Switched) 1060 2260 (Remote) 83B3 115A	CONTROL
TWX 33/35	CONTROL, CHGNTRY
7770	DFTRMLST, CHGNTRY, LERB, RESETPL, LERPRT, LOPEN
2260 (Local)	CONTROL, DFTRMLST, LERB, LERPRT
2740 (Basic)	CONTROL, DFTRMLST, CHGNTRY, RESETPL
2740 D (Dial)	CONTROL, CHGNTRY, RESETPL
2740 S (Station Control)	CONTROL
2740 SC (Station Control and Checking)	CONTROL
2740 DTC (Dial, Trans mit Control and Checking)	CONTROL
2740 C (Checking)	CONTROL, DFTRMLST, CHGNTRY
2740 DC (Dial and Checking)	CONTROL, CHGNTRY
2740 CO (Checking and OIU)	CONTROL, CHGNTRY
2740 DCO (Dial, Checking and OIU)	CONTROL, CHGNTRY
2740 DT (Dial and Transmit Control)	CONTROL
3270 (Local)	CONTROL, DFTRMLST, LERB, LERPRT, ONLTST
3270 (Remote)	CONTROL, ONLTST
WTTA	CONTROL, CHGNTRY

Figure 3. Device-Dependent Restrictions for BTAM Macro Instructions

This section provides a coding format illustration for each macro instruction described. Because the form of tabular data generated or the exact function performed varies depending on the particular remote device or network configuration, no attempt is made in this section to discuss the uses of macro instructions in detail. Rather, the Device Dependent Considerations sections present greater detail on the use of the macro instructions for each specific device or network configuration.

To describe macro instruction operands, a system of conventions is defined for use in the format descriptions:

- Both positional and keyword operands are described by a 3-part structure. Positional operands are described by a lower case name followed by a hyphen followed by a value mnemonic or a coded value.

  Example: numchars-absexp

  The lowercase name, numchars, is merely a convenient referent to the operand and, along with the hyphen and value mnemonic, is never coded by the programmer. The programmer replaces the positional operand in his coding by an expression defined by the value mnemonic as allowable.

  In the case of keyword operands, the 3-part structure consists of the keyword followed by an equal sign (both of which must be coded as shown), followed by a value mnemonic or coded value that describes what to code on the right side of the equal sign.

  Example: BUFCB=symbol

- Coded values are written in the format description as numbers or uppercase letters and must be coded as shown.

- Value mnemonics are written lowercase in the format descriptions and indicate how an operand is to be coded by the programmer. The value mnemonics used in this publication are:

  symbol
  : any symbol that the assembler accepts in the name field of an instruction.

  relexp
  : a relocatable expression (acceptable as an A-type address constant by the assembler).

  code
  : one of the coded values defined as allowable by the individual macro.

  absexp
  : 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 CSECT, and arithmetic combinations of absolute terms.

  integer
  : a decimal self-defining term.

  decchars
  : concatenated decimal digits (differs from "integer" in that each digit is assembled in binary format individually).

  hexchars
  : concatenated hexadecimal digits (the framing characters -- X' ' -- are not coded by the programmer).

  (r)
  : register notation specifying, by an absexp enclosed in parentheses, any register 2-12 and allowing the programmer to load the specified register with the appropriate value at execution time. Certain macro instructions also permit use of registers 0 and 1 by specifying (0) or (1). The 0 or 1 within parentheses must be coded literally.

  Note: When the format description indicates that register notation is an alternative to absexp, the programmer should be aware that an expression framed by parentheses is assumed to be register notation.

- {} Braces are used in two ways:

  1. To define grouping of the alternate forms of a mandatory operand. For example:

     $$\text{listype-}\begin{Bmatrix}\text{OPENLST}\\\text{WRAPLST}\\\text{DIALST}\end{Bmatrix}$$

2. To define a syntactical unit when the braces are followed by an ellipsis (three periods). For example:

   {dtfbt-symbol,}...

   The entire expression indicates that the syntactical unit (enclosed by braces) can be coded one or more times.

- [] Brackets indicate an optional operand. For example:

  [MSGL=absexp,]

  When one of the alternate forms of the operand is underlined, the function of that form is used if the operand is not coded. For example:

  [SWITCH=NO, | SWITCH=YES, | SWITCH=NEWID,]

  If none of the above is coded, SWITCH=NO is assumed.

The listed descriptive symbols are not to be coded by the programmer. However, parentheses and the apostrophes (single quote marks) that appear in the macro format illustrations must be coded as shown.

The rules for coding commas to separate operands are:

1. When the programmer omits a positional operand in the coding, the trailing comma must still be coded unless:

   - It is the last positional operand in the macro.

   - The programmer also omits all positional operands that are normally coded following the omitted positional operand.

   - The syntax in the macro format illustration indicates that the comma may be omitted along with the operand. (This is true for the device-dependent usage of DFTRMLST. See especially the use of DFTRMLST for IDLST in the Device Dependent Considerations sections of this publication.)

2. When a keyword operand is omitted, its trailing comma is also omitted.

3. The comma following the last coded operand is to be omitted.

Note: Comma usage is in accordance with the conventions defined for the DOS macro generator.

In addition to this section presenting the macro format descriptions, an additional summary of allowable values for the BTAM macros is provided in Appendix C. The macros are arranged in alphabetical order for ease of use.

BTMOD (BTAM Module) Macro Instruction

The BTMOD macro instruction (Figure 4) provides the user with the facility to assemble the BTAM routines to perform the telecommunications functions of his problem program.

Note 1: BTMOD should not be assembled in the middle of the user's code. BTMOD is reusable, but it is neither reentrant nor read only. See the Principles of BTAM

Name	Operation	Operand
[symbol]	BTMOD	[SEPASMB=NO, | SEPASMB=YES,] [BUFFER=NO, | BUFFER=REQREL, | BUFFER=YES,] [ERLOGIC=E, | ERLOGIC=N, | ERLOGIC=C, | ERLOGIC=NC,] [TERMTST=NO, | TERMTST=YES,] [SWITCH=NO, | SWITCH=YES, | SWITCH=NEWID,] [L2260=NO, | L2260=YES,] [L3277=NO, | L3277=YES,] [TST3277=NO, | TST3277=YES,] [AUDIO=NO, | AUDIO=YES,] [CANCEL=NO, | CANCEL=YES,] [TRANSL=NO, | TRANSL=YES,] [BSCS=NO, | BSCS=YES,] [SSAPL=NO, | SSAPL=YES,] [BSCMPT=NO, | BSCMPT=YES,] [BSCTEST=NO, | BSCTEST=YES,] [WTTA=NO, | WTTA=YES,] [DECBEXT=NO, | DECBEXT=YES,] [RMSR=NO, | RMSR=YES,] [RESETPL=YES, | RESETPL=NO,]

Figure 4. Format of BTMOD Macro Instruction

Register Usage section for a detailed discussion of this point.

Note 2: A separate BTMOD macro instruction must be issued for each BTAM task.

symbol
: is used to assign a name to the assembled BTAM logic module. If no symbol is coded in the BTMOD name field, a standard name (IJLBTM) is generated.

SEPASMB
: specifies how the BTAM logic module is to be assembled.

  NO
  : specifies that the BTAM module is to be assembled with the problem program (see Assembly Considerations).

  YES
  : specifies that the BTAM module is to be assembled separately. This causes the appropriate CATALR control card to be punched, thus permitting the user to catalog the module in the relocatable library for later retrieval by the linkage editor (see Assembly Considerations).

BUFFER
: specifies whether the BTAM module is to contain the buffer management routines, where:

  NO
  : specifies that the buffer management routines are not to be included.

  REQREL
  : specifies that only the REQBUF/RELBUF logic is to be included.

  YES
  : specifies that the REQBUF/RELBUF logic and the logic to read into buffers and to write from buffers are to be included.

ERLOGIC
: specifies what error handling routines are to be included in the BTAM module, where:

  E
  : specifies start-stop, local 2260, local local 3270 error recovery procedures (ERP) with no error count.

  N
  : specifies that start-stop, local 2260, and local 3270 error recovery procedures are not to be included (not allowed if RMSR or OBRSDR=YES).

  C
  : specifies that start-stop, local 2260, and local 3270 ERP and error count are to be included (not allowed if RMSR or OBRSDR=YES).

  NC
  : specifies that error count logic is to be included in the binary synchronous error recovery procedures if BSCS=YES. Start-stop, local 2260, and local 3270 ERP are to be excluded. Start-stop error count logic will not be included (not allowed if RMSR or OBRSDR=YES).

Note: This operand does not apply to the 7770 Audio Response Unit, nor to BSC except when NC is coded. The ERP unique to the local 2260 is always omitted if L2260=NO is coded. The ERP unique to the local 3270 is always omitted if L3277=NO is coded.

TERMTST
: specifies whether the online terminal test facility is to be included, where:

  NO
  : specifies that this facility is not to be included in the BTAM module.

  YES
  : specifies that this facility is to be included in the BTAM module (see the TWAIT macro instruction).

Note: This operand does not apply to the World Trade telegraph terminals or to the local or remote 3270. For the local 3270, online terminal test logic is included by specifying TST3277=YES; for the remote 3270, online terminal test logic is included by specifying BSCTEST=YES.

SWITCH
: specifies whether the BTAM logic concerning switched networks is to be included, where:

  NO
  : specifies that this logic is not to be included in the BTAM module.

  YES
  : specifies that this logic is to be included in the BTAM module.

  NEWID
  : specifies that BTAM is to provide the logic required for switched line support and that required for expanded ID verification.

L2260
: specifies whether the BTAM logic module for the local 2260 is to be included, where:

NO
specifies that this logic is not to be included in the BTAM module (2260 Local support not needed).

YES
specifies that this logic is to be included.

L3277
specifies whether the BTAM logic for the local 3270 (display station and printer) is to be included, where:

NO
specifies that this logic is not to be included in the BTAM logic module.

YES
specifies that this logic is to be included.

TST3277
specifies whether the BTAM logic for the online test facility (Request for Test) for the local 3270 display station and printer is to be included, where:

NO
specifies that this logic is not to be included.

YES
specifies that this logic is to be included.

AUDIO
specifies whether the BTAM logic for support of the 7770 Audio Response Unit is to be included, where:

NO
specifies that this logic is not to be included in the BTAM module (7770 support not needed).

YES
specifies that this logic is to be included in the BTAM module.

Note: If AUDIO=YES is coded, the error recovery procedures (ERP) unique to audio are automatically included.

CANCEL
specifies whether the BTAM CANCEL logic will cancel a job or return control to the user when a cancel condition is detected.

NO
specifies that control should be returned to the user at the instruction following the BTAM macro instruction that detected the cancel condition, along with the conditions for cancellation.

YES
specifies that a DOS CANCEL macro instruction is given when a cancel condition occurs.

TRANSL
specifies whether the Code Translation facility is to be included, where:

NO
specifies that this facility (see the TRNSLATE macro instruction) is not to be included.

YES
specifies that this facility is to be included.

BSCS
specifies whether the BTAM logic for point-to-point binary synchronous communication is to be included, where:

NO
specifies that this logic is not to be included.

YES
specifies that this logic is to be included. In addition, the ERP unique to binary synchronous communication is automatically included, independent of the ERLOGIC operand. Only ERLOGIC=C and ERLOGIC=NC are applicable to binary synchronous communication, since the error count logic is the same for start-stop and synchronous lines.

SSAPL
specifies whether the BTAM logic for the start-stop Auto Poll support is to be included, where:

NO
specifies that this logic is not to be included in the BTAM module.

YES
specifies that this logic is to be included in the BTAM module.

Note: This operand does not apply to the World Trade telegraph terminals.

BSCMPT
specifies whether the logic required to support multidropped BSC devices is to be included in the BTAM module, where:

NO
specifies that this logic is to be excluded from the assembled BTAM module.

YES
specifies that this logic is to be

included in the assembled BTAM module. This option requires that BSCS=YES also be coded in BTMOD.

BSCTEST
: specifies whether the logic for the BSC on-line terminal test is to be included in the BTAM module, where:

NO
: specifies that this logic is not to be included in the assembled BTAM module.

YES
: specifies that this logic is to be included in the assembled BTAM module. This option requires that BSCS=YES also be coded in BTMOD.

WTTA
: specifies whether the BTAM logic required to support World Trade telegraph terminals is to be included in the BTAM module, where:

NO
: specifies that this logic is not to be included in the BTAM module.

YES
: specifies that this logic is to be included in the BTAM module.

DECBEXT
: specifies whether BTMOD will include the logic to support the Extended DECB, where:

NO
: specifies that this logic is not to be included in the BTAM module and the inlist logic <u>will</u> be used.

YES
: specifies that this logic is to be included in the BTAM logic module. The user must use the DECB Extension for operations as defined in the section on the use of the DECB Extension. The inlist logic <u>will not</u> be assembled as part of the logic module.

RMSR
: specifies whether logic for RMSR is to be included in the BTAM module, where:

NO
: specifies that this logic is not to be included in the assembled BTAM module.

YES
: specifies that this logic is to be included in the assembled BTAM module. This option requires that ERLOGIC equal E.

<u>Note</u>: The RMSR operand replaces the OBRSDR operand. However, if OBRSDR=YES is specified, RMSR logic will be included.

RESETPL
: specifies whether the BTAM module is to include Reset Poll logic, where:

NO
: specifies that this logic is not to be included in the BTAM module.

YES
: specifies that this logic is to be included.

<u>BTRD (Record on Demand) Macro Instruction</u>

The BTRD macro instruction allows the user to log out all RMSR tables to disk before issuing a CLOSE macro instruction. This is to avoid losing counts.

Name	Operation	Operands
[symbol]	BTRD	{dtfbt-symbol / (r)}

dtfbt
: specifies the symbolic name of the DTFBT macro instruction whose counters are to be written to the recorder file, IJSYSRC.

<u>Return Codes</u>: The return codes for the BTRD macro instruction are shown below. For a detailed description of return codes, see <u>Posted Error Information</u> in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error occurred.
04	RMSR file not ready or RF=NO.
08	RMSR=NO or OBRSDR=NO was coded in BTMOD.

<u>CHGNTRY (Change Terminal Entry) Macro Instruction</u>

The CHGNTRY macro instruction (Figure 5) provides the means for skipping (deactivating) or reactivating a terminal entry in a list. The user must not issue CHGNTRY for SSALST, SSAWLST, AUTOLST, or AUTOWLST while the list is being polled via READ Initial (see Return Codes). This macro has no effect on an OPENLST used to address BSC multipoint terminals.

list
: specifies the symbolic name of the DFTRMLST macro instruction that defined the terminal list containing the entry to be skipped or activated. If (r) [any register 2-12] is coded, the user must have loaded the address

Name	Operation	Operands
[symbol]	CHGNTRY	{list-symbol | (r)}, listtype-{OPENLST | WRAPLST | DIALST | AUTOLST | AUTOWLST | SSALST | SSAWLST} {position-absexp | (r)},{numchars-absexp | (r)},action-{SKIP | ACTIVATE}

Figure 5. Format of CHGNTRY Macro Instruction

of the terminal list into the register specified.

listtype
: specifies the type of list specified in the DFTRMLST macro that defined the terminal list. The CHGNTRY macro is not used for the type, IDLST, so IDLST would never be coded here.

position
: specifies the position of the entry in the specified terminal list. The first entry in the list is in position 0, the second is in position 1, and so forth. The maximum position that can be specified is a function of the <u>listype</u>: OPENLST, WRAPLST, DIALST - 31; AUTOLST, AUTOWLST - 97; SSALST, SSAWLST - 98.

numchars
: specifies the number of polling or addressing characters in the specified entry. (The number does not include the control byte provided by BTAM for each entry - see Appendix A.) For SSALST and SSAWLST, this operand is omitted and only the comma is coded.

action
: specifies the operation to be performed on the specified entry, where:

SKIP
: indicates that the entry is to be skipped when polling or addressing with the list. This means the skip (S) bit is turned ON in the control byte for the entry in the case of OPENLST, WRAPLST, and DIALST. For SSALST and SSAWLST (start-stop Auto Poll open list and wraparound list) and for AUTOLST and AUTOWLST (binary synchronous Auto Poll lists), the list is physically restructured, with the entry to be skipped extracted from the list and saved at the end of the list. The value in the index byte is unchanged.

ACTIVATE
: indicates that the entry is to be reactivated. This is done by turning OFF the skip (S) bit in the control byte for the entry in the case of OPENLST, WRAPLST, and DIALST.

For SSALST, SSAWLST, AUTOLST, and AUTOWLST, the list is physically restructured, with the entry to be activated extracted from the previously-skipped entries saved at the end of the list and merged into the active entries by index order. The value in the index byte is unchanged.

<u>Return Codes</u>: The return codes for the CHGNTRY macro instruction (with SSALST, SSAWLST, AUTOLST, and AUTOWLST only) are shown below. For a detailed description of return codes, see <u>Posted Error Information</u> in the "Assembly Considerations" section.

Hex Code	Meaning
00	Action was performed or was not needed.
04	Action was not performed because CHGNTRY was issued for a terminal list that is being polled.
08	Action was not performed because the position operand is too large.

<u>CHGNTRY (for the local 2260 and local 3270) Macro Instruction</u>: Shown below is the special form of CHGNTRY for the local 2260 and the local 3270.

Name	Operation	Operands
[symbol]	CHGNTRY	{dtfbt-symbol | (r)},ATTLST, {position-absexp | (r)} action-{SKIP | ACTIVATE}

dtfbt
specifies the symbolic address of the DTFBT defining the line group.

ATTLST
specifies that the line control blocks (LCBs) of the DTFBT are to be treated logically as an "attention list."

position
specifies the relative position (0-198) within the DTFBT of the LCB for the display station to be skipped or activated on a READ Initial or READ Short operation.

action
specifies the action to be performed:

SKIP
denotes that BTAM is to ignore input from the device corresponding to the LCB.

ACTIVATE
denotes that BTAM is to resume processing the device associated with the LCB.

<u>CHGNTRY (for an Answering List) Macro Instruction</u>: Shown below is the special form of CHGNTRY for an answering list of the SWLST form.

Name	Operation	Operands
[symbol]	CHGNTRY	{list-symbol / (r)},SWLST, {position-absexp / (r)}, {ACTIVATE / DISC / POST}

list
specifies the symbolic name of the DFTRMLST macro instruction that defined the answering list containing the entry to be changed.

SWLST
specifies that the list to be changed is of the SWLST form.

position
specifies the relative position of the entry in the list. For the first entry, code a 0; for the second, code a 1; etc.

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

DISC
sets the control byte to X'01'. It specifies that BTAM is to send the disconnect signal (DLE EOT) and the 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.

POST
sets the control byte to X'02'. It specifies that BTAM is to post the Read Connect operation complete. The user program must then take the appropriate action. The control byte that is set (X'02') is for use only when ENQ alone is defined as an entry in the SWLST terminal list. This control byte is only useful in communication with terminals (2770 or 2780) that do not have the ID verification feature. The user program can issue a READ Continue macro following the READ Connect macro to communicate with the remote terminal without allowing BTAM to transmit the ID ACK-0 sequence that is defined in the SWLST terminal list referred to.

<u>CLOSE Macro Instruction</u>

The CLOSE macro instruction is used with the OPEN macro instruction to control the operative status of communication lines in the system.

The CLOSE macro instruction removes communication line groups from use. If necessary, active channel programs will be halted.

Name	Operation	Operands
[symbol]	CLOSE	{dtfname-symbol / (r),}...

dtfname
specifies the symbolic name of the DTFBT macro instructions associated with the line group to be closed. Several line groups may be closed with a single CLOSE macro by entering their DTFBT names as operands. In addition, both symbols and register notation may be mixed in the operand field of the macro if more than one DTFBT is specified. This macro may also be used to close other types of

Name	Operation	Operands
[symbol]	CONTROL	$\left\{\begin{matrix}\text{decb-symbol}\\ \text{(r)}\\ \text{(1)}\end{matrix}\right\}\text{,optype-}\left\{\begin{matrix}\text{TI}\\ \text{TD}\\ \text{TM}\end{matrix}\right\}\text{,}$ $\left[\begin{matrix}\text{dtfbt-symbol}\\ \text{(r)}\end{matrix}\right]\text{,}\left[\begin{matrix}\text{area-relexp}\\ \text{(r)}\end{matrix}\right]\text{,}$ $\left[\begin{matrix}\text{length-absexp}\\ \text{(r)}\end{matrix}\right]\text{,}\left[\begin{matrix}\text{entry-relexp}\\ \text{(r)}\end{matrix}\right]\text{,}$ $\left[\begin{matrix}\text{rln-absexp}\\ \text{(r)}\end{matrix}\right]\left[\text{,MODE=}\left\{\begin{matrix}0\\ 1\\ \text{(r)}\end{matrix}\right\}\right]\left[\text{,MF=}\left\{\begin{matrix}\text{L}\\ \text{E}\end{matrix}\right\}\right]$

Figure 6. Format of the CONTROL Macro Instruction

files. As many as 16 files may be specified in a single CLOSE macro instruction.

CONTROL Macro Instruction

The CONTROL macro instruction (Figure 6) is used for the 7770 to:

1. Enable a line to receive a call from a remote device. When connection is established, the operation is posted complete.

2. Disconnect a line after the transmission is completed.

   For binary synchronous communication, the CONTROL macro can be used for function 2 (above) for switched connections and to:

3. Set the checking mode of the binary synchronous adapter of the attached control unit, regardless of the line connection (switched or nonswitched).

Note: The CONTROL macro instruction is not applicable to the local or remote 3270.

decb
: specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing CONTROL only when MF=E is coded.

optype
: specifies which CONTROL function is to be performed, where:

  TI
  : specifies CONTROL Initial. The line is to be enabled to receive a call (7770 only).

  TD
  : specifies CONTROL Disable. The line is to be disconnected because the answering of a successfully-received call is ended and no more transmission is to occur. If the MODE operand is present, the function of CONTROL Mode (TM) is also to be executed.

  TM
  : specifies CONTROL Mode. This optype is used for binary synchronous communication only. The "mode byte" in the DECB is to be posted and the mode of the control unit adapter is to be set.

  Note: A CONTROL TM need be issued only if the checking mode is to be changed from that specified via the MODELST operand in the DTFBT.

dtfbt
: specifies the address of the DTFBT table for the line group. Register notation permits the user to load the specified register with the address of the DTFBT prior to execution.

area, length, entry
: these operands need only be used when the CONTROL macro instruction is used to create a DECB that will be subsequently used for READ or WRITE operations. Execution of the CONTROL macro does not involve the use of these operands, however, because no transfer of data is performed.

rln
: specifies the relative line number (0-198) within the line group. Default: X'FF'.

MODE
: specifies how the error information byte mode for the binary synchronous adapter or the attached control unit

is to be set. The EIB indicates the presence of either a data check or an overrun error (or no error at all). The adapter of the attached control unit is always set to perform intermediate block checking (the 2701 or 2703 always recognizes the US character as providing the end-of-intermediate block function). This operand must not be used for start-stop transmission.

0
: specifies that the non-error information byte mode is to be set. On a receive operation the 2701 or 2703 does not send an error information byte (EIB) into main storage at any time when the 2701 or 2703 is in the non-error information byte mode. If the MODE operand is omitted in the CONTROL TM form, 0 is assumed. An example of the format of a message that has been read into main storage when the binary synchronous adapter is in non-error information byte mode is illustrated below.

```
S                      E
T --text-- U --text-- T
X          S          B
```

1
: specifies that the error information byte mode is to be set. On a receive operation, the 2701 or 2703 sends an error information byte into main storage following each US, ETB, or ETX received from the line (see General Information in the 2780 Device Dependent Considerations section for the format of the EIB). An example of the format of a message that has been read into main storage when the binary synchronous adapter is in error information byte mode is illustrated below.

```
S          E           EE
T --text-- UI --text-- TI
X          SB          BB
```

BTAM support for this mode does not include analysis of E.I.B.s and precludes use of DLE US on transparent write operations. BTAM provides a READ Repeat with Leading Graphics for the user who desires to analyze EIBs and request only partial retransmission (beginning with the first block in error) of a message from a remote CPU. Through this macro the user can send leading graphics with NAK identifying the first block received in error. When this macro is used, BTAM ERP will not retry data check and overrun errors on a received message (received on the Read command in the READ TPL channel program). Similarly, when leading graphics with NAK are received (via a conversational WRITE), BTAM ERP will not automatically retransmit the previously-sent message. Instead, the user is given control. In all other cases, BTAM performs error correction through automatic total retransmission. For the 2780, partial retransmission is not possible, since EIB is machine-implemented in a different manner.

(r)
: 0 or 1 is loaded in the low-order byte of the register.

<u>Note</u>: The MODE keyword operand can be used with optypes TD and TM. If TD is specified, the line will be disconnected (disabled). The mode isthen set, only if the MODE operand is coded. Whenever the MODE operand is coded, the "mode byte" in the DECB is updated. Thus, although the MODE operand is required for the TD <u>optype</u> if the mode is to be set, it may be omitted when TM is specified, provided the "mode byte" in the DECB was initialized by a previously-issued CONTROL.

MF
: in addition to the operations performed as explained in the Creation of a DECB section, the coding of this operand results in the following actions:

MF=L
: if the MODE operand is present, the mode byte in the DECB is initialized. Register notation is not used.

MF=E
: if optype TD or TM is specified and the MODE operand is present, the code (or low-order byte in the register) is used to set the mode. If optype TM is specified and the MODE operand is not present, the previously-posted mode byte in the DECB is used to set the mode.

No MF
: the MODE operand must be present if the mode is to be set.

<u>Return Codes</u>: The return codes for a CONTROL macro instruction are shown below. For a detailed description of return codes, see <u>Posted Error Information</u> in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	Busy.
08	Invalid relative line number.

Name	Operation	Operands
symbol	DFTRMLST	listype- { OPENLST, WRAPLST, DIALST, IDLST, AUTOLST, AUTOWLST, SSALST, SSAWLST, WTTALST, SWLST } ,device dependent operands

Figure 7. Format of the DFTRMLST Macro Instruction

0C Invalid type code.
14 Line error at OPEN.
18 No buffers.
1C No buffer pool.
20 No buffer management.
24 BSC usage count exceeded limit

DFTRMLST (Define Terminal List) Macro Instruction

The DFTRMLST macro instruction (Figure 7) provides the facility for defining terminal lists.

symbol
: represents the symbolic name of the list. This symbol is mandatory because the name of the list is required in the various READ and WRITE Initials issued in the user's program.

OPENLST
: specifies that the list to be constructed is an open polling list or an addressing list for a nonswitched line.

WRAPLST
: specifies that the list to be constructed is a wraparound polling list for a nonswitched line.

DIALST
: specifies that the list to be constructed is for a line connection over a switched network.

IDLST
: specifies that the list to be constructed is for a line connection over a switched network requiring identification (ID) verification.

AUTOLST
: specifies that the list to be constructed is an open polling list used with the Auto Poll feature for binary synchronous communication.

AUTOWLST
: specifies that the list to be constructed is a wraparound polling list used with the Auto Poll feature for binary synchronous communication.

SSALST
: specifies that the list to be constructed is for an open start-stop Auto Poll list.

SSAWLST
: specifies that the list to be constructed is for a wraparound start-stop Auto Poll list.

WTTALST
: specifies that the list is to be a terminal answering list. This type of list is used with the World Trade telegraph terminals.

SWLST
: specifies that the list to be constructed is for a switched line with the expanded ID verification facility. Only SWLST may be coded when defining a list to be used by a READ Connect or WRITE Connect macro instruction.

device dependent operands
: each list type requires different operands following the listype operand. These operands also can vary with the specific device involved. The Device Dependent Considerations section presents a detailed discussion of usage of these operands by specific device. These operands provide control information to establish logical connection with the terminal.

Note: A maximum of 15 Read operations can use one Auto Poll list because the Usage Count in the list keeps track of only 15 operations for CHGNTRY purposes. This applies to the following lists: SSALST and SSAWLST. A maximum of six Read operations can use one BSC Auto Poll list because BTAM keeps track of only six operations for

```
Name    | Operation | Operands
--------+-----------+------------------------------------------------------------
        |           |                                     [ SWITCH=NO, ]
symbol  | DTFBT     | LINELST=([nnn-integer,]...),        [SWITCH=YES, ]
        |           |
        |           | CU=code,DEVICE=code,
        |           |
        |           | [FEATURE=(device dependent sublist),]
        |           |
        |           | [BUFCB=symbol,[BUFNO=absexp,BUFL=absexp,]]
        |           |
        |           | [SEPASMB=NO, ]    [MODNAME=IJLBTM,]
        |           | [SEPASMB=YES,]    [MODNAME=symbol,]
        |           |
        |           | [MSGL=absexp,]    [TERMTST=NO, ]
        |           |                   [TERMTST=YES,]
        |           |
        |           | [ERROPT=E, ]
        |           | [ERROPT=N, ]
        |           | [ERROPT=R, ] [LERBADR=symbol,] [MODELST=(sublist),]
        |           | [ERROPT=W, ]
        |           | [ERROPT=RW,]
        |           |
        |           | [         {EBCDIC} ]  [RETRY=7,     ]
        |           | [CTLCHAR= {ASCII }, ]  [RETRY=absexp,] [CONFIG=PPT,]
        |           | [         {TRNSCD} ]                  [CONFIG=MPT,]
        |           |
        |           | [MONDLY=15,    ] [EOM=WRU, ] [EOT=2EOM, ]   [LCBNUM=nnn]
        |           | [MONDLY=absexp,] [EOM=code,] [EOT=code, ]
```

Figure 8. Format of the DTFBT Macro Instruction

CHGNTRY purposes. This applies to the AUTOLST and AUTOWLST terminal lists for BSC.

__DTFBT (Define the File for BTAM) Macro Instruction__

The DTFBT macro instruction (Figure 8) permits the user to define a line group and, if necessary, associated buffers through the keyword operands of the macro. For locally attached devices, it is used to define a control unit configuration (that is, the terminals attached to a single control unit such as a 2848 Display Control or a 3272 Control Unit).

A line group is a group of communications lines that meet the following operational requirements:

1. All lines in the group are attached to the channel through the same type of telecommunications control unit.

   __Example__: 2701

2. Line connection between the control unit and the remote devices is of the same type.

   __Example__: switched network

3. All devices within the same line group have the same line control procedures, transmission code, and operating characteristics.

4. All lines within a group share the same buffer pool if one is defined.

LINELST
specifies via a sublist the correspondence between symbolic unit (SYSnnn) and relative line number. In the sublist the user codes one 3-digit number for each line in the line group. The 3-digit number is interpreted as the 'nnn' of SYSnnn. The order of coding the 3-digit numbers determines which symbolic units are associated with the individual lines in the line group.

__Example__: LINELST=(005,010,007) This results in associating:

SYS005 with relative line number 0,
SYS010 with relative line number 1, and
SYS007 with relative line number 2

in a line group comprising 3 lines.

__Note 1__: The LINELST operand has a different use for the local 2260 and local 3270. With these devices, each DTFBT macro instruction defines a control unit and each entry in the LINELST sublist defines a terminal attached to the control unit. In the above example, 005, 010, and 007 would be different devices attached to the same control unit.

Note 2: The maximum number of suboperands coded in the sublist is limited by the assembler type used and is not checked by BTAM. For instance, Assembler D allows only 31 suboperands.

SWITCH
specifies the type of line connection between the system and the remote device, where:

NO
specifies that the line connection is permanent.

YES
specifies that the line connection is through a switched network.

CU
defines the control unit attached to the channel as a 2701, 2702, 2703, 7770, 2848, or 3272. If using the System/370 with the Integrated Communications Attachment, 2703 must be specified. If a local 3270 display system is being defined, CU=3272 must be specified; if a remote 3270 is being defined, the transmission control unit (CU=2701 or CU=2703) must be specified.

Example: CU=2701

DEVICE
defines the device as a 1030, 1050, 1060, 1130, 2020 (System/360 Model 20 or 2922-1), 2260, 2740, 2780, 2848, 2972, 3277, S360 (System/370), 83B3, 115A, TW33, TW35, WTTA, BSC1, BSC2, or BSC3.

Notes:
DEVICE=2260 applies to the 2260 Local.

DEVICE=2848 applies to the 2260 Remote.

DEVICE=S360 applies to a remote System/370. The user may, however, include point-to-point lines to the 2780 or to the 1130 in the line group, provided the other operands in DTFBT are consistent in their application to these lines and provided that he does not attempt to use READ/WRITE options which are not applicable to the 2780 or to the 1130 but which are available for System/370 lines.

DEVICE=2780 applies to a remote 2780 on a point-to-point or multipoint line.

DEVICE=1130 applies to a remote 1130 on a point-to-point or multipoint line.

DEVICE=2020 applies to a remote System/360 Model 20 on a point-to-point or multipoint line or to a 2922-1.

DEVICE=WTTA applies to World Trade telegraph terminals.

DEVICE=2972 applies to a remote 2972 on a multipoint line.

DEVICE=3277 applies to the local 3270 display system (DEVICE=BSC3 is used in defining a remote 3270 display system).

DEVICE=BSC1 applies to any remote BSC station on a point-to-point nonswitched line. When this operand is coded, the user may omit the CONFIG operand of DTFBT because CONFIG=PPT is assumed. The FEATURE operand code BSC may also be omitted because it is assumed when DEVICE=BSC1.

DEVICE=BSC2 applies to any remote BSC station on a point-to-point switched line. When this operand is coded, the user may omit both the CONFIG and SWITCH operands of DTFBT because CONFIG=PPT and SWITCH=YES are assumed. The FEATURE operand code BSC may also be omitted because it is assumed when DEVICE=BSC2. However, the appropriate FEATURE operand codes (SIX, SXW, SIW, RIX, RXW, or RIW) must be coded if BSC ID verification is to be performed using the IDLST type of terminal list.

DEVICE=BSC3 applies to any remote BSC stations on a multipoint line. When this operand is coded, the user may omit the CONFIG operand because CONFIG=MPT is assumed. He may also omit the FEATURE operand code BSC because it is assumed when DEVICE=BSC3.

Note: BSC1, BSC2, and BSC3 do not apply when using 2780 Transcode.

This operand must be omitted for the 7770.

FEATURE
specifies, via a sublist, various device-dependent machine and programming special features. The features are specified by coding in the sublist the following appropriate codes:

STC
: specifies that the 2740 is equipped with the station control feature.

CHK
: specifies that the 2740 is equipped with the checking feature.

TRC
: specifies that the 2740 is equipped with the transmit control feature.

BSC
: must be coded when the DEVICE operand specifies a binary synchronous device except when the DEVICE operand is BSC1, BSC2, or BSC3.

KBL
: specifies that the 2848 control unit is equipped with the data entry feature.

  Note: The Lock option optypes are rejected as undefined when the FEATURE keyword operand is not specified as KBL.

OIU
: specifies that a 2760 Optical Image Unit is attached to an IBM 2740 with the checking feature and (optionally) the dial feature.

SIX
SXW
SIW
: if ID verification for a calling operation is to occur in a point-to-point dial system, one of these three codes must appear in the sublist. A more detailed explanation of these codes is given in the ID Verification Procedures section under System/370 to System/370 Dial.

RIX
RXW
RIW
: if ID Verification for an answering operation is to occur in a point-to-point dial system, one of these three codes must appear in the sublist. A more detailed explanation of these codes is given in the ID Verification Procedures section under System/370 to System/370 Dial.

MAS
SLV
: either MAS (Master) or SLV (Slave) may be coded in the sublist to specify whether the CPU is to be Master or Slave when contention occurs in the CPU-to-CPU Contention system (private line). If neither is coded, MAS is assumed for this system. (See the Device Dependent Considerations sections.)

  Note: If MAS is specified, the remote device is to be the slave when contention occurs. If SLV is specified, the remote device is the master. When the remote device is the 2780, SLV must always be coded. The CPU must always be the slave when contention occurs between the CPU and the remote 2780.

APL
: specifies that the Auto Poll feature is to be employed for the start-stop devices 1030, 1050 (nonswitched), 1060, 2740S, or 2740SC. If omitted, the generated channel programs for these devices will perform the standard programmed polling. When the Auto Poll feature is specified for the 2740, the station control feature must always be present but the checking feature is optional. In the example below, the FEATURE keyword operand indicates the coding for a 2740 when the Auto Poll feature is coded. When the Auto Poll feature is coded for any of the supported devices, only CU=2702 or CU=2703 may be coded. If CU=2702 is coded along with APL in the FEATURE sublist, the 2702 must be equipped with the Auto Poll feature. If the 2702 is not equipped with the Auto Poll feature, a command reject error is received when an attempt to use Auto Poll is made. If CU=2701 is coded with APL, a DTFBT MNOTE is given at assembly time. The Auto Poll feature is standard in the 2703. The switch operand must be coded SWITCH=NO.

IAM
: specifies that a World Trade telegraph terminal can ask for the computer identification by sending FIGS D (refer to the Device Dependent Considerations sections).

WRU
: specifies that both a World Trade telegraph terminal and the computer can request each other's identification by sending FIGS D. When WRU is specified, IAM is assumed. If neither IAM nor WRU is specified, no exchange of identification can be performed (refer to the Device Dependent Considerations sections).

MON
: specifies that a World Trade telegraph terminal is equipped with the Motor-On optional feature (refer to the Device Dependent Considerations sections).

Note: There is no special ordering of these codes in the sublist. Commas must be used to separate the different codes, but are not used to indicate missing ones.

Example: FEATURE=(APL,STC[,CHK])

BUFCB
: assigns a symbolic name to the buffer control block of the buffer pool used by the line group. Two or more line groups can share the same buffer pool if the DTFBT's for each specify the same name in their BUFCB operands.

BUFNO
: specifies the number of buffers to be provided in the buffer pool. Range: 2-255.

BUFL
: specifies the total length in bytes of each buffer in the pool. The length must be 12 bytes greater than the number of bytes required for data. Range: 16-32,760. For a remote 3270, 22-32,760.

Note: If the keyword operands BUFNO, BUFL, and BUFCB are omitted, no buffer pool is created. However, BUFNO and BUFL can be omitted while BUFCB is coded, if BUFCB specifies the symbolic address of a buffer pool defined by all three keywords in a previous DTFBT in the same assembly.

SEPASMB
: defines how this DTFBT will be assembled, where:

NO
: specifies that this DTFBT will be assembled with the rest of the user's code (see the Assembly Considerations section).

YES
: specifies that this DTFBT will be assembled separately. In this case a CATALR card with the name given to the DTFBT is generated ahead of the object deck (see the Assembly Considerations section).

MODNAME
: specifies the name of the BTAM logic module (BTMOD) that is to be link-edited with this DTFBT section. If MODNAME is omitted, the standard BTAM module name (IJLBTM) is used.

MSGL
: this operand is used only for the 7770 to specify the maximum input message length for any line in the line group when a WRITE invitational or WRITE conversational is issued. Range: 2-32,767. The default option is 2.

TERMTST
: specifies whether the DTFBT table will contain the test activation code, where:

NO
: specifies that the DTFBT will not contain the test activation code; therefore, no request for the on-line terminal test facility is made.

YES
: specifies that the DTFBT will contain the test activation code. This is used by the on-line terminal test facility in determining if a test request has been made. YES may be specified only for IBM terminals except for the local 2260 and 7770.

ERROPT
: specifies which ERP options (for non-binary synchronous transmission only) are to be performed, where:

E
: specifies that error recovery is to be performed without text retry.

N
: specifies that no error recovery is to be performed.

R
: specifies that error recovery is to be performed with read text retry.

W
: specifies that error recovery is to be performed with write text retry.

RW
: specifies that error recovery is to be performed with read text retry and write text retry.

Note: This operand does not apply to the 7770 Audio Response Unit or World Trade telegraph terminals.

LERBADR
: specifies the symbolic address of the Line Error Block (LERB) or the RMSR Table (RMSTAB) associated with the line group.

MODELST
: a sublist of codes is specified by the problem program (for use by BTAM at OPEN time) to define the following for each line in the line group (BSC only):

E.I.B. mode - Non-error information byte mode (non-EIB) or error information byte mode (EIB) may be selected. Only in EIB mode is the error information byte sent into main storage following a received Unit Separator (US) character, an ETB character, or an ETX character. The user must make the I/O area large enough to include the EIB characters.

Dual Communication Interface - Interface A or Interface B may be selected on the 2701 equipped with this feature.

Dual Code Feature - Code A or Code B may be selected on 2701 equipped with the Dual Code Feature.

The codes are:

Code	Meaning Mode	Interface	Code
0	non-EIB	A	A
1	non-EIB	A	B
2	non-EIB	B	A
3	non-EIB	B	B
4	EIB	A	A
5	EIB	A	B
6	EIB	B	A
7	EIB	B	B

For the 2703 or for the 2701 not equipped with the Dual Communication or Dual Code Features, a code should be used that specifies Interface A and Code A. If the MODELST operand is omitted, 0 is assumed for each line in the line group. If the MODELST operand is present, each line in the line group must be represented by a code unless 0 is implied, in which case only the trailing comma must be coded.

Example: If the line group is comprised of 5 lines, the user may code

```
MODELST=(0,4,,,) or
MODELST=(,2,,2,) etc.
```

A useful rule to remember is that the number of commas in the MODELST sublist must be equal to the number of commas in the LINELST sublist. The position of each code in the MODELST sublist corresponds to the relative line number of the line for which the code applies.

CTLCHAR
specifies the transmission code used in binary synchronous communication for the line group. For start-stop devices, this operand is not used and is ignored. It is mandatory for binary synchronous communication, since BTAM uses this operand to initialize a table of line control characters required for internal use by BTAM. The following may be coded on the right side of the equal sign.

EBCDIC
specifies that the EBCDIC line control characters are to be used.

ASCII
specifies that the USASCII line control characters are to be used.

TRNSCD
specifies that the 6-bit TRANSCODE line control characters are to be used (available only on 2780).

RETRY
specifies the number of retries by BTAM ERP for recoverable errors that occur on I/O operations for binary synchronous communication. The value of the absolute expression specified on the right side of the equal sign may range from 0 through 15. The operand may be omitted, in which case seven is assumed for binary synchronous communication.

Note 1: For BSC-programmed devices on a multipoint line, if RETRY=0 is coded and transmission is ended and the same device is immediately repolled or reselected, a possible time-out condition could exist. Therefore, it is recommended that a retry count of zero be avoided for a multipoint network.

Note 2: This operand is not applicable to non-binary synchronous devices.

CONFIG
specifies the line configuration, where:

PPT
specifies that the data link between the CPU and the remote binary synchronous device is point-to-point.

MPT
specifies that the data link between the CPU and the remote binary synchronous devices is multipoint.

This operand is not applicable to any BTAM-supported start-stop or local devices.

MONDLY
specifies the number of Mark characters to be sent to a World Trade telegraph terminal when this terminal is not equipped with the Motor-On optional feature.

MONDLY=10
corresponds to a 50-baud service, MONDLY=15 to a 75-baud service, and MONDLY=20 to a 100-baud service. When

MONDLY is omitted or exceeds 20, MONDLY=15 is assumed.

EOM
: identifies the EOM (end of message) signal recognized by a World Trade telegraph terminal.

EOM=WRU
: indicates that the WRU signal (FIGS D) is used to separate incoming messages.

EOM=X'hh' (where "hh" is the hexadecimal representation of FIGS x)
: is used only when FIGS x is not set in the WTTA as FIGS D.

EOM=X'hhlF' (where "hh" is the hexadecimal representation of FIGS y set in the WTTA)
: indicates that the FIGS y LTRS termination is used as end of message.

EOT
: identifies the EOT (end of tramsmission) signal recognized by a World Trade telegraph terminal.

EOT=2EOM
: indicates that two consecutive EOM signals are defined by the user as end of transmission.

EOT=X'hhlF'
: specifies that the FIGS y LTRS termination is used as end of transmission. Therefore, EOM=X'hhlF' cannot be used as an end of message signal.

Note: In the above descriptions of the EOM and EOT operands, "x" and "y" are the values assigned by the WTC customer.

The following rules must be observed (see the Assembly Considerations section):

1. All DTFBT statements that are to share a buffer pool must be in the same assembly.

2. If SEPASMB=YES is specified in a DTFBT statement, normally there will be no other source code (except END) in the same assembly.

Note: The user must define the line group with a DTFBT macro instruction and must activate the line group with an OPEN macro before any message can be sent or received over a communication line of the line group

LCBNUM
: specifies the number of LCBs wanted in one DTFBT csect. As many as 199 LCBs can be generated in one DTFBT csect with this operand, (Range 001-199). The LCBNUM parameter is coded on the first DTFBT only, specifying the number of LCBs wanted. The symbols field is left blank and only the LINELST and MODELST (if BSC) are coded for every continuing DTFBT. MODE will default to zero if not coded.

Example, Start-Stop:

```
Name DTFBT   LINELST=(001,002,,,,031),
             CU=2701,DEVICE=1050,,,,,
             LCBNUM=199
     DTFBT   LINELST=(032,033,,,,062)
         .
         .
         .
     DTFBT   LINELST=(187,188,,,,199)
```

Example, BSC:

```
Name DTFBT   LINELST=(001,002,,,,031),
             CU=2701,DEVICE=BSC1,
             MODELST=(1,,4),LCBNUM=199
     DTFBT   LINELST=(032,033,,,,062),
             MODELST=(1,2,,,,7)
         .
         .
         .
     DTFBT   LINELST=(187,188,,,,199),
             MODELST=(7,6,,,,0)
```

LERB (Line Error Block) Macro Instruction

The LERB macro instruction generates and initializes a table called the line error block, in which the Error Counts logic in BTMOD keeps by line the data check, intervention required, and non-text time-out error counts and the number of transmissions requested by the user (Figure 9). If any of the error counters reach their threshold values before the transmission count threshold is reached, a message is printed on the System/370 Operator's Console. The message identifies the line and contains the three error counters, the transmission counter, and their respective threshold values.

Bytes				
0	Transmission Cumulative Counter			
4	Data Check Cumulative Counter		Intervention Required Cumulative Counter	
8	Time Out Cumulative Counter		Transmissions Counter	Data Check Counter
12	Intervention Required Counter	Time Out Counter	Transmission Threshold	Data Check Threshold
16	Intervention Required Threshold	Time Out Threshold	Reserved	

Figure 9. Line Error Block

After the message is printed, the transmission counter and the three error counters are added to a separate set of four cumulative counters, after which the counters are reset. Also, whenever the transmission counter reaches its threshold without any of the error counters reaching their thresholds, the four counters are added to the four cumulative counters, and the counters are reset.

Note 1: Because the LERBADR operand of DTFBT generates a V-type address constant, the LERB macro may not appear in an unnamed control section.

Note 2: LERB counts are not supported for the local 2260 and local 3270. RMSR processing should be used instead.

Name	Operation	Operands
symbol	LERB	nlines-absexp, [{([transmct-absexp], [datack-absexp], [intreq-absexp], [nontto-absexp]),}...]

nlines
    specifies the number of lines associated with this line group (DTFBT). From 1 to 199 lines are allowed. This operand must be coded.

transmct
    specifies the number of consecutive transmission requests on a line that, when reached, causes the three error counters and the transmission counter to be added to their respective accumulators and the counters to be reset to zero. From 1 to 255 transmissions are allowed; if this operand is omitted, a 255 transmission count is assumed.

datack
    specifies the threshold value for the number of data check errors for a specified number of transmissions on a line. A value of 1 to 255 is allowed, except that the value cannot be greater than the transmission count (transmct). The default value is either 10 or the "transmct" value, whichever is less.

intreq
    specifies the threshold value for the number of intervention-required errors for a specified number of transmissions on a line. A value of 1 through 255 is allowed, except that the value cannot begreater than the "transmct". The default value is either 5 or the "transmct" value, whichever is less.

nontto
    specifies the threshold value for the number of non-text time-out errors for a specified number of transmissions on a line. A value of 1 through 255 is allowed, except that the value cannot be greater than the "transmct". The default value is either 5 or the "transmct", whichever is less.

Only the first operand "nlines" is mandatory. If the first operand is the only one coded, the default threshold values of 255, 10, 5, and 5 will be supplied for all the lines specified by "nlines". If the threshold value is coded, it remains effective for any of the following lines for which a threshold value is not coded. It will remain effective until another set of threshold values is provided.

An example of the LERB macro instruction is:

```
BTAMLERB    LERB 10,,,(200,20,,7),,,
                 (240,20,25,10)
```

This coding will result in lines 0 and 1 having default thresholds of 255 "transmct", 10 "datack", 5 "intreq", and 5 "nontto"; lines 2, 3, and 4 having thresholds of (200,20,5,7) with 5 as the default value of "intreq"; and lines 5 through 9 having thresholds of (240,20,25, 10). The parentheses are omitted if all the suboperands are omitted. The trailing commas are also omitted.

Note: A maximum of 99 LERB tables can be generated when using Assembler D. A maximum of 199 LERB tables can be generated when using Assembler F.

LERPRT (Line Error Print) Macro Instruction

The LERPRT macro instruction provides the ability to print the current values of the LERB accumulators. The accumulators are incremented each time an error threshold value is reached within a specified number of requests for transmission on a line or each time the LERB transmission counter reaches its threshold value without an error counter having reached its threshold value. When the macro instruction is issued, the current values of the error counters and the transmission counter are added to the set of four accumulators and the counters are cleared. After printing takes place, the accumulators are optionally cleared.

Name	Operation	Operands
[symbol]	LERPRT	{dtfbt-symbol | (r) | (1)} [,rln-absexp | ,(r) | ,(0)] [,CLEAR=YES | ,CLEAR=NO]

dtfbt
specifies the symbolic address of the DTFBT, which contains the address of the line error block to be processed.

rln
specifies the relative line number to indicate the entry in the line error block to be printed (0-198 allowed, but must be less than "nlines" in the LERB macro instruction). If this operand is omitted, all nonzero entries in the line error block are printed.

CLEAR
NO
specifies that the accumulators will not be reset to zero after the execution of the LERPRT macro.

YES
specifies that the accumulators will be reset to zero after the execution of this macro. If the operand is omitted, YES is assumed.

LOPEN Macro Instruction

The LOPEN macro is used to condition a line on which an error occurred during OPEN, and, when necessary, initiates a line delay.

Name	Operation	Operand
[symbol]	LOPEN	{decb-symbol | (r)}

decb
specifies the symbolic name of the DECB associated with the line to be opened. Only one line may be conditioned with a single LOPEN macro. Register notation may also appear in the operand field.

If the user issues a READ, WRITE, or CONTROL macro on a line that failed to open, a return code of X'14' is given to him in register 15. He may then issue an LOPEN macro to condition the line. If LOPEN successfully conditions the line, completion code X'7F' is posted; otherwise the completion code X'41' is posted and the line should be regarded as temporarily unusable.

Note: When the IBM 3977 Modem gets out of synchronization on a multipoint line, repeated write operations will result in transmission errors (a NAK response from the receiving station or time-out condition). After retrying the operation a user-specified number of times, the BTAM ERP will return control to the user's program, indicating the type of error. The user's program should use the LOPEN macro instruction if the 3977 is used on the line. This procedure allows the modem to resynchronize and the stations should be able to continue communication. However, the user must issue an initial operation to repoll or reselect the terminal.

ONLTST (On-Line Test) Macro Instruction

The ONLTST macro is used to generate linkage to the on-line test logic in BTMOD. This macro is issued whenever the user wishes to have BTAM initiate binary synchronous on-line terminal test. BTAM initiates on-line test by transmitting a Request-for-Test (RFT) message requesting the remote unit to perform the terminal test that is specified by the parameters of the ONLTST macro. The ONLTST macro can be issued only in communication between System/370 and System/370, or in any type of binary synchronous communication when the X operand of the macro is zero. See also the section on BSC On-Line Test.

Note: If the terminal is a remote 2715 or a local or remote 3270, this macro is not applicable; the 2715 will not accept an RFT message.

Name	Operation	Operands
[symbol]	ONLTST	{DECB=symbol | (r) | (1)} {X=absexp | (r)}, {Y=absexp | (r)}, [,DTFBT=symbol | (r)] {AREA=relexp | (r)}, [TEXT=relexp, | (r) LENGTH=absexp | (r)] [,ENTRY=relexp | (r)] [,RLN=absexp | (r)]

DECB
: specifies the address of the DECB associated with the line on which the test is to be performed.

X
: specifies the type of online test to be requested by the online test module. This value is used in the X field of the RFT message generated by the test module. Permissible values of X and their meanings are shown below. When specified as ONLTST macro operands, these values of X which are greater than 0 are only applicable in System/370-to-System/370 communication. For each case where the value of X is greater than zero, the requested message is returned Y times.

X Value	Action Requested
00	Acknowledgment of the test message Y times
01	Text received in the RFT transmitted Y times
02	Transparent EBCDIC message
03	Transparent USASCII message
04	Normal EBCDIC message
05	Normal USASCII message
06	Alphameric USASCII message
07	USASCII printer message
08	USASCII punch message
09	TRANSCODE printer message
10	TRANSCODE punch message
11	TRANSCODE multipoint message
12	EBCDIC printer message
13	EBCDIC punch message
14	EBCDIC alphameric message
15	EBCDIC weak pattern message for switched lines
16	EBCDIC weak pattern message for leased lines
17	TRANSCODE weak pattern message for switched lines
18	TRANSCODE weak pattern message for leased lines
19	EBCDIC weak pattern for DLE SYN insertion
20	EBCDIC 80-character transparent text message
21	EBCDIC 120-character transparent text message
22	EBCDIC 144-character transparent text message

Y
: specifies the number of times the requested message is to be transmitted.

DTFBT
: specifies the address of the DTFBT table for the line group.

AREA
: specifies the address of a 300-byte I/O area or an I/O area large enough to accommodate the test message. Buffers may be used, but the first buffer must be large enough to contain the on-line test message or RFT message.

TEXT,LENGTH
: the text operand specifies the address of framed text which is to be transmitted during the test. This text will be put in the RFT message only if the X operand is 1. The text must be framed by STX and ETX for normal data. Transparent data must have DLE STX as the first two characters. Ending characters for transparent data are provided by BTAM. The length operand specifies the number of bytes per test and must be coded whenever the text operand is coded. This length must not be greater than 300 bytes including control characters.

Note: Transparent data cannot be used in applications where the transparent operations are not defined.

ENTRY
: specifies the address of a DIALST, IDLST, or OPENLST provided by the DFTRMLST macro instruction in the problem program. This operand is coded on switched lines to provide dial digits and/or ID characters to BSC on-line test. For multipoint lines, entry specifies the address of an OPENLST that has the selection characters of the terminal to which the RFT message will be sent. In this case, X=0 is mandatory.

RLN
: specifies the relative line number (0-198) within the line group.

Note: The user must issue a WAIT macro instruction for the DECB before reusing that DECB.

Return Codes: The return codes for the ONLTST macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error occurred.
04	Busy.
08	Relative line number is too large.
0C	Invalid text code.
10	Invalid use of ONLTST macro.
14	Line error at OPEN time.

OPEN Macro Instruction

The OPEN macro prepares communication line groups for use and executes, when applicable, any required channel programs.

A line group must be opened before any message can be sent or received over any line in the group and before buffers from the associated buffer pool are obtained for that line group. In certain cases there is a line delay before transmission can occur on the line. In such cases, a message indicating this line delay is written on the Operator's Console.

For the local 2260 and local 3270, the OPEN macro instruction prepares for use the devices specified in the LINELST operand of the DTFBT. It erases the printer buffer if the device is a printer. It erases the local 2260 or 3270 display station buffer and unlocks the keyboard of a local 3270.

Name	Operation	Operands
[symbol]	OPEN	{dtfname-symbol, | (r)}...

dtfname
: specifies the symbolic name of the DTFBT macro instruction associated with the line group to be opened. Several line groups may be opened with a single OPEN macro by entering their DTFBT names as operands. If more than one DTFBT is specified, both symbols and register notation may appear in the operand field. For example, if the user wishes to open DTFBT1, DTFBT2, DTFBT3, and DTFBT4, he might code:

```
STEP1 OPEN DTFBT1,(2),(3),DTFBT4
```

In this case, register 2 would have been loaded with the address of DTFBT2 and register 3 with the address of DTFBT3 prior to issuing the OPEN macro. This macro instruction can be used to open other types of files. As many as 16 files may be specified in a single OPEN macro instruction.

READ Macro Instruction

A READ macro instruction (Figure 10) is issued by the problem program whenever transmission from a device is required and/or the user desires to reserve and initialize a DECB. Various options are permitted, not all of which apply to every device supported.

Figure 15 at the end of this section lists the READ options available for each device. In the Device Dependent Considerations sections, further information on specific implementation of the READ options is detailed for each supported device.

decb
: specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing a READ only when MF=E is used.

Note: With a local 3270, the DECB specified must be associated with the

```
┌─────────┬──────────┬──────────────────────────────────────────────────────────┐
│ Name    │ Operation│ Operands                                                 │
├─────────┼──────────┼──────────────────────────────────────────────────────────┤
│ [symbol]│ READ     │ {decb-symbol} ,optype-code,[dtfbt-symbol],               │
│         │          │ {   (r)     }              [    (r)     ]                │
│         │          │ {   (1)     }                                            │
│         │          │                                                          │
│         │          │ ([area-relexp][, auxarea-relexp]),                       │
│         │          │ ([    (r)    ][       (r)     ])                         │
│         │          │ ([    'S'    ][       'S'     ])                         │
│         │          │                                                          │
│         │          │ ([length-absexp], [auxlength-absexp]) ,                  │
│         │          │ ([     (r)     ]  [       (r)      ])                    │
│         │          │                                                          │
│         │          │ [entry-relexp], [rln-absexp],                            │
│         │          │ [    (r)     ]  [   (r)    ]                             │
│         │          │ [    'S'     ]                                           │
│         │          │                                                          │
│         │          │ [MF={L}]                                                 │
│         │          │ [   {E}]                                                 │
├─────────┴──────────┴──────────────────────────────────────────────────────────┤
│Note:  The parentheses around area,auxarea and length,auxlength should be coded│
│       only if both operands of the pair are coded.                            │
└───────────────────────────────────────────────────────────────────────────────┘
```

Figure 10. Format of READ Macro Instruction

same DTFBT in each READ macro instruction.

optype
: specifies the specific READ option (TI, TT, TIR, etc.). See Figure 15 for start-stop devices and the device-dependent section for individual BSC devices.

dtfbt
: for the line group.

area
: specifies the address of the first byte of the input area used to receive the message. If 'S' (apostrophes are mandatory) is coded, BTAM reads the message into the buffers of the buffer pool associated with the line group. For the local 2260 and local 3270, only one buffer will be supplied.

  Note: On completion of a READ operation with area 'S' specified, the problem program should check bit 6 of the Flag byte to see if there are unreleased buffers not used. If this bit is on, the last 3 bytes of the ECB contain the address of the first buffer not used by the problem program in the READ operation. The problem program should release the unused buffers to the buffer pool with a RELBUF macro.

auxarea
: if optype is TTL or TPL, this operand specifies the address of the first byte containing the leading graphics. See the BSC Channel Programs section for the Read Continue with Leading Graphics (TTL) instruction to obtain a description of using leading graphics.

  Note: An 'S' coded for auxarea in this case has no meaning.

length
: specifies the maximum number of bytes in the expected message. (This operand is used to determine the maximum of buffers needed when area is coded 'S'. If possible, unused buffers are returned to the buffer pool if the READ completes normally. If an error is posted on a READ operation using buffers, the user must return the acquired buffers by a RELBUF macro instruction before retrying the operation. See note above.)

  Note: Specialized use of this operand for World Trade telegraph terminals is discussed in the Device Dependent Considerations sections.

auxlength
: specifies the number of leading graphics to be transmitted to the remote terminal. The number may not exceed seven. This operand must be used in conjunction with the auxarea operand.

  Note: Area refers to an input field, while auxarea refers to an output field. Length refers to the number of bytes in the input field and auxlength refers to the number of bytes in the output field. For a WRITE macro this is reversed. (See the description of the DECB extension for WRITE.)

entry
: specifies for:

  1. A nonswitched line (OPENLST, WRAPLST, SSALST, or SSAWLST), the address of an entry within the terminal list.

  2. A switched line (DIALST), the address of the beginning of the terminal list at initial contact time and an entry afterwards.

  3. Binary synchronous applications (DIALST, AUTOLST, AUTOWLST, SWLST, IDLST), Audio Units, or World Trade telegraph terminals (IDLST), an appropriate address as specified for the specific remote device in the Device Dependent Considerations section. The 'S' option has a special use (see Figure 38) in READ Continue for CPU-to-CPU Dial. If relexp or (r) is coded, the address specified is placed in the "entry field" of the DECB and also in the "polling pointer" of the DECB.

  4. The local 3270 when READ Modified from Position (TMP) or READ Buffer from Position (TBP) is issued, the address of a data area containing the position in the device buffer from which the read operation is to begin. (See "IBM 3270 Information Display System (Local)" in the general section "Local Device-Dependent Considerations.")

     Note: The entry operand must be coded for all initial operations except for the 2260 and the local 3270. It is not used for the 2260 or for the local 3270 (except for READ TMP or READ TBP) and must be coded with consecutive commas.

rln
: specifies the relative line number within the line group. (Range 0-198 inclusive. Default: X'FF'.) Details

of the specialized use of this operand for the local 2260 and local 3270 are discussed in the device-dependent section. At the completion of a READ for the local 2260 and local 3270, the rln for the display station from which data has been read is placed in the low-order byte of the DECB polling pointer field.

MF
: see the "Creation of a DECB" section.

Return Codes: The return codes for a READ macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	Busy.
08	Invalid relative line number.
0C	Invalid type code.
10	All of the skip bits are on.
14	Line error at OPEN.
18	No buffers.
1C	No buffer pool.
20	No buffer management.
24	BSC usage count exceeded limit.
28	For local 3270 only. Printer busy.
2C	For local 3270 only. No SBA order in area pointed to by entry operand.
30	For local 3270 only. Buffer contents unreliable.
34	For local 3270 only. Device under OLTEP control.

RELBUF (Release Buffer) Macro Instruction

The RELBUF macro instruction permits the user to return to the buffer pool of the line group one or more buffers that are no longer required.

Name	Operation	Operands
[symbol]	RELBUF	{ dtfbt-symbol | (r) | (1) } ,(r)

dtfbt
: specifies the address of the DTFBT table in which the buffer pool is defined, via the DTFBT operand BUFCB.

(r)
: specifies the register containing the address of the first buffer in the chain of buffers to be returned to the buffer pool. All buffers to be returned to the buffer pool must be forward chained. If the user has altered the normal BTAM chaining, he must place in bytes 9 through 11 of each buffer the address of the next one in the chain. The last buffer in the chain to be returned must contain zeros in bytes 9 through 11.

Return Codes: The return codes for a RELBUF macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	Partially performed. Some, but not all, of the buffers specified have been returned to the buffer pool.
08	Not available. None of the buffers specified has been returned to the buffer pool.
0C	No pool. The DTFBT specified does not contain the address of a buffer pool.
10	No buffer management. The BTMOD macro did not specify either BUFFER=REQREL or BUFFER=YES.

REQBUF (Request Buffer) Macro Instruction

The REQBUF macro instruction permits the user to obtain buffers from the buffer pool of the line group. (See the section on Buffering for more details on buffers.)

Name	Operation	Operands
[symbol]	REQBUF	{ dtfbt-symbol | (r) | (1) } ,(r) [,count-absexp | ,(r) | ,(0)]

dtfbt
: specifies the address of the DTFBT table in which the buffer pool is defined, via the DTFBT operand BUFCB.

(r)
: specifies the register in which the address of the first buffer in the chain is to be returned by BTAM to the problem program.

count
: specifies the number (1-255) of buffers requested. If the count operand is omitted, one buffer is assigned.

Note: A buffer pool must have been previously assigned to the DTFBT and the DTFBT must be open. Buffers should be returned to the pool by the RELBUF macro instruction when the user has finished using them.

Return Codes: The return codes for a REQBUF macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	Partially performed. Some, but not all, of the buffers requested have been provided to the user.
08	Not available. None of the buffers requested has been provided to the user.
0C	No pool. The DTFBT specified does not contain the address of a buffer pool.
10	No buffer management. The BTMOD macro did not specify either BUFFER=REQREL or BUFFER=YES.

RESETPL Macro Instruction

The RESETPL macro instruction provides the problem program with the facility to:

1. Interrupt a polling sequence.
2. Terminate a Prepare command (2740, BSC, or WTTA).
3. Free a line for which an Enable command was issued but not completed (no call was received).
4. Terminate a wraparound Auto Poll sequence.
5. Free for other operations a local 2260 or local 3270 DTFBT for which a READ Initial or, for a local 2260, READ Short has been requested through the indicated DECB, but for which no response has been received.

Name	Operation	Operands
[symbol]	RESETPL	{decb-symbol / (r) / (1)} [,INLINE={NO / YES}]

decb
: specifies the address of the DECB associated with the line.

INLINE
: specifies that RESETPL in BTMOD will be used rather than transient RESETPL Logic.

Note 1: INLINE=YES must not be used with a DOS STXIT routine.

Note 2: It is suggested that a WAIT be issued after issuing a RESETPL. This will ensure that any outstanding ECBs will be properly posted.

Return Codes: The return codes for the RESETPL macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	RESETPL (Reset Poll Logic) is included in the BTMOD macro.
04	No RESETPL logic is included in the BTMOD macro.

RMSRTAB (Recovery Management Support Recording Table) Macro Instruction

The RMSRTAB macro instruction generates and initializes 20-byte tables called RMSR tables, in which RMSR logic in the BTAM logic module keeps count of transmissions and certain errors. These counts are kept optionally by line or by terminal for all nonswitched multipoint devices, and by line for all switched or leased-line devices. The error count is incremented each time any of these errors occurs: unit exception on Write; time out on Prepare and nontext Read; time out on Dial, Disable, and Enable; intervention required; overrun; bus out check on Write and Dial; bus out check at initial selection; data check on Write; data check on Read; data check on Poll; parity error for 2740 M2; VRC error for 2740 M2; electronic failure for 2740 M2; and Selectric failure for 2740 M2.

Transmissions and errors are each kept in a one-byte counter. When either counter reaches its threshold (255 for transmissions, 15 for errors), the count information is passed to the Supervisor for updating the system recorder file, IJSYSRC. All counters will be expanded into two-byte counters on the disk. The disk counters can be displayed between jobs by executing the DOS EREP program, that is, by entering // EXEC EREP. All counters will then be displayed on the printer. See Appendix H for more detailed information.

Name	Operation	Operands
symbol	RMSRTAB	nlines-absexp[,(pchars-absexp,...)]

nlines
    specifies the number of lines associated with this line group (DTFBT). From 1 to 99 lines are allowed using the D assembler and from 1 to 199 using the F assembler. This operand must be coded.

pchars
    specifies the hexadecimal representation of the polling or addressing characters for each terminal on which statistics are to be kept. Omitting this parameter indicates that statistics are to be kept by line. This operand must not be coded for switched or nonswitched point-to-point lines. The pchars entries must be identical to the DFTRMLST entries for selected terminals.

SDRTAB (Statistical Data Recorder Table) Macro Instruction

The SDRTAB macro instruction has been replaced by the RMSRTAB macro instruction; see the RMSRTAB macro instruction in this section. Note, however, that application programs presently using the SDRTAB macro instruction do not have to be rewritten; an RMSR table will be generated and RMSR processing will be performed by BTAM. See Appendix H, "RMSR in BTAM."

TRNSLATE (Translate) Macro Instruction

The TRNSLATE macro instruction provides the user with the facility to translate characters from EBCDIC to another transmission code or from the transmission code to EBCDIC. In order to use the TRNSLATE macro, however, the user must have coded TRANSL=YES in the BTMOD macro instruction issued for his program.

Name	Operation	Operands
[symbol]	TRNSLATE	{dtfbt-symbol | (r) | (1)}, {table-code | (r)}, {area-relexp | (r)}, {length-absexp | (r) | (0) | 'S'}

dtfbt
    specifies the address of the DTFBT for the line group.

table
    specifies the address of the translation table to be used. If a DOS BTAM table is to be used, the user must code the name of the appropriate table as given in Figure 11 (RSCI, SSCI, etc.). The user is permitted to use register notation, in which case he must load the address of the table into the specified register before issuing the TRNSLATE macro. If the address of a user-defined translation table is specified, that table must have been formatted by the user according to the requirements of the System/370 Translate (TR) instruction (See IBM System/370 Principles of Operation, GA22-7000).

area
    specifies the address of the first byte of the area containing the characters to be translated.

length
    specifies the number of bytes to be translated. Any decimal number 1-32767 or equivalent absolute expression is allowable. If the "length" operand is coded 'S' (apostrophes mandatory), then the "area" operand is assumed to point to the first buffer of a chain of buffers. All the data characters in the buffers of the chain will be translated. Register notation is permitted, provided the user loads the specified register with the binary value of the length before issuing the TRNSLATE macro. If the specified register contains binary zero, BTAM interprets this to mean the same as a coded 'S'.

Return Codes: The return codes for the TRNSLATE macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error occurred.
01	The length operand indicated buffers but the associated DTFBT was not open. (Return code is given only if CANCEL=NO appeared in BTMOD.)
04	TRANSL=NO was coded in BTMOD.
08	The length operand indicated buffers, but no buffer pool is associated with the DTFBT.

TWAIT Macro Instruction

The TWAIT macro instruction relinquishes control of the CPU when the problem program must wait for the completion of one of a number of events before further processing can be done. (See the Note for WAIT.)

Name	Operation	Operands
[symbol]	TWAIT	(r),[TERMTST,] ECBLIST={relexp / (r)}

(r)
this user-specified register contains the address of the ECB posted as complete when the TWAIT is satisfied. (An ECB consists of the first four bytes of a DECB.)

TERMTST
specifies that the online terminal test facility will be provided. If this operand is omitted, the facility will not be provided. If this operand is coded and the terminal test logic is not included in the BTAM module, a return code of 04 is placed in register 15.

If TWAIT is specified for a device that is not supported by the TERMTST facility, a test request for that device will be ignored.

ECBLIST
specifies the address of the user-

Table Name		Type of Conversion	Type of Terminal
For incoming messages:			
RC30		1030 Code to EBCDIC	IBM 1030
RC60		1060 Code to EBCDIC	IBM 1060
RC50	Note 1	1050 Code to EBCDIC	IBM 1050
RC40	Note 1	2740 Code to EBCDIC	IBM 2740
RF50	Note 1	1050 Code to EBCDIC	IBM 1050
RF40	Note 1	2740 Code to EBCDIC	IBM 2740
RASA	Note 2	USASCII Code to EBCDIC	Binary Synchronous
RSCI	Note 3	USASCII Code to EBCDIC	IBM 2848
RCT1		Baudot Code to EBCDIC	WU 33/35,WU115A
RCT2		TWX Code to EBCDIC	WU 33/35
RC80		6-Bit TRANSCODE to EBCDIC	IBM 2780
RCTW		5-Bit International Telegraph Alphabet No. 2 to EBCDIC	WTTA
RCT3		5-Bit Figure Protected Code ZSC3 to EBCDIC	WTTA
For outgoing messages:			
SD30		EBCDIC to 1030 Code	IBM 1030
SD60		EBCDIC to 1060 Code	IBM 1060
SD50		EBCDIC to 1050 Code	IBM 1050
SD40		EBCDIC to 2740 Code	IBM 2740
SASA	Note 2	EBCDIC to USASCII Code	Binary Synchronous
SSCI	Note 3	EBCDIC to USASCII Code	IBM 2848
SCT1		EBCDIC to Baudot Code	AT&T 83B3,WU115A
SCT2		EBCDIC to TWX Code	WU 33/35
SD80		EBCDIC to 6-Bit TRANSCODE	IBM 2780
SCTW		EBCDIC to 5-Bit International Telegraph Alphabet No. 2	WTTA
SCT3		EBCDIC to 5-Bit Figure Protected Code ZSC3	WTTA

Note 1: The RF50 (or RF40) translation table converts lowercase alphabetic characters from 1050 (or 2740) code to their uppercase EBCDIC equivalents. The RC50 (or RC40) translation table converts lowercase 1050 (or 2740) code to lowercase EBCDIC.

Note 2: The RASA and SASA translation tables are used for binary synchronous communication.

Note 3: The RSCI and SSCI translation tables are used with the 2848 when it is attached to a 2701 via IBM Terminal Control Type III.

Figure 11. Code Translation Tables Provided by BTAM

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 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 (r) is coded, the register specified must not be the register used for the first operand.

Note 1: The problem program must not be using register 1 at the time the TWAIT macro is issued.

Note 2: If the problem program issues a TWAIT using the same ECBLIST used by a previously completed TWAIT, the completion code of the ECB that has already been posted complete should be zeroed before issuing another READ or following the TWAIT. This will prevent the possibility of satisfying the TWAIT by a previously posted completion code.

The TWAIT macro instruction is similar to the WAIT macro instruction except that:

- TWAIT requires the completion of only one event before returning control to the problem program.
- The address of the ECB that was posted complete is returned to the user in the register specified (as the first operand of TWAIT).
- The ECB keyword is not used in TWAIT; the ECBLIST keyword is required.
- The online terminal test facility is provided if TERMTST is coded in the TWAIT macro (for IBM 1030, 1050, 1060, 2740, the remote 2848, the local 3270, and all binary synchronous devices).

Return Codes: The return codes for the TWAIT macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	TERMTST was coded in TWAIT but the terminal test logic was not included in the BTAM logic module.

WAIT Macro Instruction

The WAIT macro instruction relinquishes control of the CPU when the problem program has no further processing to do and must wait for the completion of one or more events.

Using the WAIT macro instruction for multiple reads or writes gives the user a means of overlapping input-output operations, thus achieving maximum utilization of system resources. Register 1 contains the address of a location in BTMOD that has the address of the last ECB that completed.

Note: The user is cautioned about issuing a WAIT for a channel program which includes channel commands that do not time-out. Commands such as Enable, Prepare and Poll (when polling terminals which have nothing to send and a wraparound list is being used) might not complete in a reasonable time period.

Name	Operation	Operands
[symbol]	WAIT	[count-absexp, / (r), / (0),] {ECB={symbol / (r) / (1)} / ECBLIST={relexp / (r) / (1)}}

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 is satisfied. If the count operand is omitted, 1 is assumed.

ECB
: specifies the address of an Event Control Block (ECB) representing a single event to be posted complete before processing by the problem program can continue. (An ECB consists of the first four bytes of a DECB.)

ECBLIST
: specifies the address of a variable-length list containing fullword entries, with each fullword entry containing the address of an event control block 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, the user must set the 0-bit in the high-order byte of the entry to one.

Name	Operation	Operand
[symbol]	WRITE	{decb-symbol / (r) / (1)},optype-code,[dtfbt-symbol / (r)], ([area-relexp / (r)],[auxarea-relexp / (r)]), ([length-absexp / (r) / 'S'],[auxlength-absexp / (r)]), [entry-relexp / (r) / 'S'],[rln-absexp / (r)], [MF={L / E}]

<u>Note</u>: The parentheses around area,auxarea and length,auxlength should be coded only if both operands of the pair are coded.

Figure 12. Format of the WRITE Macro Instruction

<u>WRITE Macro Instruction</u>

A WRITE macro instruction (Figure 12) is issued by the problem program whenever transmission to a device is required and/or the user wishes to reserve and initialize a DECB. Figure 15 at the end of this section lists the various WRITE options. Specific implementation of these options for each device is discussed in the Device-Dependent Considerations.

decb
: specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing a WRITE macro instruction.

  <u>Note</u>: With a local 3270, the DECB specified must be associated with the same DTFBT in each WRITE macro instruction.

optype
: specifies the option (TI, TT, TIR etc). See Figure 15 for start-stop devices and the device-dependent section for individual BSC devices.

dtfbt
: specifies the address of the DTFBT table for the line group.

area
: specifies the address of the first byte of the output area used to send the message or the address of the first buffer in a chain of buffers. For WRITE TCO, this operand is the address of the first byte of the input area for the invited response from the 2760 Optical Image Unit. This operand is ignored for WRITE TIO.

auxarea
: for conversational type macros and for Write Inquiry this parameter specifies the address of the first byte of the input area used to receive the response. If 'S' (apostrophes are mandatory) is coded, BTAM reads the response into buffers. In this case, auxlength must specify the maximum length of the expected response. BTAM performs the buffering in the same manner as discussed for the READ macro instruction when the area is coded as 'S' in that macro. The only exception is that the address of the first buffer (into which the received message is read) is placed in the <u>auxarea</u> field of the DECB extension rather than into the <u>area</u> field of the DECB.

length
: specifies the number of bytes in the outgoing message. If this operand is coded 'S' (apostrophes are mandatory), BTAM is to write a chain of buffers. The address of the first buffer in this case is specified in the area operand. It is the responsibility of the user to chain the buffers forward if the normal BTAM chaining is altered. When 'S' is coded, BTAM

generates the Write data CCW in the buffers and the remainder of the channel program in the LCB (except for the local 2260 or local 3270). One Write CCW is set up in each buffer to write out the contents of the buffer. BTAM places in the count field of each Write CCW except the last a value equal to the buffer size minus 12. If the user has not supplied a count for the last buffer (see below), BTAM also places a value equal to the buffer size minus 12 in the last Write CCW. Then, by use of data chaining and TIC commands, the contents of the full chain of buffers are transmitted. Normally, an end character appearing in the last buffer will terminate the write operation even if the count in the last Write command has not been exhausted.

A special technique (not applicable to the local 2260 or local 3270) must be employed for those cases when no end character appears in the last buffer and the user does not wish to transmit the full contents of the last buffer in the chain. After getting the required number of buffers via REQBUF, the user should place in bytes 6-7 (the 7th and 8th _physical_ bytes) of the last buffer in the chain the exact count of data characters to be transmitted from this buffer. This should be done by a Store Halfword (STH) instruction to avoid modifying the neighboring bytes, which are initialized by BTAM. After adjusting the count in the last buffer, the user can then issue a WRITE macro with _length_ coded as 'S'. This technique is useful when writing from buffers using one of the BSC transparent-WRITE macros. It is also convenient when writing from buffers to a TWX 33/35 (no end character is used). The same technique can be employed for a message-switching application where the user wants to write out from buffers just filled from a previous READ macro. (See the section on Buffering for details.)

For the local 2260 and local 3270, the message must be contained entirely within one buffer.

For WRITE TCO, this operand is the length of the input area for the invited response from the 2760 Optical Image Unit. This operand is ignored for WRITE TIO.

auxlength
: specifies the number of bytes in the message being received.

entry
: specifies for:

1. A 7770 Audio Response Unit (this operand will be used only for WRITE Invitational and WRITE Conversational), the address of the first byte of the input area. In that case, the length of the input area will be taken from the MSGL operand in the DTFBT macro instruction.

2. A nonswitched line (OPENLST), the address of an entry in the terminal list. For the 2760 Optical Image Unit on a 2740C, the address of the first of the user's three frame-change control characters, which may be set up by assembler language, DFTRMLST OPENLST, or execution.

3. A switched line (DIALST, IDLST, and SWLST), the address of the beginning of the terminal list at initial contact time and of an entry afterwards. The address specified is placed in the entry field of the DECB. For the 2760 Optical Image Unit on a 2740DC, the address of a DFTRMLST DIALST specifying as the list-entry (following the dial characters) the three frame-change characters, when the optype is TIO. On WRITE TCO, the address of the first of the frame-change characters.

4. Binary synchronous applications (DIALST, IDLST, OPENLST, and SWLST), an address as specified for the particular device in the "BSC Device-Dependent Considerations" section. Special usage is made of this operand by the Conversational-type WRITEs. In addition, the 'S' form may be used for retrying an ID verification procedure when an error occurs, without redialing (see the "ID Verification Procedures" section under "System/370 to System/370 Dial"). The address specified is placed in the "entry field" of the DECB. For the binary synchronous support it is also placed in the "address pointer" of the DECB.

Note: The entry operand must be coded for all initial operations unless the device is a local 2260 or local 3270. For local 2260 and local 3270 write operations, the entry operand must be omitted (by coding consecutive commas).

rln
: specifies the relative line number within the line group. (Range: 0-198 inclusive. Default: X'FF'.) For the local 2260 and local 3270, this operand specifies the terminal device (the relative position in the DTFBT of the LCB for the display station or printer) to which BTAM will write.

MF
: see the Creation of a DECB section.

Programming Notes

Execution of a READ, WRITE, or CONTROL macro instruction causes control to be passed to a BTAM routine that constructs channel programs. If no invalid conditions are detected, a channel program will be generated for the requested I/O operation, and control will be passed to the DOS supervisor for execution. Once the channel program has been started, control will be returned to the user's problem program with a return code of zero in register 15 and the condition code set to zero.

The I/O operation proceeds asynchronously with respect to the user's problem program execution. When the problem program wishes to continue processing only after the I/O operation has completed, a WAIT or TWAIT macro instruction may be issued.

If, as a result of a READ, WRITE, or CONTROL macro instruction, control is returned to the problem program with a nonzero return code, no I/O operation was started. Therefore, the problem program must not issue a WAIT or TWAIT macro instruction (the program would enter a permanent wait state). See the Device Dependent Considerations sections for more information on the special aspects of the READ macro instruction.

If the problem programmer intends to use the 'S' option for the entry operand in the READ Initial (TI) macro instruction, he must ensure that the polling list address is placed in the polling pointer field of the data event control block before the first execution of the READ Initial (TI) macro instruction. Once the polling pointer has been initialized, the 'S' option for the entry operand may be used freely because the polling pointer is preserved across write operations. Figures 13 and 14 show how various conditions affect the entry field, polling pointer, and address pointer in the DECB.

A Read/Write sequence on a switched network must always be terminated by a macro instruction that disconnects the line. This may be a Read or Write macro instruction specified with the reset option, a WRITE Positive Acknowledgment (TA), or a WRITE Negative Acknowledgment (TN).

The switched network user should issue a WRITE TA or a WRITE TN at the beginning of his program to reset the line.

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

When a READ or WRITE macro instruction is specified with the "Lock Option" (TIL, TOL, TBL, TLL, or TSL) for a local 2260, the manual entry keyboard retains the same locked or unlocked status that existed at the beginning of the execution of the operation requested by the macro instruction. If the keyboard is unlocked prior to the execution, the keyboard will be unlocked at the completion of the operation. A keyboard that is locked prior to the execution will remain locked at the completion; this prevents further operator entry until the ERASE key or the RESTORE key is depressed or until a BTAM macro instruction without the Lock option is issued.

Return Codes: The return codes for a WRITE macro instruction are shown below. For a detailed description of return codes, see Posted Error Information in the "Assembly Considerations" section.

Hex Code	Meaning
00	No error has occurred.
04	Busy.
08	Invalid relative line number.
0C	Invalid type code.
10	All of the skip bits are on.
14	Line error at OPEN.
18	No buffers.
1C	No buffer pool.
20	No buffer management.
24	BSC usage count exceeded limit.
28	For local 3270 only. Printer busy.
2C	For local 3270 only. No SBA order in area pointed to by entry operand.
30	For local 3270 only. Terminal buffer contents unreliable.

34 For local 3270 only. Device under OLTEP control.

EOT is recognized by the start-stop control unit (2701, 2702, 2703) and sets the control unit in control mode, but the Write command is not terminated. The Write command terminates when the residual count reaches zero. If the characters are not part of the character set recognized by the terminal adapter (2701) or the terminal control (2702, 2703), unit check with data check is set in the sense byte. For instance, X'00' will bring this condition. Therefore, it is safe to set the buffer area to a printable character for the device at the begining of the problem program.

When a WRITE macro instruction specifies "reset at completion" for a nonswitched line, the EOT sequence will be transmitted only if no error condition occurs during execution of the basic channel program.

Situation	Entry Field	Polling Pointer
READ macro expansion: entry operand coded as relexp or (r)	Absolute address of the entry specified by the macro.	Absolute address of the entry specified by the macro.
READ macro expansion: entry operand coded as 'S'.	The address in the polling pointer field of the DECB is moved into the entry field.	Unchanged.
Positive response to polling.	Address of the entry containing the polling characters of the sending device.	Address of the entry containing the polling characters of the sending device.
Negative response to polling after a RESETPL macro has terminated polling.	Address of the entry containing the polling characters of the last polled device.	Address of the first nonskipped (active) entry following the entry containing the polling characters of the last polled device. If no active entry follows, the address of the beginning of the list is used instead.
Negative response to polling after the end of an open list has been reached.	Address of the entry containing the polling characters of the last polled device.	Address of the beginning of the list.

Figure 13. Contents of the Entry Field and Polling Pointer

Situation	Entry Field	Address Pointer
WRITE macro expansion.	Absolute value of the symbolic address specified in the macro.	This field is ignored.
Negative response to addressing.	Address of the entry.	Address of the entry.
Negative response to multiaddressing.	Address of the first entry in the list.	Address of the entry that contains the addressing characters of the device that sent the negative response.
Positive response to addressing/multiaddressing.	Address of the first entry in the list.	Address of the last entry in the list.

Figure 14. Contents of the Entry Field and Address Pointer

When a READ macro instruction involving polling results in a message being received, the polling pointer in the DECB contains the address of the entry in the polling list that was last polled.

Note: The polling pointer in the DECB is not updated upon receiving a positive response to polling using an Auto Poll list. In this case the polling pointer does not contain the address of the entry in the polling list that was last polled. Thus, the problem program may determine the source of the message by inspecting the contents of the polling list entry at that address except when Auto Poll is used. Note that the polling pointer in the DECB is not modified by addressing operations. If the 'S' option for entry is coded in the next READ to be executed, polling proceeds with the device last polled. Thus, by following a successful READ macro with another READ using the 'S' option for the entry operand, the user accomplishes an exhaustive polling of a device before proceeding to poll the next device listed as an entry in the same terminal list.

WORLD TRADE TELEGRAPH TERMINAL TRANSLATION TABLES

Because the International Telegraph Alphabet No. 2 and the Figure Protected Code ZSC3 vary with countries, tables RCTW, RCT3, SCTW, and SCT3 may not fit a user's application. Therefore, four macro instructions are provided to modify these tables when necessary, and thus produce new tables to be used by the TRNSLATE macro instruction. These four macro instructions are: TRSRCTW, TRSRCT3, TRSSCTW, and TRSSCT3.

TRSRCTW and TRSRCT3 Macro Instructions

The format of these macro instructions is:

Name	Operation	Operand
symbol	TRSRCTW	Fx=hexchar,...
symbol	TRSRCT3	Fx=hexchar,...

symbol
: is the name of the translation table used in the TRNSLATE macro instruction and consists of 1-8 characters. If "symbol" is omitted, the name of the translation table is either IJLRCTW or IJLRCT3.

TRSRCTW
: specifies that table RCTW is to be modified and assembled.

TRSRCT3
: specifies that table RCT3 is to be modified and assembled.

Fx=hexchar
: specifies a modification to the table concerned.

F
: means figures shift.

x
: represents the number of the code combination to be translated. (See Appendix I.)

hexchar
: is the hexadecimal representation of this character in EBCDIC.

For TRSRCTW, the permissible values of "x" are: 1, 2, 3, 6, 7, 8, 10 through 14, 19, 22, 24, 26, and 32.

For TRSRCT3, the permissible values of "x" are: 1, 5, 8, 9, 11, 12, 14, 15, 17 through 20, 22, 24, 26, and 32.

Example: If a terminal uses 5-bit International Telegraph Alphabet No. 2, combination 6 in figures shift representing the % character does not exist in table RCTW. Therefore, the user will create the required WTTA translation table by writing:

```
TRSRCTW F6=6C
```

where 6C is the hexadecimal representation of the EBCDIC % character.

TRSSCTW and TRSSCT3 Macro Instructions

The format of these macro instructions is:

Name	Operation	Operand
symbol	TRSSCTW	Xyy=Fx,...
symbol	TRSSCT3	Xyy=Fx,...

symbol
: is the name of the translation table used in the TRNSLATE macro instruction and consists of 1-8 characters. If "symbol" is omitted, the name of the translation table is either IJLSCTW or IJLSCT3.

TRSSCTW
: specifies that table SCTW is to be modified and assembled.

TRSSCT3
: specifies that table SCT3 is to be modified and assembled.

Xyy=Fx
specifies a modification to the table concerned.

yy
is the hexadecimal representation of the EBCDIC character to be translated.

x
is the number of the code combination to be translated.

F
means figures shift.

The permissible values of "yy" are: 2A, 3F, 4A through 50, 5A through 61, 6A through 6F, 7A through 7F.

Example: If a terminal uses 5-bit International Telegraph Alphabet No. 2, and the user wishes for the EBCDIC % character (X'6C') to be translated to the ITA No. 2 % character (code combination 6 in figures shift), the required WTTA translation table is produced by writing:

```
TRSSCTW X6C=F6
```

In the same way, the user can decide that the asterisk character (X'5C') is to be sent as a % character. The required WTTA translation table is produced by writing:

```
TRSSCTW X5C=F6
```

and if the user decides that both the % character and the asterisk character are to be sent as a % character, he will write:

```
TRSSCTW X6C=F6, X5C=F6
```

Note: One of the four macro instructions above can be used to create several translation tables in the same program, provided these tables are given different names. This enables several terminals, using the same codes but with differences in their graphic arrangements, to operate in the same installation.

Option	TYPE Code	7770	Local 2260	Local 3270
READ Initial	TI	X	X	X
READ Initial with Lock	TIL		X	
READ Buffer	TB		X	X
READ Buffer with Lock	TBL		X	
READ Buffer from Position	TBP			X
READ Modified	TM			X
READ Modified from Position	TMP			X
READ Short	TO		X	
READ Short with Lock	TOL		X	
WRITE Initial	TI	X	X	X
WRITE Initial with Reset	TIR	X		
WRITE Initial with Lock	TIL		X	
WRITE Invitational	TC	X		
WRITE Continue	TT	X		
WRITE Continue with Reset	TTR	X		
WRITE Conversational	TV	X		
WRITE at Line Address	TL		X	
WRITE at Line Address with Lock	TLL		X	
WRITE Erase	TS		X	
WRITE Erase with Lock	TSL		X	
WRITE Unprotected Erase	TUS			X
CONTROL Initial	TI	X		
CONTROL Disable	TD	X		

Figure 15. READ/WRITE/CONTROL Options (Part 1 of 2) -- Audio and Local Devices

OPTION	TYPE CODE	1030	1050	Switch 1050	1060	Remote 2260	83B3 115A	TWX 33/35	2740*	WTTA
READ Initial	TI	X	X	X	X	X	X	X	1	X
READ Initial with Reset	TIR	X	X	X	X			X	2	
READ Continue	TT	X	X	X	X	X			3	X
READ Continue with Reset	TTR	X	X	X	X				3	
READ Continue with ID Exchange	TE									X
READ Continue with Leading Acknowledgment	TTA								6	
READ Conversational	TV			X				X	4	
READ Conversational with Reset	TVR			X				X	4	
READ Repeat	TP	X	X	X	X	X			3	
READ Repeat with Reset	TPR	X	X	X	X				3	
READ Buffer	TB					X				
READ Skip	TS	X	X	X	X	X	X	X	1	
WRITE Initial	TI	X	X	X	X	X	X	X	1	X
WRITE Initial with Reset	TIR	X	X	X	X	X	X	X	1	
WRITE Initial Optical	TIO								6	
WRITE Invitational Optical	TCO								6	
WRITE Continue	TT	X	X	X		X			3	X
WRITE Continue with Reset	TTR	X	X	X		X			3	
WRITE Conversational	TV			X				X	5	
WRITE Conversational with Reset	TVR			X				X	5	
WRITE Conversational Optical	TVO								6	
WRITE at Line Address	TL					X				
WRITE at Line Address with Reset	TLR					X				
WRITE Erase	TS					X				
WRITE Erase with Reset	TSR					X				
WRITE Break	TB						X			
WRITE Positive Acknowledge	TA	X	X	X	X	X			3	
WRITE Negative Acknowledge	TN	X	X	X	X			X	2	

1. All 2740
2. 2740 with C, D, DC, DT, DTC, SC, OIU
3. With C, DC, DTC, SC, OIU
4. With D, DC, DT, DTC, OIU
5. With C, D, DC, DT, DTC, OIU
6. With C and 2760 OIU, or with D, C, and 2760 OIU

Figure 15. READ/WRITE/CONTROL Options (Part 2 of 2) -- Start-Stop Devices

Use of the SEPASMB Operand

In a standard teleprocessing application using BTAM, it may be desirable to assemble separately and to catalog individually the various logical components constituting the total programmed support for the system configuration. This technique has the advantage that if one component (say, the problem program) is to be reassembled and cataloged, the others (for example, the DTFBTs and the BTAM module) need not be. For this reason, both the DTFBT and the BTMOD macro instructions include the SEPASMB keyword operand to give the user this facility.

To make effective use of the SEPASMB operand, the user should be aware of the following considerations:

1. If a BTAM macro is coded with SEPASMB=YES, a CATALR librarian-control statement is generated as the first output record of the macro expansion. For this reason such a macro call should appear normally as the first source statement to be assembled. When SEPASMB=YES is coded, no other macro call should appear in the same assembly.

   Coding of SEPASMB=YES in the BTAM macro also results in the creation of a control section in the macro expansion. The macro expansion does not restore the user's control section following the generated code because it is presumed that a user's control section has not been initiated. Other source code included in the same assembly will be included in the same control section initiated by the BTAM macro unless a named CSECT statement follows the BTAM macro call.

2. If a SEPASMB=NO is coded in a BTAM macro call, no CATALR statement is generated. Such a macro call need not appear as the first source statement in an assembly. Any number of macro calls with SEPASMB=NO may be included in a single assembly.

   Although the macro expansion also creates a control section for SEPASMB=NO, the control section preceding the macro call is restored following the macro expansion.

In abnormal usage (for example, when assembling several DTFBT macro instructions together in order to share a buffer pool and cataloging them as a single relocatable library module), the operand must be coded SEPASMB=NO or omitted in each of the macro instructions. The user must prepare his own CATALR statement to go with the object deck from the single assembly. The catalog name must be the same as the name of some entry point or CSECT in the assembly, which in turn must be the same as some name in an EXTRN statement or V-type address constant in the application program with which the cataloged relocatable module is to be used. This allows the linkage editor AUTOLINK facility to locate the assembled DTFBT module in the relocatable library.

Principles of BTAM Register Usage

Because the System/370 Disk Operating System does not permit the user to make free use of registers 0, 1, 14, and 15, BTAM does not preserve any user's values of these registers except the value placed in register 14 by a linkage to BTMOD. BTAM always preserves the user's values of registers 2-13,* inclusive. However, BTMOD requires the use of registers 1-15 as base registers or DSECT registers. Consequently, BTMOD employs USING and DROP statements involving these registers. This can cause loss of user's program addressability if BTMOD is introduced into the middle of the user's assembly. This condition can be avoided if the user does one of the following:

- Assembles BTMOD separately, and catalogs it in the relocatable library, or
- Assembles BTMOD into his program before his first USING statement, or
- Assembles BTMOD into his program after his other assembler statements (which should conclude with LTORG), and just before his END statement.

----

*Register 13 is preserved because compiler-generated DOS routines require register 13 as a save area address register. A save area register is not a universal DOS requirement, therefore BTAM does not use it for this function. Registers 2-12 are allowed as operands in BTAM macro instructions where register notation is permitted.

DEFINE THE LINE GROUPS

A line group is defined through a table generated by the DTFBT declarative macro instruction. This table contains information concerning the line connection, the control unit attached to the channel, the remote device type on the lines, and various machine and programming features used for the line group. Other information in the table concerns various programming options and address constants required by BTAM to perform the telecommunications functions desired by the user.

The individual lines in the line group are defined by the LINELST keyword operand in the DTFBT macro instruction generating the table. In coding this operand, the user associates each line in the group with a logical (or symbolic) unit which, in turn, is associated with a physical unit via a logical unit block (LUB) in a device table. Each LUB (a 2-byte entry in the device table) contains a pointer to a physical unit block (PUB) which contains information such as channel, unit, device type, etc.

The association of a LUB with a PUB is established for an installation either at system generation time or at problem program execution time. In the former case (system generation time), certain conventions are established to be followed by all programmers. In the latter case, the problem programmer can include ASSGN control statements with his JOB control cards to assign a specific logical unit (SYSnnn) to a specific physical unit. The ASSGN control statement may also be used to override the standard assignments defined at system generation time. The operator (for background or foreground programs) also may make assignments. A full discussion of device assignment may be found in the <u>DOS Version 4 System Control and System Service Programs</u> publication.

DEFINE THE TERMINAL LISTS

The DFTRMLST macro instruction defines terminal lists to be used in polling and addressing terminals, and effecting exchange of computer and terminal identifications (refer to the Device Dependent Considerations sections).

A single terminal list is normally defined for each line for all read operations. This list includes the polling or invitation-to-send codes for all terminals on the line. The order in which sending devices on a line are contacted to send data is determined by a polling list containing one or more entries for each device to be contacted. A sending device may appear one or more times in a polling list so that the stations requiring more service will be polled more frequently.

Normally, a separate list is defined for each terminal for all write operations. Each of these lists includes the addressing or call-directing code for the terminal that is to receive the message. All receiving devices must be defined in addressing lists. An addressing list may include addresses for more than one terminal or component when multi-addressing is used. BTAM contacts all terminals in the list before sending the message. Each terminal sends back a response that indicates whether it is ready to receive or not. If any terminal fails to answer or sends a negative (not ready to receive) response, the message is not sent. An addressing list is always of the open type.

When terminals on different lines have the same polling and addressing characters, the same polling and addressing list may be used for these lines. A single list may be used by more than one line at the same time if the operation is the same for the lines. Obviously, if CHGNTRY is issued for a terminal list, polling and addressing on all lines using the list would be affected.

The terminal devices connected to an audio response unit are not defined in terminal lists. Line connection is established on the subscriber's request, and no polling or addressing takes place. Once the line connection is established, exchange of information takes place.

The sections on the individual terminal types in this publication illustrate the particular DFTRMLST formats used for reading and writing on a given terminal/network configuration. Appendix A gives examples and descriptions of the generated lists.

The only choice left to the programmer is the type of list that he wants to use for <u>polling</u> terminals on nonswitched networks. For these terminals, he can choose either an open list, a wraparound list, or an Auto Poll list.

If an open list is provided for the READ macro instruction, polling ceases when the last entry in the list is polled. Each component is polled in turn, beginning with the entry whose address is given in the READ macro instruction. For each terminal or component polled, the terminal control unit returns a response. A negative response indicates that the terminal or component has no message to send to the computer. A positive response consists of the message itself. The first positive response returned stops polling. Polling of successive components ends when either:

- A positive response is returned
- An error condition from whch BTAM cannot recover is detected
- A RESETPL macro instruction is issued, or
- The end of the list is reached.

When transmission is completed, or one of the last three above conditions exists, the read operation is posted complete.

If a wraparound list is provided, polling continues indefinitely until either:

- A message is received from the terminal
- An error condition from which BTAM cannot recover is detected, or
- Polling is terminated by the RESETPL macro instruction (see the RESETPL macro instruction description).

When polling on a wraparound list is terminated by either condition 2 or 3, or when data transmission is complete, the read operation is posted complete. When a wraparound list is used, the READ Initial macro instruction may specify an 'S' as the terminal list address. In this case, BTAM will poll either:

- The same terminal, if the last polling sequence produced a message or an error condition, or
- The next active terminal in the list, if the last polling sequence produced a negative response and was stopped by a RESETPL macro instruction.

This allows the problem programmer to design a system that obtains all messages to be sent from a polled terminal, and permits ease of error retry.

DEFINE THE DECBs

A data event control block is normally defined for each line. This may be done by using the MF=L or the MF-operand-omitted form of a READ, WRITE, or CONTROL macro instruction. Normally, a single DECB per line is sufficient, but additional DECBs for specific purposes may be desired. The DECB contains data that must be used both by the programmer and by BTAM. See Figure 16 for the format of the DECB.

Creation of a DECB (Data Event Control Block)

The READ, WRITE, and CONTROL macro instructions have different functions, depending on how the MF keyword operand is coded:

1. MF=L: a DECB is reserved at assembly time and the parameters coded in the macro instruction initialize the DECB. Because this is an assembly-time function, no register notation is allowed. The function (READ, WRITE, or CONTROL) is not executed. This form of the macro should not be branched to by executable code.
2. MF=E: the parameters coded in the macro instruction update the fields of a DECB previously defined, and the function (READ, WRITE, or CONTROL) is executed. Register notation is allowed where indicated.
3. MF operand omitted: a DECB is reserved and initialized at assembly time, and the function (READ, WRITE, or CONTROL) is executed at execution time. With the execption of the decb operand, register notation may be used where indicated, and the operands so coded update the corresponding DECB fields prior to execution. If a READ, WRITE, or CONTROL of the MF=E form is issued using a DECB generated by a macro with the MF operand omitted, the values in the DECB generated at assembly time are replaced by those values specified in the MF=E form of the macro.

DECB Extension for READ: If the parameter DECBEXT is coded as 'YES' in the BTMOD macro, BTMOD will contain the logic for supporting the DECB extension. Logic for the inlist technique will not be included when this parameter is coded 'YES'. If this parameter is coded 'NO' or is omitted, the inlist logic will be included, but the DECB extension logic will not.

Programming Note: For MF=L or MF-omitted form, the DECB contains an eight-byte extension. For MF=E form, an extended DECB must have been created by previously coding an MF=L or MF-omitted form containing the auxarea and auxlength operands.

DECB Extension for WRITE: If the parameter DECBEXT is coded as 'YES' in the BTMOD macro, BTMOD will contain the logic for supporting the DECB extension. Logic for the inlist technique will not be included. Otherwise, the inlist logic will be included, but the DECB extension logic will not. When DECBEXT is coded as 'YES', the user must specify his output parameters and his input parameters for any one of the conversational-type WRITEs or WRITE Inquiry via the (area,auxarea) and (length, auxlength) operands. When both output and input parameters are coded (MF=L or MF omitted) the DECB contains an eight-byte extension. For MF=E form, the user must have created an expanded DECB by previously coding an MF=L or MF-omitted form containing the auxarea and auxlength operands.

Posting in the DECB

BTAM posts in the DECB the event associated with that DECB. An event that satisfies a WAIT condition may be one of the following:

1. A successful completion of a READ, WRITE, CONTROL, or RESETPL operation.

2. A nonsuccessful operation, as follows:

   a. Polling - when the end of an open list is reached, the polling stops.

   b. Addressing - when the addressed device is not ready, no message is sent.

   c. Multiaddressing - when one of the addressed devices is not ready, the message is not sent to the ready devices.

   d. An error condition.

3. A completion of a RESETPL macro instruction.

The following describes the DECB fields in Figure 16 for the DECB.

Bytes 0-3: Event Control Block (ECB). BTAM sets byte 0 to zero prior to execution of a macro instruction and gives a completion code in that byte when an event has been completed. (See the section on completion codes.)

Note: Under certain conditions bytes 1-3 are used. See Note 2 of Figure 16.

Byte 4: Optype qualifier bits. This byte contains various flags which are set ON as follows:

Bit-0 Initial optype coded

Bit-1 "R" added to optype of READ/WRITE when Reset option is used (not applicable for WRITE TR)

Bit-2 "L" added to optype of READ/WRITE for local 2260 when Lock option is used

Bit-3 Conversational WRITE coded

Bit-4 Start-stop Auto Poll used in current READ

Bit-5 Entry operand coded as 'S'

Bit-6 Area operand coded as 'S'

Bit-7 Length operand coded as 'S'

Byte 5: Optype code. This byte contains a hexadecimal value corresponding to the optype coded in the READ, WRITE, or CONTROL as follows:

Hex Value	Optype
00	WRITE TB
00	WRITE TD
01	READ TI
02	WRITE TI
03	READ TT
04	WRITE TT
05	READ TV
05	READ TTL
06	WRITE TV
07	READ TP
08	WRITE TA
08	WRITE TIE
09	READ TIQ
09	READ TS
0A	WRITE TN
0A	WRITE TIX
0B	READ TB
0B	READ TPL
0C	WRITE TL
0C	WRITE TIXV
0D	READ TO
0D	READ TQ
0E	WRITE TS
0E	WRITE TIV
0F	READ TE
10	WRITE TC
10	WRITE TE
11	READ TM
11	READ TRV
12	WRITE TX
12	WRITE TIO
12	WRITE TUS
13	READ TIC
13	READ TBP
14	WRITE TXV
14	WRITE TCO
15	READ TIW
16	WRITE TVO
16	WRITE TR
18	WRITE TW
19	READ TMP
1A	WRITE TQ
1B	CONTROL TI
1C	CONTROL TD
1D	CONTROL TM
20	WRITE TIC

Bytes 6-7: Length. This is the length specified by a READ or WRITE macro instruction length operand.

	0	1	2	3
0	Completion Code	Reserved (Note 2)		
4	Optype Qualifier Bits	Optype Code	Length	
8	Response Information Byte	DTFBT Address		
12	Mode Byte	Input/Output Address		
16	Sense Byte	Sense Byte for Diagnostic Write/Read	Residual Count	
20	Command Code	List Address or Entry Address		
24	Flag Byte (Note 2)	Relative Line Number	Response to Addressing	Response to LRC
28	TP Code	Error Information	Status Bytes	
32	Reserved	Addressing Pointer		
36	Index Byte	Polling Pointer		
40	Flag Bytes for DECB Extension	Reserved	Aux Length	
44	Reserved	Aux Area Address		

NOTE 1: The last eight bytes of the DECB are called the DECB extension.
NOTE 2: If bit 6 of byte 24 of the DECB is set to one, then bytes 1, 2, and 3 contain the address of the first unreleased buffer not used at the completion of a READ whose AREA operand is coded as 'S'.

Figure 16. Data Event Control Block

Byte 8: Response information byte.

Bits 0-2, reserved

Bit 3 if on indicates that a Status/ Sense message has been received from a remote 3270. This bit is sometimes referred to as the "R message received" flag.

Bits 4-5, reserved.

Bit 6 is ON when the RVI character sequence is received by the transmitting CPU as response to text.

Bit 7, reserved.

Bytes 9-10-11: DTFBT address. This is the address of the DTFBT table associated with the DECB.

Byte 12: Mode byte. This byte is used to store information for the Set Mode command for binary synchronous communication.

Bytes 13-14-15: Input/Output area address.

Byte 16: Sense byte. This byte contains the sense byte after a unit check condition occurs.

The bits are set ON when the following conditions occur:

Bit 0	Command reject
Bit 1	Intervention required
Bit 2	Bus out check
Bit 3	Equipment check
Bit 4	Data check
Bit 5	Overrun (for the local 3270 only, it has the meaning of unit specify)
Bit 6	Lost data (for the local 3270 only, it has the meaning of control check)

Bit 7 Time out (for the local 3270 only, it has the meaning of operation check)

Byte 17: Sense byte for a 2701 diagnostic Write/Read.

Bytes 18-19: Contain the residual count for the read or write data command.

Note: For World Trade telegraph terminals, the residual count contains the number of characters that have not been read. This count, however, is incremented by one when a message ends with WRU, EOM, or EOT.

Byte 20: Command code. This identifies the type of command upon which the error occurred.

Bytes 21-22-23: Entry address. For the local 3270, when READ Modified from Position or READ Buffer from Position is issued, the address of a 4-byte data area containing a WCC, an SBA, and a device buffer address; on other macro instructions for the local 3270, the entry address field is not applicable.

Byte 24: Flag byte. The bits are:

Bit 0, reserved.

Bit 1 is ON if EOT is received (binary synchronous communication or World Trade telegraph terminal).

Bit 2 is ON if DLE,EOT is received (binary synchronous communication only) or if a World Trade telegraph terminal message ends with a WRU signal.

Bit 3 is on when an error status message is received (binary synchronous communication only).

Bits 4-5, reserved.

Bit 6 is ON if the ECB contains the address of the first unreleased buffer not used at the completion of a READ (whose area operand is specified as 'S').

Bit 7 is ON when a terminal test is operating on the line.

Byte 25: Relative line number. This is the relative number of the line currently associated with the DECB.

Bytes 26-27: Response to addressing field. For start-stop transmission, the response to addressing character(s), if any, is read into byte 26. The response to the longitudinal redundancy check (LRC), if any, is read into byte 27. For binary synchronous communication, both bytes are used for responses.

Byte 28: Contains the TP code of the last executed command in a channel program that has terminated. These TP codes may be used by the problem program to identify the function of a command in a channel program when the function is not identifiable by the command code alone.

The hexadecimal TP codes have the following meaning for non-BSC channel programs:

00 Null
Prepare, Sense, Write computer-ID, and Write Pad Characters (WTTA).

01 Disable (if first command of channel program), Dial, Enable, Write pad characters, Prepare, Write Mark Characters (WTTA).

02 Write D C C C (prior to selection), Write EOT sequence (prior to polling or addressing).

03 Write polling, or addressing characters, Write turn-around sequence (TWX), Write CPU-ID sequence (TWX), or Poll.

04 Write space, Write code (2260R), Write shift (83B3), Write one (1030).

05 Read response to polling.

06 Read response to addressing.

07 Read ID response (TWX).

08 Write end-of-addressing character following addressing, Write WRU (WTTA)

09 Write response to text, Write D and 15 idle characters.

0A Write idle characters.

10 Write at Line Address (2260).

11 Read or Write text.

12 Read or Write in buffers.

20 Read response to text.

21 All Reset commands (i.e., those commands in a channel program that perform the reset function).

22 Read Skip.

23 Write Break.

The TP codes used in BSC channel programs are:

00 Not used.

01 Prepare, Enable, Dial.

02 Write special characters (DLE ETB, DLE ETX, WACK), Poll (at beginning of channel program).

03 Poll, Write addressing characters, Write ID, Write ENQ, TIC (to a Poll), I/O No-Op (in an Auto Poll channel program).

05 Read response to ENQ.

06 Read response to addressing.

07 Read ID, ACK-0.

08 Read ID, ENQ.

09 Write response to ENQ, Write response to text.

11 Read text, Write text, TIC (to a buffer).

12 Read text into buffers, Write text from buffers.

20 Read response to text.

21 Write EOT, Write DLE EOT, Set Mode, Disable.

24 Read ENQ.

31 Read Skip (ERP).

32 Write ENQ (ERP).

34 Read ENQ (ERP).

35 Write DLE ENQ (ERP).

36 Disable (ERP).

39 Write ACK-0/ACK-1 (ERP), Write NAK (ERP).

Byte 29: Error information field. This field provides the following information if ERP is specified:

Bit 0, reserved

Bit 1, should-not-occur error

Bit 2, error in ERP

Bit 3, diagnostic write/read failed

Bits 4-5, not used

Bit 6, a message of the form: STX--Text (may be zero length)--ENQ received from remote BSC device.

Bit 7, two conditions as follows:

1. NAK command received in reply to transmitted message or as a reply to a WRITE Inquiry (BSC only).
2. NAK or ID NAK received on a WRITE Connect operation.

Bytes 30-31: Status bytes. These bytes contain the status bytes after a completion occurs.

Bit 0, Attention

Bit 1, Status modifier

Bit 2, Control unit end

Bit 3, Busy

Bit 4, Channel end

Bit 5, Device end

Bit 6, Unit check

Bit 7, Unit exception

Bit 8, Program-controlled interruption

Bit 9, Incorrect length

Bit 10, Program check

Bit 11, Protection check

Bit 12, Channel data check

Bit 13, Channel control check

Bit 14, Interface control check

Bit 15, Chaining check

Byte 32: Reserved.

Bytes 33-34-35: Addressing pointer.

Byte 36: Index byte. At the normal completion of a READ Initial using start-stop Auto Poll, this byte contains the index byte of the last entry in the list to be polled.

Note: The contents of this byte will be unpredictable prior to normal completion of the READ Initial channel program.

Bytes 37-38-39: Polling pointer. At the completion of a READ for the local 2260 and local 3270, the rln for the display station from which data has been read is placed in the low-order byte of the polling pointer field (byte 39).

Byte 40: Optype Qualifier Bits for the DECB extension.

Bit 0, Auxarea operand coded as 'S'

Bit 1, Auxlength operand coded as 'S'

Byte 41: Reserved.

Byte 42-43: Auxlength.

Byte 44: Reserved.

Byte 45-46-47: Auxarea address.

BUFFERING

The buffer management logic consists of routines to:

1. Perform the functions requested through the REQBUF and RELBUF macro instructions (for obtaining and releasing buffers).
2. Read incoming messages into buffers (when the user codes the area operand as 'S' in a READ macro) or write outgoing messages from buffers (when the user codes the length operand as 'S' in a WRITE macro).

The buffer management logic, however, is optionally included in the BTAM module for a problem program. For example, if BUFFER=NO is specified in the BTMOD macro for the problem program, no buffer management is possible for that program. If BUFFER=REQREL is specified, the routines to perform (1) above are included for the program, but the routines to perform (2) are not. On the other hand, if BUFFER=YES is coded, the routines to accomplish both (1) and (2) are included for the problem program.

Through the DTFBT macro, the problem program may, if the buffer management routines are included for the program, define for the line group whether or not buffering is to be performed. This is done by coding the BUFCB, BUFNO, and BUFL operands in the DTFBT macro. The BUFNO and BUFL keyword operands in DTFBT allow the user to define the number of buffers and the length of each buffer in a buffer pool. The user may also share a buffer pool between two or more line groups.

For a line group for which buffering is specified, the BTAM OPEN routine will organize the buffer pool automatically (provided it is not a shared buffer pool previously organized and/or currently being used by another line group).

Buffer Pool

The first eight bytes of the buffer pool are the buffer control block (Figure 17).

c - Buffer pool counter; indicates the number of currently open line groups using the buffer pool.

a - Address of the first buffer available.

nd - Number of buffers as defined in the DTFBT macro instruction via the keyword operand BUFNO.

na - Number of buffers currently available. Initially, na=nd.

length - Length of each buffer as defined in the DTFBT macro instruction via the keyword operand BUFL.

Each buffer pool and each buffer within the pool begins on a doubleword boundary. The problem program makes better use of main storage if it requests buffers whose lengths are multiples of eight bytes.

Buffer Format

Each buffer has the format shown in Figure 17.

Bytes 0-8: are reserved for use by BTAM for channel commands.

Bytes 9-10-11: contain the address of the next buffer in the chain. If this buffer is the last buffer in the chain and the Reset function is not specified, bytes 9, 10, and 11 contain zero. If this buffer is the last buffer in the chain and the Reset function is specified, the address of the Reset CCW is contained in bytes 9-11.

Bytes 12 through length-1 contain:

- Zero when the buffers are in the pool and when the user gets them by a REQBUF macro instruction.
- The data read, if a READ (area operand 'S') has been issued.

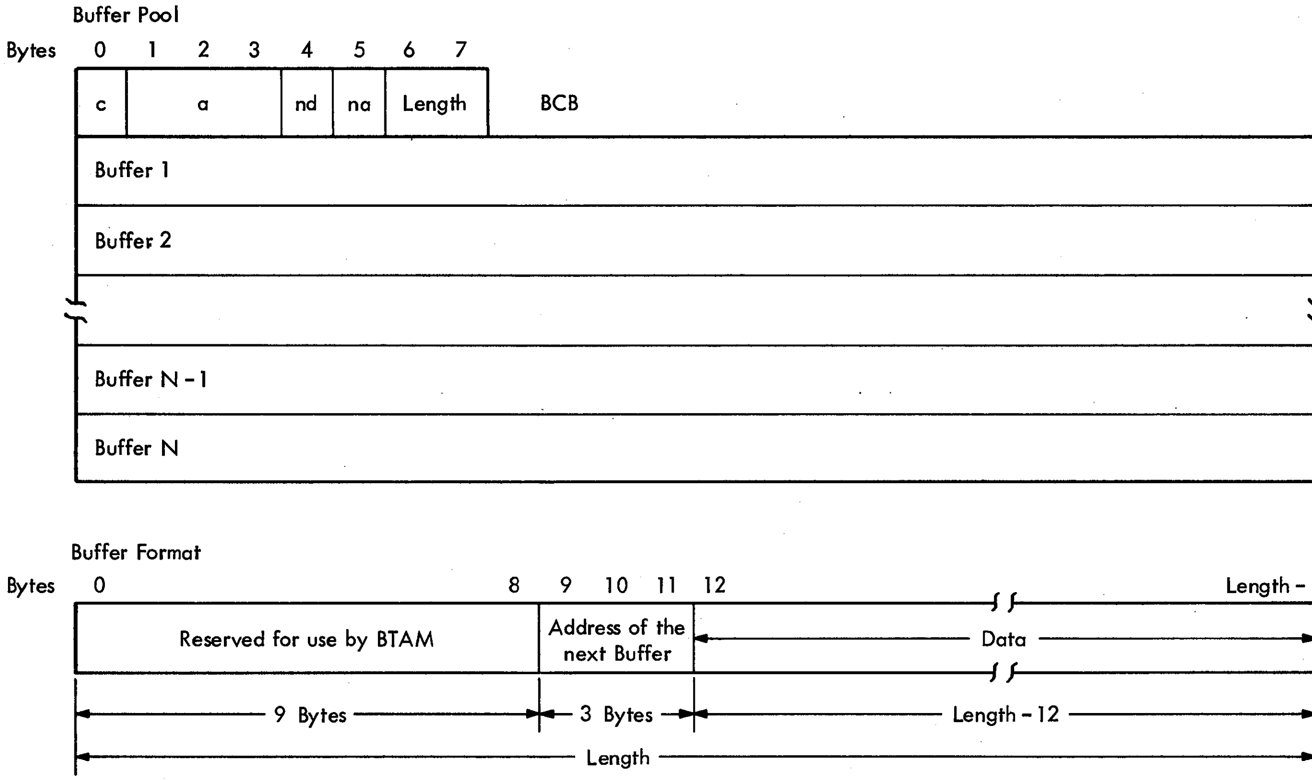


Figure 17. Buffer Pool and Buffer Formats

- The data to be written, when a WRITE (length operand 'S') is issued.

Buffers for Read Operations

If the area operand of the READ macro instruction is coded 'S', BTAM will provide the user buffers from the buffer pool of the line group. The length operand of the READ macro instruction specifies the maximum length in bytes of the expected message. BTAM will provide as many buffers as necessary, provided the required number is in the pool. These buffers will be chained. Once the operation is complete, the address of the first buffer in the chain is in the area field in the DECB for the line (bytes 12 through 15).

For the binary synchronous communication Conversational-type WRITEs (see WRITE Initial Conversational in the CPU-to-CPU Contention section). BTAM also provides buffers (optionally) for the received message. In this case, the address of the first buffer in the chain is placed in Word 1 of the user's parameter list rather than in the area field of the DECB (which contains the address of the user's output area).

If the DECB extension is being used, the address of the first buffer in the chain is placed in the auxarea field of the DECB.

For additional information on the DECB extension, see the DECB Extension for WRITE section.

The user must return the buffers whose messages have been processed. The release buffer (RELBUF) macro instruction releases buffers to the buffer pool.

If an error is posted on a Read operation using buffers, the user must return the acquired buffers to the pool before retrying the operation.

Buffers for Write Operations

A request buffer (REQBUF) macro instruction obtains a chain of buffers from the buffer pool. The address of the first buffer in the chain is returned in the register specified by the second operand in the REQBUF macro instruction. If the length operand in the WRITE macro instruction is coded 'S', BTAM will write from a chain of buffers. The user must specify the address of the first buffer in the area operand of the WRITE macro. To avoid having BTAM write out the full contents of the last buffer in the chain (when no end character is in the buffer or the end character does not terminate the Write operation), the user can use a special technique. The exact count of data characters in the last buffer can be placed in bytes 6-7 (the 7th and 8th physical bytes) of the last buffer in the chain prior to issuing the WRITE macro. For message-switching applications, if a single buffer message is read in on a READ TI or READ TIR from a start-stop multipoint terminal, the count field in the Read text CCW must be cleared before issuing the Write operation. For special considerations for World Trade telegraph terminals, refer to the Device Dependent Considerations sections.

At the completion of the WRITE macro instruction, the user must return the buffers which are no longer required. The release buffer (RELBUF) macro instruction is used to release the buffers.

Buffer Pool Updating

Step 1 - At assembly time, the DTFBT macro instruction creates the buffer control block and reserves main storage for the buffer pool.

Step 2 - At OPEN time the pool is organized, with all buffers cleared and chained forward. Assume for discussion that the first buffer available is a which is chained to b chained to c chained to d, etc.

Step 3 - If the buffers a and b are taken from the pool by a READ operation or a REQBUF macro instruction, the next buffer available is c which is chained to d, etc.

Step 4 - If the buffers m, n, o, and p are released to the pool by a RELBUF macro instruction, and the first buffer available is j, the buffer m becomes the first buffer available.

Figure 18 illustrates steps 2, 3, and 4.

The user requests buffers by issuing a REQBUF macro instruction. If the number of buffers available is exceeded by the number requested, the available buffers are still passed to the user. A return code indicating that there were either not enough buffers or not any buffers is returned in the low-order byte of register 15. The number of buffers exceeding the number available is in register 0. See the section on return codes for REQBUF.

The user may also request buffers implicitly by issuing a READ macro instruction (area coded as 'S'). If the number of buffers available is not sufficient to contain the maximum input message, then the request is ignored. In register 15 there is a return code of hexadecimal 18 indicating too few buffers available.

POSTED ERROR INFORMATION

BTAM provides information to help the user recover from error conditions occurring during the execution of a BTAM operation. These conditions may be caused by problem program errors or by temporary or permanent failure of the data link.

The error information provided by BTAM includes return codes for problem program errors and completion codes on I/O termination. The conditions of the error are posted in the DECB. For the audio response units, certain error recovery procedures are provided as standard facilities. If the ERP option has been chosen, the operation in error is retried if possible, and system-to-operator messages are sent when the number of retries reaches a retry-limit or when the error cannot be corrected. In the latter case (with ERP), comprehensive error information is posted in the DECB.

Return Codes: When BTAM detects a problem program error in the issuing of a READ, WRITE, CONTROL, REQBUF or RELBUF macro instruction, the macro is not executed, the error condition is recorded, and control is returned to the problem program at the instruction following the macro in error.

BTAM records such errors by setting the condition code to a nonzero value (so that the problem program may use a branch-on-condition instruction), and by setting a return code in bits 24 through 31 of the general register 15 (bits 0 through 23 being zero). Note that if no error has occurred, register 15 contains a return code of X'00' and the condition code is zero.

The return codes set for READ, WRITE and CONTROL are shown below.

Hex Code	Meaning
04	Busy. The specified line (or control unit for a local 2260 or local 3270) is busy with a previously requested operation.
08	Invalid relative line number. There is no line within the line group with a relative line number equal to that specified by the user in the macro instruction. For the local 2260 or local 3270, there is no device with a relative number equal to that specified by the user in the macro instruction.

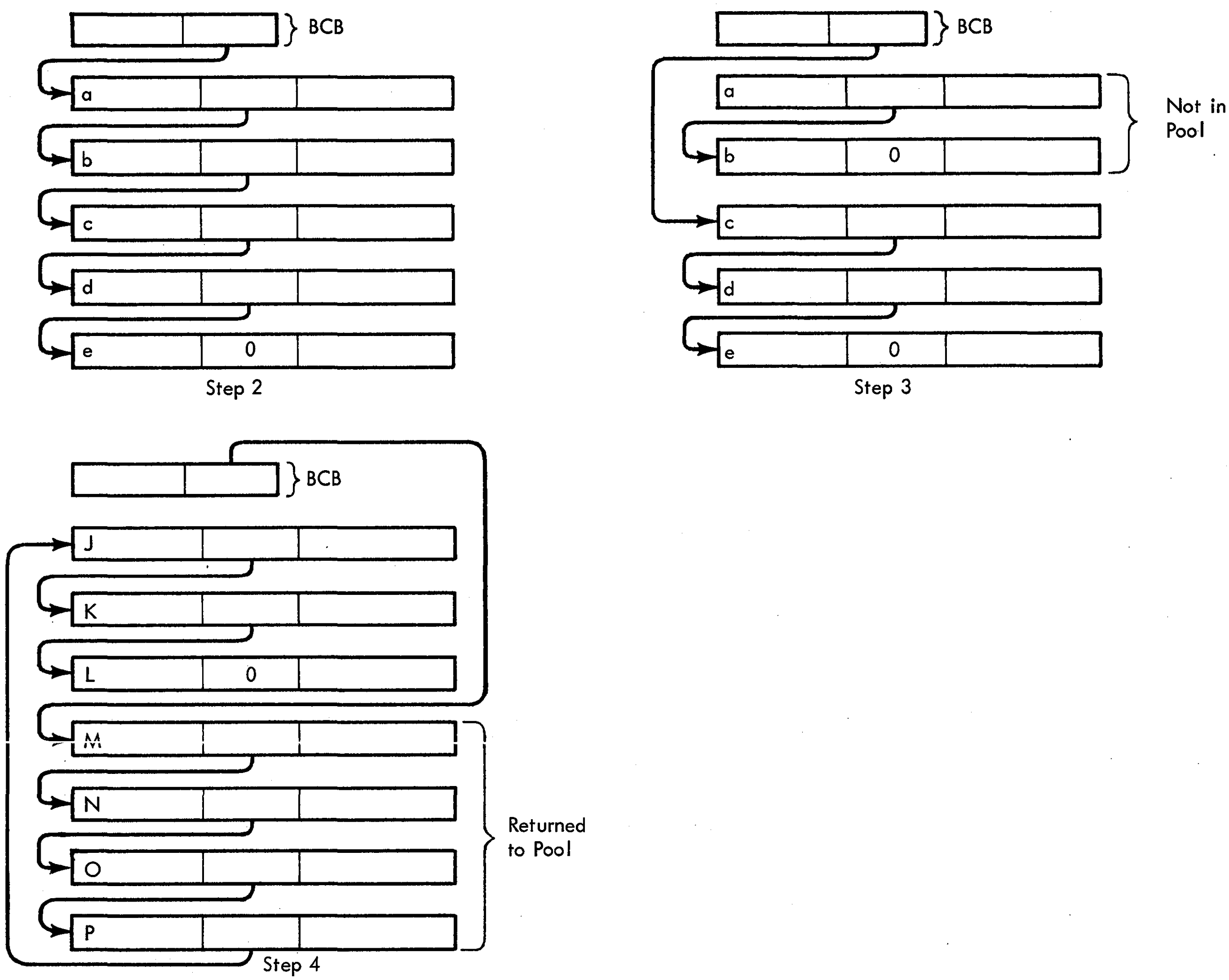


Figure 18. Buffer Pool Updating

0C Invalid type code. the code specified in the optype operand is invalid for the device, or illegally used with area coded 'S'.

10 All skip bits on. The skip bit is ON in all of the entries in the polling or addressing list specified in the macro. This return code is also used if there are no active entries in the Auto Poll list. This return code can occur if the user points to a certain entry in a polling or addressing list in which the indicated entry and all remaining entries in the list are skipped, even though preceding entries may be active.

14 Line error at OPEN, The user should cease referring to that line until he can correct any physical difficulties with the line or data set. He may then issue an LOPEN macro instruction to retry to open the one line.

18 No buffers. There are not enough buffers available in the buffer pool to satisfy the needs of a READ macro instruction with the area operand coded 'S' or a Conversational-type WRITE (binary synchronous communication) for which buffers are requested.

1C No buffer pool. The user issued a READ macro instruction with the area operand coded 'S', a WRITE macro with length coded 'S', or a Conversational-type WRITE having Word 1 of the specified parameter list containing a zero address, but the DTFBT specified does not contain the address of a buffer pool.

20 No buffer management. The user issued a READ macro with area coded 'S', a WRITE macro with length coded 'S', or a Conversational-type WRITE having Word 1 of the specified parameter list containing a zero address, or used the DECB extension with auxarea coded 'S', but the BTMOD macro for the problem program did not specify BUFFER=YES.

24 BSC Usage Count exceeded limit. More that 6 READ Initial operations using the same BSC AUTOLST or AUTOWLST terminal list have been issued by the user but have not completed. This return code is also used for illegal use of READ Continue with the 2760. The user issued a READ TT following an operation in which data was received from the 2760.

The following return codes are applicable only to the local 3270:

Hex Code	Meaning
28	Printer busy due to a previously issued WRITE macro instruction.
2C	No SBA order in the second byte of the area pointed to by the entry operand. This can occur only when the macro instruction is a READ Modified from Position or a READ Buffer from Position.
30	• For a printer, the previous print operation did not complete successfully. The application may either (1) retry the previous print operation by issuing a WRITE with a WCC with the Start Print bit set and no data stream (the data will already be in the printer buffer) and then, if a return code of X'30' is received again, reinitialize the buffer image by issuing a WRITE Erase with a WCC with the Start Print bit set and desired data to be printed in the data stream; or (2) not retry, but immediately reinitialize the buffer image and print it, issuing a WRITE Erase. • For a display station, an asynchronous I/O error has occurred since the completion of the last I/O operation to that display. The terminal buffer may not contain the expected data. The application program should reinitialize the buffer image by issuing a WRITE Erase with a data stream containing the data that should be in the terminal buffer.
34	Device temporarily under OLTEP control and therefore busy. This can be received when a READ or WRITE macro instruction other than READ Initial has been issued to the device. I/O should be attempted using other devices on the same or other control units. Periodically (but not continously), the application program should reissue the READ or WRITE macro instruction to the busy devices until the busy condition clears and the I/O operation is performed. Note: If there is no other device available on this or another control unit because all are being used by OLTEP, the application program should issue a READ Initial macro instruction and then a WAIT macro instruction. After OLTEP has relinquished control, the wait condition will be satisfied the next time an operator causes an attention interruption. If the application program continuously issues other than a READ Initial to any device attached to the control unit causing the X'34' return code, OLTEP will be prevented from executing in its lower priority background partition, will not be able to free the device, and X'34' will continue to be received by the application program.

Note: All nonzero return codes indicate that the operation was not properly performed. The problem program must not issue WAIT macro instructions for READ, WRITE, or CONTROL macros that result in nonzero return codes, because no I/O operation is initiated.

The return codes set for REQBUF and RELBUF are:

Hex Code	Meaning
04	Partially performed. Some, but not all, of the buffers have been provided to the user, or returned to the buffer pool. Register 0 contains the count of buffers not supplied or not returned to the pool. For RELBUF, the user-supplied register contains the address of the first buffer not returned.
08	Not available. None of the buffess have been provided to the user, or returned to the buffer pool. For REQBUF, the user-specified register will contain zero, and for RELBUF the user-specified register will be unchanged while register 0 will contain the count of the buffers not supplied or not returned.

Hex Code	Meaning
0C	No pool. The DTFBT specified does not contain the address of a buffer pool.
10	No buffer management. The BTMOD macro for the problem program did not specify either BUFFER=REQREL, or BUFFER=YES.

Note: If the DTFBT specified in the REQBUF/RELBUF is not open, a system-to-operator message is written and the job is canceled if the user's BTMOD specified CANCEL=YES. If CANCEL=NO, control is returned with a cancel return code of X'01'.

The return codes set for CHGNTRY (with SSALST, SSAWLST, AUTOLST, and AUTOWLST only) are:

Hex Code	Meaning
00	Action was performed or was not needed. For example, if SKIP was specified and the entry was not found to be active, then the action was not needed.
04	Action was not performed because CHGNTRY was issued for a terminal list that is being polled.
08	Action was not performed because the position operand is too large. The value of the position must be less than the number of entries.

The non-zero return code set for TWAIT is:

Hex Code	Meaning
04	TERMTST was coded in TWAIT but terminal test logic was not included in BTAM logic module.

The non-zero return codes set for TRNSLATE are:

Hex Code	Meaning
01	The length operand indicated buffers but the associated DTFBT was not open. This return code is given only if CANCEL=NO appeared in BTMOD. For CANCEL=YES a system-to-operator message is given and the job is canceled.
04	TRANSL=NO was coded in BTMOD; no translation can be performed.
08	The length operand indicated buffers but no buffer pool is associated with the DTFBT.

The return codes set for ONLTST are:

Hex Code	Meaning
00	Normal return with no error.
04	The specified line is busy with a previously-specified operation.
08	Relative line number is too large.
0C	Invalid text code. The value specified by the X operand of the ONLST macro is undefined.
10	Illegal use of ONLTST macro.
14	Line error at OPEN time.

The nonzero return codes set for BTRD are:

Hex Code	Meaning
04	RMSR file not ready or RF=NO. This return code can occur if the DOS EREP program is running in one partition and a BTRD macro is issued in another partition.
08	RMSR=NO was coded in BTMOD.

The return codes set for RESETPL are:

Hex Code	Meaning
00	RESETPL (Reset Poll Logic) is included in the BTMOD.
04	No RESETPL logic is included in the BTMOD.

Note: If the user has specified RESETPL=NO, but wishes Reset Poll logic, he must utilize the RESETPL macro.

Completion Codes: Upon termination of a Read, Write, or Control operation, BTAM sets a completion code in bits 0 to 7 of the ECB that indicates the nature of the termination.

Warning: BTAM 'ANDs' the four bit off before going to its message writer to output an error message. Care should be taken to test for the four bit (X'40') being on if the Physical Transient Overlap option is used or if no WAIT is used.

Hex Code	Meaning
7F	Normal completion. BTAM has not recognized any I/O failure due to the hardware or the problem program.
41	I/O error. The following information will be posted in the DECB:

Bytes 16-17: For the 2701, byte 16 is the sense byte (if ERP is present) for diagnostic Write/Read. For the other control units the sense byte is posted in byte 16.

Bytes 18-19: Residual count of the channel command in error.

Byte 20: Command code of the channel command in error.

Byte 28: TP Code from the channel command in error.

Byte 29: Error Information Field (posted only when ERP is present, except for bit 1, which is posted as an "unknown error" bit when ERP is absent).

Bytes 30-31: Status Bytes.

Note: For binary synchronous communication, byte 24 (Flag byte) is also posted when EOT or DLE,EOT is received. This may occur in conjunction with irrecoverable errors or normal completions (7F).

42 READ macro completed with no text transfer (local 2260 only). This completion code is posted if the local 2260 operator neglects to press the START key before keying in text and pressing the ENTER key or if the START key is depressed immediately before the ENTER key.

Note: Completion code 42 is also posted if an RFT message is received on a Read Initial operation but no TWAIT macro was issued by the BSC or local 3270 application. In this case the problem program should now issue a TWAIT macro on the ECB without removing the X'42' completion code from the ECB.

44 Received ID did not compare. Applies to WRITE TI for WU Model 33/35 TWX Terminals, to binary synchronous communication, and to READ TE for World Trade telegraph terminals. It indicates that the ID received on a dial operation or WTTA operation did not compare with the expected ID defined in the terminal list. The condition could result from:

- Op Specification of an incorrect ID or telephone number in the terminal list, or
- A transmission error which resulted in one or more of the ID characters being garbled, or
- A transmission error or switching exchange error which resulted in dialing an incorrect number (the wrong terminal).

Note: Completion code 44 indicates, for BSC, that initial contact was not successfully established between stations on leased, dial, or multipoint lines due to a garbled "bid" (ENQ) or an invalid or negative response to a bid for the line.

48 Channel program halted as a result of a RESETPL macro instruction being issued. A Halt I/O was issued to terminate a Prepare or Enable command.

50 Contention. Indicates that the slave CPU lost the line to the master CPU while attempting to transmit. Contention for World Trade telegraph terminals (refer to the Device Dependent Considerations sections).

54 Nonproductive operation that indicates:

- A negative response to polling has been received from the terminal represented by the last active (non-skip) entry in an open polling list (OPENLST);
- For the local 2260 and local 3270, RESETPL has terminated an outstanding READ Initial before the read has started;
- A negative response to polling has been received following a RESETPL instruction;
- All of the entries in a wraparound polling list (WRAPLST) are inactive (all skip bits are ON); or
- A negative response to polling has been received from the last entry in an Auto Poll list. This completion code is used for Auto Poll of AUTOLST and SSALST types of lists and for Auto Poll of AUTOWLST and SSAWLST types when a RESETPL has been issued to terminate polling.
- A negative response to addressing (start-stop only) has been received.

Note: Condition 3 can occur only as a result of one or more skip bits being turned ON after initiation of a polling operation (READ TI for all terminals, except TWX terminals, 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. For the 2740 M2 any one of six conditions can cause X'54' to be posted. Each of these conditions can be detected by checking byte 26 of the DECB for the following codes:

X'02' - terminal in enter and communicate mode

X'20' - terminal in communicate mode with document device down

X'10' - terminal in local mode

X'13' - terminal in communicate mode but out of paper

X'08' - terminal in communicate mode with printer busy

X'04' - terminal in bid mode

58 Cancel condition detected.

60 Alternating Acknowledgments. Indicates that an incorrect positive response was received. (ACK-0 instead of ACK-1 or vice versa).

61 WACK Received. Indicates the WAIT-Before-Transmitting characters were received as a reply to a message or in response to ENQ.

Note: This completion code is also posted. The RVI bit in the DECB (bit 6 of byte 8) is turned on when an RVI is received in response to a selection on a multipoint line.

62 Enquiry Character (ENQ) received in place of a message. This means the last acknowledgment sequence should be retransmitted to the remote CPU. If the ENQ character is received in response to a text message sent, this completion code will also be posted. This is an abnormal line control procedure that can occur in binary synchronous communication when "conversational" data exchange occurs (see WRITE Initial Conversational in the CPU-to-CPU Contention section). When an ENQ is received in response to an ENQ after initial connection has already been successfully established, this completion code is also used to inform the user of the abnormal condition.

When the ENQ is expected in CPU-to-CPU communication (Contention and Dial systems) for the purpose of establishing initial connection, this completion code is not used.

64 Applicable only to the local 3270. Indicates that the contents of the terminal buffer are unreliable due to RFT processing. This completion code can be returned after a READ Initial. The application program should reinitialize the terminal buffer with a WRITE Erase specifying a data stream containing the data that should be in the terminal buffer.

SERVICEABILITY FACILITIES

BTAM provides the user with serviceability facilities on an optional basis. The user specifies which facilities will be included in the BTAM logic module by coding the appropriate operands in the BTMOD and DTFBT macro instructions.

START-STOP ERROR RECOVERY PROCEDURES

Error recovery procedures (ERP) are provided on a line group basis. ERP is a standard facility (default options ERLOGIC=E in BTMOD and ERROPT=E in DTFBT); however, the user may omit ERP by coding ERLOGIC=N in BTMOD and ERROPT=N in DTFBT and write his own ERP, making use of the information posted in the DECB, except for binary synchronous communication and audio response where ERP is mandatory.

If the operand ERLOGIC=E is coded in BTMOD or omitted, the user may specify the type of error recovery he wants by coding in the DTFBT:

1. ERROPT=E for error recovery alone.
2. ERROPT=W for ERP with write-text retry.
3. ERROPT=R for ERP with read-text retry.
4. ERROPT=RW for ERP with full-text retry.

BTAM will retry transmission when feasible.

When an interruption occurs in a channel program initiated by BTAM, the supervisor gives control to the BTAM Interruption Handler. Note that in the event of a channel control check or an interface control check, the supervisor cancels the job (see the DOS Supervisor and Input-Output Macros publication.

BTAM Interruption Handler checks the conditions of the interruption. If the interruption is a normal condition, BTAM Interruption Handler takes the proper action (i.e., post completion, restart polling, etc.).

If the interruption is an abnormal condition and ERP is specified, BTAM tries to recover from the error. If the attempt to recover is successful, normal completion is posted in the DECB. If the attempt to recover is unsuccessful, a system-to-operator message is typed. The format of the message is as follows:

```
             C
[pp] 4BnnI P Text SYSyyy=mmm
TI=xxxx DC=hhhhhhhhhh DECB=aaaaaa
[pp] 4B99I CSW**=eeeeeeeeeeeee
CCW=dddddddddddddddd SN=ffff
```

If BTAM RMSR is being used, the message format is as follows:

```
[pp] 4BnnI P TEXT SYSyyy=mmm
TI=xxxx DC=hhhhhhhhhh
```

- pp for partition indicator, supplied by DOS only with a multiprogramming supervisor.
- 4B specifies BTAM message.
- nn for the decimal error code (Figure 19).
- I for operator's action (information)
- C indicates action (cancll).
- P indicates action (post).
- text for the explanation of the error.
- yyy specifies the symbolic unit.
- mmm specifies the line associated with the symbolic unit.
- aaaaaa for the DECB address in hexadecimal.
- xxxx for the terminal identification in hexadecimal. This field always indicates the first terminal in the polling list if the error occurs on a start/stop Autopoll line. This field is meaningless and should be disregarded for a DIAL line and following devices: Audio Response Units, TWX terminals, 2740 without station control, and all point-to-point BSC devices.
- hhhhhhhh for the dial characters.
- eeeeeeeeeee for the last seven bytes of the CSW in hexadecimal.
- ddd...d for the CCW in error.

Note: The CCW contents are as follows:

Bytes	Contents
0	Command code
1-3	Data address
4	Flags
5	TP code
6-7	Byte count

- ffff for the contents of the Sense byte.

Figure 19 gives the code and explanation of the ERP messages.

Note: Messages 39-42 indicate errors that occurred in the terminal on the previous message sent. Normal completion (X'7F') is posted with these messages because no errors indicated by messages 39-42 are detected by the terminal during a buffer print operation after the terminal has responded with Y to a message.

If the operand CANCEL=YES is coded in BTMOD (see BTAM Cancellation), the job is canceled for:

- Channel Data check
- Program check
- Protection check
- Command reject

The conditions of the error are posted in the DECB.

Error Detection and Error Actions

Error conditions or unusual conditions are detected by BTAM by examining the status bits in the channel status word (CSW), and, in the case of unit check, by examining the sense bits.

More than one indicator may be on in the CSW when error conditions occur. Generally, only one of these conditions properly describes the condition; the others indicate secondary effects. Similarly, control unit errors can cause more than one sense bit to be present. To insure that the primary condition is recognized and acted upon, the priority schemes indicated in Figures 20 and 21 determine the order of testing status and sense bits. The actions indicated in the figures are defined in the following subsection on actions.

Actions:

1. A system-to-operator message is provided.

Error Code	Text	Explanation
20	ERR IN ERP	Error in ERP
21	CHAN DATCK	Channel data check
22	SHOULD NOT	Should not occur error
23	CHAIN CHK	Chaining check
24	PROGRAM CK	Program check
25	PROTECT CK	Protection check
26	UNIT EXECP	Unit exception
27	EQUIPMT CK	Equipment check
28	LOST DATA	Lost data
29	TIME OUT	Time out
30	INTERVREQ	Intervention required
31	BUS OUT CK	Bus out check
32	DATA CHECK	Data check
33	OVERRUN	Overrun
34	COMMAND RJ	Command Reject
35	STX ENQ	Error in actions of problem program (if terminal is not 2780 or Model 20) Hardware malfunction on 2780 or Model 20 (see Operating Guide or 2780 Component Description)
36	STX T ENQ	Error in actions of problem program (if terminal is not 2780) Line buffer parity check or line buffer overrun on 2780
37	EOT RESPN	Error in actions of problem program (if terminal is not 2780 or Model 20) Hardware malfunction on 2780 or Model 20 (see Operator's Guide or 2780 Component Description.)
38	HDW BF OFT	EOT received in response to text
39	TRM EL ERR	2740 M2 Electronic hardware failure for last message received
40	TRM IO ERR	2740 M2 I/O hardware failure for last message received
41	VRC ERROR	2740 M2 Line MRC error for last message received
42	PARITY ERR	2740 M2 Terminal line parity error induced by terminal on a transmitted message
43	CONTROL CHK	3270 Timeout check condition detected
44	OPRATN CHK	3270 Programming error in writing data stream
50	L ERROR THRS	Line error threshold reached. The specified values for data checks, time-outs, transmission errors, and intervention required errors have been reached.
99	-- --	Appears with each of the above message codes except 39-42 to give more information.

Figure 19. Error Messages

2. A system-to-operator message is provided. The error indicates a probable program error.

3. A system-to-operator message is provided; bit 1 is set in the DECB (indicating that the system detected an I/O error that is undefined for the particular command or device).

4. If this is not a Read command, go to Action 3. If this is a Read command, a system-to-operator message is provided.

5. A system-to-operator message is provided. The error indicates a control unit failure. The line should be considered inoperative.

6. If this is not a Read command, go to Action 3. If this is a Read command and read retries are specified, a Read Repeat channel program is executed to retry the operation. On the third occurrence of the error (on the first occurrence if read retries are not specified), a system-to-operator message is provided.

7. A Read-skip command is executed to clear the line. After the Read skip completes, the original channel program is reexecuted beginning with the CCW that failed. On the third occurrence of the error, a system-to-operator message is provided. The error indicates that the line is receiving without a command.

8. If this is a Read response to polling or addressing, this is a normal condition indicating negative response. If this is a Read text, this is a normal condition indicating that an EOT sequence was received.

   If this is a Read response to text and the terminal is not a 2260, go to Action 3. If it is a 2260 and write

Priority	Bit	Condition	Action
1	44	Channel Data Check	1
2	32	Attention	3
2	33	Status Modifier	3
2	34	Control Unit End	3
2	35	Busy	3
3	38	Unit Check	See Figure 21
4	47	Chaining Check	4
5	42	Program Check	2
5	43	Protection Check	2
6	39	Unit Exception	See Figure 23
7	41	Incorrect Length	See Figure 24

Figure 20. Status Analysis (Start-Stop)

Priority	Sense Bit	Condition	Action
1	3	Equipment Check	5
2	6	Lost Data	See Figure 22
3	7	Time Out	See Figure 25
4	1	Intervention	1
5	2	Bus Out	See Figure 26
6	4	Data Check	See Figure 27
7	5	Overrun	6
8	0	Command Reject	1

Figure 21. Sense Byte Analysis (Start-Stop)

Device / Command	1030 1050 1060	83B3 115A	TWX 33/35	2740 2760	2260R
Write	3	3	3	3	3
Read	9	9	9	9	9
Sadxxx	3	3	3	3	3
Disable	3	3	3	3	3
Enable	3	3	3	3	3
Dial	18	--	18	18	--
Prepare	--	--	--	3	--
Break	--	3	--	--	--
Poll	3	--	--	3	--

Figure 22. Lost Data

Device / Command	1030 1050 1060	83B3 115A	TWX 33/35	2740 2760	2260R
Write	7	7	7	7	7
Read	8	8	8	8	8
Sadxxx	3	3	3	3	--
Disable	3	3	3	3	3
Enable	3	3	3	3	3
Dial	3	--	3	3	--
Prepare	--	--	--	3	--
Break	--	3	--	--	--
Poll	7	--	--	7	--

Figure 23. Unit Exception (Start-Stop)

Device / Command	83B3 115A	All Others
Read (response to polling)	--	3
All others	3	3

Figure 24. Incorrect Length (Start-Stop)

retries are specified, a Write continue channel program is executed to retry the operation. On the third occurrence of the error, a system-to-operator message is provided. The error indicates a buffer overflow on the 1053 printer.

9. If this is not a Read text, go to Action 3. If it is a Read text, a Read-skip command is executed and a system-to-operator message is provided. The error indicates that the input message was larger than the input area.

10. If this is a Read response to polling or addressing or a Poll, the selection sequence is retried. On the third occurrence of this condition, a system-to-operator message is provided. If line error counts are specified, each occurrence of the error is recorded in the line error block.

Command \ Device	1030 1050 1060	83B3 115A	TWX 33/35	2740 2760	2260R
Write	3	3	3	3	3
Read	10	10	10	10	10
Sadxxx	3	3	3	3	--
Disable	11	3	11	11	3
Enable	11	3	3	11	3
Dial	12	--	12	12	--
Prepare	--	--	--	12	--
Break	--	3	--	--	--
Poll	10	--	--	10	--

Figure 25. Time Out (Start-Stop)

Command \ Device	1030 1050 1060	83B3 115A	TWX 33/35	2740 2760	2260R
Write	3	3	3	3	3
Read	3	3	3	3	3
Sadxxx	3	3	3	3	--
Disable	3	3	3	3	3
Enable	3	3	3	3	3
Dial	3	--	3	3	--
Prepare	--	--	--	3	--
Break	--	4	--	--	--
Poll	3	--	--	3	--

Figure 26. Bus Out Check (Start-Stop)

Command \ Device	1030 1050 1060	83B3 115A	TWX 33/35	2740 2760	2260R
Write	4	5	5	3	3
Read	6	6	6	6	6
Sadxxx	3	3	3	3	--
Disable	3	3	3	3	3
Enable	3	3	3	3	3
Dial	3	--	3	3	--
Prepare	--	--	--	3	--
Break	--	7	--	--	--
Poll	4	--	--	4	--

Figure 27. Data Check (Start-Stop)

If this is a Read response to text and write retries are specified, the operation is retried beginning with the selection sequence. On the third occurrence of this condition, a system-to-operator message is provided. If line error counts are specified, each occurrence of the error is recorded in the line error block.

If this is a Read text, the operation is posted complete with error. If this is a Read response to polling or addressing, there was no response. For Read response to text, there was no response. For Read text, no text was received or the time between text characters was greater than 28 seconds (inter-character time-out).

11. The command is retried. On the third occurrence of this condition, a system-to-operator message is provided. If line error counts are specified, each occurrence is recorded in the line error block. The error indicates that the data set is failing to disconnect.

12. The channel program is retried. On the third occurrence of this error, a system-to-operator message is provided. If line error counts are specified, each occurrence is recorded in the line error block. The error indicates that the terminal is not answering in the time allotted.

13. If the error occurred prior to text transfer, the channel program is

retried. On the third occurrence of this condition, a system-to-operator message is provided. If this is a Write text and write retries are specified, the failing command is reexecuted. On the third occurrence of the error (on the first occurrence if write retries are not specified), a system-to-operator message is provided. The error indicates a parity error either on the command or on the data.

14. A system-to-operator message is provided. If line error counts are specified, the error is recorded in the line error block. The error indicates a transmission code parity error, or for a 2260, that data bit positions 0 and 2 are not equal.

15. If this is a Write text and write retries are specified, the channel program is retried. If line error counts are specified, each occurrence is recorded in the line error block. On the third occurrence of this condition, a system-to-operator message is provided.

    If the error occurred on a command other than Write text, the channel program is retried. If line error counts are specified, each occurrence is recorded in the line error block. On the third occurrence of this condition, a system-to-operator message is provided.

    The error indicates an echo check.

16. If the error occurred prior to text transfer, the channel program is retried beginning with the selection sequence. On the third occurrence of this condition, a system-to-operator message is provided. If line error counts are specified, each occurrence is recorded in the line error block.

    If the error occurred on a Read text and read retries are specified, a Read Repeat channel program is executed to retry the operation. If line error counts are specified, each occurrence is recorded in the line error block. On the third occurrence of the error, a system-to-operator message is provided.

    For Read response to text with write retries specified, a Write Continue channel program is executed to retry the operation. If line error counts are specified, each occurrence is recorded in the line error block. On the third occurrence of this condition, a system-to-operator message is provided.

    The error indicates one of the following:

    a. A VRC (parity) error was detected in one or more of the received characters.

    b. A LRC error was detected (the LRC character transmitted by the terminal did not match the LRC character generated by the control unit).

    c. A circle N or NAK was received as a response to text.

    d. The line was at space at stop bit time (control unit out of sync).

17. A system-to-operator message is provided. If line error counts are specified, the error is recorded in the line error block. The error indicates an echo check on the line.

18. The channel program is retried. On the third occurrence of this condition, a system-to-operator message is provided. The error indicates that a dial command was issued to a line that is already "off-hook."

START-STOP ERROR RECOVERY SUGGESTIONS

Use of the Error Recovery Procedures is strongly recommended. The operation in error is retried if possible, and the error conditions are posted in the DECB. System-to-operator messages inform the operator when an error cannot be recovered or when the number of retries per terminal exceed a user-provided threshold.

This section provides suggestions for the DOS BTAM user who intends to write his own error recovery routines. These suggestions must be followed carefully. However, the user must be aware that the lack of automatic retries for the operations in error would have a large incidence on the throughput of the system. Lack of information provided to the operator inhibits proper maintenance of the system.

The Standard Error Information posted in the DECB provides enough information to identify the error.

If the interruption is an abnormal condition, the BTAM Interruption Handler takes the following actions:

- Stores the status bytes in bytes 30 and 31 of the DECB.
- Stores the command code of the channel command in error in byte 20 of the DECB.
- Stores the TP code of the channel command in error in byte 28 of the DECB.
- Stores the residual count in bytes 18-19 of the DECB.
- Stores the sense byte in byte 16 of the DECB if the status contains a unit check.
- Posts a completion code of hexadecimal 41 in byte 0 of the DECB.
- Returns control to the supervisor.

When the user does not request Error Recovery Procedures to be included in BTMOD, it is his responsibility to check the DECB. The user should check the status bytes (Figure 28). Generally, only one of these indicators properly describes the condition; the others indicate the secondary effects.

When a unit check occurs, the sense byte (Figure 29) is stored in byte 16 of the DECB. This byte should be checked on a unit check condition.

Error Counts

If the user wishes to keep error counts with his ERP on a line group basis, he must issue an LERB macro instruction to generate and initialize a Line Error Block and he must code ERLOGIC=C in the BTMOD macro instruction. All error counts referring to a particular DTFBT are kept in one Line Error Block.

Bit	Condition
0	Attention
1	Status modifier
2	Control unit end
3	Busy
4	Channel end
5	Device end
6	Unit check
7	Unit exception
8	Program-controlled interruption
9	Incorrect length
10	Program check
11	Protection check
12	Channel data check
13	Channel control check
14	Interface control check
15	Chaining check

Figure 28. DECB Status Bytes

Bit	Explanation
0	Command reject
1	Intervention required
2	Bus out check
3	Equipment check
4	Data check
5	Overrun
6	Lost data
7	Timeout

Figure 29. DECB Sense Byte

The linkage between the DTFBT and the Line Error Block is indicated by the LERBADR operand in the DTFBT macro instruction.

If any of the error counters reach their threshold values before the transmission threshold value is reached, a message will be printed on the System/370 console typewriter. The message identifies the line and contains the three error counters and the transmission counter, together with their respective threshold values.

After the message is printed, the transmission counter and the three error counters are added to a separate set of four cumulative counters and the counters are reset. If the transmission counter reaches its threshold before any of the error counts reach their thresholds, the four counters are added to the cumulative counters and the counters are reset, but no message is printed.

To print the cumulative counters, the user may issue a LERPRT macro instruction. When the macro instruction is issued, the current values of the error counters and the transmission counter are added to the set of four cumulative counters for the line. LERPRT may be used to print the cumulative counters for one line or for the entire line group. If the user does not want his cumulative counters to be set to zero after a LERPRT macro instruction, he must code CLEAR=NO in the macro instruction.

BSC ERROR RECOVERY PROCEDURES

The error recovery procedures (ERP) for binary synchronous communication are mandatory and are automatically included in the BTAM logic module. This is in contrast to the ERP for start-stop devices, which are optionally included when specified by the ERLOGIC operand in the BTMOD macro. Thus the ERLOGIC operand has no bearing on inclusion of binary synchronous communication ERP in the BTAM module. The only relevance of ERLOGIC to binary synchronous communication consists in the C and NC options which cause the Error Count Rou-

Command \ Error Condition	Channel Data Check	Lost Data	Time Out	Intervention Required	Bus Out Check	Data Check	Overrun	Command Reject	Unit Exception	Equipment Check
Set Mode	2	2	2	2	2	2	2	11	-	2
Enable	2	2	1	2	2	2	2	11	-	2
Dial	2	4	4	4	8	2	2	11	-	2
Disable	2	2	6	2	2	2	2	2	2	2
Prepare	2	2	2	7	2	2	2	11	-	2
Write	2	2	2	7	1	2	2	10	12	2
Read	2	3	5	7	2	9	9	11	-	2
Poll	2	2	2	7	2	2	2	11	12	2
Sense	1	1	1	1	1	1	1	1	1	1

Figure 30. Guide to Error Recovery Actions Taken by BTAM (Binary Synchronous)

tines (common to both start-stop and binary synchronous communication) to be included in the BTAM module.

The purpose of these error recovery procedures is first to analyze status and sense information when I/O interruptions occur in conjunction with or as the result of errors in the initiation or execution of channel program commands. Then, if recovery of the error condition is possible, a retry procedure is attempted. The number of retries is specified by the user via the RETRY operand in the DTFBT macro instruction. When the error condition is of a class defined as unrecoverable or when the maximum number of retries fails to produce recovery, a system-to-operator message is provided, indicating the nature of the error. In addition, an error indication is posted in the appropriate DECB.

The entries in Figure 30 refer to the actions taken by ERP when listed error conditions occur during the initiation or execution of the I/O commands.

In the following description of the actions pointed to in Figure 30, n is the number of retries specified by the RETRY operand in the DTFBT macro instruction issued for the line group.

Action 1: Unrecoverable Error

A system-to-operator message is printed, indicating the nature of the error. The operation is posted complete with error (completion code: 41) and control is then returned to the problem program.

Action 2: Should-Not-Occur Error

The channel problem is retried n times if the error occurs before data transfer. If n retries fail, or if data transfer is taking place, Action 1 is performed. If data transfer is completed, the channel command is retried after the appropriate preliminary commands are executed (for example, Write ENQ, Read response).

Action 3: Lost Data on Read

If the error occurs on a text-read and the last character received is not an end character, Read commands with counts of one are issued to read the ENQ character. After the ENQ character is received, a READ Repeat is issued. If n retries fail, Action 1 is performed.

If the last character received is an end character, a READ repeat is immediately issued. Action 1 is performed if n retries fail.

Action 4: Errors on Dial

If Lost Data or Intervention Required occur during a Dial command, a Disable command is issued and the dialing operation is retried. If n retries fail, Action 1 is performed.

If time-out occurs on Dial, a Disable is issued and the operation is posted complete with a time-out indication in the DECB. Since this condition merely means that no answer to the call was received, no retries are attempted and no error message is provided.

Action 5: Time-Out on Read

If time-out occurs on a text-read, Read commands with counts of two are tried. If ENQ or DLE, ENQ is received, a READ Repeat is issued. If an EOT is received or if n consecutive retries fail, Action 1 is performed.

If time-out occurs on a Read response, a WRITE inquiry is issued. If n retries fail, Action 1 is performed.

Action 6: Time-Out on Disable

The Disable command is retried up to n times. If the error condition persists, Action 1 is performed.

Action 7: Intervention Required

If this error condition occurs on Prepare or Poll, a Disable command is issued. Then a Set Mode, Enable, and a retry of the channel program are executed. If the error condition persists, Action 1 is performed.

If the error condition occurs on Read or Write, Action 1 is performed.

Action 8: Bus Out Check on Dial

A DLE, EOT sequence is transmitted, a Disable and Set Mode are issued, and the Dial command is retried. If n retries fail to recover from the error condition, Action 1 is performed.

Action 9: Data Check or Overrun on Read

If the error occurs on a text-read, a READ Repeat is issued. If n retries fail, Action 1 is performed. If the error occurs on a Read response, a WRITE Inquiry is issued. After n unsuccessful retires, Action 1 is performed.

Action 10: Command Reject on Write

An Enable command is issued and then Action 1 is performed.

Action 11: Command Reject

A DLE, ENQ sequence is transmitted, and then Action 1 is performed.

Action 12: Unit Exception on Write or Poll

A Read command with a maximum count and the skip flag on is executed if the operation is a Poll or a Write response to text. If the operation is a line bid (Write ENQ), the read response CCW is executed. If the condition exists after n retries, Action 1 is performed.

In addition to the above actions, BTAM provides automatic retransmission of a message when a NAK response to text is received from the remote device. If n retries fail to recover, completion code 41 is set and bit 7 of byte 29 (error information field) is set to one.

If a message of the form:

STX--Text(may be zero length)--ENQ

is received, BTAM automatically responds by transmitting NAK to the remote device. If n retries fail to recover, BTAM performs Action 1. Completion code 41 is posted and bit 6 of the error information field (byte 29 of the DECB) is turned ON. Generally, n retries are not necessary, because the remote device sends EOT upon receiving the NAK. This is the standard procedure, for example, in communication with the 2780. The 2780 transmits a message of the above format in lieu of a record causing a card reader error or an internal buffer parity check. In this case, BTAM posts completion code 41, turns ON bit 6 of byte 29, and turns ON bit 1 (EOT) of the Flag byte in the DECB.

Note: A message of the above format should not occur in CPU-to-CPU communication unless (user error) the ENQ character is transmitted within text in nontransparent mode.

STANDARD ERP FOR THE 7770 AUDIO RESPONSE UNIT

The following facilities are provided for an audio response unit whether the error recovery procedures have been requested or not.

1. Hung up procedure: Whenever the unit exception bit is set in the status received at the end of an I/O operation for a 7770 (indicating that the inquirer hung up during a transaction) and the unit check bit has not been set, the line is disconnected and one of the following procedures is followed:

   a. An initial operation, that is, one starting with an Enable command (Read Initial, Write Initial and Write Invitational) is restarted and no completion posting is made.

   b. An operation specified with reset is posted normally at the completion of the reset function.

   c. If neither of the previous cases apply, the operation is posted with the I/O error completion code and the unit exception bit is set in the unit status byte field of the DECB.

2. Overlength procedure: Whenever BTAM detects that the inquirer has attempted to enter on a read operation an inquiry longer than specified in the length field of the DECB or MSGL field in the DTFBT, BTAM posts the operation with the I/O error completion code and the incorrect length indication bit set in the channel status byte field of the DECB. However, if there is an indication that the inquirer hung up, the overlength indication is disregarded.

BTAM PROGRAM CANCELLATIONS

BTAM may detect a program condition that makes it difficult or impossible to continue execution. In this situation, BTAM posts pertinent information in the cancel block and takes the proper action according to which BTAM logic routine is being executed.

The BTMOD cancel block is as follows:

Bytes	0	1	2	3
	Error Code	DTFBT Address		
	Reserved	DECB Address		
	4	5	6	7

If the BTAM interruption handler has control and a potential cancel condition is detected, the following actions are taken:

- The current operation is discontinued.
- A numeric error code, the DTFBT address, and the DECB address are posted to the BTMOD cancel block.
- A completion code of X'58' is posted to the ECB.

If any other BTAM routine has control and a potential cancel condition is detected, the following actions are taken:

- The execution of the BTAM routine that detected the cancel condition is discontinued is shown below.
- If the cancel condition has not been posted to the cancel block, a numeric error code, the DTFBT address, and the DECB address are posted to the cancel block.
- An operator-awareness message is written on SYSLOG (see format and messages
- If CANCEL=YES was coded in the BTMOD macro instruction, a DOS CANCEL macro instruction is given.
- If CANCEL=NO was coded in the BTMOD macro instruction, a return code of X'01' is set in general register 15, the condition code is set to non-zero, and general register 1 points to the location containing the address of the BTMOD cancel block. Control is then returned to the user at the instruction following the BTAM macro instruction that caused the cancel condition to be detected.

The format of the operator awareness message is:

```
            P
[pp] 4BnnI C text DTFBT=aaaaaa,
DECB=aaaaaa
```

where:

pp
    is partition indicator, supplied by

DOS when a multiprogramming supervisor is used.

nn
: is the decimal error code.

p
: indicates posting of information and return to user.

c
: indicates a Cancel message.

text
: is the explanation of the error.

aaaaaa
: is the address in hexadecimal.

The BTAM cancel messages are:

Error Code	Text
00	User referred to closed DTFBT
01	DTFBT field improperly initialized
02	DECB field improperly initialized
03	Multiple-wait count negative
04	Multiple-wait count exceeds ECBLIST size
05	Attempt to process non-BTAM buffer
06	Unexpected program error in RELBUF
07	REQBUF count negative
08	RESETPL DECB and LCB DECB not same

Note: If either the DTFBT address or DECB address is not available or is not pertinent at the point at which the potential cancel condition is detected, the corresponding cancel block field will contain zero.

Programming Note: If the user who selected the BTMOD parameter CANCEL=NO elects to continue execution despite the error condition, he must store X'FF' in Byte 0 of the Cancel Block before returning control to BTAM.

START-STOP ONLINE TERMINAL TEST

The online terminal test facility is an optional service provided only if:

1. TERMTST=YES is coded in the BTMOD macro issued for the problem program.
2. TERMTST=YES is coded in the DTFBT macro issued to define the particular line group concerned.
3. TERMTST is coded in the TWAIT macro.

The online terminal test facility services test requests initiated at the remote device by either the operator or an IBM Customer Engineer. The test requests may be used as startup procedures or for terminal checkout and terminal failure diagnosis. The tests provided operate on-line with the problem program and affect the user's problem-program operation only to the extent that line time is required by the test facility to perform the requested functions over the selected line, and CPU time is necessary to set up the requested test.

Requesting of Tests

Specific test requests are initiated only at the remote device by the transmission to the CPU of a message having a special format. The tests requested may involve message switching, comparing incoming data to a stored pattern in main storage, sending a set of characters to a specified terminal, or checking the IBM SELECTRIC typewriter element or carriage mechanisms on a remote terminal. The format of the message requesting a test is shown below. Framing characters (for example, STX) are not shown in the format.

99999	XX	FROM TYPE	TO ADDR	UNIT SELECT	TEXT	END CHARACTER

99999 (5 characters): This is the primary action code used to recognize this message as a test request.

XX (2 characters): This is the module-sequence code used to define the specific test to be executed.

FROM
TYPE (1 character): Used by the terminal-test modules to determine the type of terminal from which the test is being requested.

FROM TYPE	Device
1	1030 card reader
2	1050
3	1060
4	2740 (with or without 2760 attached)
5	1030 badge reader or manual entry

6 2260 (Remote) and 2265 (Remote)

TO
ADDR (0, 1, or 2 characters): This is the address of the terminal to which the message is to be sent.

TO ADDR Usage: This is a one-byte field for the IBM 1030 Card Reader, 1050, 2740S, and 2740SC, giving the addressing character for the selected terminal. For other 2740 combinations, the field is omitted.

The TO ADDR for 1030 badge readers and manual entry, 1060 devices, 2260, and 2265 devices is a two-byte field. The address of the receiving terminal is selected by transmitting a predefined code in these character positions. The code is as follows:

1. 1060:

   01 will define terminal A as the receiving terminal.
   02 = terminal B
   .   .
   .   .
   .   .
   26 = terminal Z

2. 1030 Badge Readers or Manual Entry:

   02 will define terminal B as the receiving terminal.
   03 = terminal C
   04 = terminal D
   .   .
   .   .
   .   .
   26 = terminal Z

   Note: If 10 is entered as the address field, this will be considered an invalid request as the corresponding address (J) is not a legal 1030 address.

3. 2848 devices:

   The TO ADDR is used to select the 2848 display control unit. The address of a display control unit can be any USASCII non-control character (i.e., any character in columns 3-7 in the USASCII code chart), therefore allowing 96 possible display-control addresses.

   Note: The TO ADDR applicable to a particular display control unit can be determined from one of its attached display stations by specifying the request-address test in the test message.

Actual 2848 USASCII address	TO ADDR (2 bytes)
$b^7...b^1$	
0100000	01
0100001	02
.	.
.	.
.	.
1111111	96

UNIT
SELECT (0, 1, or 2 characters)

Unit select is not applicable to 1030 and 2740 devices; therefore, text can start in this position.

1. 1050 and 1060 devices (1 character):

   This specifies the particular unit at the TO ADDR that is to receive the message, i.e., 1052, 1053, 1055, 1062 Printer 1 or 2, etc. The appropriate unit-select code can be determined from the publication related to the type of terminal being addressed.

2. 2260 and 2265 (or 1053 attached to the 2848 or 2845) devices (2 characters):

   2260 and 2265 Display Stations and 1053 Printers are selected by transmitting a predefined code in these character positions. The device selection can be one of twenty-five USASCII non-control characters.

Actual Unit Address	UNIT SELECT (2 bytes)
$b^7...b^1$	
1000000	01
1000001	02
.	.
.	.
.	.
1011000	25

The screen address for the 2845/2265 is 26. The printer address is 27.

Note: The predefined 2-byte code applicable to a particular display station can be determined from that display station itself by utilizing the request-address test.

END
CHARACTER

1030 = EOB
1050 = EOT
1060 = EOB
2740 = EOT
2848 = ETX

Note: The test message as transmitted from a 1060 device is entered by utilizing the data and transaction keys. The EOB character is entered by depressing the teller A or B key.

Applicable Module Sequence Codes for 1030, 1050, 1060, 2740, 2260, and 2265:

<u>01</u> <u>Message Switching</u>

This module receives a message from the requesting terminal and transmits it to the terminal (on the same line) as specified in the test message.

<u>Note</u>: The number of characters that can be switched is directly dependent upon the count field that the user specified in the problem program READ macro for the line over which the test is requested. Only this number of characters can be switched.

<u>02</u> <u>Tilt</u>

This module sends the tilt test to the requested terminal. This test is designed to check the SELECTRIC typewriter print element.

<u>03</u> <u>Rotate</u>

This module sends the rotate test to the requested terminal. This test is designed to check the SELECTRIC typewriter print element.

<u>04</u> <u>Twist</u>

This module sends the twist test to the requested terminal. This test is designed to check the SELECTRIC typewriter print element.

<u>05</u> <u>Stored Compare</u>

This module provides a means to compare the received message with a particular pattern stored in the CPU. The defined message in main storage is compatible with the transmitting capabilities of the terminal(s) involved.

The test message to be compared with the stored pattern 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, although not all of them need be specified.

The number of characters that can be compared is directly dependent upon the count field that the user specified in the problem program READ macro for the line. The I/O area must be long enough to contain the header information (99999, etc.), the characters to be compared, and the end character.

Exceptions:

1. When transmitting from a 2740 terminal with Station Control, a space character must precede the comparison data. For the 2740 Basic or 2740 with Transmit Control, two space characters are needed.

2. The comparison test for a 1060 is requested by entering the following message:

999996534210	TELLER A or B

Comparison is then made to this message. Responses to this request are printed <u>only</u> at the requesting terminal.

Messages received at the terminal are:

1. If the comparison to the stored message is valid, the following message is sent to the terminal specified in the TO ADDR field:

CMP VLD-*

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:

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.

2. If the comparison to the stored message is invalid, the data received is merely message-switched to the terminal specified in the TO ADDR.

<u>Note 1</u>: For a 2740 with checking where the ADDR field is omitted, the message is sent back to the terminal requesting the test when the stored message is invalid.

<u>Note 2</u>: The <u>stored</u> <u>compare</u> test is <u>not</u> applicable to the 1030 manual entry or badge reader.

06 <u>All Characters Test</u>

This module provides a standard all-characters test for CE terminal checkout and serves as a start-up message for the customer. Special characters are <u>not</u> used in the terminal test. Characters received at the terminal are:

1. 1030, 1060, 2848 (2260 and 1053) Numbers: 0-9, and alphabet: A-Z.
2. 1050, 2740 Numbers: 0-9, alphabet: A-Z (lowercase), and A-Z (uppercase).

07 <u>SELECTRIC Analyzer Test</u>

This module provides an exerise to analyze the capability of the SELECTRIC-typewriter-carriage mechanism to perform within defined specifications. The defined message in storage, when transmitted, will have the ability to so exercise a requested terminal. This test is not applicable to a 1053 Printer attached to a remote 2848 or 2845 Control Unit.

08 <u>Write at Line Address Test</u> (2260 and 2265 only)

This test will provide a line-selectivity test by using the first two characters after the UNIT SELECT field as a new line code. This can be followed by data which is to be switched to the terminal and written on the line specified on the display station screen. The following characters are used to select the line on the display station screen:

01 = line #1
02 = line #2
03 = line #3
12 = line #12

09 <u>Request Address Test</u> (2260 and 2265 only)

The TO ADDR and UNIT SELECT fields are <u>not</u> 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. The 9-byte message returned to the requesting display station is in the following format:

D	C	+	D	V	DC Addr	DV Addr

DC address is the predefined code necessary to select this display control unit (2 bytes). DV address is the predefined code necessary to select this display station (2 bytes).

<u>Note</u>: This test provides the TO ADDR and UNIT SELECT codes of the requesting 2260 itself. It is not a means of getting the TO ADDR and UNIT SELECT codes for some other 2260.

10 <u>2760 Optical Image Unit Frame Change Test (2740C, 2740DC)</u> See test 11.

11 <u>2760 Optical Image Unit Scan Point Test (2740C, 2740DC)</u>

These tests are requested by entering the following message from the keyboard of the associated 2740 terminal:

$$99999xx4FA_1A_2\begin{matrix}E\\O\\T\end{matrix}$$

xx = 10 for frame change test
11 for scan point test

4 = identifies test as coming from the 2740

FA_1A_2 = the 2760 control message (See 2760 Optical Image Unit Component Description)

If a frame-change test is requested, BTAM writes a frame-change message derived from the test request to the 2760.

If a scan-point test is requested, BTAM writes a frame-change message derived from the test request to the 2760. The customer engineer may then probe one or more response points, depending on the mode specified in the test-request message. The coordinates are read and printed on the IBM 2740.

On-line Terminal tests for the 2760 will generally be run by the IBM Customer Engineer during periods of inactivity or as a start-up procedure. Once the testing is complete, the customer engineer will unload the film and the operator can continue operation.

On-line Terminal tests can also be used to perform tests after the data-entry sequence has begun. However, if such an interruption occurs, the film should be unloaded and the sequence restarted. If this is not done, the filmstrip may not be at the expected position and the data entered after the test may not be the data expected.

Terminal Test Restrictions

1. The problem program I/O area must be of sufficient length so that the entire test message can be read into main storage. If buffers are 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 message through the UNIT SELECT field. In the case of the write-at-line-address test request, the two characters defining the line address must also fit into the first buffer. The TEXT characters and END CHARACTER may also appear in the first buffer.

2. If the problem program is using an answer list, none of the reset options can be specified in the READ macros. The line connection must remain established for the terminal tests.

3. 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 1030 publications).

4. The transaction code received from 1030 devices is not included as part of the test request.

5. All 1030 tests require a 1033 Printer on the same line as the requesting terminal. The printer is specified in the TO ADDR field.

6. The terminal tests will not test 1035 Badge Readers or 1030 Badge Readers in a 1035 environment.

7. In order to use online terminal test for the 2740 M2 with checking, a READ TIR operation must be used.

8. If the test pattern area in the BTMOD is busy, no test will be performed and a READ TI (READ TV for switched lines) is started.

9. The first five bytes of the user's I/O area cannot be equal to all 9's.

ONLINE TERMINAL TEST FOR BINARY SYNCHRONOUS COMMUNICATION

The online terminal test for Binary Synchronous Communication (BSC) is an optional service provided by DOS BTAM. It is provided to ensure proper operation of the system, and it may be used in the diagnosis and correction of a BSC system malfunction.

Online testing is available for all types of remote BSC stations. For System/370-to-System/370 operation, both computers may run under DOS BTAM with the online test facility, or one may run under BTAM and the other under an online diagnostic program. Operation between a System/370 and a System/360 Model 20, 2715, 2770, 2972, 2780, remote 3270, or 1800 requires the System/370 to run under BTAM or an on-line diagnostic program. For System/370 to 1130 operation, the System/370 must run under BTAM and the 1130 must run under an online diagnostic program.

In System/370-to-System/370 operation, either computer may initiate online tests. In operations between System/370 and a System/360 Model 20, 1130, 2770, 2780, 2972, or 1800, the System/370 cannot initiate the online test except for a test type (x operand) of 00. In System/370 to 2715 and System/370 to remote 3270 operation, only the 2715 or remote 3270 can initiate an online test. When the System/370 initiates the test with a 2780, the 2780 Mode switch must be set to either PRINT or PUNCH, if the 2780 is on a point-to-point line.

Figure 31 shows the test types available for each type of remote station, and whether the RFT message that initiates the test can be sent from the central computer (indicated by "C") or from the remote station (indicated by "R"), or from both. Where a - appears, the test type is not usable for that type of remote station. Figure 32 shows the test types for both the local and remote 3270s.

The BSC online terminal test is provided on two levels. First: DOS BTAM recognizes Request-for-Test (RFT) messages transmitted by the remote unit. When an RFT message is recognized, BTAM performs the requested test, which is usually transmission of a test message. Second: DOS BTAM initiates online test by transmitting an RFT message requesting the remote unit to perform a test that is specified by the parameters of the ONLTST macro. ONLTST cannot be used unless the remote devices are types equipped to receive program-initiated online tests.

The BSC online test facility recognizes Request-for-Test messages only if:

1. The request is initiated by the application program with an ONLTST macro instruction or, if the request is initiated from a terminal, when the program is executing one of these macro instructions: READ Initial, READ Initial with Tone, READ Connect, READ Connect with Tone, READ Continue (remote 3270 only).

Test Type: specified in RFT message X-field	Content of test message	S/370	S/360 Mod 20	1130	1800	2715	2770	2780	2972
00	(User-specified)	C,R	C,R	C,R	C,R	R	C,R[1]	C,R[1]	C,R[1]
01	(User-specified)	C,R	R	R	R	R	R	R	R
02	EBCDIC, all bit patterns, transparent	C,R	R	R	R	R	R[3]	-	-
03	USASCII, all bit patterns, transparent	C,R	R	-	-	-	-	-	-
04	EBCDIC, all bit patterns except data link controls	C,R	R	R	R	-	R[3]	-	-
05	USASCII, all bit patterns except data link controls	C,R	R	-	R	-	R	R[2]	-
06	USASCII, A-Z, 0-9	C,R	R	-	R	-	R	R	-
07	USASCII, printer selection code and A-Z, 0-9	C,R[4]	R[4]	-	R[4]	-	-	R[5]	-
08	USASCII, punch selection code and A-Z, 0-9	C,R[4]	R[4]	-	R[4]	-	-	R[5]	-
09	Transcode, printer selection code and A-Z, 0-9	-	-	-	-	-	-	R[5]	-
10	Transcode, punch selection code and A-Z, 0-9	-	-	-	-	-	-	R[5]	-
11	Transcode, A-Z, 0-9	-	-	-	-	-	-	R	-
12	EBCDIC, printer selection code and A-Z, 0-9	C,R[4]	R[4]	R[4]	R[4]	-	R[4]	R[5]	-
13	EBCDIC, punch selection code and A-Z, 0-9	C,R[4]	R[4]	R[4]	R[4]	-	R[4]	R[5]	-
14	EBCDIC, A-Z, 0-9	C,R	R	R	R	-	R	R	-
15	EBCDIC, 74 NUL (X'00') characters, 6 SYN (X'32') characters	C,R	R	R	R	-	R	R	-
16	EBCDIC, 40 bytes of X'AA', 40 bytes of X'55'	C,R	R	R	R	-	R	R	-
17	Transcode, 80 SOH (X'00') characters	-	-	-	-	-	-	R	-
18	Transcode, 40 N's (X'15'), 40 ESC (X'2A') characters	-	-	-	-	-	-	R	-
19	EBCDIC, 280 NUL (X'00') characters	C,R	R	R	R	R	-	-	-
20	EBCDIC, 80 characters, U-Z 0-9, X'00'-X'3F' (transparent)	C,R	-	-	-	-	R	R	-
21	EBCDIC, 120 character A-Z, 0-9, X'00'-X'53' (transparent)	C,R	-	-	-	-	R	R	-
22	EBCDIC, 144 characters A-Z, 0-9, X'00'-X'6B' (transparent)	C,R	-	-	-	-	R	R	-

C=The test can be initiated by the program using the ONLTST macro instruction.
R=The test can be initiated from a remote station.

[1]RFT messages sent from a remote 2770, 2780, or 2972 that specify X=00 must specify a transmission count (Y-field) of 1.
[2]Printer only. The print chain must be at least 120 characters.
[3]2770's with expanded buffer capability.
[4]The printer and punch codes apply to the 2780 only; the devices that contain programming (System/370, System/360 Model 20, 1130, 1800, and 2770) treat the codes as data.
[5]Point-to-point only.

Figure 31. Summary of BSC Online Tests (except for the 3270)

Test type*	Content of test message	3271 or 3272				3275		
		3277		3281 or 3286				
		Model 1	Model 2	Model 1	Model 2	Model 1	Model 2	3284 Models
23	3270 Basic (EBCDIC)	X	X	X	X	X	X	
24	3270 Model 1 (EBCDIC)	X		X		X		
25	3270 Model 2 (EBCDIC)		X		X		X	
26	3270 Orders (EBCDIC)	X	X	X	X	X	X	
27	3270 Universal Character Set (EBCDIC)			X	X			X
28	3270 NL/EOM Printer			X	X			X
29	3270 Basic (ASCII)	X	X	X	X	X	X	
30	3270 Model 1 (ASCII)	X		X		X		
31	3270 Model 2 (ASCII)		X		X		X	
32	3270 Orders (ASCII)	X	X	X	X	X	X	
33	3270 Universal Character Set (ASCII)			X	X			X
34	3270 NL/EOM Printer (ASCII)			X	X			X

*For local 3270s, only EBCDIC tests (23 through 28) are available.

X = The test type is available for the device type.

Model 1 = 480-character buffer | Model 2 = 1920-character buffer

Figure 32. Summary of Online Tests for the Local and Remote 3270

2. BSCTEST=YES is coded in the BTMOD macro being used by the application program.

3. TERMTST=YES is coded in the DTFBT macro issued to define the particular line group concerned.

The BSC online test facility services RFT messages only if:

1. The Request-for-Test is received without error.

2. The application program issued a TWAIT following the macro instruction that was executing when the RFT was received.

Programming Note: 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. On-line test messages X=04 and X=05 contain the form feed (FF) and vertical tab (VT) characters. The 2770 printer performs the indicated forms motion, FF or VT, upon detection of the FF or VT characters.

Note: The application program must ensure that the first input buffer used in any macro instruction during which an RFT may be received is large enough to accommodate both the RFT message and any test message that is sent to the device. The largest RFT message allowed is 300 bytes. Buffers may be used, but the first buffer should be large enough to accommodate the RFT message and any test message.

Terminal-Initiated RFT

In most cases, response to an RFT message consists of transmitting a fixed test message to the device a specified number of times. In these cases, the Interruption Handler turns on the terminal test flag in the DECB and posts the DECB complete without error. The online terminal test logic gets control through TWAIT. The requested test pattern is loaded into the application program I/O area and transmitted to the device. In one special case (when X=0), the test message is only acknowledged. When the required number of transmissions has been completed, an EOT is transmitted on nonswitched lines, or a DLE EOT is transmitted on switched lines, and a Disable command is issued. The results of the test are printed on the computer console, and the application program operation is restarted.

Program-Initiated RFT

When the program itself initiates a binary synchronous online test procedure, the following sequence of events occurs:

1. The program issues an ONLTST macro instruction, which generates linkage to the online test logic generated by the BTMOD macro instruction.
2. The online test logic uses the parameters supplied by the ONLTST macro to generate an RFT message.
3. The RFT message is transmitted, and test messages from the device are read into the area supplied by the online test module.
4. The results of the completion of the test are printed on the computer console. The completion code in the DECB is posted X'7F' if the test ends normally, and X'41' otherwise.
5. The application program may wait for the completion of the ONLTST macro through the TWAIT or WAIT macro. Normal operation of all other lines of the problem program will continue without interruption during the operation of the online terminal test.

Request-for-Test (RFT) Message Formats

The formats of the messages which request tests are shown below. (The tests that are provided by BTAM are described on the following pages.)

If the test is requested from the application program with the ONLTST macro instruction, BTAM constructs the request message from information provided in the ONLTST operands. If the test is requested from a terminal, the terminal constructs the request message from information provided by the terminal operator. (See the device-dependent sections of this book and applicable component description or terminal operator's publications for information on requesting a test from the terminal.)

The RFT message constructed by BTAM as a result of ONLTST or by the terminal as a result of an operator request (except for the local or remote 3270) is:

S					D	S		D	E
O%	X	Y	N	ADDR	L	T	[TEXT]	L	T
H					E	X		E	X

The RFT message constructed by the terminal for a local 3270 is:

S			S				
O	%	/	T	X	Y	N	ADDR
H			X				

The RFT message constructed by the terminal for a remote 3270 is:

S			S						E
O	%	/	T	X	Y	N	ADDR		T
H			X						X

S
O % (2 characters)
H

identifies the message as an RFT message.

or

S
O% / (3 characters)
H

identifies the message as an RFT message from a local or remote 3270. These characters, in addition to the STX for local and the STX and ETX for remote, are automatically supplied when the terminal operator presses the Test Request key.

X (2 characters)
is a two-byte zoned decimal field that defines the type of test to be performed.

Y (2 characters)
is a two-byte zoned decimal field that defines the number of times the test message is to be transmitted. If X=0 and the remote terminal is a 2770, 2780, or 2972, Y must equal 1 because these stations transmit only the RFT message. For a test for a printer attached to a 3270, Y must equal 1.

N (1 character)
is a one-byte decimal digit (0-9) that specifies the size of the ADDR field. If N≠0, then the ADDR field must be present. For the local 3270, N must be set to 3; for the remote 3270, N must be set to 4.

ADDR (0-9 characters)
is the address of the station or device to which the test message is to be sent. For a multipoint configuration, the ADDR field contains the selection address of the unit to which the test message will be sent. This unit need not be the same unit that sent the RFT message. For a point-to-point configuration, the ADDR field is limited to two characters and contains the required component selection sequence, for example, ESC x, where x indicates the component to be selected. This field is not present if N=0. For a local 3270, this is a three-character control unit-device address.

[DLE] STX (1 or 2 characters)
are framing characters for start of text in the RFT message. DLE is present if text is to be written in transparent mode.

TEXT
this field is optional and appears only when X=1.

ETX
is the framing character for end of text in the RFT message for nontransparent data. For transparent data, BTAM provides the end characters.

Note 1: When error information byte mode is used, an error information byte will be present in System/370 main storage.

Note 2: If an RFT message is transmitted from a 2780 with the Auto Turnaround feature, and the terminal test requester at the 2780 wishes to turn the line around after the transmission, blank cards should follow the RFT message with the ETX character deleted from the RFT message. This allows the Auto Turnaround feature to be utilized so that the terminal test message can be received at the punch of the 2780. This procedure cannot be used when X=0 or X=1 in the RFT message.

Test Messages

The contents of the test message used for BSC on-line test are defined by the X field of the RFT message. Allowable values of X are 00-34, and they have the following meanings:

X=00 The test message is to be acknowledged ACK if received without errors. It is acknowledged NAK if a data check is detected, and it is not responded to at all if other ending conditions are detected.

X=01 The text received with the RFT message will be transmitted Y times.

X=02 Transparent EBCDIC message:

DLE STX text DLE ETX

The text consists of all 256 EBCDIC characters in collating sequence order.

X=03 USASCII transparency message:

DLE STX text DLE ETX

The text is in USASCII code (high-order bit always zero), and consists of all 128 USASCII characters in collating sequence order.

X=04 Normal EBCDIC message:

STX SYN SYN text ETX

The text consists of the 245 non-data link control characters. The characters excluded are SOH, STX, ETX, ETB, EOT, ENQ, ACK, NAK, SYN, US, DLE.

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 X=04.

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

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 for the 2780 printer.

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 for the 2780 punch.

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 for the 2780 printer.

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 for the 2780 punch.

X=11 TRANSCODE multipoint 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 for either the 2780 punch or printer.

X=12 EBCDIC printer message:

This message has the same content as the TRANSCODE printer message, X=09. The code is EBCDIC. This message is used for the 2780 printer.

X=13 EBCDIC punch message:

This message has the same content as the TRANSCODE punch message, X=10. The code is EBCDIC. This message is used for the 2780 punch.

X=14 EBCDIC alphameric message:

This message has the same content as the TRANSCODE multi-point message, X=11. The code is EBCDIC, and it may be used for the 2780 printer or punch on multipoint lines.

X=15 EBCDIC Weak Pattern message (switched line):

STX SYN SYN text ETX

The text consists of 74 bytes of X'00' followed by 6 bytes of SYN.

X=16 EBCDIC Weak Pattern message (leased line):

STX SYN SYN text ETX

The text consists of 40 bytes of X'AA' followed by 40 bytes of X'55'.

X=17 TRANSCODE Weak Pattern message (switched line):

STX SYN SYN text ETX

The text is 80 SOH characters.

X=18 TRANSCODE Weak Pattern message (leased line):

STX SYN SYN text ETX

The text is 40 N's followed by 40 ESC characters.

X=19 EBCDIC Weak Pattern message (DLE SYN insertion):

The text is 280 bytes of X'00' followed by 10 SYN characters. This message is transmitted in transparent code.

X=20 EBCDIC 80-Character Transparent Text message:

The text consists of 80 bytes of U-Z, 0-9, and X'00'-X'3F'. The code is EBCDIC. This message is transmitted in transparent code.

X=21 EBCDIC 120-Character Transparent Text message:

The text consists of 120 bytes of A-Z, 0-9, and X'00'-X'53'. The code is EBCDIC. This message is transmitted in transparent code.

X=22 EBCDIC 144-Character Transparent Text message:

The text consists of 144 bytes of A-Z, 0-9, and X'00'-X'6B'. The code is EBCDIC. This message is transmitted in transparent code.

X=23 3270 Basic EBCDIC Test Message:

This test checks all alphameric characters at a display station or printer. It checks the use of the WCC to sound the audible alarm and allows attribute field specifications to be checked at a display station. It will start a printer, printing only 40 characters to a line.

X=24 3270 Model 1 Align EBCDIC Test Pattern:

This test checks position alignment for the 480-character display station. It also checks the WCC to sound the audible alarm. It will start a printer, printing 40 characters to a line.

X=25 3270 Model 2 Align EBCDIC Test Pattern:

This test checks position alignment for the 1920-character display station. It also checks the WCC to sound the audible alarm. It will start a printer, printing 80 characters to a line.

X=26 3270 Orders EBCDIC Test Message:

This tests 3270 orders (SF, SBA, etc.), checks the WCC to sound the audible alarm, and uses display and intensified brightness. It starts the printer, printing 64 characters to a line.

X=27 3270 EBCDIC Universal Character Set Test Pattern:

This test uses the Erase/Write command, displaying the Universal Character Set in EBCDIC. It checks the WCC to start the printer, sound the audible alarm (on a display), and print 132 characters per line on the printer. NL and EM are also tested on a printer. Display intensity is used. The SF, NL, EM, and IC orders are used.

X=28 3270 NL/EM EBCDIC Test Pattern:

This test is mainly intended to test the end of message (EM) order and multiple new line (NL) orders on the printer. The WCC is checked to start the printer, sound the alarm (on a display), and print 132 characters to a line on the printer.

X=29-34 3270 ASCII Test Patterns:

These tests correspond to tests 23-28 except that transmission is in ASCII.

Messages to the Console

The online test facility prints the results of the test on the computer console. (The local 3270 is an exception; no results are printed at the console.) Two messages are provided; one is used when BTAM is transmitting test messages (or RFT messages with X=0), and the other is used when BTAM is receiving test messages. The content of these messages is:

Transmitter	Receiver
Line Address	Line Address
Number of Transmissions (Y)	Number of Transmissions
X Field	X Field
Time Outs	Time Outs
NAKs Received	Lost Data Occurrences
Terminal ID (multipoint)	Data Checks

The formats of these messages are:

For the transmitter:

4B70I ON-LINE TEST cuu xx yy TO NK TI

For the receiver:

4B71I ON-LINE TEST cuu xx yy TO LD DC

Where:

4B70I
: identifies the messages as BSC online test results for the transmitter.

4B71I
: identifies the message as BSC online test results for the receiver.

cuu
: specifies the line in the form channel and unit.

xx
: specifies the test type. This is the X field from the RFT message.

yy
: specifies the number of transmissions. For the transmitter, this value is the value from the RFT message. For the receiver, this value is accumulated by the on-line test program. If on-line test was not successfully initiated, this field will contain zero.

TO
: specifies the number of time-out occurrences.

NK
: specifies the number of NAKs received by the transmitter.

TI
: for multipoint lines, specifies the terminal ID; for point-to-point lines, it is blank or specifies the component selection sequence received with the RFT.

LD
: specifies the number of occurrences of lost data.

DC
: specifies the number of occurrences of data check.

BSC INTERMIXED TERMINALS

BTAM permits the application program to intermix terminals on BSC multipoint and switched point-to-point lines. This means that:

1. Different BSC devices can be connected to the System/370 on the same non-switched multipoint communications line.

2. The same application program-defined switched point-to-point line can be used for communication to or from any of the various BSC devices. This permits any BSC device to call a single System/370 phone number.

This control station may be a System/370 with a 2701, 2703, or Integrated Communications Attachment (ICA) attached. Remote stations may be the System/370, System/360 Model 20, IBM 1130, 1800, 2715, 2770, 2780, 2972, or remote 3270 on the same multipoint nonswitched line. For switched networks, the remote station may be any of the above devices, except the 2972, and also the System/370 with a 2701, 2703, or ICA.

The DTFBT macro instruction provides the means by which the user describes the characteristics of a communications line group. The DEVICE operand of this macro may be coded as 1130, 2020, 2780, 2970, BSC1, BSC2, or BSC3 for any combination of the above devices on the line except for the 1800, 2715, and 2770. These devices must use the appropriate BSC code as specified in Figure 33.

Because it is no longer necessary to distinguish between types of remote BSC stations that are connected to a non-switched point-to-point, switched, or multipoint line, it is no longer necessary to code a specific device type in the DTFBT macro. The type of line configuration can be specified by BSC1, BSC2, or BSC3 alone, as shown in Figure 33.

The problem program support must consider the following requirements:

- Where BSC devices connected to the same multipoint line have addresses of different lengths, the user must define the common polling list such that all entries in the list are of equal length. This means that the shorter-length polling addresses must be extended by leading SYN characters in the user's definition (in the DFTRMLST macro instruction).

- All devices must use the same modem equipment.

- EBCDIC and USASCII are the only codes permitted on intermixed devices, with the stipulation that these codes may not be intermixed on the same multipoint nonswitched line or switched network to the same System/370 phone number. When one of the devices intermixed is an 1130 or 2972, EBCDIC must be used.

Remote BSC Station Type	BSC1 Non-switched pt-to-pt	BSC2 Switched pt-to-pt	BSC3 Non-switched multipt
S/370	X	X	
S/360 Model 20	X	X	X
1130	X	X	X
1800	X	X	X
2715 (Local)	X		
2715 (Remote)	X	X	X
2770	X	X	X
2780 (EBCDIC & USASCII)	X	X	X
2972			X
3270 (Remote)			X

Figure 33. Remote Station Types Supported for BSC1, BSC2, and BSC3.

SUPPORTED LOCAL DEVICES

BTAM supports the following terminal systems* attached directly to selector or multiplexer channels:

- IBM 2260 Display Station - IBM 2848 Display Control
- IBM 2790 Data Communications System
- IBM 3270 Information Display System (comprising the 3272 Control Unit, Model 1 or 2; the 3277 Display Station, Model 1 or 2; and the 3284 or 3286 Printer, Models 1 or 2)

LOCAL 2260-2848

General Information

An important distinction in discussing the local 2260 is the difference between the READ Initial operation and other local 2260 READ and WRITE operations. Operations other than READ Initial are directed to a specific terminal; a READ Initial operation is directed to any terminal connected to a 2848 display control unit. When a READ Initial macro instruction is issued, all terminals connected to a specified 2848 are examined for readiness to provide input (in which case the terminal operator has pressed the ENTER key); the first terminal encountered at which an attention has occurred is read. Only one terminal is read for each READ Initial macro instruction. The discussion below describes in detail how the READ Initial operation works.

When an attention interrupt occurs at the CPU, a CCB (a part of the LCB in BTAM) for that line address must be in the channel scheduler queue in order for the Attention status to be posted (see the Communication Between the Problem Program, BTAM, and the Supervisor section). If a CCB is not in the queue, the Attention interrupt is ignored by the DOS supervisor.

----

*Terminals which 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 the IBM-supplied products or programs may have on such terminals.

Only one 2848 Display Control is allowed for each DTFBT macro instruction, and only one DTFBT macro instruction is allowed for each 2848. BTAM queues each display station LCB (including the required CCB) associated with the DTFBT when the first READ Initial (or READ Short) is issued in the problem program. After queuing each LCB, BTAM returns control to the problem program. Subsequently, when an Attention interrupt occurs, the BTAM Interruption Handler requests the supervisor to start a Read for that line address. When the message is received from the 2260, BTAM posts the READ macro as complete. The LCB for the 2260 just read remains queued. However, no more Attention interrupts will be serviced until the next READ macro is issued. Only one message is read per READ macro. Attention interrupts occurring between READs are posted in the appropriate LCBs for service on subsequent READs.

The _rln_ operand has a special use in READ Initial (or READ Short). This operand specifies which LCB (that is, 2260) in the DTFBT is to be checked first for an Attention interrupt. BTAM scans all LCBs, starting with the one specified in the _rln_ operand. During the scan, BTAM checks for previously posted (but not yet serviced) Attention interrupts. Servicing of an Attention interrupt consists of reading from the 2260 that caused it. Any LCB found to be dequeued during the scan is queued again. The scan is terminated when BTAM services an LCB having an Attention interrupt posted, or when all LCBs have been scanned at least once. Once a READ Initial (or READ Short) is issued, the problem program is prevented from issuing another READ or WRITE for that DTFBT until the first READ completes or until a RESETPL is issued to terminate the first READ.

The use of the CHGNTRY macro with the SKIP option allows the user to cause BTAM to ignore a particular 2260. An LCB for "skipped" 2260 will not be queued during a scan. If it is already queued when CHGNTRY is issued with SKIP specified, a subsequent Attention interrupt caused by the associated 2260 will not be honored. CHGNTRY with ACTIVATE specified will reactivate an LCB. That is, if the LCB is currently dequeued, it will be queued during the scan for a subsequent READ Initial (or READ Short). If it is currently queued, an Attention interrupt caused by the 2260 will be serviced during the current or subsequent READ.

The other local 2260 READ/WRITE options (READ Buffer, WRITE Initial, WRITE at Line Address, and WRITE Erase) do not involve the scanning of the LCBs. For these options, the requested operation is performed on the specific display station (or the printer) specified in the rln operand. Upon completion of these macros, the specified LCB remains dequeued. CHGNTRY has no effect on the execution of these macros.

The use of RESETPL neither halts currently executing commands nor dequeues LCBs. It is used to prevent servicing of Attention interrupts occurring subsequent to its execution. Hence, the issuance of a WRITE Initial, for example, may require that BTAM first dequeue a queued LCB. Only then could the Write CCW be issued for that 2260 (or 1053).

Note: The local 2260 uses only EBCDIC code.

Terminal to CPU

Read Initial (TI): The READ Initial macro instruction is issued to read a message from any one (but only one) of the nonskipped display stations defined in the DTFBT. A READ TI causes the same channel program (a single Read CCW) to be moved into each nonskipped LCB in the DTFBT.

When the operator of the 2260 presses the ENTER key and the attention interrupt is recognized by the CPU, all characters displayed between the START symbol and the cursor, except the characters to the right of any NL symbols, are read into main storage.

- The START symbol indicates the beginning of the message. After the message is read, the START symbol is cleared from the screen automatically. If no START symbol has been keyed, no message is read.
- NL stands for new line. The characters to the right of an NL symbol are not read.
- The cursor indicates the position on the screen following the last character.

A READ Initial macro instruction allows the CPU to read any one of the 2260s defined in the DTFBT when the operator presses the ENTER key. When the READ Initial is completed, the relative position in the DTFBT of the 2260 that sent the message is stored in the low-order byte of the DECB polling pointer field.

Note: If the area operand is coded 'S', the I/O address is that of the buffer (only one buffer is received). The length of the message to be read must be less than buffer length-12. The entry operand is not used for the local 2260.

READ Short (TO): The READ Short macro instruction is similar to READ Initial, except that the START symbol is not cleared and the cursor is advanced one position after the Read. The channel program is a single Read Short CCW.

READ Buffer (TB): The READ Buffer macro instruction causes the entire buffer of the specified display station to be read, and the cursor to be placed in the top left corner. If the nondestructive cursor feature is not installed, the user should next issue a WRITE Erase (TS) to clear the buffer.

The relative line number operand for the READ Buffer macro instruction specifies the relative position in the DTFBT of the LCB corresponding to the 2260 whose buffer is to be read. The channel program is a single Read Buffer CCW.

Programming Note: If a READ Initial or READ Short is outstanding for the DTFBT, if no terminal has responded, and if the user wishes to initiate a different operation, he must first issue a RESETPL macro specifying the same SECB specified in the incomplete READ macro. The RESETPL should be followed by a WAIT macro.

CPU to Terminal

The WRITE operations for the local 2260 initiate CPU-to-terminal transmission.

WRITE Initial (TI): The WRITE Initial macro instruction is issued to transmit data to a specific display station, starting at the position on the screen indicated by the cursor. The channel program for WRITE TI is a single CCW that writes the message. If the relative line number operand refers to a 2260, the message is displayed on the screen. If the operand refers to a 1053 printer, the message is typed.

WRITE at Line Address (TL): The WRITE at Line Address macro instruction is issued to transmit data to a specific display station. The cursor is positioned at the first character position of a line on the screen as determined by the first character transmitted from the user-specified output area. Characters are displayed from that point.

WRITE Erase (TS): The WRITE Erase macro instruction is issued to erase the screen of a specific display station, and position the cursor in the upper left corner of the screen.

Figure 34 shows the local 2260 configuration.

Lock Option

BTAM provides the local 2260 user with a Lock option for each operation type. The user may request this option by adding the suffix "L" to the normal operation type codes, yielding the additional macro options:

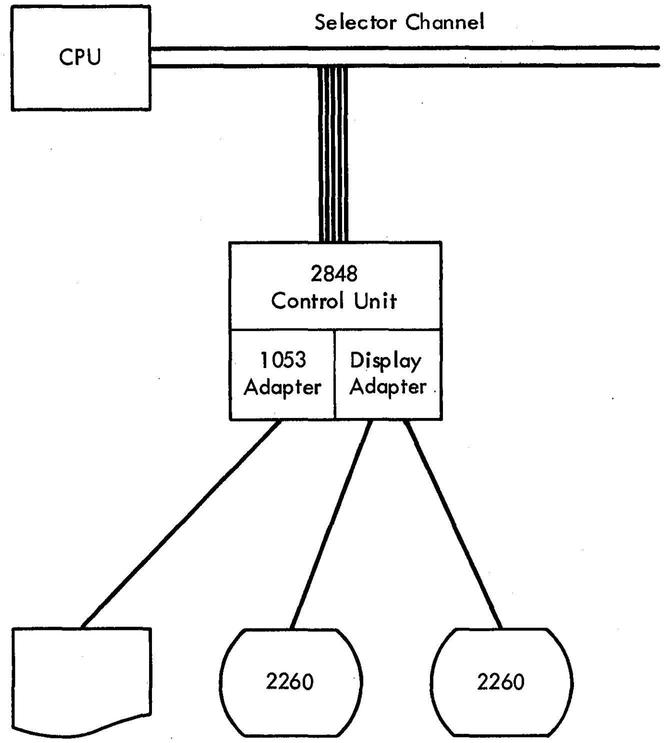
```
READ TIL
READ TOL
READ TBL
WRITE TIL
WRITE TLL
WRITE TSL
```

Each standard operation (without the Lock option) restores (unlocks) the manual entry keyboard when the operation is complete. Each operation with the Lock option functions like the corresponding standard operation, except that at the completion of the Read or Write operation, the manual entry keyboard retains the locked or unlocked status that existed at the beginning of the execution of the operation.

If the manual entry keyboard is unlocked prior to the execution of a READ TBL, WRITE TIL, WRITE TLL, or WRITE TSL operation, the keyboard will be unlocked at the completion of the operation. If the manual entry keyboard is locked prior to the execution of a READ TBL, WRITE TIL, WRITE TLL, or WRITE TSL operation, the keyboard will remain locked at the completion of the operation; this prevents further operator entry until the ERASE key or the RESTORE key is pressed, or until a BTAM macro instruction without the Lock option is issued.

The READ TIL and READ TOL operations always leave the manual entry keyboard locked. Write commands with the Lock option should be used in conjunction with Read commands with the Lock option. If an error condition is detected as a result of a Read command with the Lock option, a Write command with the Lock option should follow the Read command.

The locked keyboard gets the operator's attention and insures that the operator reads the message displayed, before further keying on the keyboard.



Note: The number of Display Adapters which may be included in a single 2848 Display Control varies.

Figure 34. Local 2260 Configuration

IBM 2790 DATA COMMUNICATIONS SYSTEM

This system may be attached directly to the multiplexer channel via a 2715 Transmission Control Unit, Model 1. However, simulation of binary synchronous transmission is provided. See the 2790 in the section, "BSC Device-Dependent Considerations."

IBM 3270 INFORMATION DISPLAY SYSTEM (LOCAL)

IN GENERAL

The 3270 Information Display System provides input/output for a wide range of applications, with display stations that may consist of an alphameric display screen (CRT) and keyboard and an associated printer. The 3270 attached directly to a multiplexer or selector channel (rather than indirectly, as a teleprocessing system, through a transmission control unit and telecommunications lines) is referred to in this book as the "local 3270." The 3270 designed for remote (teleprocessing) applications is referred to as the "remote 3270" and, because the local and remote 3270 require different programming considerations, is described separately in the general section, "BSC Device-Dependent Considerations."

USING THIS SECTION WITH THE 3270 COMPONENT DESCRIPTION

Before writing an application program using a local 3270, it is necessary to understand the local 3270's physical characteristics and capabilities, described in IBM 3270 Information Display System Component Description, GA27-2749. After reading the 3270 Component Description, the programmer can use this book, particularly this section, as a guide to the BTAM macro instructions that are used to define and control I/O for the local 3270. He must, however, use the 3270 Component Description to construct data areas (called "data streams") to be sent to the terminal to display an image or print a line, and to interpret data streams received from the terminal. The formats for these data streams are shown in this section with applicable macro instructions; however, the 3270 Component Description must be used to understand terms used within these formats.

LOCAL 3270 CONFIGURATION

A local 3270 consists of:

- One or more 3272 Control Units, Model 1 or 2
- One or more 3277 Display Stations, Model 1 or 2, attached to each Control Unit
- Optionally, one or more 3284 or 3286 Printers, Models 1 or 2

Up to 32 terminals (display stations and printers) can be attached to a single control unit. A local 3270 configuration is shown in Figure 35. For planning considerations, see An Introduction to the IBM 3270 Information Display System, GA27-2739.

DIFFERENCES BETWEEN A LOCAL AND REMOTE 3270

With a local 3270, each control unit is attached directly to a multiplexer or selector channel; with a remote 3270, each control unit is attached to the channel indirectly via a transmission control unit and telecommunications lines. In a local 3270, no control unit may be farther than 200 feet from the channel and no terminal farther than 2000 feet from a control unit; in a remote 3270, a control unit and terminals can be located thousands of miles away. (All or part of a remote 3270 configuration can also be located, if desired, at the computer installation.)

Because of these differences, BTAM logic for the local 3270 is somewhat different from that for the remote 3270. Applicable macro instructions and operands vary for the local and remote 3270s. (The macro instructions applicable to the local 3270 are described in this section; those applicable to the remote 3270 are described under "IBM 3270 Information Display System (Remote)" in the general section "BSC Device-Dependent Considerations."

In general, the local 3270 and the remote 3270 perform the same functions. Only the remote 3270 can, using a single macro instruction, copy messages from one terminal to others on the same control unit. Only the local 3270 has the facility to allow the Online Test Executive Program

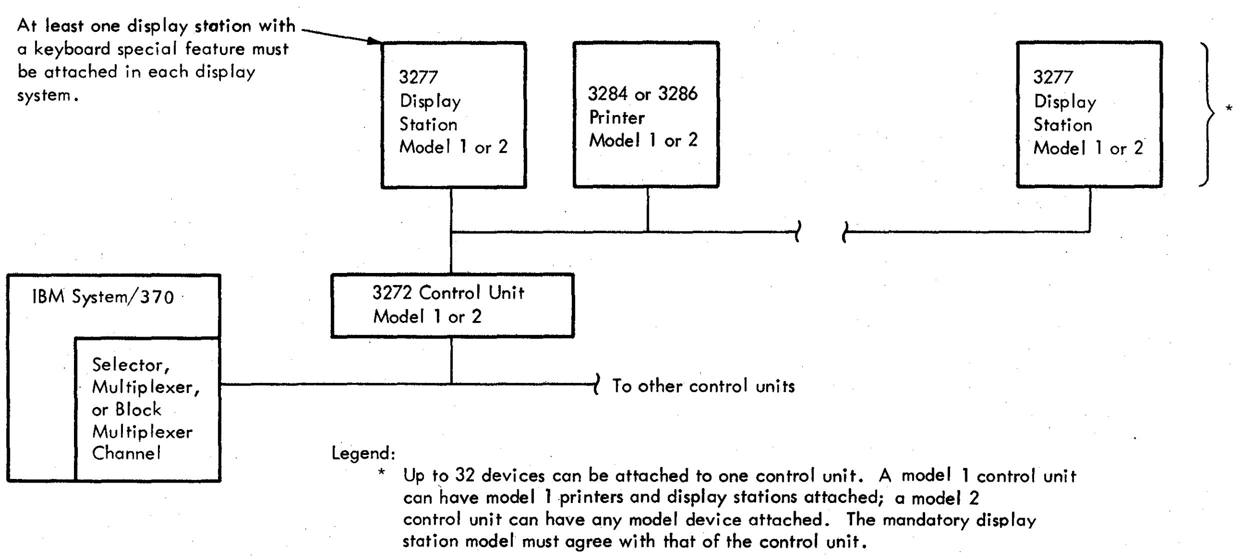


Figure 35. Locally Attached 3270 Display System

(OLTEP) to test local 3270 devices while an application program using those devices continues to run. (Remote 3270 devices can be tested by OLTEP but not concurrent with their use by an application.)

Both local and remote 3270s can be defined and controlled within the same application program.

SAMPLE LOCAL 3270 APPLICATION PROGRAM

A sample BTAM application program designed to demonstrate a local 3270 application is provided with the system. The sample program, named SAMP327L, is shown in Appendix L.

CHARACTER SETS

Display stations with keyboards having either EBCDIC or ASCII (also called "USASCII") character sets may be included in a local 3270. Since data transmission from a locally attached display station buffer can only occur in EBCDIC regardless of the keyboard character set, there is no requirement for translation by the application program, and the TRNSLATE macro instruction is not needed for this purpose.

The application programmer will need to know, however, which display stations have EBCDIC keyboards and which have ASCII. Several keys differ between the two keyboards; interpreting input and writing output correctly will depend on understanding whether the keyboard is EBCDIC or ASCII.

DEFINING A LOCAL 3270

A requirement for defining a local 3270 for an application program is to know the SYSnnn number assigned to each device to be defined as part of the local 3270. These numbers are required for use with the DTFBT macro instruction.

In writing an application program involving I/O with a local 3270 (or other devices that use BTAM), a first step is coding the BTMOD macro instruction. BTMOD is used to define the sections of BTAM logic required by the application program. BTMOD is described fully in the general section "BTAM Macro Instruction Format Descriptions." To include the local 3270 BTAM logic, the L3277 operand must be specified L3277=YES. Other BTMOD operands applicable to the local 3270 are: SEPASMB, BUFFER, ERLOGIC, TST3277, CANCEL, RMSR, and RESETPL. Additional BTMOD operands may be applicable if the application program performs I/O with other terminal systems (for example, a local 2260 or a remote 3270).

An example of a BTMOD macro instruction for the local 3270 is:

```
BTML3270  BTMOD  SEPASMB=YES,BUFFER=NO,
                 ERLOGIC=C,L3277=YES,
                 TST3277=YES
```

In the above example, a BTAM logic module named BTML3270 will be assembled separately from the application program. It will include error recovery logic with error counts, local 3270 logic, logic for online terminal tests for the local 3270, and RESETPL logic (defaulted), in addition to the basic BTAM logic always provided as a result of issuing BTMOD. It will not include buffer management or RMSR logic.

In defining the configuration itself, at least one DTFBT macro instruction must be used for each control unit in the display system. Each display station or printer is identified as a line in the LINELST operand of DTFBT. In each DTFBT, CU=3272 and DEVICE=3277 (for both display station and printer) must be specified. By using more than one DTFBT macro instruction, the same control unit may be redefined with different configurations.

The DTFBT macro instruction is described fully in the section "BTAM Macro Instruction Format Descriptions." DTFBT operands applicable to the local 3270 are: LINELST, CU, DEVICE, BUFCB, BUFNO, BUFL, SEPASMB, MODNAME, TERMTST, ERROPT, LERBADR, and LCBNUM.

An example of a DTFBT macro instruction is:

```
LOCAL1   DTFBT  LINELST=(005,006,007),
                CU=3272,DEVICE=3277,
                TERMTST=YES,LERBADR=ERTABLE
```

In the above example, devices SYS005, SYS006, and SYS007 are defined as being attached to a 3272 control unit; they may be display stations or printers (at least one, however, must be a display station). A line control block (LCB) will be created for each device; SYS005, SYS006, SYS007 will have relative line numbers 0, 1, and 2, respectively. TERMTST=YES specifies that the online terminal test facility may be used. Error statistics will be maintained in a table, ERTABLE, created by using the RMSRTAB macro instruction.

Since a terminal list is not required for the local 3270 (each terminal is, in effect, listed in the control block created by DTFBT), the DFTRMLST macro instruction is not applicable.

CONTROLLING A LOCAL 3270

Having defined the required BTAM logic with the BTMOD macro instruction and having defined the local 3270 configuration with one or more DTFBT macro instructions, READ and WRITE macro instructions are used to control input from and output to terminal devices. A description of READ and WRITE is provided in the general section "BTAM Macro Instruction Format Descriptions." A description of these macro instructions as they pertain to the local 3270 is provided in this section.

FUNCTIONS OF READ AND WRITE MACRO INSTRUCTIONS

Using READ and WRITE macro instructions, an application programmer may:

- Read modified fields from a display station buffer after a terminal operator has completed his entry and caused an attention, for instance, by pressing the ENTER key (READ Initial).
- Read modified fields from a display station buffer without waiting for an attention indication (READ Modified).
- Read only those modified fields beginning at a specified buffer location (READ Modified from Position).
- Read the entire buffer contents, both modified and unmodified data, including attribute characters (READ Buffer).
- Read the buffer contents, both modified and unmodified data, including attribute characters, beginning at a specific buffer location (READ Buffer from Position).
- Write data to any desired position in a display station or printer buffer (WRITE Initial).
- Erase the data presently at the device buffer (on the screen or in the printer buffer) and write data to any buffer location (WRITE Erase).
- Erase all unprotected fields in the display station or printer buffer (WRITE Erase Unprotected).

The optype codes used to specify these various READ and WRITE macro instructions are shown in Figure 36.

Additional functions, such as positioning the cursor, are performed by characters inserted in the output data stream and interpreted by the device itself. These functions and characters are described in the <u>3270 Component Description</u>.

Operation	Optype 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 Unprotected Erase	TUS

Figure 36. Local 3270 Optype Codes

USING THE WAIT MACRO INSTRUCTION

Each BTAM read or write operation requires two macro instructions from the application program before the program can continue execution with assurance that the operation has completed: a READ or a WRITE and a WAIT. The READ or WRITE macro instruction initiates an operation; the WAIT determines when the operation has completed, delaying further execution until it does. For efficiency, processing that does not depend on a given read or write operation should be performed in between the READ or WRITE and the WAIT for that operation. The results of each READ or WRITE macro instruction (whether or not the operation began successfully) are returned as a code in register 15; following a WAIT, a completion code is available in byte 0 of the DECB specified in the READ or WRITE. These return codes and completion codes are defined in the discussion "Posted Error Information" in the general section "Assembly Considerations."

HOW ATTENTION INTERRUPTIONS ARE HANDLED

When the operator of a local 3270 display station carries out certain actions, such as pressing the ENTER or CLEAR key, an I/O interruption, called an attention interruption, occurs. For this attention interruption to be recorded by the system, a READ Initial macro instruction specifying a DFTBT that includes the display station must have been issued by a BTAM application. The first time in a program that a READ Initial specifying a particular DTFBT is issued, the Channel Control Block (CCB), a part of the Line Control Block (LCB) that exists for each device in the DTFBT, is queued in the channel scheduler queue by the DOS Supervisor. At the time an attention interruption is received, a CCB must be queued in the channel scheduler queue; otherwise, the attention interruption is ignored. If the CCB is queued and a READ Initial has been issued and is pending (no read operation has yet taken place because no device had signalled it had input ready), when an attention interruption does occur, the read operation will take place. If the CCB is queued but no READ Initial is pending, the attention interruption will be recorded in the LCB. When a READ Initial is subsequently issued, each LCB associated with the specified DTFBT will be scanned for an attention indication. The first LCB found at which an attention indication has been recorded will cause the input from the display associated with the LCB to be read.

READ OPERATIONS

In the following discussion, each read operation is discussed separately.

Read Modified Fields After Operator Action (READ Initial)

This is the basic read operation for the application program. It is requested by issuing the READ macro instruction with the TI optype specified. It causes a message to be read from a display station whose operator, having completed an entry, has:

- Depressed one of the following keys:

  ENTER
  PF (PROGRAM FUNCTION) keys 1-12
  PA (PROGRAM ATTENTION) keys 1-3
  TEST REQUEST
  CLEAR

- Selected a detectable field with the selector pen. (See the 3270 Component Description for how to establish a field as detectable.)

- Inserted a card in the operator identification card reader and caused it to be read.

Only modified fields are read.

The READ Initial is different from other read and write operations. Macro instructions for other operations specify the particular device to which the operation is directed; the READ Initial macro instruction is issued to read any display station attached to a specified control unit. When a display is found at which an attention indication has been received, that display is read. Each READ Initial results in no more than one display being read; the first display encountered with an attention indication is the one read. If, at the time the READ Initial is issued, no display has an attention indication, a return code of X'00' will still appear in register 15, meaning I/O has been initiated. Subsequently, during other processing or after WAIT has been issued, when an attention interruption is caused at a display (by, for instance, the operator pressing the ENTER key), that display will be read. The attention interruption will not be apparent to the application program; if the read operation completed successfully, a code of X'7F' will be the completion code in the DECB on return from WAIT, and the rln (relative position of the device in the DTFBT defined for its control unit) will be located in the low-order byte of the polling pointer field (byte 39) of the DECB.

Applicable operands for the READ Initial macro instruction for the local 3270 are: decb, optype, dtfbt, area, length, rln, and MF. Note that the entry operand is not applicable. The rln operand specifies the first display to be checked for readiness to provide input (with a pending attention indication). All displays (each is specified in the sublist of the LINELST operand of the DTFBT macro instruction) are scanned until an attention indication is found.

If the operator caused an attention by pressing the ENTER key or one of the PF keys or by selecting a detectable field with the selector pen (all conditions referred to as a "normal read"), the data read is, unless the screen is completely unformatted or completely protected, in this format:

AID	CURSOR ADDRESS	SBA	BUFFER ADDRESS	TEXT	SBA	BUFFER ADDRESS	TEXT

If the terminal buffer is unformatted, the input message is:

AID	CURSOR ADDRESS	TEXT

If the operator caused an attention by pressing the CLEAR key or one of the PA keys, a short read will occur and the message is simply:

AID

If the operator caused an attention by inserting a card in the identification card reader, the data read is in this format:

AID	0-37 TEXT CHARACTERS	EOR EOI	LRC

If the operator caused an attention by pressing the TEST REQUEST key, the data read is in this format:

SOH	%	/	STX	TEXT

In the above case, the program will not usually be aware of the message; it will be intercepted and handled by BTAM. The program will only be aware of this message if RFT logic in BTAM has not been requested or if TERMTST=NO is not specified in the DTFBT macro instruction.

The meanings of the characters illustrated above in the input data formats are provided in IBM 3270 Information Display System Component Description, GA27-2749.

An example of a READ Initial macro instruction is:

```
INPUT1    READ  DECB1,TI,DTFBT1,INAREA,256,,
                3,MF=E
```

In the above example, device 4 (rln 3) connected to the control unit identified by DTFBT1 will be checked for input, then the other devices in turn. DECB1 may previously have been created by a non-executable READ with MF=L specified. DECB1 must be associated with DTFBT1 in all other READ and WRITE macro instructions. When data is read, a maximum 256 bytes will be read into the input area, INAREA. Only fields modified by the terminal operator or formatted by the program as modified will be read. If part of a field has been modified, the entire field will be read, except that all nulls (X'00') in the field are suppressed (that is, not read). If the application program determines that more data remains to be read from the terminal buffer (more than the 256 bytes read initially), the additional data may be read either by issuing one or more READ Modified from Position macro instructions or by issuing a READ Modified, specifying a larger input area.

The CCB for the display read will remain queued after the operation.

The channel program generated and executed for a READ Initial is:

```
Select          CC
Read Modified   SLI   (into user area)
```

Skipping or Activating A Terminal: CHGNTRY

The CHGNTRY macro instruction allows a particular display to be ignored during subsequent READ Initial operations, or, if it has been previously ignored, to be considered for input once again. (See the special format for the local 2260 and local 3270 CHGNTRY macro instruction in the general section "BTAM Macro Instruction Format Descriptions.")

Each READ Initial macro instruction causes all displays defined by a DTFBT macro, beginning with the display specified by the rln operand, to be scanned for input, and the first display for which an attention interruption has been received to be read. To have a particular display ignored during one or more READ Initial operations, CHGNTRY may be issued with the SKIP operand specified. The display will then be ignored for any pending or subsequent READ Initial operations; an attention interruption from that display will go unrecognized. To reactivate a display for input consideration, CHGNTRY is issued with the ACTIVATE operand specified. After the display has been reactivated, it will be scanned during subsequent READ Initial operations.

Terminating A Read Initial: RESETPL

The RESETPL macro instruction allows the application program to terminate a pending READ Initial operation so that another read or write operation may be performed with a device on the control unit. (The format of RESETPL is described in the general section "BTAM Macro Instruction Format Descriptions.")

Following the RESETPL, a WAIT macro instruction must be issued. Following the WAIT, a completion code of X'54' in the DECB indicates that the READ Initial operation was terminated before a read took place. A completion code other than X'54' indicates that a read operation had already begun at the time RESETPL was issued and the read operation was completed successfully or unsuccessfully. In either case, any device on the control unit will now be free for another READ or WRITE macro instruction.

Read Modified Fields (TM)

This operation is similar to READ Initial except that the read operation is directed to a specific display and is performed immediately; it does not depend on an attention interruption first occurring at the display station. This operation reads all modified fields in the display buffer. It is requested by issuing the READ macro instruction with the TM optype specified.

By using a READ Modified with an associated READ Initial, input area requirements may be minimized while awaiting an attention interruption. A READ Initial may be issued specifying an area length of 2, and only two bytes reserved as an input area by the application program. When, following an attention, data is read by the READ Initial, a longer area may now be obtained and the complete message read by issuing a READ Modified to the display at which the attention occurred.

Applicable operands for the READ Modified macro instruction for the local 3270 are: decb, optype, dtfbt, area, length, rln, and MF. Note that the entry operand is not applicable. The rln operand specifies the display that is to be read. If the READ Modified is used in conjunction with a READ Initial, the rln should be the same as that placed in the low-order byte (byte 39) of the polling pointer field in the DECB on completion of the READ Initial.

The data read by a READ Modified will be in one of the formats shown for the READ Initial.

An example of a READ Modified macro instruction is:

```
READMOD  READ  DECB1,TM,DTFBT1,INAREA,200,,
               4,MF=E
```

In the above example, all modified fields in the buffer of device 5 (rln 4) connected to the control unit defined by DTFBT1 will be read into the 200-byte input area, INAREA. DECB1 will have been created by a previous READ macro instruction with MF=L specified.

The CCB for the display read will remain queued after completion of the operation.

The channel program generated and executed for a READ Modified is the same as for READ Initial.

Read Modified Fields from Position (TMP)

This operation reads all modified fields beginning at a specified position in the display buffer. It is requested by issuing the READ macro instruction with the TMP optype specified. This macro instruction can be used to read in remaining modified fields when, following a previous READ Initial or READ Modified from Position, it is determined that there was more data to be read than the input buffer could hold.

Applicable operands for the READ Modified from Position macro instruction for the local 3270 are: decb, optype, dtfbt, area, length, entry, rln, and MF. Note that the entry operand is applicable only for this and the READ Buffer from Position macro instructions. The entry operand must specify the address of a four-byte area containing the buffer location from which the read is to start. The four-byte area must have this format:

WCC	SBA	BUFFER ADDRESS	

It is suggested that a space character (X'40') be used as the WCC (Write Control Character) since this setting will not affect display conditions. The WCC, SBA, and buffer address formats are explained in IBM 3270 Information Display System Component Description, GA27-2749.

As with READ Modified, the rln operand specifies the device to be read.

The message read into the input buffer will have one of the formats shown for READ Initial.

An example of a READ Modified from Position macro instruction is:

```
RDMDPS   READ  DECB1,TMP,DTFBT1,INAREA,100,
               ENTRY1,5,MF=E
```

In the above example, ENTRY1 is the address of a four-byte area that contains the buffer address from which all modified fields are read. These fields are read from device 6 (rln 5) attached to the control unit identified by DTFBT1 into the 100-byte input area, INAREA.

The CCB for the display read will remain queued after completion of the operation.

The channel program generated and executed for READ Modified from Position is:

Select	CC	
Write	CC,SLI	(to set the buffer address)
Read Modified	SLI	(into the user area)

Read Buffer (TB)

This operation reads the entire contents of a specified terminal buffer, including modified and unmodified fields, attribute characters, and nulls (X'00'). It is intended primarily for diagnostic purposes. It is requested by issuing the READ macro instruction with the TB optype specified.

Applicable operands are: decb, optype, dtfbt, area, length, rln, and MF. Note that the entry operand is not applicable.

The rln operand specifies the device that is to be read.

All data beginning at location 0 in the device buffer is read. In addition, a special character (SF) is inserted into the input data stream to indicate the beginning of each field. The application program must add these SF characters to the amount of device buffer data he plans to read to determine the size of the input area. The input data stream is in this format in the input area (unless the device buffer is completely unformatted):

AID	CURSOR ADDRESS	SF	ATTR CHAR	TEXT	SF	ATTR CHAR	TEXT

If the device buffer is unformatted, no SF characters are inserted since there are no fields. The input consists of the AID character, the cursor address, and all character locations in the device buffer, including nulls.

An example of a READ Buffer macro instruction is:

```
DIAG    READ  DECB1,TB,DTFBT1,INAREA,600,,1,
              MF=E
```

In the above example, the input area, INAREA, has been made large enough (600 bytes) to accommodate an entire 480-character display buffer and allow 120 bytes for the AID, cursor address, and 117 SF characters. (To read a 1920-character display buffer, subsequent READ Buffer from Position macro instructions could be issued, or the input area could be made larger.) In this example, the buffer of device 2 (rln 1) attached to the control unit identified by DTFBT1 is read into INAREA. DECB1 was created by a previous READ macro instruction specifying MF=L.

The CCB for the device read will remain queued after completion of the operation.

The channel program generated and executed for a READ Buffer macro instruction is:

```
   Select       CC
   Read Buffer  SLI    (into user area)
```

<u>Read Buffer from Position (TBP)</u>

This operation reads in all device buffer contents beginning at a specified buffer location. It can be used to continue the reading of an entire buffer begun with a READ Buffer macro instruction. To perform this operation, a READ macro instruction is issued with the TBP optype specified.

The same operands that apply to READ Modified from Position also apply to READ Buffer from Position. The entry operand specifies a data area (shown under READ Modified from Position) that contains the buffer address from which the buffer is to be read. The rln operand specifies the device whose buffer is to be read.

An example of a READ Buffer from Position macro instruction is:

```
RDBFPOS   READ  DECB1,TBP,DTFBT1,INAREA,300,
                ENTRY1,2,MF=E
```

In the above example, ENTRY1 will be the address of a data area containing the buffer location from which data will be read. Up to 300 bytes of data and SF characters (inserted in front of each field) are read into INAREA from device 3 (rln 2) attached to the control unit identified by DTFBT1. DECB1 will have been created previously by a READ macro instruction with MF=L specified.

The CCB for the device read will remain queued after completion of the operation.

The channel program for a READ Buffer from Position macro instruction is:

```
   Select       CC
   Write        CC,SLI   (to set the buffer
                          address)
   Read Buffer  SLI      (into the user
                          area)
```

WRITE OPERATIONS

As with read operations, each write operation requires two macro instructions from the application program before the program can continue execution with assurance that the write operation has completed: a WRITE and a WAIT. The WRITE macro instruction initiates an output operation; the WAIT determines when the operation has completed, delaying execution until it does. Processing that does not depend on a given write operation should be performed between the WRITE and WAIT for that operation. The results of each WRITE macro instruction are returned as a code in register 15; the results of the WRITE operation are available after the WAIT in a completion code placed in the DECB specified by the WRITE and, if the completion code is X'41', by sense information placed in the DECB. These return codes and completion codes are defined in the discussion "Posted Error Information" in the general section "Assembly Considerations." Sense information for the local 3270 is discussed later in this section under "Error Recovery."

In the following discussion, each write operation is discussed separately.

Write Initial (TI)

This operation writes data (device control information and text) to a specified display station or printer. The WRITE Initial may either rewrite the entire buffer or only parts of it, leaving the rest of the buffer as it was. It is requested by issuing the WRITE macro instruction with the TI optype specified.

Operands applicable to the WRITE Initial macro instruction are: decb, optype, dtfbt, area, length, rln, and MF. Note that the entry operand is not applicable. The rln operand specifies the device to receive the data.

Data in the output area must be in this format:

```
┌─────┬───┐  ┌─────────┐
│     │   │  │ ORDERS  │
│ WCC │   │  │AND TEXT │
└─────┴───┘  └─────────┘
```

The WCC and order sequences are explained in IBM 3270 Information Display System Component Description, GA27-2749.

An example of a WRITE Initial macro instruction is:

```
WRITINT   WRITE  DECB1,TI,DTFBT1,OUTAREA,
                 150,,4,MF=E
```

In the above example, a 150-byte message is written from OUTAREA to the terminal buffer of device 5 (rln 4) attached to the control unit identified by DTFBT1. DECB1 may have been defined by a previous WRITE macro instruction with MF=L specified.

The CCB for the device written to will remain queued after completion of the operation.

The channel program generated and executed is:

```
Select   CC
Write    SLI   (from the user area)
```

Programming Note: If a terminal operator has made an entry and pressed the ENTER key but no READ Initial has yet been issued, a WRITE Initial to the display may nullify the operator input. This situation may be avoided by reserving areas of the display for operator input only (nothing will be written to these areas) and then setting the RMD (Reset Modified Data) bit to zero (meaning "do not reset modified data tags") in the WCC (Write Control Character) of the WRITE Initial message. Setting the RMD bit off in the WCC is required because if modified data tags are reset as part of the WRITE Initial, the pending attention will not be honored since there will be an indication that no fields have been modified.

Write Erase (TS)

This operation first erases the entire buffer of a specified device and then writes a message to it. It can also be used just to erase the buffer. The erasure consists of setting all terminal buffer character positions to nulls (X'00'). The operation is requested by issuing a WRITE macro instruction with the TS optype specified.

Applicable operands are the same as those for the WRITE Initial macro instruction. The output message format is the same as that shown for WRITE Initial; if the buffer is simply to be erased and no data written, the output message should be the WCC only and the length should be specified as 1.

An example of the WRITE Erase macro instruction is:

```
WRITE1    WRITE  DECB1,TS,DTFBT1,OUTAREA,
                 130,,3,MF=E
```

In the above example, device 4 (rln 3) of the control unit identified by DTFBT1 is selected. Its buffer is first erased, then the 130-byte message in OUTAREA is written to the device buffer. DECB1 may have been created by a previous WRITE macro instruction specifying MF=L.

The CCB of the device written to will remain queued after completion of the operation.

The channel program generated and executed is:

```
Erase Write   SLI   (from the user area)
```

Write Unprotected Erase (TUS)

This operation erases all unprotected text fields in a specified terminal buffer. It also resets all unprotected fields with a modified attribute to nonmodified, restores the keyboard if the device is a display station, and positions the cursor to the first character position of the first unprotected field. This operation can be used to allow a terminal operator to make a succession of entries without the program's having to rewrite the same display screen format. It is requested by issuing the WRITE macro instruction with the TUS optype specified.

All operands of the WRITE macro instruction are applicable except the entry operand. The rln operand specifies the device. Although no output message is written, the area operand must specify a real address. The length operand must be specified as 1.

An example of the WRITE Unprotected Erase macro instruction is:

```
WRUNERS   WRITE  DECB1,TUS,DTFBT1,OUTAREA,
                 1,,2,MF=E
```

In the above example, device 3 (rln 2) of the control unit identified by DTFBT1 is selected and all unprotected fields in the device buffer are erased and marked nonmodified, the keyboard is restored, and the cursor is positioned to the first character position of the first unprotected field.

The CCB of the device will remain queued after completion of the operation.

The channel program generated and executed is:

Erase All Unprotected

SIX-BIT STRUCTURED DATA

Certain data in 3270 data streams has a six-bit structure and the two high-order bits for eight-bit structure are determined by the setting of the six low-order bits in the byte (see Appendix K). Six-bit structured data includes the WCC, attribute character, and cursor and buffer addresses; for more information, see IBM 3270 Information Display System Component Description, GA27-2749.

BUFFERING

Either programmer-defined I/O areas or buffering provided by BTAM may be used by the local 3270 application program. See the heading "Buffering" in the general section "Assembly Considerations."

ERROR RECOVERY

For device errors, BTAM provides automatic error retry for the local 3270 if, in the BTMOD macro instruction, L3277=YES and either ERLOGIC=E or ERLOGIC=C have been specified. The application program is not notified if error recovery was invoked and recovery was successful. The operation is posted complete with a successful completion code.

For program errors and unrecoverable device errors, information is returned to the application program by BTAM. This information consists of:

- Return codes in register 15 following READ and WRITE macro instructions.
- A completion code placed in the DECB.
- If the completion code is X'41', sense information in the DECB.

Return codes and completion codes are described under "Posted Error Information" in the general section "Assembly Considerations." Completion codes and sense byte information for the local 3270 are summarized in Figure 37. Related recovery actions are described in Figure 38. The contents of the DECB are described under "Posting in the DECB" in the general section "Assembly Considerations."

ERROR RECORDING

Facilities are provided in DOS for recording hardware errors. If RMSR=YES is specified in the BTMOD macro instruction, the RMSRTAB macro instruction can be used to create RMSR tables. RMSR (Recovery Management Support Recording) is described in Appendix H.

ONLINE TESTING: RFT

BTAM provides the facility, known as Request-For-Test (or RFT), to test devices, including the 3270 display stations and printers, to assure that they are working properly. These online tests are requested from a display station by the terminal operator. (For the local 3270, tests may not be requested from the program using the ONLTST macro instruction; this macro instruction is not applicable for the local 3270.)

The requested tests may be sent to the display station from which they were requested or to another device attached to the same control unit and defined in the same DTFBT macro instruction. The procedure for requesting an online test is described later in this section.

Any of six EBCDIC tests, 23 through 28, may be requested. Some are for different models of the display station and some are for the printer. The tests for the local 3270 are described with those for the remote 3270 in the section "Online Terminal Test for Binary Synchronous Communication" in the general section "Assembly Considerations."

COMPLETION CODE	SENSE BYTE	CAUSE	CONSOLE MSG	ACTION (See Figure 38)
X'41'	X'01'	OPERATION CHECK - Invalid 3270 data stream. For example, illegal buffer address.	4B44I	3
X'41'	X'02'	CONTROL CHECK - The device fails to respond within a specified time period to CU attempts to communicate.	4B43I	1
X'41'	X'08'	DATA CHECK - 1. Parity check detected by the CU on data or control words received from the device. 2. A CU detection of a cursor check. 3. An internal CU parity check.	4B32I	1
X'41'	X'0C'	DATA CHECK, UNIT SPECIFY - 1. Parity check detected by the device on data or control words received from the CU. 2. A display station detected a cursor check. 3. An internal parity check on the device buffer.	4B32I	4
X'41'	X'20'	BUS OUT CHECK - CU detects a parity check on a command or any data byte received from the channel.	4B31I	1
X'41'	X'40'	INTERVENTION REQUIRED - The device is not ready.	4B30I	2
X'41'	X'80'	COMMAND REJECT - The CU received an invalid channel command.	4B34I	5
X'42'	N/A	A Request-For-Test message has been received on a READ Initial operation.	N/A	6
X'54'	N/A	RESETPL successfully terminated a READ Initial with no data transfer. (Non-productive operation)	N/A	7
X'58'	N/A	Cancel condition detected by BTAM.	N/A	8
X'64'	N/A	Buffer contents are unreliable due to RFT processing.	N/A	9
X'7F'	N/A	Normal completion.	N/A	7

<u>NOTE</u>: Printer operations are posted complete at Channel End; therefore DOS/BTAM is unable to:

1. Retry failing printer operations which occur after CE, or

2. Notify the application of an I/O error because the operation has already been posted complete.

The application will be informed of a printer error upon the next issuance of a READ or WRITE to that printer. A return code of X'30' will be returned in register 15 indicating that the previous print operation did not complete successfully. (See X'30' return code description.)

Figure 37. Completion Code and Sense Byte Information for the Local 3270

ACTION NUMBER	ACTION
1	Retry the failing operation a desired number of times. If the problem persists, handle as an unrecoverable error.
2	Periodically retry the operation until the situation is corrected.
3	Analyze and correct the output data stream and reissue the macro instruction. (See Operation Check message.) This may require recoding and reassembly.
4	Reconstruct the entire buffer image and retry the failing I/O operation. If unsuccessful after a desired number of retries, handle as an unrecoverable error.
5	Handle as an unrecoverable error.
6	Issue a TWAIT macro with TERMTST specified.
7	Proceed normally.
8	Correct the cancel condition. This may require recoding and reassembly.
9	Reinitialize the entire buffer with a WRITE Erase macro instruction.

Figure 38. Error Recovery Actions for the Local 3270

BTAM support for the online testing facility is optional; it may be requested by coding in the application program:

- TST3277=YES in the BTMOD macro instruction
- TERMTST=YES in the DTFBT macro instruction
- TERMTST in the TWAIT macro instruction

In order to receive standard IBM maintenance, the online test facility must be included in the application program.

A test request message will be received only as the result of a READ Initial macro instruction. When a completion code of X'42' is found, following return from the WAIT macro instruction associated with the READ Initial, a test request message has been received. The application program must now issue a TWAIT macro instruction with the TERMTST operand specified. As a result of TWAIT, BTAM will send the requested tests and then reinitiate the original READ Initial operation. It may be simpler to use TWAIT with TERMTST specified in all cases where WAIT is desired.

On return from TWAIT, the original READ Initial is completed with the data that was read now in the input area; the program may continue as though the test request had not been received.

Since writing the test to the device will have changed the buffer contents, the application program will be informed on the first attempt to read or write to the tested device so that the program can reinitialize the terminal buffer. This information is provided by a completion code of X'64' following the TWAIT if the device on which the READ Initial was performed was the same device to which tests had just been written, or by a return code of X'30' in register 15 if the first attempt to perform I/O to the device is on a subsequent READ or WRITE macro instruction. During the time that the test is in progress, no device attached to the same control unit will be accessible to the program.

Minimum Input Area Sizes for RFT

If an application program is handling its own input/output areas, in order for RFT to be performed by devices controlled by that program, the minimum input area size must be 300 bytes (13 bytes plus the largest test pattern, 287 bytes). If dynamic buffering is being performed, the minimum buffer size that can be specified in the BUFL operand is 300.

How to Request An Online Test from a Local 3270

To initiate an online test, a display station operator must:

1. Assure himself that the screen is unformatted (one way to do this is to press the CLEAR key, then the RESET key).

2. With the cursor now at location 0, type in a message in the format:

X	X	Y	Y	N	ADDR		

where XX is a number 23 through 28 specifying the desired test; YY is a number 01 through 99 specifying the number of times the test is to be written to the device (if the device is a printer, the test can only be sent one time); N is the number 3 (indicating the length of ADDR); and ADDR is a three-digit hexadecimal number specifying the CUU (channel-unit) address of the device to which the test is to be sent. Alphabetic characters in the hexadecimal number must by typed in uppercase.

3. Press the TEST REQUEST key.

The test should now appear at the selected display station or printer.

ONLINE TESTING: CONCURRENT USE OF OLTEP

A local 3270 testing facility in addition to Request for Test (RFT) is the Online Test Executive Program (OLTEP). Whereas RFT (for the local 3270) can be initiated only from display terminals, OLTEP is initiated only from the system console. A special BTAM facility allows OLTEP to test local 3270 devices (control units and terminals) controlled by an application program. (Control units and terminals attached either locally or remotely can also be tested by OLTEP without use of this facility; however, they cannot currently be in use by a BTAM application program.) For OLTEP to run concurrently with the BTAM application program, it must be run in the background partition while the BTAM application is in a foreground partition.

To allow the BTAM application to continue running concurrently with OLTEP processing, the console operator invokes the BT3270SC program prior to beginning OLTEP. The BT3270SC program will allow local 3270 devices under the control of a BTAM application to be temporarily placed under OLTEP control. While OLTEP has control, the devices will simply appear "busy" to the BTAM application program. After the initial BT3270SC processing is complete, OLTEP is invoked. (How to run OLTEP is explained in DOS OLTEP, GC24-5086.) After OLTEP is complete, the system operator must run the BT3270SC program again to restore control of the devices to the BTAM application.

Preparing the Program for Concurrent Use of OLTEP

The BTAM application programmer must anticipate that it may be desirable to run OLTEP concurrently with the application. When a READ or WRITE macro instruction is issued and an X'34' is returned in register 15, the terminal or possibly the control unit is under the control of OLTEP. The program should interpret this to mean the device will be busy until OLTEP completes and continue processing using other devices. The READ or WRITE should be reissued periodically for the local 3270 device under control of OLTEP until a code other than X'34' is returned. Note: If the application program has no other devices with which to perform I/O, or if the program finds all devices on a control unit returning X'34', it should not continuously issue a READ or WRITE when the X'34' is returned. To do so will not allow the other, lower priority partitions to be selected by the supervisor. The program should issue a READ Initial for the control unit and then issue a WAIT. After OLTEP control is relinquished, the read will be performed and control returned to the program.

How to Set Up and Run the BT3270SC Program

The following DOS job stream is required to place the BT3270SC program in the core image library so that it may be executed when desired by the system console operator:

```
// JOB CATALOG
// OPTION CATAL
   PHASE BT3270SC,*
   INCLUDE BT3270SC
/*
// EXEC LNKEDT
/&
```

After BT3270SC has been placed on the core image library, it may be invoked at any time either by card input or from the system console.

The card input required is:

```
// JOB EXECUTE
// EXEC BT3270SC
/*
/&
```

The system console input required is:

```
// JOB EXECUTE   (optional)
// EXEC BT3270SC
```

BT3270SC will then prompt the system console operator for information it requires. After BT3270SC has performed its initiation function, the system operator can run the OLTEP program on local 3270 control units and/or terminals being used by a BTAM application program. After OLTEP is complete, the system operator must once again invoke the BT3270SC program for processing that will allow the BTAM application to regain control of the tested devices.

How to Respond to the BT3270SC Program

After BT3270SC has been invoked, the message

```
4BS1D  INDICATE INITIATION OR COMPLETION
```

will appear on the system console. The operator should respond

INITIATION (or any abbreviation thereof)

if BT3270SC is doing processing prior to OLTEP; or

COMPLETION (or any abbreviation thereof)

if BT3270SC is doing processing following OLTEP.

Next, the message

4BS2D INDICATE DEVICE(S) TO BE PROCESSED

will appear. Acceptable responses are either

X'xyz'

or X'xyz' - X'xyz'

where the first response indicates a single device to be reserved for OLTEP testing (if BT3270SC is being run prior to OLTEP) or to be restored to the BTAM application program control (if BT3270SC is being run following OLTEP), and the second response indicates an inclusive range of devices. x indicates the channel to which the control unit is attached. yz indicates the device (00-FF). Although up to 32 devices may be attached to a single control unit, the number of devices processed at one time by BT3270SC may not exceed 16. If more than 16 devices on one control unit are to be processed, the devices in excess of 16 have to be specified in another range immediately after the first range has been processed. In this case, y must be an even number from 0 through E in the first range, and one more than that number in the second range. For example,

X'120' - X'12F'
X'130' - X'13F'

would reserve 32 devices attached to a control unit on channel 0.

Normally, the message

4BS3I ALL DEVICES PROCESSED

will appear after successful processing of the devices by BT3270SC. On initiation only, the foreground partition and the channel-unit address of the processed devices will appear on the console. The message

4BS7D RESPOND PROCEED OR TERMINATE

will then appear. The response from the console

PROCEED (or any abbreviation thereof)

will indicate more devices are to be processed;

TERMINATE (or any abbreviation thereof)

will indicate that processing is complete. Note: Approximately 15 percent of the local online tests invocable through OLTEP are designed to test display stations or printers. If only these tests are to be run under OLTEP, then only the affected display stations or printers should be indicated in the reply to

INDICATE DEVICE(S) TO BE PROCESSED.

The remaining tests are designed to test the 3272 control unit; these are sometimes called "card calling tests." If any of these tests are to be run by OLTEP, all the devices attached to a control unit must be specified in response to INDICATE DEVICE(S) TO BE PROCESSED. If all devices are not specified, OLTEP does not execute the tests on the control unit.

What to Do If the BTAM Application Ends During OLTEP

On completion of OLTEP, it is possible that the BTAM application in the foreground partition to which the tested devices are to be reassigned has terminated. The BTAM application may have come to end-of-job or been canceled or otherwise ended. In this case, it will still be necessary to assign the devices specified when BT3270SC was run prior to OLTEP back to the BTAM partition. These devices must be processed by BT3270SC completion processing before they are assigned to any other use or specified as unassigned.

If a HOLD was issued on the devices, UNA or RELSE should not be issued until BT3270SC completion processing restores the devices back to the foreground partition. If a HOLD was not issued, the devices must first be restored to the foreground partition by BT3270SC completion processing; then they must be made unassigned by issuing DVCDN and DVCUP statements for each device.

Assigning Devices During OLTEP

All device assignments performed during OLTEP processing should be temporary background assignments (// ASSGN).

Examples of BT3270SC Processing

In these examples, the devices are all defined by an application program running in the foreground 1 (F1) partition. Devices may also be processed that are under control of an application program in the F2 partition.

Here is a sequence of messages and replies that might appear on the system console when BT3270SC is run before OLTEP:

PROGRAM:	4BS1D	INDICATE INITIATION OR COMPLETION
OPERATOR:	INIT	
PROGRAM:	4BS2D	INDICATE DEVICE(S) TO BE PROCESSED
OPERATOR:	X'140' - X'14F'	
PROGRAM:	4BS3I	ALL DEVICES PROCESSED
	4BS6I	F1 DEVICES: X'140', X'141', etc.
	4BS6I	F1 DEVICES: X'146', X'147', etc.
	4BS6I	F1 DEVICES: X'14C',X'14D',X'14E', X'1F'
	4BS7D	RESPOND PROCEED OR TERMINATE
OPERATOR:	PROC	
PROGRAM:	4BS2D	INDICATE DEVICE(S) TO BE PROCESSED
OPERATOR:	X'150' - X'156'	
PROGRAM:	4BS3I	ALL DEVICES PROCESSED
	4BS3I	F1 DEVICES: X'150', X'151', etc.
	4BS3I	F1 DEVICES: X'156'
	4BS7D	RESPOND PROCEED OR TERMINATE
OPERATOR:	TERM	

Here is a sequence of messages and replies that might appear on the system console when BT3270SC is run after OLTEP:

PROGRAM:	4BS1D	INDICATE INITIATION OR COMPLETION
OPERATOR:	COMP	
PROGRAM:	4BS8D	INDICATE PARTITION
OPERATOR:	F1	
PROGRAM:	4BS2D	INDICATE DEVICES TO BE PROCESSED
OPERATOR:	X'140' - X'14F'	
PROGRAM:	4BS3I	ALL DEVICES PROCESSED
	4BS7D	RESPOND PROCEED OR TERMINATE
OPERATOR:	PROC	
PROGRAM:	4BS8D	INDICATE PARTITION
OPERATOR:	F1	
PROGRAM:	4BS2D	INDICATE DEVICES TO BE PROCESSED
OPERATOR:	X'150' - X'156'	
PROGRAM:	4BS3I	ALL DEVICES PROCESSED
	4BS7D	RESPOND PROCEED OR TERMINATE
OPERATOR:	TERM	

2260 TO 3270 CONVERSION

A special facility is provided in BTAM to allow BTAM application programs written for local or remote 2260s to be modified for use with the local or remote 3270. Although not all features of the 3270 can be utilized, this facility does have the advantage of allowing quick conversion from the 2260 to the 3270 without requiring the 2260 data processing portions of the application program to be rewritten.

Essentially, this facility provides a macro instruction (SCANREQ) that invokes a routine to translate the 3270 data stream into the 2260 format on input and translate the 2260 data stream format to that of the 3270 before output.

This facility is described fully in IBM 2260 BTAM and 2260 GAM to IBM 3270 BTAM Conversion Guide, GC27-6975.

SUPPORTED DEVICES

BTAM supports the following Audio Response Unit attached directly to a multiplexer channel:

- IBM 7770 Audio Response Unit

IBM 7770 AUDIO RESPONSE UNIT

GENERAL INFORMATION

- Input Messages: Input messages received by the IBM Audio Response Unit are transferred to main storage as 8-bit representations of the telephone company A B C code (except those messages dialed from IBM 3944 terminals, which are transferred to the processor directly in EBCDIC representation). Input messages are interpreted by the user's program to format an appropriate response.

- Output Messages: Output messages from the System/370 are converted to audio response by the 7770 and routed to the telephone terminals that originated inquiries.

  The 7770 accepts an output message in the form of drum addresses that are used to locate specific words stored on the 7770 audio drum.

  The user is advised against adding pause words at the end of an audio message that is to be followed by a Read (as in WRITE Invitational or WRITE Conversational) so that the caller is not likely to enter the inquiry while still in the Write operation. This would cause an overrun and loss of data. The user should add one or more pause words at the end of an audio message to be followed by a disconnect (reset option or CONTROL Disable) in order to avoid the click of the disconnect being heard right after the last word.

- Vocabulary: The 7770 provides a basic vocabulary of 32 words. With an expansion of words in groups of 16, the 7770 can offer a maximum vocabulary of up to 128 words per unit.

- Configuration: The IBM 7770 Audio Response Unit is attached to a multiplexer channel. It presents to the channel a unique interface regardless of the data sets and the terminal to which it is connected. Since code conversion is the user's responsibility, the handling of the control unit is independent of the terminals used. Therefore, any reference to the 7770 also implies the terminals to which it is connected.

PROGRAMMING INFORMATION

BTAM does not support the 7770 Audio Response Unit on a nonswitched line. Therefore, SWITCH=YES must be specified in BTMOD when AUDIO=YES is specified. Since the IBM 7770 terminal cannot be polled or addressed, the following macro instructions must not be used with audio response units:

DFTRMLST
RESETPL
CHGNTRY

The 'entry' operand in the WRITE macro instruction will be used to specify the address where the input message should be read when the conversational or invitational options are used. The maximum length of the input message is specified in the MSGL operand of the DTFBT macro instruction.

Buffer management is not permitted because it does not serve any useful purpose with audio response units. Because these units have real-time requirements, the user cannot afford to wait for a buffer to be released from the pool on a read operation. For this reason, an audio application program must set aside the I/O area it needs rather than use a buffer pool.

In order to save main storage space while retaining an efficient usage of the Voice Code Translators (VCT), the user may decide to use only as many output areas as he has output channels. No problem arises if the conversational optype is not used. If it is used, the need arises for a method of using the output area during the time used by an input operation. No FEATURE option is allowed for the IBM 7770. When the P-bit is posted in the DECB after a WRITE Conversational macro instruction, it signals that the Read operation has started and, consequently, that the output buffer used by the previous Write operation is no longer used and can be filled with an audio message for another line.

Terminal to CPU

There is one macro instruction available for communication from an audio response unit to the CPU. The optype for this macro instruction is:

TI - READ Initial

READ Initial (TI): The problem program issues a READ TI to enable the line to receive a call and to read the inquiry message.

The channel program generated and executed is:

1. Enable
2. Read message

CPU to Terminal

There are six macro instructions available for communication from the CPU to an audio response unit. The optypes for these macro instructions are:

TI	- WRITE Initial
TIR	- WRITE Initial with Reset
TC	- WRITE Invitational
TT	- WRITE Continue
TTR	- WRITE Continue with Reset
TV	- WRITE Conversational

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to receive a call and send a specified audio message to the caller.

The WRITE TI functions as follows. The line is enabled to receive a call. When the call is received, the user-specified audio message is sent to the terminal that called. When the reset option is specified and no error occurs, the channel program is reinitialized automatically and no posting is made. This loop is stopped when the line group is closed. The same procedure is followed in case the caller hangs up. If an error occurs, the line will be disconnected and the error will be posted. No reinitialization takes place after an error.

The WRITE Initial macro instruction allows the user to send the first segment of a multisegment audio message when the complete data connot be loaded into main storage. A READ macro instruction must not follow the WRITE TI, since the audio response units require that a Read CCW be chained to the previous CCW to avoid possible loss of input characters. If the user needs a WRITE Initial followed by a read sequence, he may use the WRITE Invitational macro instruction.

When specified with reset, the WRITE Initial macro instruction implements the information mode. When a user dials the audio response unit, no inquiry is expected; rather a fixed-format audio message (for example, a weather forecast) is expected. Then the line is disconnected and made ready for another call. To change the mode in which the line is used, the line group must be closed, the change made, and the line opened again, unless no further activity is required for the line group.

The channel program generated and executed is:

1. Enable
2. Write message
3. Disable -- TIR only
4. TIC to (1) -- TIR only

WRITE Invitational (TC): The problem program issues a WRITE TC macro instruction to enable the line, write a message to the calling terminal, and read an input message into main storage.

The channel program generated and executed is:

1. Enable
2. Write message
3. Read inquiry

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR to send an audio message in multiple segments. The last segment can be sent with a WRITE TTR to disconnect the line at the end of the transaction.

The channel program generated and executed is:

1. Write message
2. Disable -- TTR only

WRITE Conversational (TV): The problem program issues a WRITE TV macro instruction to send an audio message and to read the next inquiry.

The channel program generated and executed is:

1. Write message

2. Read inquiry

CONTROL Macro Instructions

CONTROL Initial (TI): The problem program issues a CONTROL TI to enable the line to receive a call. When a call is received, the connection is established and the operation is posted as complete.

The channel program generated and executed is:

1. Enable

CONTROL Disable (TD): The problem program issues a CONTROL TD to disconnect the line when the transaction has been completed. This macro instruction must be used when the reset option cannot be used.

The channel program generated and executed is:

1. Disable

WRITE Positive Acknowledgment (TA) - Display or Printer: The problem program issues a WRITE TA to indicate to the sending station that the message was received without error and to end the operation.

The channel program generated and executed is:

1. Write STX

2. Write EOT sequence

WRITE Negative Acknowledgment (TN) - Display or Printer: The problem program issues a WRITE TN to indicate to the sending station that the message was received in error. The operation is ended.

The channel program generated and executed is:

1. Write EOT sequence

SUPPORTED START-STOP DEVICES

BTAM supports the following devices on non-switched lines attached to a multiplexer channel through the IBM 2701 Data Adapter Unit, IBM 2702 Transmission Control, or IBM 2703 Transmission Control.

- IBM 1030 Data Collection System
- IBM 1050 Data Communication System
- IBM 1060 Data Communication System
- IBM 2260 Display Station - IBM 2848 Display Control attached through the 2701 only
- IBM 2740 Communication Terminal and IBM 2760 Optical Image Unit (optional attachment)
- WU 83B3 Selective Calling Stations
- Western Union Plan 115A Outstations
- World Trade telegraph terminals

BTAM supports the following devices on switched networks attached to a multiplexer channel through an IBM 2701, 2702, or 2703:

- IBM 1050 Data Communication System
- IBM 2740 Communication Terminal and IBM 2760 Optical Image Unit (optional attachement)
- WU Model 33/35 Teletypewriter Exchange Terminal

Note: Terminals which 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 the IBM-supplied products or programs may have on such terminals.

WESTERN UNION PLAN 115A TERMINALS

Terminal to CPU

There is one macro instruction available for communication from the 115A to the CPU. The optype for this macro instruction is:

TI - READ Initial

Each terminal on a line is "invited to send" with a two-character code as specified in the DFTRMLST macro instruction. The first character is always an 'X'. The second character identifies the terminal. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST}, ({xxyy-hexchars,}...)

The list may be an open list or a wrap-around list. The four characters xxyy are the hexadecimal representation of the "invitation to send" code as it appears in shifted Baudot code (see Appendix B). Thus, xx is always 17; yy is the letter that selects the station.

Read Initial (TI): The problem program issues a READ TI macro instruction to start or restart polling (by sending the "invitation to send") and to read a message.

The READ TI macro first puts all the terminals on the line into control mode. Then the "invitation to send" code is sent to poll the first (or next) terminal in the terminal list. The response to polling is read into the first byte of the input area. If the response is negative (single character V or M), the next terminal in the list is polled. If the end of an open list is reached before a positive response is received, the operation is posted as complete (X'54') in the ECB.

A message itself is considered as a positive response. The message is read until an EOT is received, and the operation is posted complete in the ECB. If the user-specified count reaches zero before an EOT is received, a lost data error condition will be indicated.

The channel program generated and executed is:

1. Write EOT sequence
2. Write invitation code
3. Read response to polling
4. Read message

CPU to Terminal

There is one macro instruction available for communication from the CPU to a 115A. The optype for this macro instruction is:

TI - WRITE Initial

Each terminal on a line is addressed by a two-character call code as specified in the DFTRMLST macro instruction. The first character is the circuit call. The second character identifies the terminal. Normally, the addressing list contains a single call code. However, a message can be sent to multiple terminals on a line. The format of the DFTRMLST used is;

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

The four characters xxyy are the hexadecimal representation of the call code as it appears in shifted Baudot code (see Appendix B). Thus, xx is the circuit call and yy is the station identifier.

WRITE Initial (TI): The problem program issues a WRITE TI macro instruction to send a message to one or several stations on a line. The WRITE TI first puts all the terminals on the line into control mode. Then the call code from the terminal list is sent to address the first terminal in the list.

The response is read into the DECB response field. If the response is negative (no response), the operation is posted as complete-with-error in the ECB. If the response is positive (single character V or M) and the end of the list has not been reached, the next terminal in the list is addressed.

When the end of the list is reached, the message is transmitted. The format of the message should be as follows:

SP text FIGS H LTRS LTRS
LTRS LTRS LTRS LTRS LTRS

The length specified in the WRITE macro instruction must be exact (including control characters).

The channel program generated and executed is:

1. Write EOT sequence
2. Write call code
3. Read response to addressing
4. Write message

Terminal to Terminal

BTAM does not support terminal-to-terminal traffic on a line. However, there is one macro instruction available for use if such terminal-to-terminal traffic does occur. The optype for this macro instruction is:

TB - WRITE Break

WRITE Break (TB): The WRITE TB macro instruction is not used for normal message processing. It should be issued when a data check occurs due to terminal-to-terminal traffic. It is used to stop a terminal that is sending data. The transmission control unit sends continuous space signals to the line adapter, thus causing a "break" in the line.

The length operand specifies the number of space signals to be sent. The area and entry operands are not required for a WRITE Break macro instruction.

WU 83B3 SELECTRIC CALLING STATIONS

Terminal to CPU

There is one macro instruction available for communication from the 83B3 to the CPU. The optype for this macro instruction is:

TI - READ Initial

Each terminal on a line has a unique two-character transmitter start code (TSC) as specified in the DFTRMLST macro instruction. Normally, all of the terminals on a line are polled with a single terminal list. The format for the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST}, ({xxyy-hexchars,}...)

The list may be either an open list or a wraparound list. The four characters xxyy are the hexadecimal representation of the TSC as it appears in the shifted Baudot code (see Appendix B).

READ Initial (TI): The problem program issues a READ TI macro instruction to start or restart polling and to read a message from the 83B3. The READ TI first puts all the terminals on the line into control mode. Then the transmitter start code from the terminal list is sent to poll the first terminal in the list. The response to polling is read into the first byte of the input area. If the response is negative (single character V or M), the next terminal on the list is polled. If the end of an open list is reached before a positive response is received, the operation is posted as complete (X'54') in the ECB. A message itself is considered as a positive response. The message is read until an EOT is received, and the operation is posted as complete in the ECB. If the user-specified count reaches zero before an EOT is received, a lost data error condition will be indicated.

The message will appear in main storage as follows:

CR LF LTRS text H

The channel program generated and executed is:

1. Write EOT sequence
2. Write TSC
3. Read response
4. Read message

CPU to Terminal

There is one macro instruction available for communication from the CPU to an 83B3. The optype for this macro instruction is:

TI - WRITE Initial

Each terminal on a line has a unique two-character call directing code (CDC) as specified in the DFTRMLST macro instruction. Normally, the terminal list contains a single CDC. Messages can be sent to multiple terminals on a line. The format for the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, (xxyy-hexchars,...)

The four characters xxyy are the hexadecimal representation of the CDC as it appears in shifted Baudot code (see Appendix B).

WRITE Initial (TI): The problem program issues a WRITE TI macro instruction to send a message to one or several stations. The WRITE TI first puts all the terminals on the line into control mode. Then the call directing code from the terminal list is sent to address the first (or next) terminal in the list. A letters shift character is sent to trigger the response to addressing.

The response is read into the DECB response field. If any terminal on the line fails to answer (negative response), the operation is posted as complete-with-error in the ECB. If the response is positive (single character V or M) and the end of the list has not been reached, the next terminal in the list is addressed.

When the end of the list is reached, the message is transmitted. The format of the message sent should be as follows:

CR LF LTRS text FIGS H LTRS

The length specified in the WRITE macro instruction must be exact (including the control characters).

The channel program generated and executed is:

1. Write EOT sequence
2. Write CDC

3. Write LTRS

4. Read response

5. Write message

<u>Terminal to Terminal</u>

BTAM does not support terminal-to-terminal traffic on a line. However, there is one macro instruction available for use if such terminal-to-terminal traffic does occur. The optype for this macro instruction is:

TB - WRITE Break

<u>WRITE Break (TB)</u>: The WRITE TB macro instruction is not used for normal message processing. It should be issued when a data check occurs due to terminal-to-terminal traffic. It is used to stop a terminal that is sending data. The transmission control unit sends continuous space signals to the line adapter, thus causing a "break" in the line.

The length operand specifies the number of space signals to be sent. The area and entry operands are not required for a WRITE Break macro instruction.

WU MODEL 33/35 TWX TERMINALS

Terminal to CPU

There are five macro instructions available for communication from the TWX terminal to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TV - READ Conversational

TVR - READ Conversational with Reset

TS - READ Skip

TWX terminals are not polled. The line connection is established on terminal request; the terminal calls the CPU through a switched network, and the CPU answers the call. When transmission is to be originated by the terminal operator dialing the CPU (READ Initial), the format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	IDLST,0,numsent-integer, sentchar-hexchars

The 0 specifies that this is an answer list. Numsent is the number of characters to be sent, and sentchar is a string of characters sent when the terminal dials in. The recommended sequence is: timing character (null), carriage return (CR), line feed (LF), delete, subscriber-selected characters, carriage return, line feed, and transmitter on (Xon). Each of these characters is coded as two hexadecimal digits that represent the character as it appears in the eight-bit data interchange code (see Appendix B).

When the transmission from terminal to CPU (READ Conversational) is to follow a READ or WRITE Initial or a WRITE Conversational, the format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	IDLST,0, nocnsent-integer, cntrlseq-hexchars

The 0 specifies that this is an answer list. Nocnsent is the number of characters in the control sequence. Cntrlseq is the combination of control characters used to prepare the terminal to transmit. A commonly-used sequence is: Xon, 1 to 4 subscriber characters, Xoff.* This may be used provided that the READ TV is preceded by a WRITE Initial. The TD call-in key at the terminal must be on if the above sequence is used. If a READ Initial was the first sequence, Xon alone may be used. These control sequences start the tape transmitter.

In reading data keyed in by an operator from the TWX keyboard, the problem program may want to send a meaningful sequence of characters to be printed at the TWX. Such a sequence could alert the TWX operator to the fact that the CPU is ready for the next keyboard entry.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR when the terminal operator is to originate transmission. The READ Initial functions as follows. The line is disabled in case this was not done previously. The line is then conditioned to receive incoming calls. Before this is done, the line is busy to incoming calls. When a call is received, fifteen pad characters are sent. The control sequence is then sent just as it appears in the terminal list, and the message is read. If READ TIR was specified and no transmission error occurred, the terminal is put into control mode and the line connection is broken.

The channel program generated and executed is:

1. Disable
2. Enable
3. Write pad characters
4. Write control sequence
5. Read message
6. Write EOT -- TIR only
7. Disable -- TIR only

The message transmitted from the terminal must end with one of the following: WRU, Xon, Xoff, or the EOT sequence. If the message is terminated by WRU, Xon, or Xoff, a READ Conversational can be executed. The EOT sequence breaks the line connection, requiring that the next operation be another READ Initial or a WRITE Initial.

----

*This sequence is optional and must be wired into the terminal during installation.

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR to read a message when line connection has been previously established by a READ Initial or WRITE Initial. READ TV can also follow a WRITE Conversational.

The READ Conversational functions as follows. The control sequence, as it appears in the terminal list, is sent to prepare the terminal to transmit. Then the message is read. If READ TVR was specified and no transmission error occurred, the terminal is put into control mode and the line connection is broken.

The channel program generated and executed is:

1. Write control sequence
2. Read message
3. Write EOT -- TVR only
4. Disable -- TVR only

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read from a terminal to clear the line, but it is not received in main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are five macro instructions available for communication from the CPU to the TWX terminal. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TN - WRITE Negative Acknowledgment

TWX terminals are not addressed. Line connection is established upon CPU request; the CPU calls the remote terminal through a switched network. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numsent-integer, tidseq-hexchars

The dialcount is the number of digits in the phone number of the terminal. Dialchars are the actual digits in the phone number. Numsent is the number of characters in the terminal ID sequence. Tidseq is the string of characters that is to be compared to a terminal identification sequence sent from the terminal. This ID is sent automatically by the terminal and is determined at the time the terminal is installed.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to send a message when transmission is to be originated by the CPU calling the terminal. In addition, the WRITE TIR resets the terminal.

The WRITE Initial functions as follows. The line is disabled in case this was not done previously. The terminal is dialed using the dial digits of the phone number contained in the terminal list. If the terminal ID sequence is not equal to the sequence in the terminal list, the operation is posted as complete with error. If the sequences are equal, the message is written. The message sent should not end with a control character or sequence. If WRITE TIR was specified, and no error is detected during transmission, the terminal is put into control mode and line connection is broken.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Read terminal ID sequence
4. Write message
5. Write EOT -- TIR only
6. Disable -- TIR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR to send a message following a READ Initial, READ Conversational, WRITE Initial, or another WRITE Conversational. The message, which must not end with a control character, is sent to the terminal. If WRITE TVR was specified and no error occurred in transmission, the terminal is put into control mode and the line is reset.

The channel program generated and executed is:

1. Write message
2. Write EOT -- TVR only
3. Disable -- TVR only

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN to turn off the terminal motors and disconnect the line. It may be used to disconnect the line after any READ or WRITE operation issued without reset.

The channel program generated and executed is:

1. Write EOT
2. Disable

IBM 1030 TERMINALS

Terminal to CPU

There are seven macro instructions available for communication from the 1030 to the CPU. The optypes for these macro instructions are:

- TI - READ Initial
- TIR - READ Initial with Reset
- TT - READ Continue
- TTR - READ Continue with Reset
- TP - READ Repeat
- TPR - READ Repeat with Reset
- TS - READ Skip

Each IBM 1031 Input Station on a line is polled with one polling character as specified in the DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST}, ({xx-hexchars,}...)
symbol	DFTRMLST	{SSALST, SSAWLST}, {xx-hexchars,}...

The list may be an open list, a wraparound list, or an Auto Poll list (open or wraparound). The two characters xx are the hexadecimal representation of the terminal polling character as it appears in the six-bit BCD transmission code (see Appendix B).

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to start or restart polling and to read a message from the 1030. In addition, the READ TIR resets the line. The channel programs for READ TI and READ TIR vary according to the type of terminal list specified in the DFTRMLST macro instruction.

OPENLST or WRAPLST: When an open list or wraparound list is specified, the READ Initial functions as follows. First, all of the terminals on the line are put into control mode. Then the polling character from the terminal list is sent to poll the first entry in the list. The response to polling is read into the first byte of the input area. If the response is negative, the next terminal in the list is polled. If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (EOA), and the next character received is not an EOB or EOT, the remainder of the text is read. If the second character received is an EOB or EOT, the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred during transmission, the line is reset.

The channel program generated and executed using OPENLST or WRAPLST is:

1. Write EOT sequence
2. Write polling character
3. Read response to polling
4. Read message
5. Write positive response and EOT sequence -- TIR only

SSALST: When an open start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling character from the terminal list is sent to the starting entry (or next entry). If the end of the list is reached without a positive response, the channel program is halted. On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOB is received, and the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSALST is:

1. Write EOT sequence
2. Poll starting entry
3. NOP
4. Read response to polling
5. Read message
6. Write positive response and EOT sequence -- TIR only

SSAWLST: When a wraparound start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling character from the terminal list is sent to the starting entry (or next entry). If the end of the list is reached before a positive response is received, polling is restarted with the first entry in the list. On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOB is received, and the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSAWLST is:

1. Write EOT sequence
2. Poll starting entry
3. TIC to Command (5)
4. TIC to Command (7)
5. Poll first entry
6. TIC to Command (5)
7. Read response to polling
8. Read message
9. Write positive response and EOT sequence -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR to send a positive response after a successful READ Initial. The READ TT or READ TTR then functions exactly as a READ TI or READ TIR. The channel program for READ Continue varies according to the type of terminal list specified in the DFTRMLST macro instruction. The channel program generated and executed is:

1. Write positive response and EOT sequence
2. The remaining commands are the same as commands 2-5, 2-6, or 2-9 of the corresponding READ Initial.

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR to send a negative response after an unsuccessful READ TI, TT, or TP. The READ TP or READ TPR then functions exactly as a READ TI or READ TIR. The channel program varies according to the terminal list specified in the DFTRMLST macro instruction. The channel program generated and executed is:

1. Write negative response and EOT sequence
2. The remaining commands are the same as commands 2-5, 2-6, or 2-9 of the corresponding READ Initial.

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user. The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are six macro instructions available for communication from the CPU to the 1030. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

Each 1033 printer on a line is addressed with a single character code as specified in the DFTRMLST macro instruction. An addressing terminal list may contain only one printer address. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xx-hexchars)

The two characters xx are the hexadecimal representation of the addressing character as it appears in the six-bit BCD transmission code (see Appendix B). The list must be defined as an open list.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to establish connection with the printer and to send a message block to it. In addition, the WRITE TIR resets the line.

The WRITE Initial functions as follows. A special character sequence (EOT sequence and address select) is sent to condition the 1031 to receive the printer addressing character. Then the addressing character is sent as it appears in the terminal list. Another character (X'02') is sent to prepare the 1031 to send the response character.

The response to addressing is read into the first byte of the DECB response field. If the response is negative, the operation is posted as complete in the ECB. If the response is positive, the message is written.

The response to checking is read into the DECB response field. If the response is positive and TIR was specified, the line is reset.

The channel program generated and executed is:

1. Write EOT sequence and address select
2. Write addressing character
3. Write "1"
4. Read response to addressing
5. Write message
6. Read response to checking
7. Write EOT sequence -- TIR only

Each message written must begin with EOA and end with EOB. Pad characters (X'DF') must be included between all message and/or control characters. A three-character delay is used between the output printing characters. Additional delay is required for carriage return, line feed, tabs, etc.

<u>WRITE Continue (TT) and WRITE Continue with Reset (TTR)</u>: The problem program issues a WRITE TT or WRITE TTR macro instruction to send a message to the printer after initial contact has been established by a WRITE TI. In addition, the WRITE TTR resets the line.

The WRITE Continue functions as follows. The message block is written to the 1033. The response to checking is read into the DECB response field. If the response is positive and WRITE TTR was specified, the line is reset.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT sequence -- TTR only

The first character of a message sent by a WRITE Continue should be a text character, not an EOA. Other information concerning message format is the same as for WRITE Initial.

<u>WRITE Positive Acknowledgment (TA)</u>: The problem program issues a WRITE TA to send a positive response and to stop line activity after a successful READ operation.

The channel program generated and executed is:

1. Write positive response and EOT sequence

<u>WRITE Negative Acknowledgment (TN)</u>: The problem program issues a WRITE TN to send a negative response and to stop line activity after a WRITE operation or an unsuccessful READ operation.

The channel program generated and executed is:

1. Write EOT sequence

IBM 1050 TERMINALS ON NONSWITCHED NETWORKS

Terminal to CPU

There are seven macro instructions available for communication from the 1050 to the CPU. The optypes for these macro instructions are:

- TI - READ Initial
- TIR - READ Initial with Reset
- TT - READ Continue
- TTR - READ Continue with Reset
- TP - READ Repeat
- TPR - READ Repeat with Reset
- TS - READ Skip

Each component on a line is polled with a two-character code as specified in the DFTRMLST macro instruction. The first of these characters identifies the terminal. The second either identifies a single component of that terminal or is the common polling character (0). The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST}({xxyy-hexchars}...)
symbol	DFTRMLST	{SSALST, SSAWLST},{xxyy-hexchars}...

The list may be an open list, a wraparound list, or an Auto Poll list (open or wraparound). The four characters xxyy are the hexadecimal representation of the polling characters as they appear in six-bit BCD transmission code (see Appendix B). Thus, xx identifies the terminal and yy the component.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to start or restart polling and to read a message from the 1050. In addition, the READ TIR resets the line. The channel programs for READ TI and READ TIR vary according to the type of terminal list specified in the DFTRMLST macro instruction.

OPENLST or WRAPLST: When an open list or wraparound list is specified, the READ Initial functions as follows. First, all of the terminals on the line are put into control mode. Then the polling characters from the terminal list are sent to poll the first entry in the list. The response to polling is read into the first byte of the input area. If the response is negative, the next terminal in the list is polled. If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (EOA), and the next character received is not EOB or EOT, the remainder of the text is read. If the second character received is an EOB or EOT, the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred during transmission, the line is reset.

The channel program generated and executed using OPENLST or WRAPLST is:

1. Write EOT sequence
2. Write polling characters
3. Read response to polling
4. Read message
5. Write positive response and EOT sequence -- TIR only

SSALST: When an open start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling characters from the terminal list are sent to the starting entry (or next entry). If the end of the list is reached without a positive response, the channel program is halted. On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is then read into the input area until an EOB is received, and the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSALST is:

1. Write EOT sequence
2. Poll starting entry
3. NOP
4. Read response to polling
5. Read message

6. Write positive response and EOT sequence -- TIR only

SSAWLST: When a wraparound start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling characters from the terminal list are sent to the starting entry (or next entry). If the end of the list is reached before a positive response is received, polling is restarted with the first entry in the list. On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is then read into the input area until an EOB is received, and the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSAWLST is:

1. Write EOT sequence
2. Poll starting entry
3. TIC to Command (5)
4. TIC to Command (7)
5. Poll first entry
6. TIC to Command (5)
7. Read response to polling
8. Read message
9. Write positive response and EOT sequence -- TIR only

<u>READ Continue (TT) and READ Continue with Reset (TTR)</u>: The problem program issues a READ TT or READ TTR after a successful READ Initial, READ Continue, or READ Repeat to send a positive response and read subsequent blocks of data from the same component without repolling.

The READ Continue functions as follows. A positive response is written and the message is read. If READ TTR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed is:

1. Write positive response
2. Read message
3. Write positive response and EOT sequence -- TTR only

<u>READ Repeat (TP) and READ Repeat with Reset (TPR)</u>: The problem program issues a READ TP or READ TPR* macro instruction following an unsuccessful read operation to reread the last message from the same component into the same storage area as in the previous operation.

The READ repeat functions as follows. A negative response is written and the message is read. If READ TPR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed is:

1. Write negative response
2. Read message
3. Write positive response and EOT sequence -- TPR only

For paper tape or card input, READ TP requires the line correction feature. For these components, READ TP can be used only twice in succession, for a total of three tries. Furthermore, it can be used with the 1054 Paper Tape Reader only for messages that are equal to or less than 312 characters long.

<u>READ Skip (TS)</u>: The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

<u>CPU to Terminal</u>

There are six macro instructions available for communication from the CPU to the 1050. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

Each terminal on a line is addressed with a two-character code as specified in the DFTRMLST macro instruction. The first of these two addressing characters identifies the terminal. The second selects one or all of the components of that terminal and conditions the component(s) to receive a message. The terminal list may contain more than one component address. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

The four characters xxyy are the hexadecimal representation of the addressing characters as they appear in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the terminal and yy identifies the component. The list must be defined as an open list.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to address one or more components of a terminal and to send a message to it. In addition, the WRITE TIR resets the line.

The WRITE Initial functions as follows. All terminals on the line are put into control mode. The addressing characters from the terminal list are sent to address the device. The response to addressing is read into the first byte of the DECB response field. A negative response from any component causes the operation to be posted as complete-with-error in the ECB.

When all entries in the list have responded positively, an EOA is sent to terminate addressing. The message is then sent to the terminal. The response to checking is read into the DECB response field. If WRITE TIR was specified and there was no error in transmission, the terminal is put into control mode.

The channel program generated and executed is:

1. Write EOT sequence
2. Write addressing characters
3. Read response to addressing
4. Write EOA
5. Write message
6. Read response to checking
7. Write EOT sequence -- TIR only

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR to send additional messages (without readdressing) after a successful WRITE Initial. In addition, WRITE TTR puts the terminal into control mode.

The WRITE Continue functions as follows. The message block is written to the 1050. The response to checking is read into the DECB response field. If the response is positive and WRITE TTR was specified, the line is reset.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT sequence -- TTR only

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA to stop receiving messages from a terminal before the end of transmission is received. A positive response is sent to the terminal and the line is reset.

The channel program generated and executed is:

1. Write EOA and EOT sequence

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN to send a negative response and reset the line. The WRITE TN may also be issued after a WRITE Initial or WRITE Continue to reset the line. This is the same function provided by WRITE TIR and WRITE TTR.

The channel program generated and executed is:

1. Write EOT sequence

Programming Notes

To read all of a message, it may be necessary to issue at least two READ macro instructions to poll a terminal component. The first macro instruction must be a READ Initial. The second macro instruction must be a READ Continue. Figure 39 illustrates channel, line, and 1050 activity for READ Initial and READ Continue.

A WRITE Continue macro instruction can be issued after any READ, if the last character received is not an EOT. The first character of the message must be an EOA, which selects all ready units of the terminal. Figure 40 illustrates channel, line, and 1050 activity for READ Initial and WRITE Continue.

After READ operation, the user may decide to stop terminal activity by issuing:

1. A WRITE Positive Acknowledgment, if the READ operation was successful.

2. A WRITE Negative Acknowledgment, if the READ operation was not successful.

3. A READ or WRITE Initial if the READ operation was not successful, but the user wants to restart activity on the line.

The last macro instruction of a write sequence can be a WRITE Initial with Reset, a WRITE Continue with Reset, or a WRITE Negative Acknowledgment, if the user wants to reset the line. However, a WRITE Initial or a READ Initial will reset the line and restart activity.

The user must provide some of the line control characters that pertain to the messages he wishes to send. See Figure 41.

Program	Channel	Line	1050
READ Initial	1. Write C C C	C C C	
	2. Write polling characters	A6	
	3. Read response		D
	4. Read message block		Block 1 B LRC
READ Continue	1. Write Y	Y	
	2. Read message block		Block 2 B LRC
READ Continue	1. Write Y	Y	
	2. Read message block		C

Figure 39. READ TI and READ TT Macro instructions

Program	Channel	Line	1050
READ Initial	1. Write C C C	C C C	
	2. Write polling characters	A6	
	3. Read response		D
	4. Read message block		Block 1 B LRC
WRITE Continue	1. Write message block	D Block 2 B LRC	
	2. Read response		Y

Figure 40. READ TI and WRITE TT Macro Instructions

MACRO Instruction	I/O Area	
	First character provided by user	Last character provided by user
WRITE initial	First text character	B
WRITE initial with reset	First text character	B
WRITE continue	First text character	B
WRITE continue with reset	First text character	B
WRITE continue (conversational mode)	D	B
WRITE continue with reset (conversational mode)	D	B

Figure 41. Line Control Characters for Messages

IBM 1050 TERMINALS ON SWITCHED NETWORKS

Terminal to CPU

There are nine macro instructions available for communication from the 1050 to the CPU. The optypes for these macro instructions are:

- TI - READ Initial
- TIR - READ Initial with Reset
- TT - READ Continue
- TTR - READ Continue with Reset
- TP - READ Repeat
- TPR - READ Repeat with Reset
- TV - READ Conversational
- TVR - READ Conversational with Reset
- TS - READ Skip

Normally, terminal-to-CPU transmission occurs when the problem program issues a READ Initial referring to an answering-polling terminal list to enable a line. Prior to the enable, no calls can be received by the transmission control unit. The terminal operator dials the CPU when he has something to send. This completes the enable command. BTAM polls the terminal that dialed, using the polling characters specified in the DFTRMLST macro instruction. The format of the DFTRMLST used for an answering-polling list is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0, ({xxyy-hexchars,}...)

The 0 indicates that the list is an answering list. The four characters xxyy are the hexadecimal representation of the component polling characters as they appear in six-bit BCD transmission code (see Appendix B). The answering-polling list must contain polling characters that are appropriate for all terminals that can call in on the line. Normally, the common polling character 0 is used to poll all components of a terminal. All terminals should have the same identification character.

Another possible, though not normally appropriate, technique allows terminal-to-CPU transmission to occur by dialing the terminal and then polling. This technique is useful primarily for unattended operations using card or paper tape input. The problem program issues a READ Initial to dial the terminal and then to poll a component. If ready, the component can then begin sending data. The format of the DFTRMLST used for this calling-polling list is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars, ({xxyy-hexchars,}...)

Dialcount is the number of digits in the phone number of the terminal. Dialchar is the actual phone number of the terminal. The four characters xxyy are the hexadecimal representation of the two polling characters for the component or components.

<u>READ Initial (TI) and READ Initial with Reset (TIR)</u>: The problem program issues a READ TI or READ TIR to start polling and to read a message from the 1050. In addition, the READ TIR breaks the line connection. The channel programs for READ TI and READ TIR vary according to the type of terminal list specified in the DFTRMLST macro instruction.

<u>Answering-polling list</u>: When the terminal list specified is an answering-polling list, the READ Initial functions as follows. First, the line is reset. Then the line is conditioned to receive incoming calls. Before this is accomplished, the line is busy to incoming calls. When a call is received, fifteen pad characters are sent. The terminal that dialed is then placed into control mode. Polling characters from the terminal list are sent to poll the first component, and the response is read into the first byte of the input area.

If the response to polling is negative, the next component is polled. If the end of the list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (EOA character) and if the first text character is not EOB or EOT, the remainder of the message is read. If the second character (first text character) is EOB or EOT, the operation is posted as complete in the ECB. The message is read until an EOB is received and the operation is posted as complete (with or without error) in the ECB.

If READ TIR was specified and no error occurred during transmission, the terminals are placed in control mode and the line connection is broken.

The channel program generated and executed using an answering-polling list is:

1. Disable
2. Enable
3. Write 15 pad characters
4. Write EOT sequence
5. Write polling characters
6. Read response to polling
7. Read message
8. Write EOA and EOT -- TIR only
9. Disable -- TIR only

Calling-Polling list: When the terminal list specified is a calling-polling list, the READ Initial functions as follows. First, the line is reset in case the line was not disconnected after the previous operation. Then, the terminal is dialed using the call digits as specified in the calling-polling terminal list. This READ Initial then functions exactly as a READ TI with an answering-polling list does after a call is received.

The channel program generated and executed using a calling-polling list is:

1. Disable
2. Dial call digits
3. Write 15 pad characters
4. Write EOT sequence
5. Write polling characters
6. Read response to polling
7. Read message
8. Write EOA and EOT -- TIR only
9. Disable -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR macro instruction following a successful READ TI, TT, TP, or TV to read another message from the same component. No polling is performed. In addition, the READ TTR puts the terminal into control mode.

The READ Continue functions as follows. A positive response is written to the terminal. The message is read until an EOB is received. The data received will be either the next message or an EOT that indicates the end of transmission. If READ TTR was specified, the terminal is placed into control mode and the line connection is broken.

The channel program generated and executed is:

1. Write positive response
2. Read message
3. Write EOA and EOT -- TTR only
4. Disable -- TTR only

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR macro instruction following an unsuccessful read operation for retransmission of data that was received in error. In addition, the READ TPR puts the terminal into control mode.

The READ Repeat macro instruction functions as follows. A negative response is written to the terminal. The message is read until an EOB is received. If READ TPR was specified, the terminal is placed in control mode and the line connection is broken.

The channel program generated and executed is:

1. Write negative response
2. Read message
3. Write EOA and EOT -- TPR only
4. Disable -- TPR only

For paper tape or card input, READ Repeat requires the line correction feature. For these components, it can only be used twice in succession for a total of three tries. Furthermore, it can be used with the 1054 Paper Tape Reader only for messages that are no more than 312 characters long.

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR to poll one of the components of a terminal when line connection is already established. In addition, the READ TVR macro instruction puts the terminal into control mode.

The READ Conversational macro instruction functions as follows. The terminal is placed into control mode. Polling characters from the terminal list are sent to poll the first component, and the response is read into the first byte of the input area.

If the response to polling is negative, the next component is polled. If the end of the list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (EOA character) and if the first text character is not EOB or EOT, the remainder of the message is read. If the second character (first text character) is EOB or EOT, the operation is posted as complete in the ECB. The message is read until an EOB is received and the operation is posted as complete (with or without error) in the ECB.

If READ TVR was specified and no error occurred during transmission, the terminals are placed in control mode and the line connection is broken.

The channel program generated and executed is:

1. Write EOT sequence
2. Write polling characters
3. Read response to polling
4. Read message
5. Write EOA and EOT -- TVR only
6. Disable

The terminal list referred to by a READ Conversational must be an open-type polling list. It may be a separate list, defined with the OPENLST operand in the DFTRMLST macro, or it may be the same list used with READ Initial (defined as DIALST). To use the original dial list, the entry operand of the READ TV must refer to "symbol+n+1," where symbol is the name field of the DFTRMLST and n is the number of dial digits specified.

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The READ Skip macro instruction functions as follows. The rest of the message is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are eight macro instructions available for communication from the CPU to the 1050. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

Normally, CPU-to-terminal transmission occurs when the problem program issues a WRITE Initial referring to a calling-addressing list to dial the terminal and to address the component. The format of the DFTRMLST used to define this calling-addressing list is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars, ({xxyy-hexchars,}...)

Dialcount is the number of dial digits in the terminal's phone number. Dialchar is the actual phone number of the terminal. The four characters xxyy are the hexadecimal representation of the terminal and component identification as it appears in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the terminal and must be the same for all entries in the list, and yy identifies the component.

Another possible, though not normally appropriate, technique allows CPU-to-terminal transmission to occur when the problem program issues a WRITE Initial referring to an answering-addressing terminal list to enable the line. The terminal operator dials the CPU. One or more of the components of the terminal that dialed are then addressed using the addressing characters specified in the DFTRMLST macro instruction. A positive response must be received from all components addressed before the message is sent. The format of the DFTRMLST used for an answering-addressing list is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0, ([xxyy-hexchars,]...)

The four characters xxyy are the hexadecimal representation of the addressing characters for the components as they appear in the six-bit BCD transmission code (see Appendix B). If this technique is used, the terminal list must be appropriate for all terminals that can call in on the line.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to address a component and to send the first message. In addition, the WRITE TIR resets the terminal and disables the line. The channel programs for WRITE TI and WRITE TIR vary according to the type of terminal list used.

Calling-Addressing List: If a calling-addressing list is used, the WRITE Initial functions as follows. The line is disconnected and the terminal is dialed using the dial characters specified in the DFTRMLST macro instruction. Fifteen pad characters are sent and the terminal is then placed into control mode. The first component of the terminal dialed is addressed using the addressing characters specified in the DFTRMLST.

The response to addressing is read into the first byte of the DECB response field. If the response is positive, and the end of the list has not been reached, the next component is addressed in the same manner. If the response is negative for any component in the list, the operation is posted as nonproductive in the ECB. If the response is positive and the end of the list has been reached, an EOA is sent to terminate addressing.

Next the message is written, and the response to checking is read into the DECB response field. If there was no error in transmission and WRITE TIR was specified, the terminal is reset and the line is disabled.

The channel program generated and executed when using a calling-addressing list is:

1. Disable
2. Dial call digits
3. Write pad characters
4. Write EOT sequence
5. Write addressing characters
6. Read response to addressing
7. Write EOA
8. Write message
9. Read response to checking
10. Write EOT -- TIR only
11. Disable -- TIR only

Answering-Addressing List: If an answering-addressing list is used, the WRITE Initial functions as follows. The line is first disconnected and then it is enabled. When the terminal operator dials the CPU, this WRITE Initial then functions exactly as the WRITE Initial using a calling-addressing list after the terminal is dialed.

The channel program generated and executed when using an answering-addressing list is:

1. Disable
2. Enable
3. Write pad characters
4. Write EOT sequence
5. Write addressing characters
6. Read response to addressing
7. Write EOA
8. Write message
9. Read response to checking
10. Write EOT -- TIR only
11. Disable -- TIR only

The message sent by a WRITE Initial or WRITE Initial with Reset must begin with an EOA and end with an EOB. Since BTAM provides the EOA character, the user's message should begin with a text character and end with an EOB.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR macro instruction following a successful READ TI, TT, TV, or TP, or WRITE TI, TT, or TV to send a message to all receiving devices (after a READ) or to the component or components that have been addressed (after a WRITE). No terminal list is required for a WRITE Continue.

The WRITE TT macro instruction functions as follows. The message is written to the appropriate components (see paragraph above), and the response to checking is read. If there was no error in transmission and WRITE TTR was specified, the terminal is reset and the line is disabled.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT -- TTR only
4. Disable -- TTR only

When the WRITE Continue macro instruction is issued after a Write operation (except when retrying a message sent in error), the first character of the message must be a text character. After a Read operation or when retrying a message sent in error, the first character must be an EOA.

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after a read operation to send a positive response, address a component, and send a message. The WRITE Conversational macro instruction functions as follows. A positive response is sent, and the terminal is put into control mode. The first component of the terminal is addressed using the addressing characters specified in the DFTRMLST.

The response to addressing is read into the first byte of the DECB response field. If the response is positive and the end of the list has not been reached, the next component is addressed in the same manner. If the response is negative for any component in the list, the operation is posted as nonproductive in the ECB. If the response is positive and the end of the list has been reached, an EOA and the message are written. The message must end with EOB.

The response to checking is read into the DECB response field. If there was no error in transmission and WRITE TVR was specified, the terminal is reset and the line is disabled.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Write addressing characters
3. Read response to addressing
4. Write EOA
5. Write message
6. Read response to checking
7. Write EOT -- TVR only
8. Disable -- TVR only

The terminal list referred to by the WRITE Conversational macro instruction must be an open-type list. It may be a separate list defined with the OPENLST operand, or it may be the same list used with WRITE Initial (defined as DIALST). To use the original dial list, the entry operand of the WRITE TV must refer to "symbol+n+1," where symbol is the name field of the DFTRMLST and n is the number of dial digits specified.

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA macro instruction to stop line activity after a successful Read operation. A positive response and EOT sequence are sent. The line connection is then broken.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Disable

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN after an unsuccessful Read operation to send a negative response and break the line connection, or after a Write operation to break the line connection.

The channel program generated and executed is:

1. Write EOT sequence
2. Disable

Programming Notes

Line Connection: Line connection on a switched network must be established by a WRITE Initial or a READ Initial. Connection must be broken by a READ/WRITE with the reset option, a WRITE Positive Acknowlegment, or a WRITE Negative Acknowledgment before another READ/WRITE Initial can be issued for this line.

Conversational Mode at Terminal Component Level: A WRITE Continue macro instruction can be issued after any READ macro if the message received is not EOT. The first character of the message must be an EOA. This option provides the problem program with the facility to answer an inquiry.

Conversational Mode at Terminal Level: The READ/WRITE Conversational allows the problem program to initiate further activity on any other component of the same terminal without reestablishing initial contact.

<u>Polling</u>: To poll, the first macro instruction must be either a READ Initial (or READ Initial with Reset) if the line connection is to be established, or a READ Conversational (or READ Conversational with Reset) if the line connection has already been established.

When data is received in error, the line connection is not broken. The user may issue one of the following macro instructions:

1. A READ Repeat, to send a negative response and ask for retransmission.
2. A WRITE Negative Acknowledgment, to reset the line (negative response) and break the line connection.
3. A READ to reset the line (negative response) and restart activity.
4. A READ Conversational with Reset will have the same effect as above, but at the end of the operation, the line connection will be broken.

When data is received correctly and the reset option has been specified in the first macro instruction, the line connection is broken. The user must issue a READ Initial or WRITE Initial macro instruction to restart activity. If the reset option was not specified, the user may issue one of the following macro instructions:

1. A READ Continue, to send a positive response and to ask for transmission of the next block of the message.
2. A READ Continue with Reset will have the same effect as above, but at the end of the operation, the line connection will be broken.
3. A WRITE Continue (first character sent is EOA), to send a positive response to the sending device and then a message to all receiving devices of the terminal.
4. A WRITE Continue with Reset will have the same effect as above, but at the end of the operation the line connection will be broken.
5. A WRITE Positive Acknowledgment, to send a positive response and to break the line connection.
6. A WRITE Conversational to send a positive response and address a terminal component.

<u>Addressing</u>: The first macro instruction of an addressing sequence may be:

1. A WRITE Initial or a WRITE Initial with Reset, if the line connection is to be established.
2. A WRITE Conversational or a WRITE Conversational with Reset, if the line connection is already established.

When the terminal receives a message block in error, the problem program may issue a WRITE Continue or a WRITE Continue with Reset macro instruction to retry transmission. However, the first character sent must be an EOA.

IBM 1060 TERMINALS

Terminal to CPU

There are seven macro instructions available for communciation from the 1060 to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TT - READ Continue

TTR - READ Continue with Reset

TP - READ Repeat

TPR - READ Repeat with Reset

TS - READ Skip

Each IBM 1062 Teller Terminal on a line is polled with a two-character code as specified in the DFTRMLST macro instruction. The first of these two polling characters identifies the IBM 1061 Control Unit (normally only one per line, though up to 26 are possible). The second character identifies the 1062 Teller Terminal (one or two per 1061). The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xxyy-nexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars,}...

The list may be an open list, a wraparound list, or an Auto Poll list (open or wraparound). The four characters xxyy are the hexadecimal respresentation of the polling characters as they appear in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the 1061 and yy the 1062.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to start or restart polling and to read a message from the 1060. In addition, the READ TIR resets the line. The channel programs for READ TI and READ TIR vary according to the type of terminal list specified in the DFTRMLST macro instruction.

OPENLST or WRAPLST: When an open list or wraparaound list is specified, the READ Initial functions as follows. First, all of the terminals on the line are put in control mode. Then the polling characters from the terminal list are sent to poll the first entry in the list. The response to polling is read into the first byte of the input area. If the response is negative, the next terminal in the list is polled. If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If a positive response and the first text character are received, the remainder of the text is read. When an EOB is received, the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred during transmission, the line is reset.

The channel program generated and executed using OPENLST or WRAPLST is:

1. Write EOT sequence
2. Write polling characters
3. Read response to polling
4. Read message
5. Write positive response and EOT sequence -- TIR only

SSALST: When an open start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling characters from the terminal list are sent to the starting entry (or next entry). On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOB is received. The operation is posted as complete in the ECB. If the end of the list is reached without a positive response, the channel program is halted.

If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSALST is:

1. Write EOT sequence
2. Poll starting entry
3. NOP
4. Read response to polling

5. Read message

6. Write positive response and EOT sequence -- TIR only

SSAWLST: When a wraparound start-stop Auto Poll list is specified, the READ Initial functions as follows. First, all the terminals on the line are put into control mode. Then the polling characters from the terminal list are sent to the starting entry (or next entry). On a positive response to polling, the index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOB is received. If the end of the list is reached before a positive response is received, polling is restarted with the first entry in the list. The operation is posted as complete in the ECB. If READ TIR was specified and no error occurred in transmission, a positive response is sent to indicate that the operation occurred without error, and the line is reset.

The channel program generated and executed using SSAWLST is:

1. Write EOT sequence

2. Poll starting entry

3. TIC to Command (5)

4. TIC to Command (7)

5. Poll first entry

6. TIC to Command (5)

7. Read response to polling

8. Read message

9. Write positive response and EOT sequence -- TIR only.

<u>READ Continue (TT) and READ Continue with Reset (TTR)</u>: The problem program issues a READ TT or READ TTR to send a positive response after a successful READ TI, TT, or TP. The READ TT or READ TTR then functions exactly as a READ TI or READ TIR. The channel program for READ Continue varies according to the type of terminal list specified in the DFTRMLST macro instruction. The channel program generated and executed is:

1. Write positive response and EOT sequence

2. The remaining commands are the same as commands 2-5, 2-6, or 2-9 of the corresponding READ Initial.

<u>READ Repeat (TP) and READ Repeat with Reset (TPR)</u>: The problem program issues a READ TP or READ TPR to send a negative response after an unsuccessful READ TI, TT, or TP. The READ TP or READ TPR then functions exactly as a READ TI or READ TIR. The channel program varies according to the terminal list specified in the DFTRMLST macro instruction.

The channel program generated and executed is:

1. Write negative response and EOT sequence

2. The remaining commands are the same as commands 2-5, 2-6, or 2-9 of the corresponding READ Initial.

<u>READ Skip (TS)</u>: The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user. The channel program generated and executed is:

1. Read skip

<u>CPU to Terminal</u>

There are four macro instructions available for communication from the CPU to the 1060. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

Each 1062 on a line is addressed with a two-character code as specified in the DFTRMLST macro instruction. An addressing terminal list may contain only one terminal address. The first of the two addressing characters identifies the 1061 Control Unit. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xxyy-hexchars)

The four characters xxyy are the hexadecimal representation of the addressing characters as they appear in the six-bit BCD transmission code (see Appendix B). The list must be defined as an open list.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to establish connection with the 1062 and to send a message block to it. In addition, the WRITE TIR resets the line.

The WRITE Initial functions as follows. First, all the terminals on the line are put into control mode. Then the addressing characters are sent as they appear in the terminal list.

The response to addressing is read into the first byte of the DECB response field. If the response is negative, the operation is posted as complete in the ECB. If the response is positive, the message is written.

The response to checking is read into the DECB response field and the operation is posted as complete (with or without error) in the ECB. If the response is positive and TIR was specified, the line is reset.

The channel program generated and executed is:

1. Write EOT sequence

2. Write addressing character

3. Read response to addressing

4. Write message

5. Read response to checking

6. Write EOT sequence -- TIR only

Each message written must begin with EOA and end with EOB. Delete, idle, or other nonprintable characters must be included in the output message, where necessary, to allow time for tab, carriage return, and line feed operations.

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA after a successful READ operation to send a positive response and to stop line activity.

The channel program generated and executed is:

1. Write positive response and EOT sequence

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN after a WRITE operation or an unsuccessful READ operation to send a negative response and to stop line activity.

The channel program generated and executed is:

1. Write EOT sequence

IBM 2740 COMMUNICATION TERMINALS

The IBM 2740 Communication Terminal may be tailored for the user's application. BTAM includes support for the basic 2740 and the checking, dial, transmit control, station control, and OIU features that are available on the 2740. The different features are referred to as follows: 'C' for checking, 'D' for dial, 'T' for transmit control, 'S' for station control, and 'O' for the 2760 Optical Image Unit.

Two restrictions exist in installing the 2740 features: station control and dial are mutually exclusive and dial is a prerequisite for transmit control.

The following ten 2740 configurations are supported by BTAM:

2740	(basic)
2740C	(with checking)
2740CO	(with checking and the 2760 OIU)
2740D	(with dial)
2740DC	(with dial and checking)
2740DCO	(with dial, checking, and the 2760 OIU)
2640DT	(with dial and transmit control)
2740DTC	(with dial, transmit control, and checking)
2740S	(with station control)
2740SC	(with station control and checking)

The checking feature is the only one that informs the problem program that a message has been received. Without the checking feature, a READ macro instruction may be associated with a WRITE macro instruction. The receipt of a message by a 2740 without checking is indicated by the terminal operator, who sends either a positive or negative response.

The first character received from the 2740 is not placed in main storage. This character is normally the end-of-addressing character that is generated when the user presses the BID key and sets the line to control mode. Therefore, when the problem program issues a READ macro instruction that conditions the control unit to receive a message and reads a byte of sense information, the problem program should restore the end-of-addressing character before making an attempt to write the message.

The following condition applies to a 2740 with the checking feature on a switched line: following transmission to the terminal, an EOT should be transmitted to put the line in control mode before issuing an operation beginning with a Disable command.

2740 BASIC

Terminal to CPU

There are three macro instructions available for communication from a 2740 basic to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Leading Reset

TS - READ Skip

Point-to-point operations are supported for the 2740 basic using a nonswitched line. These operations are performed on a contention basis with a terminal seizing the line when the operator presses the BID key. The BID key sends an end-of-addressing character to indicate to the CPU that the terminal has data to be read.

The DFTRMLST macro instruction is not used because there is no polling of the 2740 basic.

READ Initial (TI) and READ Initial with Leading Reset (TIR): The problem program issues a READ TI or READ TIR to read a message from the 2740. In addition, the READ TIR places all terminals in standby mode before reading the message.

The READ Initial functions as follows. The control unit is conditioned to receive a message, and a byte of sense information is read into the DECB response field. Then the message is read. The EOA, sent by the transmitting terminal when the operator presses the BID key, is deleted. The text, received by the CPU and all terminals on the line, ends with an EOT, which causes the 2740 to revert to standby mode. Another terminal can then seize the line when its operator presses the BID key.

The READ TIR provides a means of resetting the 2740 before allowing the remote operator to enter a message. This permits the problem program to leave the 2740 in text mode after a WRITE operation without reset until the problem program is ready to receive a message. The problem program can then issue a READ TIR to return the terminal to control mode. This prevents any characters from being entered before the program is ready to receive them.

The channel program generated and executed is:

1. NOP -- Write EOT for TIR

2. Prepare

3. Sense

4. Read data

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are four macro instructions available for communication from the CPU to the 2740 basic. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

The DFTRMLST macro instruction is not used because there is no addressing of the 2740 basic.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to send a message to a terminal. In addition, the WRITE TIR sets the terminal in control mode.

The WRITE Initial functions as follows. An EOA character and fifteen idle characters are sent over the line. The message is then written to the terminal and must be followed by an EOB. If WRITE TIR was specified, the terminal is placed in standby mode. Successive uses of WRITE TI after the terminal has been placed in text mode by the first WRITE TI cause the EOA to be printed as a pound sign (#).

The channel program generated and executed is:

1. Write EOA and 15 idle characters

2. Write message

3. Write EOT sequence -- TIR only

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT OR WRITE TTR to send additional message blocks after a WRITE Initial. In addition, WRITE TTR sets the terminal in control mode.

The WRITE Continue functions as follows. The message is sent to the terminal. If WRITE TTR was specified, the terminal is placed in standby mode. The WRITE TT and WRITE TTR macro instructions can be used only after the terminal has been placed in text mode by a WRITE TI.

The channel program generated and executed is:

1. Write message

2. Write EOT sequence -- TTR only

2740 WITH CHECKING

The 2740 Communication Terminal with the checking feature allows point-to-point operations between a terminal and the CPU. The DFTRMLST macro instruction is not needed because there is no polling or addressing of the 2740 with checking.

Terminal to CPU

There are seven macro instructions available for communication from the 2740 with checking to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Leading Reset

TT - READ Continue

TTR - READ Continue with Reset

TP - READ Repeat

TPR - READ Repeat with Reset

TS - READ Skip

The line is seized by depressing the BID key of the SELECTRIC typewriter, which sends an EOA over the line. The message is sent and is followed by an EOB.

READ Initial (TI) and READ Initial with Leading Reset (TIR): The problem program issues a READ TI or READ TIR to read a message from the terminal.

The READ Initial functions as follows. The control unit is conditioned to receive a message, and a byte of sense information is read into the response field of the DECB. The EOA sent by the transmitting terminal is deleted and does not appear in the input area. The message, which ends with EOB, is read into main storage.

The channel program generated and executed is:

1. NOP-Write EOT for TIR

2. Prepare

3. Sense

4. Read message

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR to read another message from the same terminal after a successful Read operation. In addition, READ TTR Resets the line.

The READ Continue macro instruction functions as follows. A positive response is sent to the terminal to indicate that the previous message was received successfully. Either a message block or an EOT is read. EOT indicates that transmission is ended. If READ TTR was specified and no errors occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write positive response

2. Read message

3. Write EOA and EOT sequence -- TTR only

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR for retransmission of a message that was previously received in error. In addition, READ TPR resets the line.

The READ Repeat functions as follows. A negative response is sent to indicate that an error occurred during transmission of the previous message. The message is then reread. If READ TPR was specified and no errors occurred in transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write negative response

2. Read message

3. Write EOA and EOT sequence -- TPR only

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read to clear the line, but is not received in main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are eight macro instructions available for communication from the CPU to the terminal. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

The CPU sends an EOA to put the terminal in receive mode, then sends the message, which must end with an EOB.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to send a message to the terminal and read a response to checking. In addition, WRITE TIR resets the line.

The WRITE Initial functions as follows. The terminal is placed in receive mode and fifteen idle characters are sent to give the terminal motors time to reach operating speed. The message is sent and the response to checking is read. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOA and idle characters

2. Write message

3. Read response to checking

4. Write EOT sequence -- TIR only

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR to send a message after a successful Write operation. An EOA is not sent before the message. WRITE TTR also resets the line.

The WRITE Continue macro instruction functions as follows. The message, which must end with EOB, is sent to the terminal. The response to checking is read. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT sequence -- TTR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after a successful Read operation, or for reestablishing connection with the SELECTRIC typewriter after using the OIU. In addition, WRITE TVR resets the line.

The WRITE Conversational functions as follows. The terminal is placed in receive mode. The message, which must end with EOB, is sent and the response to checking is read. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOA
2. Write message
3. Read response to checking
4. Write EOT sequence -- TVR only

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA after a successful Read operation to send a positive response and turn the terminal motors off.

The channel program generated and executed is:

1. Write EOA and EOT sequence

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN after a Read or Write operation. After a Read operation, WRITE TN sends a negative response that turns the terminal motors off. After a Write operation, it turns the terminal motors off.

The channel program generated and executed is:

1. Write EOT sequence

2740 WITH CHECKING AND OIU

The 2740 Communication Terminal with the checking and OIU features allows point-to-point operations between a terminal and the CPU. The DFTRMLST macro instruction is not needed because there is no polling or addressing of the 2740 with checking and OIU.

Terminal to CPU

There are eight macro instructions available for communication from the 2740 with checking and OIU to the CPU. The optypes for these macro instructions are:

TI	- READ Initial
TIR	- READ Initial with Leading Reset
TT	- READ Continue
TTA	- READ Continue with Leading Acknowledgment
TTR	- READ Continue with Reset
TP	- READ Repeat
TPR	- READ Repeat with Reset
TS	- READ Skip

The operator seizes the line by pressing the BID key of the SELECTRIC typewriter or by pressing the light pen against a response point of the 2760 OIU, which sends an EOA over the line. The message is sent and is followed by an EOB.

Following the receipt of a message from the 2740 or 2760, a message may be sent to either the 2740 (text must be preceded by EOA and followed by EOB) or the 2760 (text must be preceded by EOA, the prefix character, and the letter O, and followed by EOB).

Following the receipt of a message from the 2740, a message may be read from the 2760 after the line is put in control mode. Conversely, following the receipt of a message from the 2760, a message may be read from the 2740 after the line is put in control mode.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is active but prior or concurrent with an operation issued by the problem program. The operation may be posted in error. Alternative actions the user may take when this condition occurs are:

- Issue a WRITE TCO to move the filmstrip to an error handling frame which will aid the operator in recovering the lost data.
- Issue a WRITE TV to write an error message on the 2740 keyboard.
- Issue a READ TIR (nonswitched line) or READ TV (switched line) to allow the 2760 operator to retry. The operator should be instructed to reprobe if he does not hear the audible tone within a reasonable period of time.

READ Initial (TI) and READ Initial with Leading Reset (TIR): The problem program issues a READ TI or READ TIR to read a message from the terminal.

The READ Initial macro instruction functions as follows. The control unit is conditioned to receive a message, and a byte of sense information is read into the response field of the DECB. The EOA sent by the transmitting terminal is deleted and does not appear in the input area. The message, which ends with EOB, is read into main storage.

The channel program generated and executed is:

1. NOP -- Write EOT for TIR
2. Prepare
3. Sense
4. Read message

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR to read another message from the same terminal after a successful Read operation. In addition, READ TTR resets the line.

The READ Continue macro instruction functions as follows. A positive response is sent to the terminal to indicate that the previous message was received successfully. Either a message or an EOT is read. EOT indicates transmission is ended. If READ TTR was specified and no errors occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write positive response
2. Read message
3. Write EOA and EOT sequence -- TTR only

Note: The READ TT and READ TTR operations are not supported following an operation in which data was received from the 2760 OIU.

READ Continue with Leading Acknowledgment (TTA): The problem program issues a READ TTA to give a positive response to data received from either the 2740 or the 2760, to put the line in control mode, and to read again, from either device.

If a message read from the 2760 is to be followed by additional data from either the 2740 or the 2760, READ TTA should be used. The EOA causes the audible tone and releases the probe interlock.

The READ TTA functions as follows. A positive response is sent to the terminal to indicate that the previous message was received successfully. The terminal is placed in control mode and input can be received from either the 2740 or 2760. The EOA received is deleted. A byte of sense information is read into the DECB response field. The message, which ends with EOB, is read into main storage.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Prepare
3. Sense
4. Read message

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR for retransmission of a message that was previously received in error. In addition, READ TPR resets the line.

The READ Repeat functions as follows. A negative response is sent to indicate that an error occurred during transmission of the previous message. The message is then reread. If READ TPR was specified and no errors occurred in transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write negative response
2. Read message

3. Write EOA and EOT sequence -- TPR only

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read to clear the line, but is not received in main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are ten macro instructions available for communication from the CPU to the terminal. The optypes for these macro instructions are:

TI	- WRITE Initial
TIR	- WRITE Initial with Reset
TIO	- WRITE Initial Optical
TT	- WRITE Continue
TTR	- WRITE Continue with Reset
TV	- WRITE Conversational
TVR	- WRITE Conversational with Reset
TCO	- WRITE Invitational Optical
TA	- WRITE Positive Acknowledgment
TN	- WRITE Negative Acknowledgment

The CPU sends an EOA to put the terminal in receive mode, then sends the message, which must end with an EOB.

A message sent to the 2740 may be followed by a message to the 2760. If the line is not put in control mode after writing to the 2740, the message to the 2760 must be preceded by the prefix character and the letter O, and followed by an EOB. If the line is put in control mode after writing to the 2740, the message to the 2760 must be preceded by EOA, the prefix character, and the letter O, and followed by an EOB.

A message sent to the 2760 may be followed by a message to the 2740. If the line is not put in control mode after writing to the 2760, the message to the 2740 must begin with text and must be followed by an EOB. If the line is put in control mode after writing to the 2760, the message to the 2740 must be preceded by an EOA and be followed by an EOB.

Following a WRITE to the 2740, a message may be read from the 2760 after the line is put in control mode.

The user may issue a WRITE TCO to write to the 2760 and then read from either the 2740 or 2760. If WRITE TIO (switched or nonswitched line) or WRITE TVO (switched line) is used to write to the 2760, and if input is then expected from either the 2760 or 2740, the user should issue a READ TV (switched line) immediately after the WRITE TIO or TVO completes.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is active but prior or concurrent with an operation issued by the problem program. The operation may be posted in error. Alternative actions the user may take when this condition occurs are:

- Issue a WRITE TCO to move the filmstrip to an error handling frame which will aid the operator in recovering the lost data.
- Issue a WRITE TV to write an error message on the 2740 keyboard.
- Issue a READ TIR (nonswitched line) or READ TV (switched line) to allow the 2760 operator to retry.

The operator should be instructed to reprobe if he does not hear the audible tone within a reasonable period of time.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to send a message to the terminal and read a response to checking. In addition, WRITE TIR resets the line.

The WRITE Initial functions as follows. The terminal is placed in receive mode and one idle character is sent to give the terminal motors time to reach operating speed. The message is sent and the response to checking is read. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOA and idle character

2. Write message

3. Read response to checking

4. Write EOT sequence -- TIR only

WRITE Initial Optical (TIO): The problem program issues a WRITE TIO to send a control message to the 2760 OIU and read the response.

The WRITE Initial Optical macro instruction functions as follows. The terminal is placed in receive mode and the message is identified as being for the 2760 OIU by the prefix character and letter O (not coded by the user). Three frame change characters are sent to the OIU. The F character describes the direction of film movement and end of message mode, and the A_1A_2 characters describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro instruction. The length operand is ignored. An EOB (not coded by the user) is then sent to the 2760.

The response to checking is read. If the response is positive, an EOT is sent to the 2760 to enable it to execute the function specified in the control message. If the response is negative, EOT is not sent and the error is indicated to the user. Successful completion of this macro instruction leaves the line in control mode. If the macro is concluded in error, the user should issue a WRITE TN to clear the error condition before retrying the WRITE TIO.

The channel program generated and executed is:

1. Write EOA, PRE, O
2. Write frame change
3. Write EOB
4. Read response to checking
5. Write EOT sequence

Note: Buffers may not be used with the WRITE TIO macro instruction.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR to send a message after a successful Write operation. An EOA is not sent before the message. WRITE TTR also resets the line.

The WRITE Continue functions as follows. The message, which must end with EOB, is sent to the terminal. The response to checking is read. If WRITE TTR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT sequence -- TTR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after a successful Read operation, or for reestablishing connection with the SELECTRIC typewriter after using the OIU. In addition, WRITE TVR resets the line.

The WRITE Conversational functions as follows. The terminal is placed in receive mode. The message (which must end with EOB) is sent and the response to checking is read. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOA
2. Write message
3. Read response to checking
4. Write EOT Sequence -- TVR only

WRITE Invitational Optical (TCO): The problem program issues a WRITE TCO to send a control message to the OIU and to read a message from it.

The WRITE Invitational Optical functions as follows. The terminal is placed in receive mode and the message is identified as being for the 2760 OIU by the prefix character and letter O (not coded by the user). Three frame change characters are sent to the OIU. The F character describes the direction of film movement and end of message mode, and the A_1A_2 characters describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro instruction. An EOB (not coded by the user) is then sent to the 2760.

The response to checking is read. If the response is positive, an EOT is sent to the 2760 to enable it to execute the function specified in the control message. If the response is negative, EOT is not sent and the error is indicated to the user. A message is read from the OIU. (The EOA in front of the message is deleted.) The address of the input area should be coded in the area operand, and the length operand of the WRITE macro instruction. The length may be 7 bytes or variable, depending on the end of message mode. If buffers are used, the area operand, rather than the length operand, must be coded as 'S'.

The channel program generated and executed is:

1. Write EOA, PRE, O
2. Write frame change
3. Write EOB
4. Read response to checking
5. Write EOT sequence
6. Prepare
7. Sense
8. Read message

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA after a successful Read operation to send a positive response and turn the terminal motors off.

The channel program generated and executed is:

1. Write EOA and EOT sequence

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN after a Read or Write operation. After a Read operation, WRITE TN sends a negative response that turns the terminal motors off. After a Write operation, it turns the terminal motors off.

The channel program generated and executed is:

1. Write EOT sequence

2740 WITH DIAL

With the dial-up adapter feature installed, the 2740 communications terminal can communicate over a common-carrier dial network.

Terminal to CPU

There are five macro instructions available for communication from the 2740 with dial to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TV - READ Conversational

TVR - READ Conversational with Reset

TS - READ Skip

Each terminal that has the dial feature must first establish the line connection by calling the CPU. After this is accomplished, the terminal's BID key can be depressed, thus the EOA character. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies that this is an answer list.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to receive a call and read a message. In addition, the READ TIR resets the line.

The READ Initial functions as follows. The line is reset in case it was not previously disabled. The line is enabled, and the control unit is conditioned to receive a call. The EOA sent from the terminal is deleted. A byte of sense information is then read into the DECB response field, and the message is read. The operation is posted as complete (with or without error) in the ECB. If READ TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable
2. Enable
3. Prepare
4. Sense
5. Read message
6. Write EOT -- TIR only
7. Disable -- TIR only

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR to perform a Read operation after a successful Write operation. In addition, the READ TVR resets the line.

The READ Conversational functions as follows. The control unit is conditioned to receive a message and a byte of sense information is read into the DECB response field. The message is then read. If READ TVR was specified and no errors occurred, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Prepare
2. Sense
3. Read message
4. Write EOT -- TVR only
5. Disable -- TVR only

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but it is not received in main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are five macro instructions available for communication from the CPU to a 2740 with dial. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TN - WRITE Disconnect

The CPU may call the terminal using dial digits specified in a DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

The dialcount is the number of digits in the phone number of the terminal. Dialchar is the actual phone number of the terminal.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to dial a terminal and send a message to that terminal. In addition, the WRITE TIR resets the line.

The WRITE Initial functions as follows. The line is disabled in case this was not done previously. The terminal is called using the dial digits specified in the DFTRMLST. Fifteen pad characters are sent over the line, and the message is sent to the terminal. The message must begin with EOA and end with an EOB. The operation is posted as complete (with or without error) in the ECB. If WRITE TIR was specified and no error occurred during transmission, the terminal motors are turned off and the line is disabled.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Write pad characters
4. Write message
5. Write EOT -- TIR only
6. Disable -- TIR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after a successful Read operation to send a message to the terminal with which the connection has been established.

The WRITE Conversational functions as follows. The message is written to the terminal. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write message
2. Write EOT -- TVR only
3. Disable

WRITE Disconnect (TN): The problem program issues a WRITE TN to disconnect the line after a Read or Write operation. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOT
2. Disable

2740 WITH DIAL AND CHECKING

The 2740 with dial and checking features requires that each terminal have the dial-up adapter special feature installed. The line connection is established by either the terminal calling the CPU or the CPU calling the terminal.

The checking feature consists of VRC (vertical redundancy check), which checks each character transmitted for parity, and LRC (longitudinal redundancy check), which checks each message block for correct transmission.

Terminal to CPU

There are nine macro instructions available for communication from the 2740 with dial and checking to the CPU. The optypes for these macro instructions are:

- TI - READ Initial
- TIR - READ Initial with Reset
- TT - READ Continue
- TTR - READ Continue with Reset
- TP - READ Repeat
- TPR - READ Repeat with Reset
- TV - READ Conversational
- TVR - READ Conversational with Reset
- TS - READ Skip

The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies that the terminal list is an answering list.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to answer a call from the terminal, and to read a message.

The READ Initial functions as follows. The line is disabled in case this was not done previously. The line is conditioned to receive a call. When a call is received, the control unit is conditioned to receive a message. A byte of sense information is read into the DECB response field.

The message is read into main storage. The EOA sent by the terminal is deleted. The message will end with an EOB, which is sent when the terminal operator presses the EOB key. The operation is posted as complete (with or without error) in the ECB. If READ TIR was specified and no errors occurred during transmission, a positive response is sent, the terminal motors are turned off, and the line is disabled.

The channel program generated and executed is:

1. Disable
2. Enable
3. Prepare
4. Sense
5. Read message
6. Write EOA and EOT -- TIR only
7. Disable -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR after a successful Read operation to send a positive response and to read more data. The line connection must have already been established.

The READ Continue functions as follows. A positive response is sent to indicate that the previous message block was successfully received. The message is then read. The message ends with an EOB. If READ TTR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is disabled.

The channel program generated and executed is:

1. Write Positive response
2. Read message
3. Write EOA and EOT -- TTR only
4. Disable -- TTR only

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR to reread the previous message block if an error occurred during transmission. In addition, READ TPR breaks the line connection.

The READ Repeat functions as follows. A negative response is sent to indicate that the previous message was received in error. The message is then read. If READ TPR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write negative response
2. Read message
3. Write EOA and EOT -- TPR only
4. Disable -- TPR only

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR to read a message after a successful Write operation. In addition, the READ TVR resets the line.

The READ Conversational macro instruction functions as follows. The line is placed in control mode and the control unit is conditioned to receive a message. The EOA received is deleted. A byte of sense information is read into the DECB response field.

The message, which ends with EOB, is read into main storage. If READ TVR is specified and no errors occurred during transmission, and EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is reset.

The channel program generated and executed is:

1. Write EOT sequence
2. Prepare
3. Sense
4. Read message
5. Write EOA and EOT -- TVR only
6. Disable -- TVR only

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

CPU to Terminal

There are eight macro instructions available for communication from the CPU to the 2740 with dial and checking. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TA - WRITE Positive Acknowledgment and Disconnect

TN - WRITE Negative Acknowledgment and Disconnect

The CPU calls the terminal using the number specified in a DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

The terminal list is a calling list. Dialcount is the number of digits in the terminal phone number. Dialchar is the phone number of the terminal.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to establish line connection and send a message to the terminal.

The WRITE Initial functions as follows. The line is disabled in case this was not done previously. The CPU dials the terminal using the digits specified in the DFTRMLST. Fifteen pad characters are sent over the line, followed by EOA, and the message is written to the terminal. The text must end with an EOB. The response to checking is read into the DECB response field. IF WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable

2. Dial call digits

3. Write pad characters

4. Write EOA

5. Write message

6. Read response to checking

7. Write EOT -- TIR only

8. Disable -- TIR only

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR after a Write operation to send a message to the terminal when line connection has already been established. In addition, WRITE TTR resets the line.

The WRITE Continue functions as follows. The message is written to the terminal. The response to checking is then read into the DECB response field. If WRITE TTR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write message

2. Read response to checking

3. Write EOT -- TTR only

4. Disable -- TTR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after each successful Read operation to send a message to the terminal. In addition, WRITE TVR resets the line.

The WRITE Conversational functions as follows. An EOA is sent to the terminal as a positive response to the previous message read. The message is written to the terminal. The response to checking is read into the DECB response field. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOA

2. Write message

3. Read response to checking

4. Write EOT -- TVR only

5. Disable -- TVR only

WRITE Positive Acknowledgment and Disconnect (TA): The problem program issues a WRITE TA after a successful Read operation to send a positive response to the message read and to reset the line.

The WRITE TA functions as follows. An EOA is sent to indicate that the previous message was received without error. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOA and EOT

2. Disable

WRITE Negative Acknowledgment and Disconnect (TN): The problem program issues a WRITE TN after an unsuccessful Read operation to indicate an error in transmission and reset the line, or after a Write operation to reset the line.

The WRITE TN functions as follows. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOT

2. Disable

2740 WITH DIAL, CHECKING, AND OIU

The 2740 with dial, checking, and OIU features requires that each terminal have the dial-up adapter special feature installed. The connection is established either by the terminal calling the CPU or by the CPU calling the terminal.

The checking feature consists of VRC (veritcal redundancy check), which checks each character transmitted for parity, and LRC (longitudinal redundancy check), which checks each message block for correct transmission.

Terminal to CPU

There are ten macro instructions available for communication from the 2740 with dial, checking, and OIU to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TT - READ Continue

TTA - READ Continue with Leading Acknowledgment

TTR - READ Continue with Reset

TP - READ Repeat

TPR - READ Repeat with Reset

TV - READ Conversational

TVR - READ Conversational with Reset

TS - READ Skip

The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies that the terminal list is an answering list.

Following the receipt of a message from the 2740 or 2760, a message may be sent to either the 2740 (text must be preceded by EOA and followed by EOB) or the 2760 (text must be preceded by EOA, the prefix character, and the letter O, and followed by EOB).

Following the receipt of a message from the 2740, a message may be read from the 2760 after the line is put in control mode. Following the receipt of a message from the 2760, a message may be read from the 2740 after the line is put in control mode.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is active but prior or concurrent with an operation issued by the problem program. The operation may be posted in error. Alternative actions the user may take when this condition occurs are:

- Issue a WRITE TCO to move the filmstrip to an error handling frame which will aid the operator in recovering the lost data.
- Issue a WRITE TV to write an error message on the 2740 keyboard.
- Issue a READ TIR (nonswitched line) or READ TV (switched line) to allow the 2760 operator to retry. The operator should be instructed to reprobe if he does not hear the audible tone within a reasonable period of time.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to answer a call from the terminal, and to read a message.

The READ Initial functions as follows. The line is disabled in case this was not previously done. The line is conditioned to receive a call. When a call is received, the control unit is conditioned to receive a message. A byte of sense information is read into the DECB response field.

The message is read into main storage. The EOA sent by the terminal is deleted. The message will end with an EOB, which is sent when the terminal operator presses the EOB key, or automatically by the 2760 OIU when in Auto EOM mode. The operation is posted as complete (with or without error) in the ECB. If READ TIR was specified and no errors occurred during transmission, a positive response is sent, the terminal motors are turned off, and the line is disabled.

The channel program generated and executed is:

1. Disable
2. Enable
3. Prepare
4. Sense
5. Read message
6. Write EOA and EOT -- TIR only
7. Disable -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR after a successful Read operation to send a positive response and to read more data. The line connection must have already been established.

The READ Continue functions as follows. A positive response is sent to indicate that the previous message block was successfully received. The message is then read. The message ends with an EOB. The operation is posted as complete (with or without error) in the ECB. If READ TTR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is disabled.

The channel program generated and executed is:

1. Write Positive response
2. Read message
3. Write EOA and EOT -- TTR only
4. Disable -- TTR only

Note: READ TT and READ TTR are not supported following an operation in which data was received from the 2760.

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR to reread the previous message block if an error occurred during transmission. In addition, READ TPR breaks the line connection.

The READ Repeat functions as follows. A negative response is sent to indicate that the previous message was received in error. The message is then read. If READ TPR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write negative response
2. Read message
3. Write EOA and EOT -- TPR only
4. Disable -- TPR only

READ Continue with Leading Acknowledgment (TTA): The problem program issues a READ TTA to give a positive response to data received from either the 2740 or 2760, to put the line in control mode, and to read again, from either device.

If a message read from the 2760 is to be followed by additional data from either the 2740 or 2760, READ TTA should be used. The EOA causes the audible tone and releases the probe interlock.

The READ Continue with Leading Acknowledgment functions as follows. A positive response is sent to the terminal to indicate that the previous message was received successfully. The terminal is placed in control mode and input can be received from either the 2740 or 2760. The EOA received is deleted. A byte of sense information is read into the DECB response field. The message, which ends with EOB, is read into main storage.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Prepare
3. Sense
4. Read message

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR after a successful Write operation. It can be used to read from the 2740 after writing to the 2760 or to read from the 2760 after writing to the 2740. In addition, the READ TVR resets the line.

The READ Conversational macro instruction functions as follows. The line is placed in control mode and the control unit is conditioned to receive a message. The EOA received is deleted. A byte of sense information is read into the DECB response field.

The message, which ends with EOB, is read into main storage. If READ TVR is specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is reset.

The channel program generated and executed is:

1. Write EOT sequence
2. Prepare
3. Sense
4. Read message
5. Write EOA and EOT -- TVR only
6. Disable -- TVR only

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read skip

CPU to Terminal

There are twelve macro instructions available for communication from the CPU to the

2740 with dial, checking, and OIU. The optypes for these macro instructions are:

- TI - WRITE Initial
- TIR - WRITE Initial with Reset
- TIO - WRITE Initial Optical
- TT - WRITE Continue
- TTR - WRITE Continue with Reset
- TV - WRITE Conversational
- TVR - WRITE Conversational with Reset
- TVO - WRITE Conversational Optical
- TCO - WRITE Invitational Optical
- TA - WRITE Positive Acknowledgment and Disconnect
- TN - WRITE Negative Acknowledgment and Disconnect

The CPU calls the terminal using the number specified in a DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars [,(ffaaaa-hexchars)]

The terminal list is a calling list. Dialcount is the number of digits in the terminal phone number. Dialchar is the phone number of the terminal. The ffaaaa represents FA_1A_2, the three frame change characters sent to the OIU to describe film movement.

If it is necessary to send to the 2740 and to the 2760 as consecutive operations, the write to the 2760 should be issued first.

A message sent to the 2760 may be followed by a message to the 2740. If the line is not put in control mode after writing to the 2760, the message to the 2740 must begin with text and must be followed by an EOB. If the line is put in control mode after writing to the 2760, the message to the 2740 must be preceded by an EOA and be followed by an EOB.

Following a WRITE to the 2740, a message may be read from the 2760 after the line is put in control mode. Following a WRITE to the 2760, a message may be read from the 2740 after the line is put in control mode.

The user may issue a WRITE TCO to write to the 2760 and then read from either the 2740 or 2760. If WRITE TIO (switched or nonswitched line) or WRITE TVO (switched line) is used to write to the 2760, and if input is then expected from either the 2760 or 2740, the user should issue a READ TI (nonswitched line) or READ TV (switched line) immediately after the WRITE TIO or TVO completes.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is active but prior to or concurrent with an operation issued by the problem program. The operation may be posted in error. Alternative actions the user may take when this condition occurs are:

- Issue a WRITE TCO to move the filmstrip to an error handling frame which will aid the operator in recovering the lost data.
- Issue a WRITE TV to write an error message on the 2740 keyboard.
- Issue a READ TIR (nonswitched line) or READ TV (switched line) to allow the 2760 operator to retry. The operator should be instructed to reprobe if he does not hear the audible tone within a reasonable period of time.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR)</u>: The problem program issues a WRITE TI or WRITE TIR to establish line connection and send a message to the terminal.

The WRITE Initial functions as follows. The line is disabled in case this was not done previously. The CPU dials the terminal using the digits specified in the DFTRMLST. Fifteen pad characters are sent over the line, followed by EOA, and the message is written to the terminal. The text must end with an EOB. The response to checking is read into the DECB response field. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Write pad characters

4. Write EOA

5. Write message

6. Read response to checking

7. Write EOT -- TIR only

8. Disable -- TIR only

WRITE Initial Optical (TIO): The problem program issues a WRITE TIO to make initial contact with the terminal, transmits a control message to the 2760 OIU, and reads its response.

The WRITE Initial Optical functions as follows. The line is disabled in case this was not done previously. The CPU calls the terminal using the digits specified in the DFTRMLST. Fifteen pad characters are sent to allow time for the terminal motors to reach operating speed.

The terminal is put in receive mode and the message is identified as being for the 2760 OIU by the prefix character and letter O (not coded by the user). The frame change characters specified in the DFTRMLST are then sent to the OIU. F describes the direction of film movement and end of message mode, and A_1A_2 describe the amount of film movement. An EOB (not coded by the user) follows the frame change characters.

The response to checking is read into the DECB response field. If the response is positive, an EOT is sent to the OIU to enable it to execute the function specified in the control message. If the response is negative, the operation is halted and the error is indicated to the user. Successful completion leaves the line in control mode.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Write pad characters
4. Write EOA, PRE, O
5. Write frame change
6. Write EOB
7. Read response to checking
8. Write EOT sequence

Note: Buffers may not be used with the WRITE TIO macro instruction.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR after a Write operation to send a message to the terminal when line connection has already been established. In addition, WRITE TTR resets the line.

The WRITE Continue functions as follows. The message is written to the terminal. The response to checking is then read into the DECB response field. If WRITE TTR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT -- TTR only
4. Disable -- TTR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR after each successful Read operation, or for reestablishing connection with the SELECTRIC typewriter after using the 2760 OIU, to send a message to the terminal. In addition, WRITE TVR resets the line.

The WRITE Conversational functions as follows. An EOA is sent to the terminal as a positive response to the previous message read. The message is written to the terminal. The response to checking is read into the DECB response field. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel problem generated and executed is:

1. Write EOA
2. Write message
3. Read response to checking
4. Write EOT -- TVR only
5. Disable -- TVR only

WRITE Conversational Optical (TVO): The problem program issues a WRITE TVO to send a control message to the 2760 OIU and to read the response.

The WRITE Conversational Optical functions as follows. The terminal is placed in receive mode and the message is identified as being for the 2760 OIU by the prefix character and letter O (not coded by the user). Three frame change characters are sent to the OIU. F describes the direction of film movement and end of message mode, and A_1A_2 describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro instruction. The length operand is ignored. An EOB (supplied by BTAM) is then sent to the 2760.

The response to checking is read. If the response is positive, an EOT is sent to the 2760 to enable it to execute the function specified in the control message. If the response is negative, EOT is not sent and the error is indicated to the user. Successful completion of this macro instruction leaves the line in control mode. If the macro is concluded in error, the user should issue a WRITE TN to clear the error condition before retrying the WRITE TVO.

The channel program generated and executed is:

1. Write EOA, PRE, O
2. Write frame change
3. Write EOB
4. Read response to checking
5. Write EOT sequence

Note: Buffers may not be used with the WRITE TVO macro instruction.

WRITE Invitational Optical (TCO): The problem program issues a WRITE TCO to send a control message to the OIU and to read a message from it.

The WRITE Invitational Optical macro instruction functions as follows. The terminal is placed in receive mode and the message is identified as being for the 2760 OIU by the prefix character and the letter O (not coded by the user). Three frame change characters are sent to the OIU. F describes the direction of film movement and end of message mode, and A_1A_2 describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro instruction. An EOB is then sent to the 2760.

The response to checking is read. If the response is positive, an EOT is sent to the 2760 to enable it to execute the function specified in the control message. If the response is negative, EOT is not sent and the error is indicated to the user. A message is read from the OIU. (The EOA is front of the message is deleted.) The address of the input area should be coded in the area operand, and the length in the length operand of the WRITE macro instruction. The length may be 7 bytes or variable, depending on the end-of-message mode. If buffers are used, the area operand, rather than the length operand, must be coded as 'S'.

The channel program generated and executed is:

1. Write EOA, PRE, O
2. Write frame change
3. Write EOB
4. Read response to checking
5. Write EOT sequence
6. Prepare
7. Sense
8. Read message

WRITE Positive Acknowledgment and Disconnect (TA): The problem program issues a WRITE TA after a successful Read operation to send a positive response to the message read and to reset the terminal.

The WRITE TA functions as follows. An EOA is sent to indicate that the previous message was received without error, the terminal motors are turned off, and the line connection is broken.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN): The problem program issues a WRITE TN after an unsuccessful Read operation to indicate an error in transmission and reset the line, or after a Write operation to reset the line.

The WRITE TN functions as follows. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOT sequence
2. Disable

2740 WITH DIAL AND TRANSMIT CONTROL

The 2740 with dial and transmit control permits the CPU to control the send and receive status of the terminal. The transmit control feature can only be installed when the dial-up feature is also installed. Provided with the tranmsit control feature is the transmit control switch located on the left side of the 2740 cabinet. This switch (MTC,OFF) permits the terminal operator to set up the terminal for proper line control when communicating with the CPU. The switch must be set to MTC when using the transmit control code (/ and space).

Terminal to CPU

There are five macro instructions available for communication from the 2740 with dial and transmit control to the CPU. The optypes for these macro instructions are:

- TI - READ Initial
- TIR - READ Initial with Reset
- TV - READ Conversational
- TVR - READ Conversational with Reset
- TS - READ Skip

The CPU can control the sending status of a remote terminal with the two-character control code. When the / space is sent, the terminal transmit light comes on if the operator has the transmit control switch on MTC. The terminal automatically sends a EOA in response to the / space. If the terminal does not follow this EOA with data within 15 seconds, a CPU time-out occurs and the terminal standby light comes on. For communication from the terminal to the CPU, the CPU may call the terminal or may answer a call from the terminal. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, {0 dialcount-integer, dialchar-decchars}

The terminal list may be an answering list or a calling list. The 0 specifies that it is an answering list. The dialcount is the number of digits in the terminal phone number in a calling list. Dialchar is the actual phone number of the terminal.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to establish line connection and read a message. In addition, the READ TIR breaks the line connection. The READ Initial functions differently according to the type of terminal list specified.

CALLING LIST: If the CPU is to call the terminal, the READ Initial functions as follows. The line is disabled in case this was not done previously. The CPU calls the terminal using the dial digits specified in the DFTRMLST macro instruction. Fifteen pad characters are then sent to the terminal. The transmit control code (/ space) is sent to allow the terminal to send a message. The response from the terminal is read. If no data is received within 15 seconds, a time-out occurs. The remainder of the message is read into main storage. If READ TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line is disabled.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Write pad characters
4. Write transmit control code
5. Read response
6. Read message
7. Write EOT -- TIR only
8. Disable -- TIR only

ANSWERING LIST: When the CPU is to answer a call from the terminal, the READ Initial functions as follows. The line is disabled in case this was not done previously. The line is enabled to receive a call. When a call is received, fifteen pad characters are sent over the line. The transmit control code is sent to tell the terminal that it may begin sending data. The response from the terminal is read. If no data is received within 15 seconds, a time-out occurs. The remainder of the message is then read. If READ TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable

2. Enable

3. Write pad characters

4. Write transmit control code

5. Read response

6. Read message

7. Write EOT -- TIR only

8. Disable -- TIR only

<u>READ Conversational (TV) and READ Conversational with Reset (TVR)</u>: The problem program issues a READ TV or READ TVR after a successful Read operation to read a message without reestablishing initial contact. The READ Conversational functions as follows. The transmit control code is sent to the terminal to indicate that it can now send data to the CPU. The response is read from the terminal. If no data is received within 15 seconds, a time-out occurs. The remainder of the message is then read from the terminal. If READ TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line is disabled.

The channel program generated and executed is:

1. Write transmit control code

2. Read response

3. Read message

4. Write EOT -- TVR only

5. Disable

<u>READ Skip (TS)</u>: The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

<u>CPU to Terminal</u>

There are five macro instructions available for communication from the CPU to a 2740 with dial and transmit control. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TV - WRITE Conversational

TVR - WRITE Conversational with Reset

TN - WRITE Disconnect

The CPU dials the terminal, using the dial digits specified in a DFTRMLST macro instruction, and then sends a message. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

Dialcount is the same number of dial digits as the phone number of the terminal. Dialchar is the actual phone number of the terminal.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR)</u>: The problem program issues a WRITE TI or WRITE TIR to call a terminal and send a message to that terminal. In addition, the WRITE TIR disconnects the line.

The WRITE Initial functions as follows. The line is disabled in case this was not done previously. The CPU dials the terminal using the digits specified in the DFTRMLST macro instruction. Fifteen pad characters are sent over the line, and the message is written to the terminal. The text must begin with an EOA and end with an EOB. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable

2. Dial call digits

3. Write pad characters

4. Write message

5. Write EOT -- TIR only

6. Disable -- TIR only

<u>WRITE Conversational (TV) and WRITE Conversational with Reset (TVR)</u>: The problem program issues a WRITE TV or WRITE TVR after a successful read operation to send data in answer to the data read. In addition, the WRITE TVR turns off the terminal motors and breaks the line connection.

The WRITE Conversational macro instruction writes a message to the terminal. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write message
2. Write EOT -- TVR only
3. Disable -- TVR only

WRITE Disconnect (TN): The problem program issues a WRITE TN to reset the line. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOT
2. Disable

2740 WITH DIAL, TRANSMIT CONTROL, AND CHECKING

Line connection for a 2740 with dial, transmit control, and checking is established by dialing and is maintained until the CPU disables the line or until the dial-disconnect key at the terminal is depressed.

The transmit control feature allows the CPU to control the sending and receiving status of the terminal. A transmit control code (/ space) is sent to indicate that the terminal may begin sending data. The transmit control feature can be installed only when the dial feature is installed. The transmit control switch at the terminal (MTC/OFF) must be set to MTC when using the transmit control code.

The checking feature consists of VRC (vertical redundancy check), which checks each character transmitted for parity, and LRC (longitudinal redundancy check), which checks each message block for correct transmission.

Terminal to CPU

There are nine macro instructions available for communication from the 2740 with dial, transmit control, and checking to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TT - READ Continue

TTR - READ Continue with Reset

TP - READ Repeat

TPR - READ Repeat with Reset

TV - READ Conversational

TVR - READ Conversational with Reset

TS - READ Skip

The CPU controls the sending status of the terminal with the transmit control code (/ space). Either the CPU or the terminal may dial to establish line connection. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, $\begin{Bmatrix}\text{0}\\\text{dialcount-integer,}\\\text{dialchar-decchars}\end{Bmatrix}$

The 0 specifies that the terminal list is an answering list. If dialcount and dialchar are specified, the terminal list is a calling list. Dialcount is the number of digits in the terminal phone number. Dialchar is the actual phone number of the terminal.

READ Initial (TI) and READ Initial with Reset (TIR): The problem program issues a READ TI or READ TIR to dial the terminal, or answer a call from the terminal, and to read a message. The READ Initial functions differently according to the type of terminal list specified.

ANSWERING LIST: When the terminal list specified is an answering list, the line is disabled in case this was not done previously. The line is conditioned to received a call. When a call is received, fifteen pad characters are sent over the line. The transmit control code is sent to allow the terminal to begin sending.

The response is read. If EOT is received, the operation is posted as complete in the ECB. The remainder of the message is read and the operation is posted as complete (with or without error) in the ECB. If READ TIR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable

2. Enable

3. Write pad characters

4. Write transmit control code

5. Read response

6. Read message

7. Write EOA and EOT -- TIR only

8. Disable -- TIR only

CALLING LIST: When the terminal list specified is a calling list, the READ Initial functions as follows. The line is disabled in case this was not done previously. The CPU dials the terminal using the number specified in the DFTRMLST macro instruction. Fifteen pad characters are sent over the line. The transmit control code is sent to the terminal to allow it to begin sending data.

The response is read. If EOT is received, the operation is posted as complete. The remainder of the message is read and the operation is posted as complete (with or without error) in the ECB. If READ TIR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is reset.

The channel program generated and executed is:

1. Disable

2. Dial call digits

3. Write pad characters

4. Write transmit control code

5. Read response

6. Read message

7. Write EOA and EOT -- TIR only

8. Disable -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR after a successful Read operation to read more data from the same component. The line connection must have already been established.

The READ Continue functions as follows. A positive response is sent to indicate that the previous message block was successfully received. The message is then read. The message received may be a message block, or it may be an EOT, which terminates transmission. The operation is posted as complete (with or without error) in the ECB. If READ TTR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is reset.

The channel program generated and executed is:

1. Write positive response

2. Read message

3. Write EOA and EOT -- TTR only

4. Disable -- TTR only

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR to reread the previous message block if an error occurred during transmission. In addition, READ TPR breaks the line connection.

The READ Repeat functions as follows. A negative response is sent to indicate that the previous message was received in error. The message is then read. If READ TPR was specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write negative response

2. Read message

3. Write EOA and EOT -- TPR only

4. Disable -- TPR only

READ Conversational (TV) and READ Conversational with Reset (TVR): The problem program issues a READ TV or READ TVR after a Write operation to read data without reestablishing initial contact with the terminal. In addition, the READ TVR resets the line.

The READ Conversational macro instruction functions as follows. The line is placed in control mode and the transmit control code (/ space) is sent to allow the terminal to send a message. The response from the terminal is read. If no data is received within 15 seconds, a time-out occurs. The remainder of the message is read into main storage. If READ TVR is specified and no errors occurred during transmission, an EOA is sent to indicate a good transmission. The terminal motors are turned off and the line is reset.

The channel program generated and executed is:

1. Write EOT sequence
2. Write transmit control code
3. Read response
4. Read message
5. Write EOA and EOT -- TVR only
6. Disable -- TVR only

READ Skip (TS): The problem program issues a READ TS to recover from a lost data error condition. The remainder of the message is read from the terminal to clear the line, but is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

CPU to Terminal

There are eight macro instructions available for communication from the CPU to the 2740 with dial, transmit control, and checking. The optypes for these macro instructions are:

TI	-	WRITE Initial
TIR	-	WRITE Initial with Reset
TT	-	WRITE Continue
TTR	-	WRITE Continue with Reset
TV	-	WRITE Conversational
TVR	-	WRITE Conversational with Reset
TA	-	WRITE Positive Acknowledgment and Disconnect
TN	-	WRITE Negative Acknowledgment and Disconnect

The CPU can either condition the line to receive a call or the CPU can call the terminal. The CPU controls the receiving status of the terminal. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, {0 | dialcount-integer, dialchar-decchars}

The 0 specifies that this is an answering list. If dialcount and dialchar are specified, this is a calling list. The dialcount operand is the number of digits in the terminal phone number. The dialchar operand is the actual phone number of the terminal.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to establish line connection and send a message to the terminal. The function of WRITE Initial varies according to the type of DFTRMLST specified.

CALLING LIST: When a calling list is specified, the WRITE Initial functions as follows. The line is disabled in case this was not done previously. The CPU dials the terminal using the digits specified in the DFTRMLST. Fifteen pad characters are sent over the line, and the message is written to the terminal. The text must begin with an EOA and end with an EOB. The response to checking is read into the DECB response field. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable
2. Dial call digits
3. Write pad characters
4. Write message
5. Read response to checking
6. Write EOT -- TIR only
7. Disable -- TIR only

ANSWERING LIST: When an answering list is specified, the WRITE Initial functions as follows. The line is disabled in case this was not done previously. The line is enabled to receive a call. Fifteen pad characters are sent over the line and the message is written to the terminal. The text must begin with an EOA and end with an EOB. The response to checking is read into the DECB response field. If WRITE TIR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Disable
2. Enable
3. Write pad characters
4. Write message
5. Read response to checking
6. Write EOT -- TIR only
7. Disable -- TIR only

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR after a Read or Write operation to send a message to the terminal when line connection has already been established. In addition, WRITE TTR resets the line. The WRITE Continue functions as follows. The message is written to the terminal. The response to checking is then read into the DECB response field. If WRITE TTR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT -- TTR only
4. Disable -- TTR only

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The problem program issues a WRITE TV or WRITE TVR to send a message to the terminal after a successful Read operation. In addition, WRITE TVR resets the line. The WRITE Conversational functions as follows. An EOA is sent to the terminal as a positive response to the previous message read. The message is written to the terminal. The response to checking is read into the DECB response field. If WRITE TVR was specified and no errors occurred during transmission, the terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOA
2. Write message
3. Read response to checking
4. Write EOT -- TVR only
5. Disable -- TVR only

WRITE Positive Acknowledgment and Disconnect (TA): The problem program issues a WRITE TA to send a positive response to the message read and to reset the line. The WRITE TA functions as follows. An EOA is sent to indicate that the previous message was received without error. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOA and EOT sequence
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN): The problem program issues a WRITE TN after an unsuccessful operation to indicate an error in transmission and reset the line, or after a Write operation to reset the line. The WRITE TN functions as follows. The terminal motors are turned off and the line connection is broken.

The channel program generated and executed is:

1. Write EOT sequence
2. Disable

2740 WITH STATION CONTROL

The station control feature of the 2740 provides the terminal with the ability to be addressed and polled. All the necessary polling and addressing is done by the CPU with a one-character identification code. If there is no transmission for a fifteen-second time period, a time-out occurs and the terminal is placed in standby mode.

Terminal to CPU

There are two macro instructions available for communication from the 2740 with station control to the CPU. The optypes for these macro instructions are:

TI - READ

TS - READ Skip

The CPU polls the terminals to see if they have data to send using the identification code specified in a DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST} ({xx-hexchars}...)
symbol	DFTRMLST	{SSALST, SSAWLST} {xxyy-hexchars}...

The list may be an open list, a wraparound list, or an Auto Poll list (open or wraparound). Each terminal on the line is polled with a one-character code that identifies the terminal. The xx is the hexadecimal respresentation of the polling character as it appears in the six-bit BCD transmission code (see Appendix B). The yy is the hexadecimal representation of the space character.

READ Initial (TI): The problem program issues a READ TI to begin polling on a line and to read a message. The function of the READ Initial varies according to the type of terminal list used.

OPENLST or WRAPLST: If the terminal list specified is an open list or wraparound list, the READ Initial functions as follows. The terminal is placed in control mode. The polling character specified in the DFTRMLST is sent to poll the terminal. A space character is then sent. A two-byte response to polling is read into the input area. If the response is negative, the next terminal in the list is polled. If there is no response within fifteen seconds, a time-out occurs and the next terminal is polled. If the end of an open list is reached before a positive response (EOA followed by text) is received, the operation is posted as complete. When a positive response is received, the remainder of the message is read.

The channel program generated and executed is:

1. Write EOT sequence
2. Write polling character
3. Write space character
4. Read response to polling
5. Read message

SSALST: If the terminal list specified is an open start-stop Auto Poll list, the READ Initial functions as follows. All the terminals on the line are put in control mode. The starting entry (or next entry) in the terminal list is polled using the polling character from the DFTRMLST macro. If a positive response (an index byte and one text byte) is received, the remainder of the message is read until an EOT is received. The operation is posted as complete in the ECB. If the end of the list is reached before a positive response is received, the operation is halted.

The channel program generated and executed is:

1. Write EOT sequence
2. Poll starting entry
3. NOP
4. Read response to polling
5. Read message

SSAWLST: If the terminal list specified is a wraparound start-stop Auto Poll list, the READ Initial functions as follows. All the terminals on the line are put in control mode. The starting entry (or next entry) polling characters from the terminal list are sent to poll an entry in the list. If the end of the polling list is reached before a positive response is received, polling is restarted at the beginning of the list.

When a positive response to polling is received, an index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOT is received. The operation is posted as complete in the ECB.

The channel program generated and executed is:

1. Write EOT sequence
2. Poll starting entry
3. TIC to (5)
4. TIC to (7)
5. Poll first entry
6. TIC to (5)
7. Read response to polling
8. Read message

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the data is read from the terminal to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

CPU to Terminal

There are two macro instructions available for communication from the CPU to the 2740 with station control. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

Each 2740 with station control is addressed with a single-character code as specified in a DFTRMLST macro instruction. When Auto Poll is not used, the format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xx-hexchars,}...)

The xx is the hexadecimal representation of the addressing character as it appears in the six-bit BCD transmission code (see Appendix B).

When Auto Poll is used, the format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars}...)

The xx is the hexadecimal representation of the addressing character. The yy is the hexadecimal representation of the space character.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to address a terminal and send a message. In addition, WRITE TIR resets the line.

The WRITE Initial functions as follows. The terminal is placed in control mode and the address select character is sent to indicate that addressing is being done. The addressing character specified in the DFTRMLST macro instruction is sent to the terminal. Then the space character is sent.

The response to addressing is read into the first byte of the DECB response field. If the response is negative, the operation is posted as complete with error in the ECB. If the response is positive, the message is sent to the terminal. The message format must be EOA, text, EOB. If WRITE TIR was specified and no errors occurred during transmission, the terminal is placed in control mode. Otherwise, the operation is posted as complete (with or without error) in the ECB.

The channel program generated and executed is:

1. Write EOT sequence and address select
2. Write addressing character
3. Write space character
4. Read response to addressing
5. Write message
6. Write EOT sequence -- TIR only

Note: When the Auto Poll feature is used, the channel program generated for a WRITE Initial does not write the space character that follows the addressing character. This space character is defined in the DFTRMLST as shown above.

2740 WITH STATION CONTROL AND CHECKING

The station control feature of the 2740 provides the CPU with the ability to poll or to address a terminal. The terminals are polled and addressed with a single-character identification code. The checking feature consists of both LRC and VRC.

Terminal to CPU

There are seven macro instructions available for communication from the 2740 with station control and checking to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TIR - READ Initial with Reset

TT - READ Continue

TTR - READ Continue with Reset

TP - READ Repeat

TPR - READ Repeat with Reset

TS - READ Skip

The CPU polls each terminal with a single character that is unique for that terminal on that line. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xx-hexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars.}...

The xx is the hexadecimal representation of the polling character as it appears in the six-bit BCD transmission code (see Appendix B). The yy is the hexadecimal representation of the space character. The list may be an open list, a wraparound list, or an Auto Poll list (open or wraparound).

<u>READ Initial (TI) and READ Initial with Reset (TIR)</u>: The problem program issues a READ TI or READ TIR to begin polling on a line and to read a message. In addition, READ TIR resets the line. The function of the READ Initial varies according to the type of terminal list used.

OPENLST or WRAPLST: If the terminal list specified is an open list or a wraparound list, the READ Initial functions as follows. The terminal is placed in control mode. The polling character specified in the DFTRMLST is sent to poll the terminal. A space character is then sent. A two-byte response to polling is read into the input area. If the response is negative, the next terminal in the list is polled. If there is no response within fifteen seconds, a time-out occurs and the next terminal is polled. If the end of an open list is reached before a positive response (EOA followed by text) is received, the operation is posted as complete. When a positive response is received, the remainder of the message is read. If READ TIR was specified and no errors occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOT sequence
2. Write polling character
3. Write space character
4. Read response to polling
5. Read message
6. Write EOA and EOT sequence -- TIR only

SSALST: If the terminal list specified is an open start-stop Auto Poll list, the READ Initial functions as follows. All the terminals on the line are put in control mode. The starting entry (or next entry) in the terminal list is polled using the polling character from the DFTRMLST macro. If a positive response (an index byte and one text byte) is received, the remainder of the message is read until an EOB is received. The operation is posted as complete in the ECB. If the end of the list is reached before a positive response is received, the operation is halted. If READ TIR was specified and no errors occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOT sequence
2. Poll starting entry
3. NOP
4. Read response to polling
5. Read message
6. Write EOA and EOT sequence -- TIR only

SSAWLST: If the terminal list specified is a wraparound start-stop Auto Poll list, the READ Initial functions as follows. All the terminals on the line are put in control mode. The starting entry (or next entry) polling characters from the terminal list are sent to poll an entry in the list. If the end of the polling list is reached before a positive response is received, polling is restarted at the beginning of the list.

When a positive response to polling is received, an index byte (corresponding to the responding terminal) and one text byte are read into the input area. The message is read into the input area until an EOB is received. The operation is posted as complete in the ECB. If READ TIR was specified and no errors occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write EOT sequence
2. Poll starting entry
3. TIC to (5)

4. TIC to (7)

5. Poll first entry

6. TIC to (5)

7. Read response to polling

8. Read message

9. Write EOA and EOT sequence -- TIR only

READ Continue (TT) and READ Continue with Reset (TTR): The problem program issues a READ TT or READ TTR after a successful READ TI or READ TT to read a subsequent message from the same component without repolling. In addition, the READ TTR resets the line.

The READ Continue functions as follows. A positive response is sent to indicate that the previous message was received without error. Then the message is read. If READ TTR was specified and no errors occurred during transmission, a positive response is sent to the terminal and the terminal motors are turned off.

The channel program generated and executed is:

1. Write positive response

2. Read message

3. Write EOA and EOT sequence -- TTR only

Note: When using the 2740 M2, the READ Continue should only be used to read the EOT sent by the terminal. (The 2740 M2 does not send multiblocked records). Ideally, the READ Initial with Reset should be used for this purpose.

READ Repeat (TP) and READ Repeat with Reset (TPR): The problem program issues a READ TP or READ TPR after an unsuccessful read operation to reread a message received in error. In addition, READ TPR resets the line.

The READ Repeat functions as follows. A negative response is sent to the terminal to indicate an error in the previous message. The message is then read. If READ TPR was specified and no error occurred during transmission, a positive response is sent and the terminal motors are turned off.

The channel program generated and executed is:

1. Write negative response

2. Read message

3. Write EOA and EOT sequence -- TPR only

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read from a terminal to clear the line, but it is not received in main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

CPU to Terminal

There are six macro instructions available for communication from the CPU to the 2740 with station control and checking. The optypes for these macro instructions are:

TI - WRITE Initial

TIR - WRITE Initial with Reset

TT - WRITE Continue

TTR - WRITE Continue with Reset

TA - WRITE Positive Acknowledgment

TN - WRITE Negative Acknowledgment

Each terminal is addressed with a single-character code as specified in a DFTRMLST macro instruction. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xx-hexchars)

The xx is the hexadecimal representation of the addressing character. It appears in the six-bit BCD transmission code (See Appendix B).

When Auto Poll is used, the format of the DFTRMLST used is:

Name	operation	Operand
symbol	DFTRMLST	OPENLST, (xxyy-hexchars)

The xx is the hexadecimal representation of the addressing character. The yy is the hexadecimal representation of the space character.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The problem program issues a WRITE TI or WRITE TIR to address a terminal and send a message. The transmission is also checked for errors. In addition, WRITE TIR resets the line.

The WRITE Initial functions as follows. The terminal is placed in control mode and the address select character is sent to indicate that addressing is being done. The addressing character from the DFTRMLST macro and a space character are written to the terminal.

The response to addressing is read into the first byte of the DECB response field. If the response is negative or if no response is received in fifteen seconds, the operation is posted as complete with error.

If the response is positive, an EOA is sent to trigger checking on the Write operation. The message is then written to the terminal and must be followed by an EOB. The response to checking is read into the DECB response field. If the response is positive and WRITE TIR was specified, the terminal motors are turned off. Otherwise the operation is posted as complete (with or without error) in the ECB.

The channel program generated and executed is:

1. Write EOT sequence and address select
2. Write addressing character and space character
3. Read response to addressing
4. Write EOA
5. Write message
6. Read response to checking
7. Write EOT sequence -- TIR only

Programming Note: Multiblock messages should not be sent to the 2740 M2 with the record checking and buffer receive features. The message transmitted to the 2740 M2 is not printed until an EOT is received, so the WRITE TIR operation should be used with the 2740 M2, unless an EOT is sent to the terminal after a WRITE TI operation.

When a nonproductive operation completion code (X'54') is posted for a WRITE TI or WRITE TIR operation to the 2740 M2, the printer on the terminal is busy, the printer is out of paper, or the terminal is in the local mode. The operation should be retried after waiting a suitable period of time for the printer to complete its operation.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The problem program issues a WRITE TT or WRITE TTR after a successful Write or Read operation to send a message block. In addition, WRITE TTR resets the line.

The WRITE Continue functions as follows. The message is sent to the terminal. The response to checking is read into the DECB response field. If WRITE TTR was specified and the response is positive, the terminal motors are turned off.

The channel program generated and executed is:

1. Write message
2. Read response to checking
3. Write EOT sequence -- TTR only

WRITE Positive Acknowledgment (TA): The problem program issues a WRITE TA to send a positive response to the terminal and turn the terminal motors off.

The channel program generated and executed is:

1. Write EOA and EOT sequence

WRITE Negative Acknowledgment (TN): The problem program issues a WRITE TN after a Write operation or an unsuccessful Read operation. After the read, the terminal interprets this as a negative response and turns the terminal motors off. After the write, this macro instruction resets the line.

The channel program generated and executed is:

1. Write EOT sequence

IBM 2260-2848 DISPLAY COMPLEX AND IBM 2265-2845 DISPLAY COMPLEX*

This section describes the remotely attached 2260-2848 and 2265-2845 display complexes. See the general section "Local Device-Dependent Considerations" for information on the locally attached 2260-2848. The information in this section applies equally to the IBM 2260-2848 display complex and the IBM 2265-2845 display complex, except that the 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).

Terminal to CPU

There are five macro instructions available for communication from the remote 2260 to the CPU. The optypes for these macro instructions are:

TI - READ Initial

TT - READ Continue

TP - READ Repeat

TB - READ Buffer

TS - READ Skip

Message transmission from a remote 2260 is initiated by the CPU through a READ macro instruction that polls with a two-character code. The character selects the 2848 Display Control; the second character selects the 2260 Display station. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST, WRAPLST}, ({xxyy-hexchars,}...)

The xxyy is the hexadecimal representation of the polling characters (see Appendix B). The list may be an open list or a wrap around list.

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

STX	device address	[text]	[CAN]	ETX	LRC

STX
: Start-of-Text character.

device address
: character that identifies the sending unit (display station or printer).

text
: the printer status message has a zero-length text.

CAN
: a wrong-parity character sent to force an error condition only if the display control detects an internal operation error when transmitting the message. The READ may be followed by a WRITE Erase macro instruction to clear the error.

ETX
: End-of-Text character.

LRC
: Longitudinal Redundancy Check character. The LRC accumulation is started by the STX character and terminated by the ETX character.

  Note: The LRC does not enter main storage.

READ Initial (TI) - Display or Printer: The problem program issues a READ Initial to start or restart polling on a line or to request printer status.

The READ TI functions as follows. The IBM 2848 Display Control address selection is reset to control mode, nonselect status, and all 2848s on the line are prepared to receive a polling sequence. The polling characters are sent to the terminal, specifying a specific poll of a display station, a general poll of a display control, or a request of a printer status. The READ MI control unit code specifies that the function of the 2848 Display Control is to transmit data from the polled 2260 Display Station. This is one of five control unit codes that are unique to the remote 2848. These control unit codes specify the function that is to be performed by the 2848 Display Control in connection with the 2260 Display Station.

If the response to polling is negative and the end of an open list is reached, the operation is posted as complete-with-error in the ECB. If the response is negative and a wraparound list is specified, polling is restarted. On a positive response, the message is read into main storage. The operation is posted as complete (with or without error) in the ECB.

The channel program generated and executed is:

1. Write EOT

2. Write polling characters

3. Write READ MI code

4. Read response to polling

5. Read message

Specific Poll of Display Station: On positive response to polling (STX), the message is read. On negative response (EOT), an interruption occurs. BTAM detects the polling restart TP code, initializes the channel program to poll the next entry within the list, and returns control to the supervisor.

Request of a Printer Status: If the printer is ready and the buffer is empty, a reservation is set on the printer buffer that prevents transmission of messages from the display stations to the printer buffer. If a message is received indicating these conditions, the message is read. The problem program may then issue a WRITE Continue to print the message. The next EOT resets the reservation condition.

A negative response is either NAK, which indicates that the printer is not ready, or EOT, which indicates that the printer is ready but the buffer is not empty. If a NAK response is received, the problem program may send a message to an operator. Both negative responses set the printer request condition by which the Display Control, upon receipt of a General Poll, can sense if the printer is ready and the buffer is empty.

General Poll of a Display Control: In this entry, the polling list must specify a general poll; the second byte is a hexadecimal FF. If the printer has a status pending as a result of a previous request (printer status or WRITE Initial), this message is transmitted and the message is read.

If the printer is not ready or no request has been made, the display stations are scanned for a message. If a message is pending, it is sent. If there is no message waiting for transmission, a negative response, EOT, is received. The channel program is interrupted; BTAM detects the polling Read response TP code, updates the channel program to poll the next entry, and returns control to the supervisor.

READ Continue (TT): The problem program issues a READ TT after a successful READ TI, TT, or TP to read another message into main storage. The macro instruction should be issued after a general poll.

The READ Continue macro functions as follows. A positive response is sent to the station to indicate that the previous message was received without error. The message is then read into the input area.

The channel program generated and executed is:

1. Write ACK

2. Read message

READ Repeat (TP) - Display Only: The problem program issues a READ TP to request retransmission of a message received in error.

The READ Repeat macro functions as follows. A negative response is sent to the display station. The message that was received in error is then reread.

The channel program generated and executed is:

1. Write NAK

2. Read message

READ Buffer (TB) - Display only: The READ Buffer macro instruction is used primarily for diagnostic applications. No polling takes place. The information received is the entire buffer of the specified display station. The 2848s on the line are reset and are prepared to receive a polling sequence. The polling characters are sent to select the 2260 Display Station whose buffer is to be read. The display station is not being polled to see if it has any message to transmit, but is really being selected so that its buffer in the 2848 can be read. The READ BUFFER control unit code specifies that the function of the 2848 Display Control is to transmit the buffer of the 2260 Display Station indicated. The message is read into the input area and the operation is posted as complete (with or without error) in the ECB.

Note: The user should issue a WRITE Erase macro after the READ Buffer macros.

The channel program generated and executed is:

1. Write EOT sequence

2. Write polling characters -- from DFTRMLST

3. Write READ BUFFER command code

4. Read message

READ Skip (TS): The problem program issues a READ TS macro instruction to recover from a lost data error condition. The remainder of the message is read to clear the line, but it is not received into main storage. The number of characters read is the count specified by the user.

The channel program generated and executed is:

1. Read Skip

CPU to Terminal

There are ten macro instructions available for communication from the CPU to the remote 2260. The optypes for these macro instructions are:

TI	-	WRITE Initial
TIR	-	WRITE Initial with Reset
TL	-	WRITE at Line Address
TLR	-	WRITE at Line Address with Reset
TS	-	WRITE Erase
TSR	-	WRITE Erase with Reset
TT	-	WRITE Continue
TTR	-	WRITE Continue with Reset
TA	-	WRITE Positive Acknowledgment
TN	-	WRITE Negative Acknowledgment

Two characters, as specified in the DFTRMLST macro instruction, are used to address the 2260 Remote. The format of the DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

The xx is the hexadecimal representation of the 2848 address character, and yy is the device address character.

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

STX	text	ETX	LRC

Note: The STX and ETX characters are supplied by the user in his output area.

WRITE Initial (TI) - Display or Printer: The problem program issues a WRITE Initial macro instruction to send the first block of a message. The WRITE TI functions as follows. The 2848 Display Control address selection is reset to control mode, nonselected status, and all the 2848s on the line are prepared to receive an addressing sequence.

The addressing characters that indicate a 2848 and a display station or printer are sent. The WRITE command code is sent to specify that the function of the 2848 is to receive data.

If a printer is addressed, the addressing sequence response is read. NAK, which indicates that the printer is not ready, or EOT, which indicates that the printer is ready but the buffer is not empty, sets a printer request. If the response is positive (ACK), indicating that the printer is ready and available, the message is sent. If a transmission error occurs, the operation is stopped, and the printer buffer is cleared. The user may retry transmission by issuing a WRITE Continue.

If a display station is addressed, the addressing sequence response is read (normally ACK), and the message is sent. If a transmission error occurs, the user may retry by issuing a WRITE Continue, but the erroneous message will not be cleared. The user may issue a WRITE Erase to resend the same message, or a READ Buffer and WRITE Erase if several messages were displayed.

The channel program generated and executed is:

1. Write EOT sequence
2. Write addressing sequence
3. Write WRITE command code
4. Read response to addressing
5. Write message
6. Read response to checking

WRITE at Line Address (TL) - Display only: The problem program issues a WRITE TL macro instruction to position the cursor to start on a specified line. The line address should be the first text character after the STX character. Characters are displayed starting from that point.

The WRITE at Line Address functions as follows. The 2848 address selection is reset to control mode, nonselected status, and all 2848s are prepared to receive an addressing sequence. The addressing characters are sent to indicate a 2848 and a display station. The WRITE LINE command code is then transmitted. Responses to addressing are handled as discussed for the WRITE Initial macro instruction. If a transmission error occurs, the user may try sending the same message again with a WRITE Continue. Data will be displayed starting at the same line.

The channel program generated and executed is:

1. Write EOT
2. Write addressing sequence
3. Write WRITE LINE code
4. Read response to addressing
5. Write message
6. Read response to checking

Note: The following values are the possible line address values.

2848/2260 Character	Hexadecimal Code Representation	Display Line
0	X'50'	1
1	X'51'	2
2	X'52'	3
3	X'53'	4
4	X'54'	5
5	X'55'	6
6	X'56'	7
7	X'57'	8
8	X'58'	9
9	X'59'	10
:	X'5A'	11
;	X'5B'	12

WRITE Erase (TS) - Display only: The problem program issues a WRITE TS macro instruction to erase the screen and to display a message starting at the upper left hand corner. The WRITE Erase functions as follows. The 2848 address selection is reset to control mode, non-selected status, and all 2848s are prepared to receive an addressing sequence. The addressing characters are sent to indicate a 2848 and a display station. The ERASE command code is then transmitted. Responses to addressing are handled as discussed for the WRITE Initial macro instruction. If a transmission error occurs, the user may issue another WRITE Erase to send the same message.

The channel program generated and executed is:

1. Write EOT
2. Write addressing sequence
3. Write ERASE code
4. Read response to addressing
5. Write message
6. Read response to checking

WRITE Continue (TT) - Display or Printer: The problem program issues a WRITE TT after any Read or Write operation where the reset option is not specified to write a message.

If the previous operation was a WRITE Initial addressing a printer, the WRITE Continue retries transmission of the message. If the previous operation was a WRITE at Line Address or a WRITE at Line Address followed by one or several WRITE Continues, the cursor is positioned at the beginning of the line specified in the message.

If a transmission error occurs, the user may retry by issuing WRITE Continue. In all other cases, the characters are displayed on the station starting at the cursor position. When text has been received, an answer may be provided by a WRITE Continue without reselecting the display.

The channel program generated and executed is:

1. Write message
2. Read response to checking

WRITE with Reset: All of the previously described WRITE macro instructions can be issued with reset. The additional command sends an EOT to set the line in control mode, non-select status. If the data is received in error, the reset function is not performed and the line is not reset.

- WRITE Initial with Reset (TIR) establishes line connection, sends the first message block, and resets the terminal.
- WRITE at Line Address with Reset (TLR) positions the cursor, displays a message, and resets the terminal.
- WRITE Erase with Reset (TSR) erases the CRT, displays the message, and resets the terminal.
- WRITE Continue with Reset (TTR) operates the same as WRITE Continue and resets the terminal.

WRITE Positive Acknowledgment (TA) - Display or Printer: The problem program issues a WRITE TA to indicate to the sending station that the message was received without error and to end the operation.

The channel program generated and executed is:

1. Write STX

2. Write EOT sequence

__WRITE Negative Acknowledgment (TN) - Display or Printer__: The problem program issues a WRITE TN to indicate to the sending station that the message was received in error. The operation is ended.

The channel program generated and executed is:

1. Write EOT sequence

WORLD TRADE TELEGRAPH TERMINALS

GENERAL INFORMATION

The name World Trade (WT) telegraph terminals refers to various European teletypewriters using a start-stop 5-level code with two shifts (letters shift and figures shift) to transfer data over leased point-to-point telegraph lines at 50, 75, or 100 baud (bits per second). The code used is either the International Telegraph Alphabet No. 2 or the Figure Protected Code ZSC3 (both shown in Appendix B). No line control is provided; therefore, when a WT terminal tries to send a message (input message) at the same time as the computer tries to send a message (output message), both transmissions immediately stop. This is called a contention situation.

WT terminals are always ready to receive or send messages. Normally, the motor of the terminal is off and the first "letters shift" character (LTRS) sent or received by the terminal starts the motor. The motor needs 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; in this case, the terminal is said to be operating in Motor-Off mode. Optionally, the terminal can be equipped with a heavy-duty motor that 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 DTFBT macrio instruction (refer to the DTFBT Macro Instruction section) enables the user to specify the number of Mark characters corresponding to the 1.5-second period mentioned above. When a WRITE macro instruction is issued, BTAM determines the actual status of the terminal (motor off or motor on) and generates a "letters shift" character (which can be followed by a user-specified number of Mark characters), which 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 string 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 combination 4 in figures shift (FIGS D).

The World Trade Telegraph Adapter recognizes two end conditions: FIGS x and FIGS y LTRS, which are set when the IBM 2701, 2702, or 2703 is installed, and where "x" and "y" are assigned by the WTC 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 FIGS D character is referred to as the WRU signal. Otherwise, any other code combination can be selected.

The user must select different code combinations for x and y. The FIGS y character immediately followed by a LTRS character causes a read operation to end. Therefore, this FIGS y character can be sent by a terminal as data, provided it is not followed by a LTRS character.

The above terminations of a read operation can be used as an end-of-message (EOM) signal and, except for FIGS D, as an end-of-transmission (EOT) signal.

The World Trade Telegraph Adapter deletes all incoming LTRS or FIGS characters and updates a shift bit (S), which is added to each character transferred to main storage. The adapter examines the shift bit of each outgoing character, and automatically generates a LTRS or FIGS character, whenever necessary.

0	1	2	3	4	5	6	7	System/370 byte configuration
x	x	S	1	2	3	4	5	Telegraph-character congiguration

Message Format

A message must begin with 12 letters shift characters. The maximum length of a message depends on the length defined in a READ macro instruction. The message ends with one of the terminations described above.

When input messages are pre-punched into paper tape, the 12 letters shift characters are required. However, the WRU, EOM, and EOT characters must never be pre-punched; they must be sent by the terminal operator.

Contention Resolution

Contention occurs when a terminal tries to send an input message at the same time as the computer tries to send an output message to that terminal; for example, when a WRITE TI, WRITE TT, or READ TE macro instruction is issued. The completion code is X'50'. Contention is resolved by the user's coding and the local operator's action, according to one of the following procedures:

a. Priority to the computer: the operator must wait and the user's program must try again to execute the stopped macro instruction.

b. Priority to the terminal: the operator continues sending his message and the user's program must issue a READ Continue macro instruction.

CHANNEL PROGRAMS

Terminal to CPU

World Trade Telegraph terminals are not polled. BTAM does not consider the EOM, EOT, and WRU signals as parts of an input message.

READ Initial (TI): After a READ Initial macro instruction has been issued the computer can receive a character from the terminal. The channel program executed for READ TI is:

1. Prepare
2. Sense
3. Read message

The Prepare command terminates when the first character is assembled by the line adapter. The Sense command reads the adapter sense byte in the first byte of the DECB response field to determine at any given time whether a message is being sent over the line. The input message is read until the computer receives an EOM, EOT, or WRU signal, or until a time-out occurs.

READ Continue (TT): A READ Continue macro instruction is issued to read an input message following an EOM signal (that is, when the EOT flag is not present in the DECB), or to receive an input message when the terminal was given priority over the computer in case of contention. The channel program generated and executed is:

1. Read message

The input message is read until the computer receives an WOM, EOT, or WRU signal, or unitl a time-out occurs.

READ Continue with Identification Exchange (TE): A READ TE macro instruction is issued when a WRU signal ends the execution of a previous READ Continue or READ Initial macro instruction and when at least one IAM or WRU feature is coded in the DTFBT, or when the user's program starts an identification exchange. In these cases the terminal lists used are the following:

1. Feature WRU is present in the DTFBT.

Name	Operation	Operands
[symbol]	DFTRMLST	WTTALST,0, numrec-integer, ridseq-hexchars, numsent-integer, tidseq-hexchars

numrec
: specifies the number of characters in the terminal ID sequence (must be a multiple of 4 if the control unit is a 2703, LTRS and FIGS not being included in this count).

ridseq
: is a string of characters that is to be compared to a terminal identification sequence sent from the terminal.

numsent
: specifies the number of characters in the CPU ID sequence.

tidseq
: is a string of from 7 to 20 characters (CPU ID) sent over the line after the terminal sends the WRU signal.

2. Feature IAM is present in the DTFBT, WRU is absent.

Name	Operation	Operands
[symbol]	DFTRMLST	WTTALST,0,0,0, numsent-integer, tidseq-hexchars

numsent
: specifies the number of characters in the CPU ID sequence

tidseq
: is a string of from 7 to 20 characters (CPU ID) sent over the line after the terminal sends the WRU signal.

Examples are given in Appendix A.

The channel program generated and executed is:

1. Write Mark characters
2. Write computer identification -- tidseq sequence
3. Write WRU signal -- LTRS if FEATURE=(WRU) is not coded

4. Read terminal identification sequence -- only for FEATURE=(WRU)

5. Read message

One LTRS character plus 'm' Mark characters are sent as follows:

1. When the terminal is equipped with the Motor-On optional feature, 'm' is always zero.

2. When the terminal is not equipped with the Motor-On optional feature, 'm' can take one of the following values: m=0 if the previous operation was a WRITE or READ operation ended with an EOM or WRU siganl, or m=the value given to the MONDLY operand of the DTFBT macro instruction.

If upon verification of the terminal identification the received ID and the expected result are not equal, the operation is posted complete with the completion code X'44'.

CPU to Terminal

World Trade Telegraph terminals are not addressed.

WRITE Initial (TI): The channel program generated and executed is:

1. Write Mark caracters

2. Write 12 pad characters

3. Write message

The number of Mark characters is determined as explained under the READ Continue with Identification Exchange macro instruction.

WRITE Continue (TT): The channel program generated and executed is:

1. Write Mark characters

2. Write message

The number of Mark characters is determined as explained under the READ Continue with Identification Exchange macro instruction.

Note: For purposes of message switching, it is not necessary for the user to update the count in the last CCW when using a WRITE macro with length operand coded as 'S'.

WORLD TRADE TELEX TERMINALS

GENERAL INFORMATION

The name World Trade Telex terminals refers to various European teleprinters using a start-stop 5-level code with two shifts (letters shift and numbers shift) to transfer data over switched point-to-point telegraph lines at 50 baud in a World Trade Telex network. The operator at the remote Telex terminal must dial the computer to establish a connection. BTAM provides support for auto-answering, but not for auto-dialing. After a connection is established, BTAM's support of World Trade Telex terminals is identical to its support of World Trade Telegraph terminals on a leased line.

Even though World Trade Telex terminals operate on switched lines, the SWITCH parameter of the DTFBT macro instruction must be coded as NO or omitted. Upon completion of transmission, the operator at the remote terminal may disconnect the line, or the computer may issue a WRITE Break macro instruction to disconnect the terminal automatically.

CHANNEL PROGRAMS

Terminal to CPU

Macro instructions for World Trade Telex terminals are coded identically to those for World Trade telegraph terminals for terminals-to-CPU communication.

CPU to Terminal

Macro instructions for World Trade Telex terminals are coded identically to those for World Trade Telegraph terminals for CPU-to-terminal communication with one addition, the WRITE Break macro.

WRITE Break (TB)

The channel program generated and executed is:

1. Write Break

The WRITE Break macro is issued on a switched Telex line to disconnect a World Trade Telex terminal from a Telex switching center. This macro causes BTAM to transmit a break signal. The length of the break signal is specified in the length operand of the WRITE Break macro and is dependent upon the country in which it is to be used. The area operand of the WRITE Break macro specifies the break signal to be transmitted. The break signal used must not be defined as the character X'DF'.

For binary synchronous transmission, one of the following transmission control units (TCU) must be attached to the System/370 channel unless the local 2715 is used:

- IBM 2701 Data Adapter Unit with Synchronous Data Adapter-Type II (SDA-II). The 2701 may be attached to either the multiplexer (switched or nonswitched lines) or the selector channel (nonswitched lines only).

- IBM 2703 Transmission Control unit with Synchronous Base I. The 2703 may be attached to the multiplexer channel only.

Binary synchronous communications (BSC) is supported for IBM Models 25 and above communicating with the following stations:

1. Point-to-point, nonswitched lines:

   IBM 1130 Computing System
   IBM 1800 Process and Control System
   IBM 2715 Transmission Control Unit
   IBM 2770 Data Communications Terminal
   IBM 2780 Data Transmission Terminal
   IBM 3735
   IBM System/3
   IBM System/360 Model 20
   IBM System/370

2. Point-to-point, switched lines:

   IBM 1130
   IBM 1800
   IBM 2715
   IBM 2770
   IBM 2780
   IBM 3735
   IBM System/3
   IBM System/360 Model 20
   IBM System/370

3. Multipoint, nonswitched lines:

   IBM 1130
   IBM 1800
   IBM 2715
   IBM 2770
   IBM 2780
   IBM 2972
   IBM 3735
   IBM 3270 Information Display System
   IBM System/3
   IBM System/360 Model 20

The channel programs for (BSC) vary depending on the line connection (that is, point-to-point contention, point-to-point dial, or multipoint). The variance is most apparent for those channel programs that are generated and executed for the purpose of establishing initial contact with the remote device. For example, the channel programs for WRITE Initial are constructed so that the following conditions occur:

1. On a point-to-point leased line, an ENQ character is transmitted to bid for control of the line;

2. On a point-to-point dial line, the appropriate phone number is dialed before transmitting the ENQ character (with optional ID verification);

3. On a multipoint line, the line is placed in control mode and the remote device is addressed before data transmission.

Once initial contact has been established, the channel programs do not vary greatly. Device dependencies do not affect the structure of channel programs; the channel programs for each line configuration apply to all BSC devices supported under that configuration. (Except for the 6-bit Transcode 2780).

In the following sections, the BSC channel programs are discussed according to the line connection. General information concerning each device is given prior to the channel program discussion. Only normal completions are described in the following sections. The Completion Codes section discusses posting of irregular completions.

SYSTEM/370 TO SYSTEM/370

CONTENTION

BTAM supports point-to-point Binary Synchronous Communication over a nonswitched line between the CPU and a remote CPU that is also a System/370 using BTAM support. The transmission codes available are EBCDIC and USASCII. The Read/Write/Control options available for CPU-to-CPU communications are listed in Figure 46.

The CPUs operate in contention. On rare occasions, both CPUs may simultaneously bid for control of the line by executing a WRITE Initial macro instruction. The user must coordinate the problem programs in both CPUs so that one is programmed to act as the master and the other as the slave (in the contention situation only). This is accomplished by coding FEATURE=MAS in the DTFBT macro instruction for one CPU and FEATURE=SLV for the other. (See the DTFBT macro instruction section.)

Transmission is initiated when one of the CPUs successfully sends the Inquiry (ENQ) character to the other CPU. Because polling and addressing are not used in this application, the *entry* operand in the READ and WRITE macro instructions is not used except for READ TTL, READ TPL, WRITE TQ, and the conversational WRITEs.

The user must provide the text framing characters SOH, STX, US, ETB, ETX, and DLE STX. BTAM provides DLE ETB, DLE ETX, and all other control characters.

Note: Since both SOH and STX, when beginning a transmission, reset but are not included in the block check, the following precaution is recommended. Include as the first character of the heading, following SOH, a uniquely-chosen non-control character that is never used as the first character following STX in a nontransparent text transmission. The user may select any character other than a data link control character or the percent sign (%) for this purpose. Consistent use of this practice avoids the processing of text as heading or heading as text due to a line error that changes STX to SOH or vice versa. When no error is indicated on completion of a text-read, absence or presence of the user-selected character is necessary and sufficient evidence of text or heading.

To establish the direction of transmission between the two CPUs, the CPU wishing to transmit bids for the line by transmitting an ENQ character (using an initial WRITE) and reading the response. If the proper alternating acknowledgment (ACK-0) is received, the CPU wishing to transmit has successfully gained control of the line and may transmit data.

When contention occurs, the FEATURE field in the DTFBT is examined by BTAM to determine the recovery procedure to be used. BTAM immediately ends the WRITE Initial for the slave CPU with completion code 50. The slave CPU problem program should issue the READ Initial or READ Initial Inquiry macro instruction. Meanwhile, BTAM retries the bidding sequence automatically for the master CPU. Should the retries be unsuccessful, BTAM posts completion code 44 and ends the WRITE Initial. Figure 42 lists the various completion codes posted when error conditions occur on WRITE Initial. Retry options are also specified.

DIAL

BTAM supports Binary Synchronous Communication over a switched network between the CPU and a remote CPU that is also a System/370 using BTAM support. The transmission codes available are EBCDIC and USASCII.

Transmission is initiated when either the CPU or the remote CPU performs a calling operation. For this reason, BTAM provides both calling and answering functions for the CPU. Optional Identification (ID) Verification procedures are provided by BTAM in place of polling and addressing procedures, which are not used for CPU-to-CPU Dial communication.

When transmission is initiated by either the CPU or remote CPU, a standard basic line control procedure is always used:

1. The calling CPU dials and sends the Inquiry (ENQ) character.

2. The called CPU answers, reads the ENQ character, and responds with a positive acknowledgment (ACK-0) if the line connection is valid, or does not respond if the line connection is invalid. No response is sent by the called CPU, for example, when the ENQ character is received with I/O error. The calling CPU would then retry sending the ENQ character.

3. On a positive response to the ENQ, the calling CPU transmits the first message.

Figures 43 and 44 list the various completion codes that may be returned for possible errors in READ Initial. Figure 45 lists the various completion codes that may be returned for the possible errors in WRITE Initial. The listed retry options may be employed when the user desires further attempts to recover without having

CPU	Completion Code	Meaning	Retry Options
Master	44	ENQ, NAK, or I/O Error received in response to the ENQ sent to the remote CPU.	1
	41 with Data Check	(TIV, TIXV only) Message received from remote CPU was in error.	2
	60	Response received to message sent to remote CPU was invalid acknowledgment.	3
	41	NAK received in response to text (Bit 7 of byte 29 in the DECB is set if n retries fail).	4
Slave	50	Bid for line unsuccessful. Master CPU is also bidding for the line.	5
	44	NAK or invalid response received in reply to ENQ.	1
	41 with Data Check	Same as for Master CPU.	2
	60	Same as for Master CPU.	3
	41	Same as for Master CPU.	4

Retry Options

1. Retry the WRITE Initial macro.
2. Initiate a READ Repeat macro.
3. Initiate a WRITE Inquiry macro.
4. Initiate a WRITE Continue macro.
5. Issue a READ Initial or a READ Initial Inquiry.

Figure 42. Retry Options for Completion Codes on WRITE Initial in CPU-to-CPU Contention

to disconnect and reenable the line. However, the user is reminded that BTAM automatically retries all of the listed errors the number of times specified in the retry operand of the DTFBT macro issued for the line group. The completion codes listed are posted by BTAM only if this number of retries fails to recover from the indicated errors.

ID Verification Procedures

Three optional BTAM-supported ID verification procedures are available to provide the user with greater circuit assurance and security protection. The following discussion is applicable to all BSC devices; however, devices that cannot be programmed (the 2770 and the 2780) do not verify or analyze ID sequences.

I AM/Who Are You? (IAM/WRU): Using this procedure, both the calling device and the called device exchange ID graphic characters when line connection is being established:

1. The calling device dials and sends the ENQ, preceded by a maximum of 15 graphic characters, to the called device, as proof of identity.

2. The called device answers, reads the ID-ENQ sequence, verifies it, and if the ID is:

    a. valid, sends the ACK-0 characters, preceded by a maximum of 15 ID graphic characters of its own, to the calling device.

    b. invalid, does not reply. Instead ERP restarts the channel program at the Read CCW, awaiting the retry of the ID, ENQ by the calling device.

    c. invalid, after n retries (n = number specified by RETRY in DTFBT), may issue a WRITE Disconnect or a CONTROL Disable to break the line connection.

3. The calling device then:

    a. verifies the ID-ACK-0 response sequence and, if valid, transmits the message. If the ID sequence is invalid, the calling device responds by retrying its ID-ENQ sequence again or by sending the disconnect signal, depending on the user's option.

b. if the response is NAK, may either retry the ID-ENQ sequence or send the disconnect signal.

c. if the response is the disconnect signal (DLE, EOT), disables the line by issuing the CONTROL Disable macro.

I Am (IAM): This procedure requires that the calling device send the ID graphic characters preceding the ENQ, and that the called device verify and send only a positive acknowledgment (ACK-0) in response to a valid ID sequence. Invalid IDs are handled in the same manner as discussed for IAM/WRU.

Who Are You (WRU): This procedure requires that the calling device send the ENQ character only, and that the called device respond with an ID sequence preceding ACK-0. Recovery is similar to that discussed for IAM/WRU.

Prerequisites: The user chooses the ID verification procedures desired by coding the appropriate codes in the sublist of the FEATURE keyword operand in the DTFBT macro instruction as shown below.

Procedure	Code for Calling Operation	Code for Answering Operation
IAM	SIX	RIX
WRU	SXW	RXW
IAM/WRU	SIW	RIW

The code is dependent only upon what procedure is to be employed by the calling device. The code for the calling operation at the calling device must correspond to the code for the answering operation at the called device. For example, if the device is calling the remote device and employing the IAM procedure, the problem program must code SIX for the calling operation in the FEATURE sublist. Since each device can

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
Read Initial (with Answering List)	44	ENQ not received properly	No ID Verif. WRU	1 1
		Received ID, ENQ did not compare to expected ID, ENQ in IDLST	IAM IAM/WRU	1 1
	62	ENQ or ID, ENQ was received instead of the text message, meaning the remote (calling) CPU either did not receive or rejected the positive acknowledgment sequence (ACK-0 or ID, ACK-0) sent.	No ID Verif. WRU IAM IAM/WRU	2 or 5 2S or 5 2 or 5 2S or 5
	41 with lost data, data check, or overrun	Text message received with I/O error.	No ID Verif. WRU IAM IAM/WRU	3 or 5 3 or 5 3 or 5 3 or 5
	41 with time out	No text message received.	No ID Verif. WRU IAM IAM/WRU	1, 4, or 5 1, 4, or 5 1, 4, or 5 1, 4, or 5

Retry Options: The S following the number in the table means the user must code 'S' for the entry operand in the retry macro. (BTAM performs the necessary ID Verification again.)

1. Issue a CONTROL Disable (TD) to disconnect the line.
2. Issue a READ Continue (TT) (2S: Code 'S' for entry operand).
3. Issue a READ Repeat (TP).
4. Issue a READ Inquiry (TQ).
5. Issue a WRITE Disconnect (TD) to break the connection.

Figure 43. Error Completion Codes and Retry Options for READ Initial (answering) in CPU-to-CPU Dial

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
	44	NAK or invalid ID received in response to sent ENQ.	No ID Verif. WRU	1 or 5 1S or 5
	44	NAK or invalid ID received in response to sent ID, ENQ.	IAM IAM/WRU	1S or 5 1S or 5
	60	Wrong acknowledgment received (ACK-1 instead of ACK-0).	No ID Verif. WRU IAM IAM/WRU	1 or 5 1S or 5 1S or 5 1S or 5
READ Initial (with Calling list)	41 with time out	No response received to ENQ or ID, ENQ (the sent ID may be invalid). The Read CCW has a TP code of hex 05 (for No ID or IAM) or hex 07 (for WRU or IAM/WRU).	No ID Verif. WRU IAM IAM/WRU	1 or 5 1S or 5 1S or 5 1S or 5
	41 with time out	No text message received. The Read CCW has a TP Code of hex 11 (no buffers) or hex 12 (buffers).	No ID Verif. WRU IAM IAM/WRU	2, 3, or 5 2, 3, or 5 2, 3, or 5 2, 3, or 5
	41 with lost data, data check, or overrun	Text message received with I/O error. The READ CCW has a TP code of hex 11 (no buffers) or hex 12 (buffers).	No ID Verif. WRU IAM IAM/WRU	4 or 5 4 or 5 4 or 5 4 or 5

Retry Options: The S following the number in the table means the user must code 'S' for the entry operand in the retry macro. (BTAM performs the necessary ID Verification again.)

1. Issue a WRITE Inquiry (TQ) (1S: Code 'S' for entry operand).
2. Issue a CONTROL Disable (TD) to disconnect the line.
3. Issue a READ Inquiry (TQ) to monitor the line for an ENQ from the remote station.
4. Issue a READ Repeat (TP).
5. Issue a WRITE Disconnect (TD) to break the connection.

Figure 44. Error Completion Codes and Retry Options for READ Initial (calling) in CPU-to-CPU Dial

both call and answer calls from each remote device, it is possible that in calling the procedure would be IAM, and in answering, WRU. The user would code:

```
FEATURE=(BSC,SIX,RWX)
```

The user program at the remote device would correspondingly code:

```
FEATURE=(BSC,RIX,SWX)
```

If one of the calling/answering ID codes is omitted in the sublist, ID verification is not performed for the relevant operation (calling or answering). If the calling/answering codes are omitted completely, this means the user has chosen not to use ID Verification procedures for either calling or answering operations on that line.

Note: It is the user's responsibility to know which procedure, if any, is expected by the remote device.

For reliability purposes, each ID sequence may be defined with a minimum length of two characters. It is recommended that the BTAM user define the second character in each ID sequence identical to the first, and establish the first character of each ID sequence in the problem program as a unique character. This procedure gives the problem program more protection from the possibility of a valid ID sequence being changed into another valid ID sequence by line errors. This idea may be extended for a longer ID sequence by defining identical pairs of characters.

Expanded ID Verification

Improved BTAM includes an expanded ID verification facility in which the user can designate many different ID sequences that can be accepted from remote stations. This allows each station to be assigned a unique ID sequence; these sequences may be of different lengths. (The maximum length of an ID sequence, excluding the ENQ, ACK-0, or NAK, is fifteen characters.) With the expanded facility, the central computer can communicate, over the same switched line termination, with all BSC stations, regardless of whether they employ ID sequences or not.

To provide the expanded facility, BTAM provides three macro instructions -- READ Connect, READ Connect with Tone, and WRITE Connect -- and a form of terminal list designated as SWLST. Unlike the single-entry format of the terminal lists used with the READ Initial and WRITE Initial macros, a list of the SWLST format may contain multiple entries. Each entry has a field containing a valid ID sequence that can be accepted from a remote station, and a control byte. Each entry may have an optional user data area.

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. The user program typically uses this address to determine which remote station called or answered the central computer.

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
Initial-type WRITE'S	44	NAK or invalid ID received in response to sent ENQ.	No ID Verif. WRU	1 or 4 1S or 4
		NAK or invalid ID received in response to sent ID, ENQ.	IAM IAM/WRU	1S or 4 1S or 4
	60	Wrong acknowledgment received in response to ENQ or ID, ENQ. Read CCW has TP code of hex 05 (no ID or IAM) or hex 07 (WRU or IAM/WRU).	No ID Verif. WRU IAM IAM/WRU	1 or 4 1S or 4 1S or 4 1S or 4
	60	Wrong acknowledgment received in response to the transmitted message. Read CCW has a TP code of hex 20.	No ID Verif. WRU IAM IAM/WRU	1 or 4 1 or 4 1 or 4 1 or 4
	41	NAK received in response to text message sent. Bit 7 of DECB byte 29 is set.	No ID Verif. WRU IAM IAM/WRU	2 or 4 2 or 4 2 or 4 2 or 4
	41 with lost data, data check, or overrun	Text message received with I/O error (TIV,TIXV only).	No ID Verif. WRU IAM IAM/WRU	3 or 4 3 or 4 3 or 4 3 or 4
	41 with time out	No response received to the sent message.	No ID Verif. WRU IAM IAM/WRU	1 or 4 1 or 4 1 or 4 1 or 4

Retry Options: The S following the number in the table means the user must code 'S' for the entry operand in the retry macro. (BTAM performs the necessary ID Verification again.)

1. Issue a WRITE Inquiry (TQ) (1S: Code 'S').
2. Issue a WRITE Continue-type macro to retransmit the message.
3. Issue A READ Repeat (TP).
4. Issue a WRITE Disconnect (TD) to break the connection.

Figure 45. Error Completion Codes and Retry Options for Initial-type WRITEs in CPU-to-CPU Dial

OPTION	TYPE CODE	CPU-CPU Contention	CPU-CPU Dial
Read Initial	TI	X	X
Read Initial Inquiry	TIQ	X	
Read Connect	TIC		X
Read Connect with Tone	TIW		X
Read Initial with Tone	TIZ		X
Read Continue	TT	X	X
Read Continue with Leading Graphics	TTL	X	X
Read Repeat	TP	X	X
Read Repeat with Leading Graphics	TPL	X	X
Read Inquiry	TQ	X	X
Read Interrupt	TRV	X	X
Write Initial	TI	X	X
Write Connect	TIC		X
Write Initial Transparent Block	TIE	X	X
Write Initial Transparent Text	TIX	X	X
Write Initial Transparent Conversational	TIXV	X	X
Write Initial Conversational	TIV	X	X
Write Continue	TT	X	X
Write Transparent Block	TE	X	X
Write Transparent Text	TX	X	X
Write Transparent Conversational	TXV	X	X
Write Conversational	TV	X	X
Write Disconnect	TD		X
Write End-of-Transmission	TR	X	X
Write Wait-Before-Transmitting	TW	X	X
Write Inquiry	TQ	X	X
Control Disable	TD		X
Control Mode	TM	X	X

Figure 46. READ/WRITE/CONTROL Options for CPU-to-CPU Communication

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 complete.

By setting the control byte before the READ Connect or WRITE Connect 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.

The optional user data field is a four-byte area. The user may place a relocatable expression in this field as an address to be associated with the ID sequence (or ENQ character) contained in that entry. The user data field is typically used for the address of a subroutine to be called when the remote station represented by the ID establishes contact with the CPU.

For more detailed information on use of the expanded ID verification facility, see the explanations of DFTRMLST, format for READ Connect, and WRITE Connect in the section on Terminal Lists.

Terminal Lists

READ Initial: When performing the answering function, READ Initial answers an incoming call, verifies the received ID (IAM,IAM/WRU), sends an ID sequence (WRU, IAM/WRU) preceding ACK-0, and reads in the message block received from the remote device.

If no ID Verification is used (that is, none of the codes, RIX, RXW or RIW, is coded in the FEATURE sublist of the DTFBT for the line group), the entry operand in the READ initial must specify the address of a terminal list constructed by the following macro.

Name	Operation	Operands
symbol	DFTRMLST	DIALST,0

The 0 specifies an answer list.

When ID Verification procedures are used, it is necessary that the user specify the address, via the entry operand, of an IDLST type terminal list defined by a DFTRMLST macro instruction in the problem program.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,0,numrec-integer, [idrec-hexchars,] numsent-integer, [idsent-hexchars]

IDLST and 0 must be coded as shown. The 0 indicates an answer list. The numrec operand specifies the number of ID characters expected from the calling terminal. This would be zero for the WRU procedure. For the IAM and IAM/WRU procedures this number would be equal to the number of ID characters plus one (for ENQ character). The idrec operand is omitted for WRU. For IAM and IAM/WRU it specifies the hexadecimal representation (the X' ' framing characters are not coded) of the transmission code for the ID characters plus the ENQ character. Thus, if four ID characters and an ENQ are expected, numrec would specify 5 and idrec would be specified by 10 hexadecimal digits.

The numsent operand is 0 for the IAM procedure. For WRU and IAM/WRU it specifies the number of ID characters to be sent to the remote (calling) device. The number specified does not include the ACK-0 characters, since they are supplied by BTAM. For IAM the idsent operand is omitted. For WRU and IAM/WRU the idsent operand specifies the hexadecimal respresentation (without X' ') of the transmission code for the ID characters being sent. If four ID characters are to be sent, numsent would specify 4 and idsent would be coded as the 8 hexadecimal digits representing the ID characters.

When performing the calling function, READ Initial dials the telephone number of the remote device, establishes valid line connection, indicates to the remote device that the call was made for the purpose of receiving messages, and then receives the first message block (if any) that the remote device transmits.

If the user does not use ID Verification (that is, none of the codes, SIX, SXW, or SIW, is coded in the FEATURE sublist of the DTFBT macro for the line group), the entry operand specifies the address of a terminal list defined via DFTRMLST as follows.

Name	Operation	Operands
symbol	DFTRMLST	DIALST, dialcount-integer, dialchars-decchars

The dialcount specifies the number of dialing characters to be used (example: 7) and dialchars specifies the actual dialing characters (example: 1234567).

If ID Verification is used in the READ initial (one of the codes, SIX,SXW, or SIW, is coded in the FEATURE operand sublist of the DTFBT for the line group), the user must provide, via the entry operand, the address of an IDLST constructed by a DFTRMLST macro.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numrec-integer, [idrec-hexchars,] numsent-integer [,idsent-hexchars]

The nonzero dialcount indicates a calling list. This operand specifies the dialing characters to be used. The numrec operand specifies the number of ID characters to be received from the remote CPU. The idrec operand specifies the actual characters expected. For example, if A,B, C,ACK-0 is expected, numrec would be coded as 5 and idrec (EBCDIC transmission code) would be coded C1C2C31070. The numsent operand specifies the number of ID characters to be transmitted to the remote CPU. The idsent operand specifies the actual ID characters to be transmitted. Neither numsent nor idsent includes provision for ENQ, which is supplied by BTAM. For example, if A,B,C,ENQ is to be transmitted, numsent specifies 3, while idsent is coded C1C2C3. Depending upon which procedure (IAM,WRU or IAM/WRU) is employed, idrec or idsent is optionally coded.

<u>WRITE Initial</u>: If no ID Verification is employed, the user must specify, via the entry operand, the address of a DIALST type terminal list constructed by a DFTRMLST macro instruction in the problem program. If the number to be dialed is 12345678, for example, the user would code:

```
DLIST DFTRMLST    DIALST,8,12345678
```

The inlist positional operand is used only for WRITE Initial Conversational (TIV) and WRITE Initial Transparent Conversational (TIXV) and is explained in a later section.

Name	Operation	Operands
symbol	DFTRMLST	DIALST, dialcount-integer, dialchars-dechars [,inlist-relexp]

If ID Verification is used in the WRITE Initial, the user must provide, via the entry operand, the address of an IDLST constructed by a DFTRMLST macro.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numrec-integer, [idrec-hexchars,] numsent-integer [,idsent-hexchars] [,inlist-relexp]

For the IAM/WRU procedure, all operands are coded, except inlist, which is used only for WRITE Initial Conversational (TIV) and WRITE Initial Transparent Conversational (TIXV).

The numrec and idrec operands refer, respectively, to the number of ID characters expected from the remote (called) device and the hexadecimal representation for them. The user is required to include the ACK-0 characters in these operands. This means, if 4 ID characters plus ACK-0 are expected, numrec specifies 6 and idrec is coded as 12 hexadecimal digits, the last 4 of which represent ACK-0. The numsent and idsent operands refer, in analogous manner, to ID characters to be sent to the remote device since this is provided by BTAM. So if 4 ID characters are to be sent preceding the ENQ, numsent specifies 4 and idsent is coded as 8 hexadecimal digits.

The IAM and WRU procedures require similar coding except that for:

- IAM, numrec is coded 0 and the idrec operand is omitted.
- WRU, numsent is coded 0 and the idsent operand is omitted.

In addition to having the ability to use automatic dial, the user now has the capability of manual dial. The operator at the CPU may manually dial a remote device upon request. Once the line connection has been established, BTAM will initiate transmission to the remote device. If ID Verification is not being used, the user must specify a DIALST with a dialcount of zero. The format of the DFTRMLST used for no ID verification is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

If ID verification is being used, the user must specify an IDLST. The dial count in the list must be coded as MD (manual dial). The format of the DFTRMLST used for ID verification is:

Name	Operation	Operand
symbol	DFTRMLST	IDLST,MD,numrec-integer, [idrec-hexchars,] numsent-integer [,idsent-hexchars]

The numrec and idrec operands refer, respectively, to the number of ID characters expected from the remote device and the hexadecimal representation for them. The numsent and idsent operands refer to the ID characters to be sent to the remote device.

Manual dial is supported for all of the Initial WRITEs.

<u>READ Connect and READ Connect with Tone</u>: READ Connect allows initial contact to be established with a remote BSC station and performs a specific action based on the ID sequence, if any, received from the remote station. The possible actions include reading a message block, disconnecting the line, and immediately returning control to the user program.

READ Connect with Tone allows initial contact to be established with a remote BSC station which has a data set that does not generate a tone. The macro performs a specific action based on the ID sequence, if any, received from the remote station. The possible actions include reading a message block, disconnecting the line, and immediately returning control to the user program.

These macros are used only when the expanded ID verification facility is employed. The <u>entry</u> operand of the READ Connect or READ Connect with Tone macro must specify the name of an answering list of the SWLST format, as defined by a DFTRMLST macro.

Name	Operation	Operands
symbol	DFTRMLST	SWLST,AN, entrywidth-integer, [userlength-{<u>0</u> | 4}], idcount-integer, idsent-hexchars [{(authsequence-hexchars) [,[controlvalue-{<u>0</u> | 1 | 2}], [userdata-relexp]])}...]

SWLST and AN must be coded as shown. SWLST specifies a list structure for expanded BSC ID verification. AN indicates an answer list.

The entrywidth operand specifies the number of bytes to be allocated for each list entry containing a user-defined authorized ID ENQ sequence. The integer specified should be equal to 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 shorter than the maximum length are assembled left-justified within the fixed-length ID field allocated per entry. Each userdata field (if any) and control byte have the same offset within all entries. The value specified may be zero if no sequence other than ENQ, alone, is expected and ENQ is not put in the list.

The userlength operand 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 idcount operand specifies the length (in bytes) of the field required to accommodate the ID ACK-0 sequence defined by the <u>idsent</u> operand. The allowable range is 2 (ACK-0 alone) through 17; up to 15 ID characters may be specified.

The idsent operand 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 the user's 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 sends out the ID ACK-0 sequence if the checked value is 0.

The authsequence operand specifies the hexadecimal representation of an authorized ID ENQ sequence. Each ID ENQ sequence is defined in a separate sublist. The user 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.

The controlvalue operand specifies the value of the control byte associated with each list entry assembled for an authsequence sequence. This value determines the automatic BTAM action to be performed when the sequence is received on a READ Connect operation using the list. The value and corresponding BTAM actions are:

0

specifies that BTAM should send the idsent sequence and read a message block (if any) from the calling station. If a controlvalue operand is omitted 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.)

1

specifies that BTAM should send the DLE EOT sequence (BTAM-provided), break the line connection, and restart the channel program at the Enable command (to await a new call). The user 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).

2

specifies that BTAM should post normal completion of the READ Connect macro immediately with the address of the entry containing ENQ stored in the first word of the answering list. This control byte value is for use only when ENQ alone is defined as an entry in the SWLST. This can be used for communication with terminals (such as certain 2770s or 2780s) that do not support ID verification. This control byte value permits control to be returned to the user program when ENQ is received on a READ Connect operation with ENQ defined in the SWLST terminal list. By then issuing a READ Continue macro, the user program can communicate with the remote terminal without having BTAM transmit the ID ACK-0 sequence that is defined in the answering SWLST. This control byte value should not be specified for any ID ENQ sequence other than ENQ alone.

The userdata operand specifies the relocatable expression to be assembled right-justified in the userdata field of the associated list entry. If this operand is omitted in a sublist and userlength specifies 4, four noninitialized bytes are allocated for the corresponding list entry. No boundary alignment can be assumed for the user data field.

Note: A maximum of 94 sublists can be coded for an answering list of the SWLST form. 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.

WRITE Connect: 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.

This macro is used only when the expanded ID verification facility is employed. 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.

Name	Operation	Operands
symbol	DFTRMLST	SWLST,{AD}, {MD} [dialcount-integer, dialchars-decchars,] entrywidth-integer, [userlength-{0}], {4} idcount-integer, idsent-hexchars [{,(authsequence- hexchars [,controlvalue-{0} {1} [userdata-relexp]])}...]

SWLST specifies a list structure for extended BSC ID verification and must be coded as shown.

AD specifies an auto-dial calling list. When AD is used, 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 a manual-dial calling list. When MD is used, tha dialcount and dialchars operands are omitted 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.

The dialcount operand specifies the number of dial characters (bytes) used in the auto-dialing operation. This operand is coded only if AD is coded as the preceding operand.

The dialchars operand specifies the decimal digits of the telephone number to be dialed. This operand is coded only if AD is coded.

The entrywidth operand 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 accomodate the authorized ID ACK-0 sequence of maximum length, plus the userdata field (0 or 4), plus one for the entry's control byte. Authorized ID ACK-0 sequences shorter than the maximum are assembled left-justified within the fixed-length ID field allocated per entry. Each userdata field (if any) and control byte have the same offset within all entries.

The userlength operand 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 idcount operand specifies the length (in bytes) of the field required to accommodate the ID characters (if any) and ENQ defined by the idsent operand. The allowable range is 1 (ENQ alone) through 16; up to 15 ID characters may be specified.

The idsent operand specifies the hexadecimal representation of the ID characters (if any) and ENQ to be sent to the remote station. Typically, the ID characters will convey station identification.

The authsequence operand 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). The user should code a separate ID ACK-0 sequence for each sequence that can possibly be received from remote (answering) stations. ACK-0 must be coded following each ID sequence; it must not be coded when no ID sequence is used.

The controlvalue operand specifies the value of the control byte associated with each list entry assembled for an authsequence sequence. 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 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 station is not ready to receive, and the user wishes to retry in the expectation that the remote station will shortly be ready to receive. The number of retries performed for this control byte value is limited to the number specified in the RETRY operand of the DTFBT macro defining the line used. If more retries than this are desired, the user can reissue the WRITE Connect macro; BTAM bypasses the initial Enable or Dial command if the line connection is already established.

The userdata operand specifies the relocatable expression to be assembled right-justified in the userdata field of the associated list entry. If this operand is omitted in a sublist and userlength is specified as 4, four noninitialized bytes are allocated for the corresponding list entry. No boundary alignment can be assumed for the user data field.

Note: A maximum of 92 sublists can be coded for a calling list of the SWLST form. The control byte values for a calling list cannot be changed by use of the CHGNTRY macro. The first fullword of the SWLST list is the area in which BTAM stores (before 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.

IBM 2780 DATA TRANSMISSION TERMINAL

BTAM supports binary synchronous communication with the remote IBM 2780 Data Transmission Terminal over both point-to-point and multipoint line connections. The remote 2780 may be connected point-to-point over either a switched or nonswitched network.

There are three available transmission codes for the 2780; 6-Bit TRANSCODE, EBCDIC, and USASCII. Code charts for these codes are shown in Appendix F.

The 2-character data link control functions in the three codes vary in the second character used, as shown in in the following table:

	Characters used in:		
Function	USASCII	EBCDIC*	TRANSCODE
ACK-0	DLE,0	DLE,X'70'	DLE,-(hyphen)
ACK-1	DLE,1	DLE,X'61'	DLE,T
WACK	DLE,;	DLE,X'6B'	DLE,Z

*The second characters in the EBCDIC code are shown in their hexadecimal representation.

In certain cases (for example, polling/addressing sequences), the user is required to define the hexadecimal equivalents of the transmission code characters in main storage. If the transmission code is EBCDIC there is, of course, no problem. For USASCII and TRANSCODE the user must remember that each 7-bit or 6-bit character, respectively, is to be represented hexadecimally within an 8-bit System/370 byte, all eight bits of which are sent over the System/370 channel to the control unit on a write operation. The following rules apply:

- In main storage, bits 1-7 in a System/370 byte correspond to bits b^7-b^1, respectively, of the USASCII character. The zero-bit in the System/370 byte is always zero (OFF). When the control unit receives a byte over the channel, a parity bit is sent over the line along with bits 1-7 of the byte. Conversely, when 7 bits plus a parity bit are received by the control unit from the line, the 7 (data) bits are read into main storage right-justified in a byte, and the zero-bit is set to zero.

  Example: The hexadecimal equivalent (in main storage) of the USASCII character A is 41.

- For TRANSCODE, a similar rule holds. The hexadecimal equivalent is right-justified in a System/370 byte (bits 2-7) and the zero-bit and 1-bit are always set to zero (OFF). Only bits 2-7 are sent over the line.

  Example: The hexadecimal equivalent of the TRANSCODE character A is 01.

For the 2780 there are 4 characters in each code that are not data link control characters, but that provide secondary end-to-end control functions. These are:

1. BEL - The Bell character is used only in the 2780-to-2780 communication and not in CPU-to-2780 communication. Its function in the terminal-to-terminal environment is to cause the audible alarm to sound when it is received by the 2780.

2. EM - The "end of medium" character has a twofold function. It causes the card reader to stop reading a card and it causes a punch eject when received by the punch (in normal text mode for both cases). It does, however, appear on the line or the punched card along with the data characters.

3. ESC - The escape character is the first character in a 2-character sequence providing a dual function:

   - On a point-to-point line the escape sequence is used for component selection (i.e., of the printer or the punch). Component selection is discussed in the Component Selection section. For multipoint lines the techniques of polling and addressing are used and the function of component selection through escape sequences is not applicable.

   - On either point-to-point or multipoint lines the escape sequence is used for vertical forms control of the printer. Vertical forms control is achieved by specifying one of the following 2-character sequences as the first two characters in a block of data (that is, a print line record) transmitted to the printer.

Sequence		Forms Motion
USASCII	*EBCDIC/TRANSCODE	AFTER Print
ESC,Q	ESC,/ (Slash)	Single Space
ESC,R	ESC,S	Double Space
ESC,S	ESC,T	Triple Space
ESC,A	ESC,A	Skip to 1
ESC,B	ESC,B	Skip to 2
ESC,C	ESC,C	Skip to 3
ESC,D	ESC,D	Skip to 4
ESC,E	ESC,E	Skip to 5
ESC,F	ESC,F	Skip to 6
ESC,G	ESC,G	Skip to 7
ESC,H	ESC,H	Skip to 8

*In EBCDIC the PRE (Prefix) character = ESC

When single spacing is desired and no component selection is required (for a point-to-point connection), the escape sequence (ESC,/ or ESC,Q) may be omitted. For the other vertical forms controls, the appropriate escape sequence must be sent at the beginning of each record, so that the desired forms motion can be accomplished for that record block.

4. HT - The horizontal tab character is used following ESC to indicate the start of a format record. A format record is stored by the 2780 and used to format subsequent blocks received for the printer. HT is used within a format record to indicate where a stop is desired on the print line. A space character in the format record indicates that a stop is not desired at the print line position corresponding to the position occupied by the space character in the format message. No character other than space (SP) or horizontal tab (HT) is permitted in a format message.

In the example of a format message shown in Figure 47 there are no stops defined beyond print position 16. A format message myst always be sent in nontransparent mode. A format record is maintained until:

- A new format record is received by the 2780.
- Power is removed from the terminal.
- A card is read by the card reader (EBCDIC and USASCII only).
- A record is sent to the punch (EBCDIC and USASCII only).

For terminals using 6-Bit TRANSCODE, the format record is retained even if card reading or punching occurs.

The HT character is also used in subsequent data blocks (once a format record has been stored) to define a skip to the next stop position on the print line. For example, if the user desired to print two characters (A,B) at print positions 1 and 2, and three characters (C,D,E) at print positions 12, 13, and 14, he could send the message (with the format record in Figure 31 stored):

STX A B HT HT C D E ETB

If HT appears in nontransparent data when no format record is stored, or if it appears in the data beyond the position defined as the last stip on the print line, an error occurs (overrun of the print line).

Format records can be stored and used by the 2780 only if it is equipped with the Printer Horizontal Format Control feature. Formatting of blocks transmitted to the 2780 in transparent mode is not performed, because the HT character is not recognized as an end-to-end control character. Instead, it would be treated as a data character. If the 2780 is not equipped with the Printer Horizontal Format Control feature, the HT character is not recognized apart from data characters in transparent or nontransparent mode.

The 2780 has a 400-position line buffer providing service for the data link with the CPU. The basic 2780 has the capability of receiving and sending two records per transmission (without line turnaround until after the second record). The first record is ended by the unit separator (US) character (in all codes), which provides the ITB

$$\text{STX}\ \underbrace{\text{ESC HT}}_{1}\ \underbrace{\text{SP SP SP SP}}_{2}\ \underbrace{\text{HT}}_{3}\ \underbrace{\text{SP SP SP SP SP SP}}_{4}\ \underbrace{\text{HT}}_{5}\ \underbrace{\text{SP SP SP}}_{6}\ \underbrace{\text{HT}}_{7}\ \text{ETB}$$

1. Two-character sequence introducing the format record.
2. Indicates print positions 1-4 do not contain a stop.
3. Indicates a stop at print position 5.
4. Indicates no stop in print positions 6-11.
5. Indicates a stop at print position 12.
6. Indicates no stop in print positions 13-15.
7. Indicates a stop at print position 16.

Figure 47. Example of a Format Record

function, while the second record is ended by ETB or ETX. Thus, in nontransparent mode, messages appear on the line in the format:

STX---Text---US---Text---ETB(or ETX)

In transmission to the 2780, only the "text" is printed or punched. Note that US, ETB, and ETX provide the function of punch eject without being punched themselves (unlike EM).

The STX (Start-of-Text) character is mandatory at the beginning of the first record, and optional at the beginning of the second (after US). When a 2-record message is received by the 2780 from the CPU, each record is individually checked. If the first intermediate block is valid and the second is invalid, the first record is printed or punched, and the second is not. A subsequent retransmission of the 2-record message results in the printing or punching of the second record only, if it is valid. If the first record is invalid, neither record is printed or punched, to avoid processing out of sequence.

In transparent mode (may be used only with the EBCDIC transmission code) BTAM supports only single-record transmission to the 2780. The format of such a message is:

DLE, STX---Text---DLE, ETB (or DLE, ETX)

However, the 2780 transmits (from the card reader) transparent messages of the form:

DLE, STX---Text---DLE, US DLE,
STX---Text---DLE, ETB (or DLE, ETX)

The distinction between ETB and ETX is that ETX is used to end the last block in a transmission. Transparent mode is not used in communication with the 2780 when the transmission code is USASCII or 6-bit TRANSCODE.

If the 2780 is equipped with the Multiple Record Transmission feature, up to seven records may be sent or received by the 2780 in nontransparent mode. In transparent mode (EBCDIC), the 2780 sends up to four records per transmission. Only one record per transmission may be sent to the 2780 using BTAM support in transparent mode. Figure 48 summarizes block number and block length considerations.

Although escape sequences cannot perform the function of component selection (see the Component Selection section) in transparent mode, they may be used in transparent transmission 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 (80, 120, or 144, depending on the printer used). 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 escape (ESC).

When messages are received by the 2701 or 2703 control unit, an error information byte (EIB) is inserted following each US, ETB, or ETX and sent over the System/370 channel into main storage if the 2701 or 2703 control unit is in error information byte mode. The EIB has the following significance:

- No bits ON = No error in intermediate block.
- 4-bit ON = Data check in the block.
- 5-Bit ON = Overrun.

	Number of records per transmission	Number of data characters[1] per record
Nontransparent Mode	Maximum: 2 without MRT[2] 7 with MRT	Variable record length Maximum: 80 (to punch) 80 (from reader) print line[3] (to printer)
Transparent Mode	Maximum: 1 (to 2780) 2 (from 2780 without MRT) 4 (from 2780 with MRT)	Fixed record length 80 (to punch) print line (to printer) 80(from reader)

[1]Not including control characters (STX, US, etc.) or escape sequences.
[2]MRT=Multiple Record Transmission feature of the 2780.
[3]The print line may be 80, 120 or 144 positions.

Figure 48. Block Number and Block Length Considerations

The user generally need not concern himself with the EIBs because BTAM automatically attempts recovery when an error occurs. A completion code in the DECB informs the user when an irrecoverable error occurs. The completion code fully defines the error. The user needs to consider the presence of EIBs in applications involving message switching, in which he would not want to send the EIBs to some other terminal. To avoid reception of an EIB by the remote terminal, change the EIB to a SYN character before transmitting it. The remote binary synchronous control unit automatically strips out the SYN character when it is received.

Messages from the 2780 are always sent in intermediate block check mode.

When a message is received in transparent mode, the 2701 (or 2703) strips out the DLE character preceding US, ETB or ETX, and does not send it over the channel into main storage. The EIB, of course, is inserted following the US, ETB, or ETX, as before. The EIB character is sent into main storage under the same (CCW) count control as any other character.

PROGRAMMING NOTES

The following considerations are important to the programmer:

- In nontransparent mode, maximum record length is 169 data characters. Records exceeding 169 data characters cause an I/O buffer overrun error. This error results in the 2780 sending an EOT reply to the message. The 2780 would then be in control mode.

  Records exceeding 80 characters (for the punch) or the print line (for the printer) are truncated without error when received by the 2780, as long as they do not exceed 169 characters.

- In nontransparent mode, variable-length records may be sent to the 2780. In transmitting to a 2780 equipped with the multiple record transmission feature, the programmer may send up to 7 records in a transmission, provided the number of characters does not exceed the 400 positions in the line buffer. The STX character preceding a record is stripped out by the receiving 2780 control unit and does not reach the line buffer. All other characters, including escape sequences, end-to-end controls, and US, ETB, or ETX, occupy positions in the buffer.

- If blank cards (for subsequent punching) are in the 2780 card reader hopper behind a deck being transmitted to the CPU, the first blank card will cause the 2780 (if equipped with the Auto Turnaround Feature) to stop reading and send an ETX over the line. The blanks are not sent over the line. In this case the last message from the 2780 will be:

  STX---Text---US ETX or
  STX ETX.

- Nontransparent and transparent modes cannot occur in the same transmission to the 2780. That is, a message of the form:

  STX---Text---US, DLE,
  STX---Text---DLE, ETB

  is not permitted.

- There are two responses by the 2780 to abnormal conditions occurring while the 2780 is receiving a message from the CPU: NAK and EOT.

  NAK is used for the class of errors that can result from line errors. For example, the block check character(s) (inserted by the 2701 or 2703 control unit) following a message block transmitted to the 2780 can be invalid, indicating to the 2780 that the received message block was in error. A NAK response by the 2780 requests a retransmission. Another type of line error can result in a message block received by the 2780 appearing to contain too many records or too many characters (although a problem program error can produce the same result). A NAK response by the 2780 requests retransmission to correct the line error. BTAM ERP automatically attempts recovery in these cases (see the RETRY operand of the DTFBT). Of course, if the problem program has actually transmitted too many records or characters within a message block, a correction of the problem program is required for successful recovery.

  EOT is the response used by the 2780 for detectable errors in the problem program (i.e., using horizontal format control without first storing a format record, transmitting a single record having too many characters in a message block, etc.), and also for the occurrence of the 2780 internal errors. The latter errors can be the result of punch jam, printer forms check, buffer parity check, etc. When they occur, the 2780 generally responds immediately with an EOT, thereby informing the transmitting CPU of a condition at the 2780 requiring the operator intervention before normal operation can be resumed.

Recovery from such a disruption in the normal operation is dependent on the system discipline of the particular application. That is, the problem program restart and recovery procedures and the instruction guide for the 2780 operator can be instructed to prepare the 2780 to expect retransmission of the message to which EOT was responded, or alternatively, the operator can prepare the 2780 to receive a new message unrelated to the previous message. The problem program should adhere to the same descipline.

In certain instances the 2780 will not immediately respond with EOT when an internal error occurs. This occurs only if the error condition arises during the printing or punching of the last record in a received message block. The 2780, upon verifying that the last record in a received block is free of error, responds with the appropriate acknowledgment to the CPU and overlaps the processing of the last record with subsequent transmission. If an internal error occurs during the processing of the last record, the 2780 responds to the subsequent transmission with an EOT. The operator can intervene to process the record on which the error actually occurred without retransmission of that record. Thus, problem program restart and recovery procedures can begin by retransmitting the subsequent transmission.

A description of internal errors and operator procedures may be found in the publication IBM 2780 Data Transmission Terminal -- Component Description, GA27-3005.

When the 2780 is transmitting to the CPU, special use is made of the ENQ character to denote the occurrence of a 2780 internal error during the transmission. If the 2780 transmits a message of the form:

STX---Text--US--Text--ENQ or
STX---Text--ENQ

it indicates that a buffer parity or overrun error occurred in the message being transmitted by the 2780. Such a message is invalid, and BTAM ERP automatically responds with an NAK (negative acknowledgment) to the 2780. The 2780, in turn, transmits the EOT character to the CPU, indicating that transmission is aborted until operator intervention corrects the error condition.

If the 2780 transmits a message of the form:

STX ENQ (no text)

it indicates that an internal error occurred in the processing of a record being readied for transmission to the CPU. The exchange of NAK (CPU-to-2780) and EOT (2780-to-CPU) proceeds as discussed in the preceding paragraph.

BTAM posts completion code 41 and sets bit 6 of the error information field and bit 1 of the flag byte in the DECB to inform the user of the error condition and the reception of EOT. Subsequent transmission by the 2780, after contact is resumed between the CPU and the 2780, normally begins with the record on which the error occurred.

Component Selection

Point-to-point communication with the 2780 may occur over switched or nonswitched networks. In point-to-point communication, the programmer must perform component selection when transmitting messages to the 2780.* Component selection is accomplished by a 2-character escape sequence appearing at the beginning of a nontransparent record. To select the punch, ESC, 4 is sent; to select the printer, any one of the vertical forms control escape sequences may be sent (e.g., ESC /, ESC A, ESC B). Thus, the vertical forms control escape sequences perform the dual function of selecting the printer and providing control of forms motion for the record blocks in which they appear. Once a component is selected it need not be reselected in successive record blocks intended for it. Because component selection occurs only in nontransparent mode, the first message in a transmission to the 2780 cannot be in transparent mode. Figure 49 illustrates component selection.

Point-to-Point Contention

In a contention system, either the CPU or the 2780 may initiate transmission by sending the inquiry signal (ENQ) over the line. When a true contention situation occurs (both the CPU and the 2780 are bidding for control of the line), the 2780 must be declared the master station. For this reason, the programmer must code SLV in the

----

*That is, if the Mode Switch on the 2780 is on Transmit or Receive, component selection is mandatory. If the Mode Switch is on Print or Punch, component selection is ignored and all messages received by the 2780 are automatically printed or punched. The Mode Switch on the 2780 is manually set by the operator.

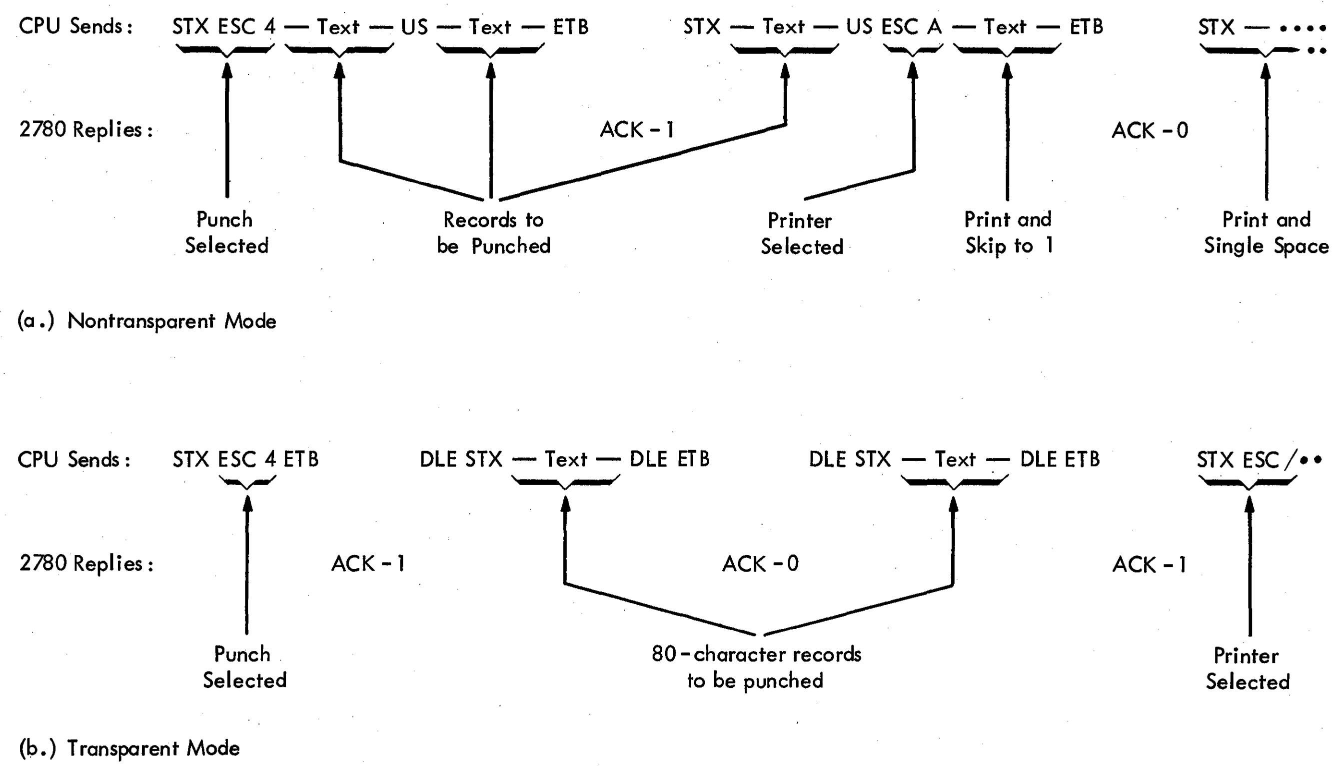


Figure 49. Examples of Component Selection

FEATURE operand sublist in the DTFBT for the line. Thus, when an ENQ is received by the CPU in reply to an ENQ transmitted to the 2780 via a WRITE initial, BTAM posts completion code 50 in the ECB and returns control to the user. By then issuing a READ initial, the user can receive from the remote 2780.

If the CPU successfully gains control of the line (ACK-0 received as reply to ENQ), the first message with the component selection sequence is sent to the 2780. Figure 45 illustrates typical transmission sequences.

When the 2780 is the transmitting station, it sends ETX (or DLE,ETX) at the end of the last record block transmitted. Upon receiving a positive reply it transmits EOT.

When the 2780 is equipped with the Automatic Turnaround feature, the card punch automatically obtains ready status after the card reader reads a blank card. This permits the CPU (upon receiving EOT) to bid immediately for the line and select the punch for a punching operation. Without this feature, selection of the card punch following a card reader operation requires operator intervention. Operator intervention is not required for the printer to be selected following a card reader operation.

Figure 50 lists the READ, WRITE, and CONTROL options available for communication with the 2780 in a contention system. A detailed explanation of channel programs may be found in the Point-to-Point Contention under BSC Channel Programs section.

Point-to-Point Dial

Transmission over a switched network is initiated when either the CPU or the operator at the remote 2780 performs a calling operation. For this reason both answering (via READ Initial) and calling (via READ Initial or WRITE Initial) operations are available for communication with the 2780 over a switched network. If ID verification is used, see the Dial section under System/370-to-System/370 for a description of ID verification procedures. The user should note that the ID received by the 2780 will send back a fixed ID sequence, which is established at installation of the 2780.

The 2780 normally does not transmit the disconnect signal (DLE, EOT) to the CPU. This permits the problem program to determine when the connection is to be broken. Thus, when the 2780 is the transmitting

Option	Type Code	Point-to-Point		Multipoint
		Contention	Dial	
READ Initial	TI	X	X	X
READ Initial Inquiry	TIQ	X		
Read Connect	TIC		X	
Read Connect with Tone	TIW		X	
READ Initial with Tone	TIZ		X	
READ Continue	TT	X	X	X
READ Interrupt	TRV	X	X	X
READ Repeat	TP	X	X	X
READ Inquiry	TQ	X	X	X
WRITE Initial	TI	X	X	X
Write Connect	TIC		X	
*WRITE Initial Transparent Block[1]	TIE			X
*WRITE Initial Transparent Text[1]	TIX			X
WRITE Continue	TT	X	X	X
*WRITE Transparent Block[1]	TE	X	X	X
*WRITE Transparent Text	TX	X	X	X
WRITE End-of-Transmission	TR	X	X	X
WRITE Wait-before-Transmitting[1]	TW	X	X	X
WRITE Inquiry	TQ	X	X	X
WRITE Disconnect	TD		X	
CONTROL Disable	TD		X	
CONTROL Mode	TM	X	X	X

*Available for EBCDIC transmission code only.
[1]Not available for Transcode transmission code.

Figure 50. READ/WRITE/CONTROL Options for 2780

station, it sends EOT after receiving a positive reply to its last message bit transmitted. The problem program may then send the disconnect signal or bid for the line (via WRITE Inquiry) to subsequently transmit to the 2780. When the problem program sends EOT on a dial line, the 2780 reacts in either of two ways:

- If the card reader is ready to send, the 2780 bids for the line by sending ENQ to the CPU. BTAM posts a completion code of 7F for this case. The user may then follow the completed WRITE TR by a READ Continue (TT) if the program is ready to receive.

- If the card reader is not ready to send, the 2780 will not respond to the EOT. Instead, the 2780 continues to monitor the line after receiving the EOT. After a nominal twenty seconds, the 2780 (with the Auto Answer feature) will go "on-hook" if no bid (ENQ) is received from the CPU.

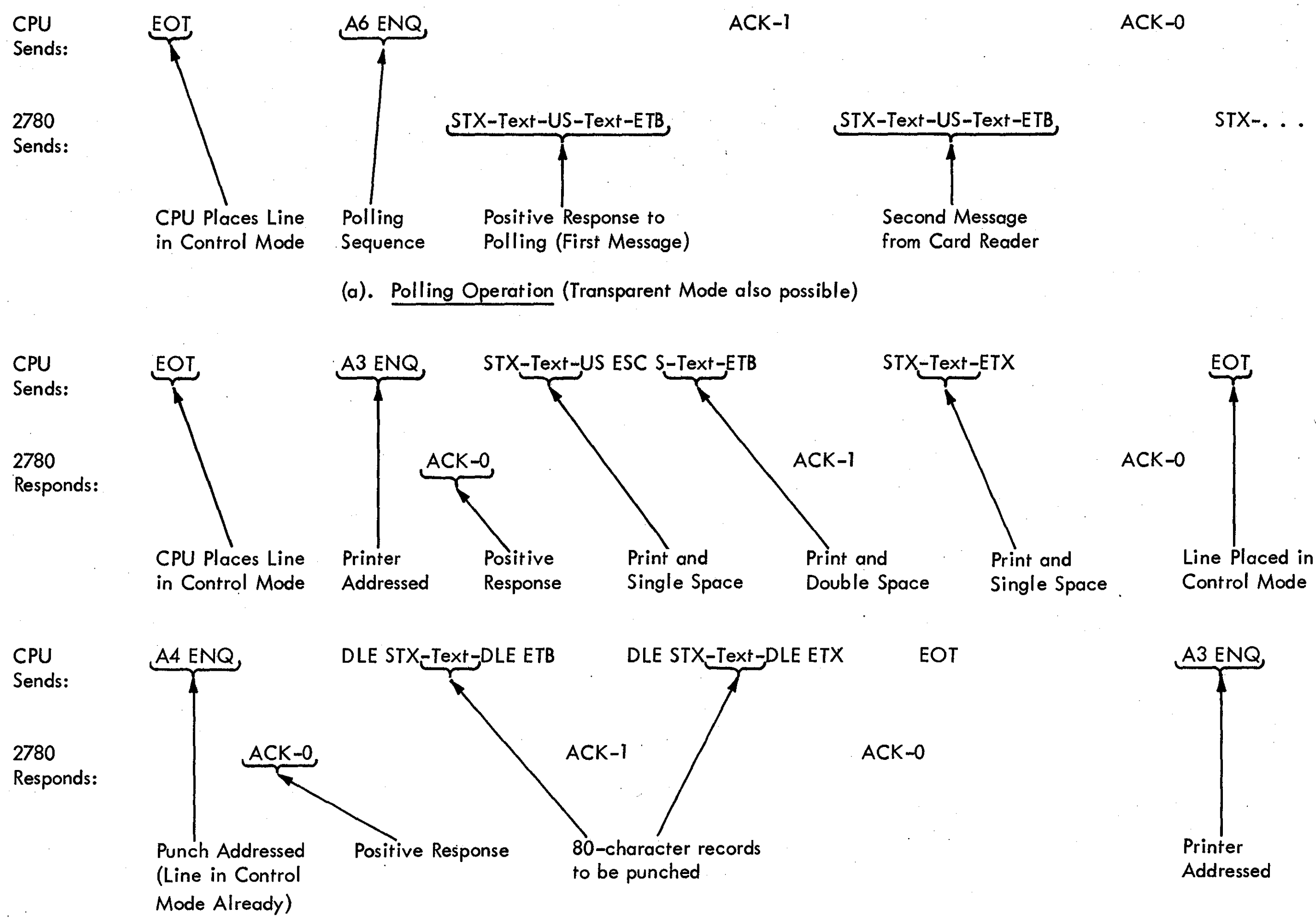


Figure 51. Polling/Addressing Operations for 2780

Once a line connection is established between the CPU and a 2780 over a switched network, message transmission proceeds in the same fashion as described for a contention system. Component selection and auto turnaround operate the same in both systems. The READ/WRITE/CONTROL options for a dial operation are listed in Figure 50. A complete description of the channel programs may be found in the section Point-to-Point Dial under BSC Channel Programs.

Multipoint Communication

Transmission of messages from or to a particular 2780 on a multipoint line is initiated by the CPU, which places the line in control mode and then sends the appropriate polling or addressing sequence (Figure 51).

Message transmission from a remote 2780 to the CPU is initiated by the CPU through a READ Initial macro instruction. Execution of the READ Initial channel program results in the sending of a polling sequence by the CPU and a subsequent response by the polled 2780. Polling is accomplished using Auto Poll-automatic negative response polling. A negative response to polling does not cause a CPU interruption until:

1. The last entry in the AUTOLST type terminal list is polled.

2. A RESETPL macro is issued by the problem program and the last entry in the AUTOWLST type terminal list is polled.

Name	Operation	Operand
symbol	DFTRMLST	listtype{AUTOLST / AUTOWLST}, eotsyn-hexchars, {pollchars-hexchars,}...

In issuing the DFTRMLST macro in the form shown, the user is required to specify, via eotsyn, the bit configurations of the EOT and SYN characters in the transmission code used. These are:

Code	Bit Configuration of EOT,SYN (in hex)
EBCDIC	X'3732'
USASCII	X'0416'
TRANSCODE	X'1E3A'

For the polling sequence, the user specifies the hexadecimal equivalent of a 3-character sequence of the form:

x 6 ENQ

where the x may be any uppercase or lowercase character used as the station address. This is set at the time of installation of the 2780. The second character is always a 6 to indicate the card reader. Up to 98 entries may be specified in the terminal list. The list generated by BTAM is shown in Appendix A.

Message transmission from the CPU to a multidropped 2780 is initiated by the CPU through an initial WRITE (TI, TIE, TIX) macro instruction. Execution of one of these WRITEs results in the sending of an addressing (selection) sequence and subsequent transmission of the first message upon receiving a positive response to selection from the appropriate 2780. Component selection in the form used for point-to-point communication is not operative on a multipoint line. Therefore a component (printer or punch) must be selected, via an addressing sequence, prior to any transmission of messages to it.

Vertical forms control, however, does function in precisely the same fashion on both point-to-point and multipoint connections.

The addressing sequences are specified by the user through a DFTRMLST macro instruciton. The address of the first character of the addressing sequence is specified by the entry operand of the WRITE macro.

Name	Operation	Operands
symbol	DFTRMLST	OPENLST,({addrchars-hexchars,}...)

An addressing sequence for the 2780 consists of a 2-character sequence followed by ENQ, where:

x 3 ENQ = Addressing sequence for the printer

x 4 ENQ = Addressing sequence for the punch

Again, the x may be any uppercase or lowercase character corresponding to the station address set at installation time.

The DOS assembler restriction of 127 characters maximum in a sublist limits the number of addrchars operands coded in a single DFTRMLST macro to a maximum of eighteen (because 2 hexadecimal digits are required to represent a single addressing character).

IBM 1130 COMPUTING SYSTEM

BTAM supports binary synchronous communication with the remote IBM 1130 Computing System for both point-to-point and multipoint line connections. The remote 1130 may be connected point-to-point over either a switched or nonswitched network.

POINT-TO-POINT COMMUNICATION

The BTAM support for communication with the 1130 on a point-to-point line is a compatible subset of the BTAM support for CPU-to-CPU communication. Communication with the 1130 does not permit the following:

1. Conversational communication.

2. USASCII transmission code (EBCDIC must be used).

Figure 52 (READ/WRITE/CONTROL options for 1130) lists the options available for communication with the 1130 over both nonswitched and switched networks. The channel programs generated and executed are described in the sections on Point-to-Point Contention and Point-to Point Dial. If ID verification is used, see the Dial section under System/370-to-System/370 for a description of ID verification procedures.

MULTIPOINT COMMUNICATION

The CPU initiates transmission of messages from or to a particular 1130 on a multipoint line by placing the line in control mode and then sending the appropriate polling or addressing sequence to the 1130.

Message transmission from a remote 1130 to the CPU is initiated by the CPU through a READ Initial macro instruction. This results in the sending of a polling sequence by the CPU and a subsequent response by the polled 1130. Polling is accomplished using Auto Poll (automatic negative response polling). No CPU interruption occurs on a negative response to polling until:

1. The last entry in the AUTOLST type terminal list is polled.

2. A RESETPL macro instruction is issued by the problem program and the last entry in the AUTOWLST type terminal list is polled.

Name	Operation	Operands
symbol	DFTRMLST	listype-{AUTOLST | AUTOWLST},
		eotsyn-hexchars,
		{pollchars-hexchars,}...

The user specifies the polling characters to be used by coding a DFTRMLST macro. The listype specified may be either AUTOLST or AUTOWLST, depending upon whether the list generated is to be an open or a wraparound polling list, respectively.

The eotsyn operand is coded as an absolute expression representing the EBCDIC bit configuration of the EOT and SYN control characters. For example, X'3732' is the hexadecimal equivalent of the EBCDIC EOT and SYN (37 for EOT and 32 for SYN) characters. Any absexp (hexadecimal, decimal, binary, etc.) coded is interpreted by BTAM to represent two bytes: the first byte representing EOT, the second, SYN. The eotsyn operand is required by BTAM for internal use in generating the list.

The particular entries to be generated in the terminal list are specified via one or more (maximum of 98) pollchars operands. Each pollchars operand represents a polling sequence for a specific 1130 multidropped on the line. A polling sequence for the 1130 is defined as one EBCDIC polling character followed by the ENQ control character. Thus, in coding the pollchars operand, the user would code 4 hexchars of the form xxyy, where xx represents the hexadecimal equivalent of the polling character and yy is always 2D (the hexadecimal equivalent of ENQ in EBCDIC). The polling character used for a particular 1130 depends upon the character assigned by the problem program in that 1130 as the polling character.

Appendix A shows the list generated for a DFTRMLST macro having AUTOLST or AUTOWLST coded. Note that the list has one entry for each pollchars operand coded. In addition to the characters specified in a pollchars operand, a list entry includes an automatically-generated index byte.

The index byte can be used by the problem program to identify which entry in the list corresponds to the device that responds to polling with a message. The first byte of the user-specified input area (READ Initial only) will contain the index value of the list entry for the device that sends a message. The received message will appear in the input area immediately following the index. For those applications involving message switching, the user should move the SYN (synchronous idle) character into the byte containing the index value prior to switching a received message to some other remote device. This is particularly important when the user switches a message received into buffers, because in this case the output area address is the same as the input area address. If the user transmits the index byte, an error will occur.

If the polling list is exhausted (end of the list is reached and no wraparound is to occur) and all polled devices have devices

Option	Type Code	1130 Point-to-Point		1130 Multipoint
		Contention	Dial	
READ Initial	TI	X	X	X
READ Initial Inquiry	TIQ	X		
Read Connect	TIC		X	
Read Connect with Tone	TIW		X	
READ Initial with Tone	TIZ		X	
READ Continue	TT	X	X	X
READ Repeat	TP	X	X	X
READ Inquiry	TQ	X	X	X
READ Interrupt	TRV	X	X	X
WRITE Conversational	TV	X	X	X
WRITE Initial	TI	X	X	X
Write Connect	TIC		X	
WRITE Initial Transparent Block	TIE	X	X	X
WRITE Initial Conversational	TIV	X	X	X
WRITE Initial Transparent Conversational	TIXV	X	X	X
WRITE Initial Transparent Text	TIX	X	X	X
WRITE Continue	TT	X	X	X
WRITE Transparent Block	TE	X	X	X
WRITE Transparent Conversational	TXV	X	X	X
WRITE Transparent Text	TX	X	X	X
WRITE Disconnect	TD		X	
WRITE End-of-Transmission	TR	X	X	X
WRITE Wait-Before-Transmitting	TW	X	X	X
WRITE Inquiry	TQ	X	X	X
CONTROL Disable	TD		X	
CONTROL Mode	TM	X	X	X

Figure 52. READ/WRITE/CONTROL Options for 1130

have responded negatively, no index byte will appear in the input area. Instead, completion code 54 identifies this case.

Message transmission from the CPU to a remote 1130 is initiated by the CPU through a WRITE Initial macro instruction, which causes an addressing sequence to be sent selecting the appropriate 1130. If the selected 1130 responds positively to addressing, the message is transmitted from the CPU to the 1130.

The user specifies the addressing sequences through a DFTRMLST macro instruction.

Name	Operation	Operands
symbol	DFTRMLST	OPENLST, (addrchars-hexchars,...)

An addressing sequence for the 1130 consists of one addressing character followed by ENQ. Each addressing sequence is specified through an <u>addrchars</u> operand in the sublist as shown in the format illustration above. Each <u>addrchars</u> is coded by the user as a <u>hexchars</u> sequence of the form xxyy, where xx represents the hexadecimal equivalent of the addressing character and yy is always 2D (the hexadecimal equivalent of the ENQ character).

The DOS Assembler restriction of 127 characters maximum in a sublist limits the number of <u>addrchars</u> operands coded in a DFTRMLST macro to a maximum of 25. The format of an OPENLST type terminal list is shown in Appendix A.

<u>Note</u>: The address of an 1130 is set by the problem program in the 1130.

IBM SYSTEM/3

BTAM supports binary synchronous communication with remote IBM System/3 RPG II support for point-to-point and multipoint lines. Both EBCDIC and ASCII transmission codes are available to System/3 RPG II support. Code charts are given in Appendix F.

POINT-TO-POINT COMMUNICATION

The System/3 point-to-point configuration may be either switched or non-switched lines. BTAM support is identical to the support for CPU-to-CPU, with certain limitations. These limitations are discussed below under Programming Considerations. Functions supported for non-switched lines are the same as those described in the CPU-TO-CPU Dial section.

MULTIPOINT COMMUNICATION

The CPU initiates transmission from or to a tributary System/3 by placing the line in control mode and then sending the appropriate polling or addressing sequence. The polling and addressing sequence sent to the System/3 must consist of two identical non-control characters preceding ENQ. Polling and addressing characters must be paired as discussed in the Double Addressing section.

PROGRAMMING CONSIDERATIONS

IBM System/3 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

- Transparent, ITB:

  DLE-STX-data-ITB-DLE-STX-data-ITB-DLE-STX-data-ETX or ETB.

Data can be either blocked or unblocked but must be of fixed record length. Fixed record length and unblocked implies non-ITB mode and requires that all blocks of data between ITBs be of the same length.

Conversational Mode

IBM System/3 RPG II support allows only one conversational response to data sent by the System/3. Also, System/3 will send only one conversational response to data sent to System/3. To maintain proper line discipline, the System/3 will accept or send a NULL message (an STX-ETX sequence), in lieu of a data transfer following a conversational response.

Error Retry Count

The System/3 error recovery retry count is a fixed value of seven for all retries except autocall sequences. The autocall retry count is determined by the operator.

Leading Graphics

System/3 will accept leading graphics preceding acknowledgments, but will not send leading graphics preceding acknowledgments. Leading graphics received by System/3 are ignored.

Note: Figure 53 (READ/WRITE/CONTROL Options for Model 20 and System/3) lists the available options for communications with System/3 over both switched and non-switched networks.

SYSTEM/360 MODEL 20

BTAM supports binary synchronous communication with the remote System/360 Model 20 for both point-to-point and multipoint line connections. The Model 20 may be connected point-to-point over a either a switched or nonswitched network.

There are two available transmission codes for the Model 20: EBCDIC and USASCII. Code charts for these codes are shown in Appendix F.

The 2-character data link control functions in the above codes vary in the second character used, as shown in Appendix F.

POINT-TO-POINT COMMUNICATION

The BTAM support for communication with the System/360 Model 20 on a point-to-point line is identical to the BTAM support for CPU-to-CPU communication, if the Model 20 is considered as another CPU. The functions supported for nonswitched lines are the same as those described in the CPU-to-CPU Contention section, and the functions supported for dial lines are the same as those described in the CPU-to-CPU Dial discussion. Figure 53 (READ/WRITE/CONTROL Options for Model 20) lists the options available for communication with the Model

20 over both switched and nonswitched networks.

Note: If a BTAM problem program receives a transparent block of data from the Model 20, text cannot be transmitted as a response.

MULTIPOINT COMMUNICATION

The CPU initiates transmission of messages from or to a specific Model 20 on a multipoint line by placing the line in control mode and then sending the appropriate polling or addressing sequence to the Model 20. The polling or addressing sequence that is sent to the Model 20 may consist of a maximum of seven non-control characters preceding the ENQ character, but only the first character is used by the Model 20 hardware. Any additional characters in the polling or addressing sequence are used informationally by the Model 20 problem program and hence are optional. Since double addressing is provided in the Model 20 program, the first two characters in the polling or addressing sequence must be identical.

OPTION	TYPE CODE	CPU-Model 20 Contention	CPU-Model 20 Dial	CPU-Model 20 Multipoint
Read Initial	TI	X	X	X
Read Initial Inquiry	TIQ	X		
Read Connect	TIC		X	
Read Connect with Tone	TIW		X	
Read Initial with Tone	TIZ		X	
Read Continue	TT	X	X	X
Read Continue with Leading Graphics	TTL	X	X	X
Read Repeat	TP	X	X	X
Read Repeat with Leading Graphics	TPL	X	X	X
Read Inquiry	TQ	X	X	X
Read Interrupt	TRV	X	X	X
Write Initial	TI	X	X	X
Write Connect	TIC		X	
Write Initial Transparent Block	TIE	X	X	X
Write Initial Transparent Text	TIX	X	X	X
Write Initial Transparent Conversational	TIXV	X	X	X
Write Initial Conversational	TIV	X	X	X
WRITE Continue	TT	X	X	X
Write Transparent Block	TE	X	X	X
Write Transparent Text	TX	X	X	X
Write Transparent Conversational	TXV	X	X	X
Write Conversational	TV	X	X	X
Write Disconnect	TD		X	
Write End-of-Transmission	TR	X	X	X
Write Inquiry	TQ	X	X	X
Control Disable	TD		X	
Control Mode	TM	X	X	X
Write Wait-Before-Transmitting	TW	X	X	X

Figure 53. READ/WRITE/CONTROL Options for System/360 Model 20 and System/3

IBM 2972 GENERAL BANKING TERMINAL SYSTEM

BTAM supports binary synchronous communication with the remote IBM 2972 for multipoint line connections only. The 2972 uses a subset of the EBCDIC transmission code. A chart of this code is shown in Appendix G.

MULTIPOINT COMMUNICATION

The CPU initiates transmission of messages from or to a specific 2972 on a multipoint line by placing the line in control mode and then sending the appropriate polling or addressing sequence to the 2972. The polling or addressing sequence that is sent to the 2972 may consist of a maximum of seven characters preceding the ENQ character. Double addressing (first two characters the same) is mandatory for the 2972.

Read, Write, and Control options available for communication with the 2972 are shown in Figure 54.

OPTION	TYPE CODE
READ Initial	TI
READ Interrupt	TRV
READ Continue	TT
READ Continue with Leading Graphics	TTL
READ Repeat	TP
READ Repeat with Leading Graphics	TPL
READ Inquiry	TQ
WRITE Initial	TI
WRITE Initial Conversational	TIV
WRITE Continue	TT
WRITE Conversational	TV
WRITE End-of-Transmission	TR
WRITE Inquiry	TQ
WRITE Wait-Before-Transmitting	TW
CONTROL Mode	TM

Figure 54. READ/WRITE/CONTROL Options for 2972

IBM 2790 DATA COMMUNICATIONS SYSTEM

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 the IBM 2790 Component Description manual, GA27-3015.

The configurations supported by BTAM for the 2715 are:

- Point-to-point nonswitched lines.
- Point-to-point switched lines.
- Multipoint nonswitched lines.

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 55.

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 DOS assembler, generate 21 types of tables. These tables contain pointers and index values, as well as parameters used by the microcoded routines. The tables are sent to the 2715 by the user's BTAM program. Use of the DOS Assembler D requires the 14K version.

A sample 2790 program is provided in Appendix N.

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.

MACRO	Type Code	Point-to-Point Contention and Local	Point-to-Point Switched	Multipoint
READ Initial	TI	X	X	X
READ Initial Inquiry	TIQ	X		
READ Continue	TT	X	X	X
Read Connect	TIC		X	
Read Connect with Tone	TIW		X	
READ Continue with Leading Graphics	TTL	X	X	X
READ Repeat	TP	X	X	X
READ Repeat with Leading Graphics	TPL	X	X	X
READ Inquiry	TQ	X	X	X
READ Interrupt	TRV	X	X	X
WRITE Initial Transparent Text	TIX	X	X	X
WRITE Initial Transparent Block	TIE	X	X	X
WRITE Initial Transparent Conversational	TIXV	X	X	X
WRITE Connect	TIC		X	
WRITE Transparent Text	TX	X	X	X
WRITE Transparent Block	TE	X	X	X
WRITE Transparent Conversational	TXV	X	X	X
WRITE Wait-Before-Transmit	TW	X	X	X
WRITE Disconnect	TD		X	
WRITE End-of-Transmission	TR	X	X	X
WRITE Inquiry	TQ	X	X	X
CONTROL Mode	TM	X	X	X
CONTROL Disable	TD		X	

Figure 55. READ/WRITE/CONTROL Options for 2715

- 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.

Count 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 12 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 re-initialize 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 scheduleon an undividual 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 enbble 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 re-initialize 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. 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 through 80 of a card or 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 and 80 of a card or columns 19 through 22 of a badge. For the 2795, the user must specify the starting and ending counter addresses in columns 77 and 80 of a card or columns 19 through 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 core) 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

The assembly output of the user macros is an object deck of tables necessary for 2715 internal operation. 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 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 Sequence Table
- GDU Area Station Table
- GDU Sequence Table
- Identification Table
- Translate Table

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 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 maximum size of the Counter Table is 2048 bytes: 2 bytes for each counter plus 2 additional bytes of control information. The Counter Table is defined by CTRGROUP macro instructions. Each CTRGROUP 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.

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. 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 will contain a transaction header, a variable number of GDU steps (up to 16), a byte of zeros, 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 GDULIST 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.

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.

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 56). The user must provide DLE STX in front of the message; BTAM provides ending characters (Figure 56). All cards must be transmitted, one at a time, to the 2715.

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 "step 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:

D L E	S T X	Message Header 2 bytes	Transaction Header 8 bytes	Object Card 80 bytes

Figure 56. Format for User Transmission of Table Object Cards to the 2715

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)
- CTRGOUP (optional)
- CTRSCHED (optional)
- GDUTRANS (optional)
- PARAMNUM (optional)
- PARMLIST (optional)
- DISPGUID (optional)
- TRANSLAT (optional)
- TRLIST
- ASLIST
- DEULIST (optional)
- CTRLIST (optional)
- GDULIST (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 counter 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	[,CORE=$\left\{\begin{matrix}\underline{16}\\32\end{matrix}\right\}$][,PC=$\left\{\begin{matrix}\underline{NO}\\YES\end{matrix}\right\}$] [,GDU=$\left\{\begin{matrix}\underline{NO}\\YES\end{matrix}\right\}$] [,FUNCERR=(absexp,...)] [,ENDERR=(absexp,...)] [,MONERR=(absexp,...)] [,GETID=absexp] [,STORID=absexp] [,IDCOUNT=absexp] [,INQDISP=$\left\{\begin{matrix}\underline{NO}\\YES\end{matrix}\right\}$]

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 57. The size will be the total of all macros used.

PC
: 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.

FUNCERR
: 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.
- 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 60.)

ENDERR
: 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 check specified by the PARMLIST 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 60.)

Macro	Minimum 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 CKLNGTH=. 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 57. 2715 Macro storage Size Estimates

MONERR
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 60.)

GETID
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 61.)

STORID
specifies the Store identification character that is used for the Store Identifier function. The value of this operand is the hexadecimal equi-

valent of any of the characters: A-Z, 0-9, and any of the special characters: . $ & - / , # @ " = : ? ! ; * + TAB NEWLINE SPACE LINEFEED. (See Figure 61.)

IDCOUNT
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
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 DEGROUP operand). The format of the AS macro is:

Name	Operation	Operand
[symbol]	AS	ID=absexp[,ASGROUP= symbol] [,DEGROUP=(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 58). 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.

ASGROUP
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 is used to build an entry in the AS table.

DEGROUP
symbol
the name of the transaction group table 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" is used to build 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 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 54 for examples.)

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 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.

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 58. AS and GDUAS Macro ID Parameter Decimal and Hexadecimal Equivalents

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 58). 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
: 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 is used to define 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 name field is required for all TGROUP macros. 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. and the name field of the second TGROUP macro is required. Furthermore, the first operand of both TGROUP macros must be identical. (Only one operand need be coded for the first macro.)

Figure 59 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.

Name	Operation	Operands
* DEFINING TRANSACTION GROUPS FOR BOTH 2795 AND 2796 DATA ENTRY UNITS		
	AS	ID=59,DEGROUP=(GROUP1,5) (TOTAL OF 5 DEUs)
GROUP1	TGROUP	TC1=ATTENDNC,TC2=SETUP,TC3=PRODN,... (2795s)
GROUP2	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2796s)
* DEFINING TRANSACTION GROUP TABLE FOR 2796 AND DUMMY TRANSACTION GROUP TABLE FOR 2795		
	AS	ID=59,DEGROUP=(GROUP1,2) (TOTAL OF 2 2796s)
GROUP1	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2795s)
GROUP3	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2796s)
* DEFINING TRANSACTION GROUP TABLE FOR 2975 ONLY		
	AS	ID=59,DEGROUP=(GROUP1,3) (TOTAL OF 3 2795s)
GROUP1	TGROUP	TC1=ATTENDNC,TC2=SETUP,TC3=PRODN,... (2795s)
* DEFINING TRANSACTION GROUP FOR 2797 AND DUMMY TRANSACTION GROUP FOR 2795 AND 2796		
	AS	ID=59,DEGROUP=(GROUP1,3) (TOTAL OF 3 2797s)
GROUP1	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2795s)
GROUP2	TGROUP	TC1=MESSAGE (DUMMY TABLE FOR 2796s)
GROUP3	TGROUP	TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,...(2797s)

Figure 59. Examples of Defining Transaction Group Tables for Data Entry Units

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 add 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 referef 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. The maximum size of the Counter Table is 2048 bytes. 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-absexp, [sro-absexp], [cttest-absexp], 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 <u>not</u> 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 prodiction 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=RESET 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-absexp,...

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
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
: 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
: 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.

PARMLST
: 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 PARMLST 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	[CKLNGTH=(length-absexp, errguidance-absexp,...)] [,CKMONKY= $\left\{\begin{matrix}\underline{NO}\\ YES\end{matrix}\right\}$] [,CKMOD11=(length-absexp,position-absexp, errguidance-absexp, ...)] [,CKRANGE= (position1-absexp, position2-absexp, hilowchars-absexp,...)] [,LOWGUID=(absexp,...)] [,HIGUID=(absexp,...)] [,RNGETST= $\left\{\begin{matrix}\underline{ERROR}\\ DATA\end{matrix}\right\}$] [,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, ...)] [,TRANSL= $\left\{\begin{matrix}\underline{NO}\\ YES\end{matrix}\right\}$] [,IDENT= $\left\{\begin{matrix}\underline{NO}\\ YES\end{matrix}\right\}$]

symbol
: the name field must be specified and must be the same name as defined by the PARMLST operand in the PARAMNUM macro.

CKLNGTH
: 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.

  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 specifies the length any characters to be entered from the GDU keyboard he must assign a value of 1 to the length suboperand.

errguidance
specifies the error guidance that is returned to the 2798 GDU if the data entry 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 60.)

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.

CKMONKY
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.

CKMOD11
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 Component Description: IBM 2790 Data Communication System, GA27-3015.

length
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.

position
specifies the starting position of the modulus 11 check field. The value of this suboperand may be between 2 and 16.

errguidance
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 60.)

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.

CKRANGE
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.

position1
specifies the starting position of the field for which the range check is performed. The value may be from 2 to 17.

position2
specifies the last position of the field for which the range check is performed. The value may be from 2 to 17.

hilowchars
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 turned 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 exists. (See Figure 60.)

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 60.)

Example: From the example associated with the CKRANGE operand, the user may code HIGGUID=(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
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
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
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
specifies the starting position of the modulus 10 check field. The value of this suboperand may be from 2 to 16.

errguidance
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 60.)

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
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 61.)

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
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 60.)

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
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
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
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 character 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 61.)

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
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 61.)

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
causes the 2715 to check a specified field to ensure that no numeric characters are received.

position1
specifies the starting position of the field to be checked. The value may be from 2 to 17.

position2
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
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 60.)

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 nonnumeric. 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
causes the 2715 to check a specified field to ensure that all numeric characters are received.

position1
specifies the starting position of the field to be checked.The value may be from 2 to 17.

position2
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
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 60.)

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.

Left Panel	Right Panel
1	9
2	10
3	11
4	12
5	13
6	14
7	15
8	16

Figure 60. 2798 GDU Guidance Panels

TRANSL
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 not be used. This operand cannot be used if any other operand in the PARMLIST macro except CKMONKY is used. The TRANSL and IDENT operands are mutually exclusive.

Keyboard Character	Hexadecimal Equivalent	Keyboard Character	Hexadecimal 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 61. 2798 GDU Keyboard Character Conversion

IDENT
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 CKLNGTH 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 CKLNGTH 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' [,SUPPRES= {YES / NO}]

symbol
the name field of this macro is optional.

DISPMSG
defines a user specified display guidance message. The text must not exceed 16 characters.

SUPPRES
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
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 61.)

TRANTXT
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 has 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, the Data Entry Unit List macro, and the GDU 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. When a transaction is initiated by a 2798 GDU, the Transaction List macro is followed by one or more GDULIST macros. 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, GDULIST, 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 list and can be referred to by the GDUTRANS macro.

Message routing can be specified (NULL or absexp1 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 the 1053 printer on an area station. The only ways the user can specify message routing are to code ROUTE=NULL or ROUTE=absexp1 in the TRLIST macro. Coding ROUTE=DISK, CPU, or LOG does not imply message routing, but that the transaction is to be routed to the specified ROUTE parameter. Transaction routing does <u>not</u> 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={<u>NO</u> | YES}] [,INQDISP={<u>NO</u> | YES}] [,DEMOD10={<u>NO</u> | YES}] [,DEMOD11={<u>NO</u> | YES}] [,GDU={<u>NO</u> | 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 127 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
: 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 respresntation of the address of an area station (see Figure 53). 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
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 INQDIST=YES requires one extra GDULIST macro to end this transaction list. See GDULIST macro description for details.

DEMOD10
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 DEMOD10=YES specifies that modulus self checking will be performed on a data entry in this transaction. Coding DEMOD10=YES requires CORE=32 to be coded in the CONFIGUR macro. This operand is mutually exclusive with the DEMOD11 operand. This operand does not apply to the 2798 GDU.

DEMOD11
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
allows for 100 additional transactions 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 nonunique 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 transaction and the associated GDULIST macros should be the last macros coded before STEND. (See Appendix N.)

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={1 / M}] [,MSG='text'] [,INQDISP=absexp] [,MODULUS=(absexp1, absexp2,absexp3)] [,SELTRAN={NO / YES}]

symbol
the name field of this macro instruction is optional.

device-code
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
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 62). (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 62.)

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.

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 55.)

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
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
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
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.

absexp3
specifies which guidance light is turned on if the modulus check fails. This value must be from 1 to 31.

Note: The MODULUS operand cannot be coded unless DEMOD10=YES is coded in the TRLIST macro. This operand is mutually exclusive with the DIGIT operand.

SELTRAN
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 transaction. 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 cannot 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:

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 TRANSACTION
ONLINE	REPEAT CLEAR	IN PROCESS	CARD IN

Figure 62. Matrix of ASLIST Operand Values for Guidance Lights by Position on the Area Station

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, thered 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+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
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 to 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. The value can be from 1 to 15. This length does not include the self-check character.

DIGIT2
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
specifies 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 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 coded 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']        │
│         │          │ [        {1}]        │
│         │          │ [,ENTRY= {M}]        │
└─────────┴──────────┴──────────────────────┘
```

symbol
: the name field in this operand is optional.

PARAMNO
: 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
: 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
: 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
: 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 this 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 DISPMSG 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.

<u>Example</u>: 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).

<u>Note</u>: The user is made aware of the fact that he has not stored any text in a particular Identifier since he performs 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 the GDULIST macro and sees the text 'NOT USED,' he should realize that he has never stored any text in the Identifier.

ENTRY
: 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.

<u>CTRLIST (Counter List) Macro Instruction</u>

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 STRLIST 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 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/370 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-ITER KEY **	RIGHT KNOB	10 DATA	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
  TGROUP
  ASCTR (optional)
  CTRGROUP (optional)
  CTRSCHED (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 DEGROUP operand is coded in the AS macro for this area station. See Figure 54 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 or DLE ETB 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 63.

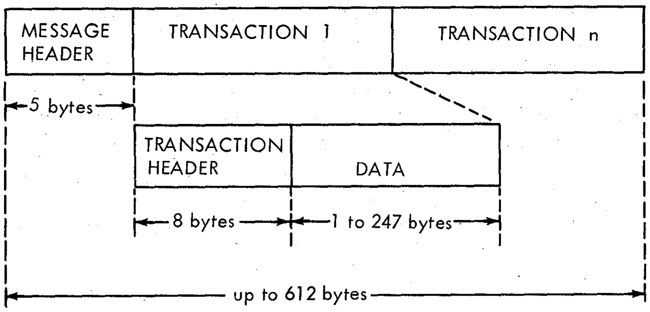
Message Header -- System/370 to 2715

The message header is two bytes and has the following format:

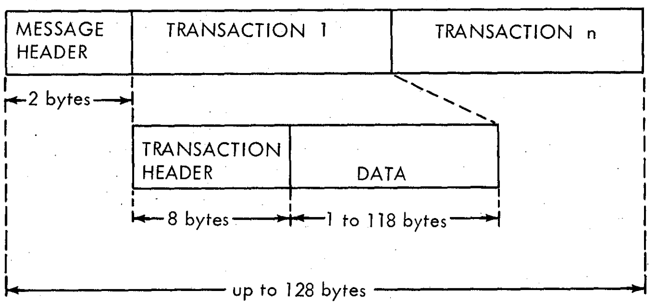
Byte 0	Message length
Byte 1	Message control byte

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.

Message Control Byte: The message control byte is used to indicate one of three possible destinations for output data, as shown in Figure 64.



2715 to System/370



System/370 to 2715

Figure 63. Message Formats

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.

MESSAGE DESTINATION	CONTROL BYTE
1053 Printer or Pulse Count	X'01'
2715 Control	X'02'
2740 Terminal	X'04'

Figure 64. System/370 to 2715 Message Control Byte Usage

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 58. If the value in a control transaction is not recognized by the 2715, a message is returned to the System/370.

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 53).

- 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 53). 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.

Message Type and Function	Control Byte	Data Bytes ($d_1,d_2,\ldots,d_n$)
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_{80}= one object card from the user's assembly of 2715 macro instructions
User table load end.	X'D3'	none
CPU restart recovers the 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 unrecoverable error or stop.	X'D5'	d_1-d_2 = deferred restart number log (in hexadecimal) d_3-d_4 = priority restart number log (in hexadecimal)

Figure 65. Transmission Control Byte Usage (Part 1 of 2)

Message Type and Function	Control Byte	Data Bytes ($d_1, d_2, \ldots, d_n$)
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
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
<u>All Pulse Count Transactions</u>	X'FC'	d_1 = counter control byte in EBCDIC
Read Group functions reads the contents or one or more counters.		d_2-d_3 = EBCDIC value of last counter in the group.
Set functions and Read and Set functions sets the counter to a predetermined count.		d_2-d_6 = EBCDIC value to which the counter is to be set
2715 to System/370 <u>Data Transactions</u>	X'00' X'7F'[1]	
<u>Control Transactions</u>		
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'F0' X'F9'	
Automatically initiated response.	X'FF'	
Unsolicited 2715 response.	X'FD'	
<u>Special Pulse Counter Transactions</u>		
Positive response to CPU request	X'CA'	d_1 = counter control request d_2 = control definition
Invalid response to CPU request.	X'CC'	same as X'CA'
Positive response to operator initiated pulse counter control request.	X'DA'	d_1 = counter request d_2 = control definition
Pulse transactions for CPU.	X'ED'	
CE-initiated response for pulse counters.	X'EF'[2]	

[1]This is the value of the transaction identifier.
[2]A CE can run diagnostics 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, the CE can issue a diagnostic code to route all the counter values to the System/370. After the CE has returned the counters to the system, they may be restored to the original values by the user program.

Figure 65. Transmission Control Byte Usage (Part 2 of 2)

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.

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 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.

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 analyzing 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. 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:

Counter Control Operation	Counter Control Byte (in Hex)
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.

- Read Group.
- Read Group Residual.
- No-Count Test Failure
- Unassigned Production Test Failure.
- Overflow Interrupt.
- Power Interrupt.

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:

Operation	Message
No-Count Test Failure	NCTF
Unassigned Production Test Failure	UPTF
Overflow Interrupt	OVFL
Power Interrupt	POWR

CONTROL TRANSACTIONS: 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
  CE-initiated response for pulse counters.

In addition to the above transactions, there are two types of messages that are transparent to the user (i.e., non-user data).

1. Error records are recorded by BTAM on a disk file; and
2. Diagnostic information (automatic or resulting from Customer Engineer 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 N-Count
- Set Counter and Reset No-Count Testing and Set Unassigned Production Testing
- Set Counter and Reset all count testing function
- Read 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 66.

Overflow Interrupt

An overflow interrupt message is transmitted to the user-defined routing indication (specified in the ASCTR macro) 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). 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 <u>and</u> 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

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, read 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 NCP, 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 66. 2790 Pulse Counter Control Operation

defines the type of alarm message, as follows:

Control Byte Value	Alarm Message Type
X'CD'	Alarm
X'CE'	Text
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:

BID	D1	D2	D3	Text	EOT

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 normally 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 initiation, 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 The SPACE character indicates that nothing is punched in this column of the card.

Message Type	First Byte	Second Byte
Alarm	BELL	SPACE
Text	SPACE	BELL
Alarm and Text	BELL	BELL

<u>2740 TRANSACTIONS</u>: The 2740 Communication 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.

TERMINAL ERROR STATUS MESSAGE (IBM 2715)

This message provides the results of a scan of the error file in an IBM 2715 Transmission Control Unit. The message is in one of four possible formats. 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 formats shown 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 (24-hour system) 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.

4B72I cuu xx tttt yy ERS z

This message reports the results of an error scan by the 2715 when five or more ofthe 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).

No operator response is necessary for this error.

4B72I cuu xx tttt THRESHOLD

This message indicates that the threshold value of eight has been reached for the area station whose address is xx, but no device attached to the station accounted for as many as five of the errors.

No operator response is necessary for this error.

4B72I cuu xx tttt yy eeee zzzz yy eeee zzzz yy eeee zzzz yy eeee zzzz

This message is issued whenever an error scan for a particular area station is manually requested at the 2715, 2740 attached to the 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 area station xx.

yy
: is the address of a device (hexadecimal).

eeee
: is the error data for device yy (hexadecimal).

zzzz
: is the time (24-hour system) that the error data was recorded on the 2715 disk (decimal).

No operator response is necessary.

4B72I cuu ww tttt eeeeeee zzzz eeeeeee zzzz eeeeeee zzzz eeeeeee zzzz

This message is issued whenever an error scan for a particular 2715 adapter is manually requested by the 2715, 2740 attached to the 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 (24-hour system) that the error data was recorded on the 2715 disk (decimal).

No operator response is necessary.

Note: The system continues processing in all cases.

IBM 2770 DATA COMMUNICATIONS SYSTEM

The IBM 2770 is a multipurpose terminal providing a selection of input/output devices and a choice of speeds of operation to meet a wide variety of functional and volume requirements. Communication with the 2770 may be on either point-to-point nonswitched, point-to-point switched, or multipoint nonswitched lines.

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.

The READ, WRITE, and CONTROL options available for the 2770 are listed in Figure 67.

PROGRAMMING -- GENERAL INFORMATION

Record Length

In nontransparent mode the maximum record length is 128 characters for the basic 2772, and 256 characters for the 2772 equipped with the Expanded Buffer feature. Records exceeding buffer size cause an I/O buffer overrun error. This error causes the 2772 to send a NAK. BTAM retries transmission with the number of retries depending on the value of the RETRY operand of the DTFBT macro instruction.

If data for a card punch is not formatted into 80-character records with the IRS character, a new card is punched after 80 characters have been sent to the punch. When the end of a block is reached, the card being punched is ejected and a new card is fed.

If data for the printer is not formatted into print lines (132 characters or less) with the IRS or NL characters, a new line is begun after 132 characters have been sent to the printer or after a margin stop is reached.

Records exceeding the display line are not truncated but are continued on the next 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 go into 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 buffer size.

Blank Cards

Basic 2772: In transparent and nontransparent modes, blank cards are read into the 2772 Control Unit from the card reader and transmitted as any other data cards.

2772 with Expanded Buffer Feature: In nontransparent mode, data from the card reader is packed. The card is read into the buffer and is scanned from column 80 backwards until a data character is reached. An IRS character is inserted in the buffer in the next position. Thus, card definition is maintained while unnecessary blanks are stripped out. Hence blank cards are not transmitted by a 2772 with the expanded buffer feature.

In transparent mode, data is not packed and blank cards are transmitted.

Error Status Messages

Whenever the 2770 causes the terminal operator to perform an error recovery procedure and the recovery procedure specifies that an error message be sent to the processor because the remote station cannot recover without more information from the terminal, the error message specified by the recovery procedure must be transmitted. The error message to be sent is selected from one of the following messages:

1.	Check Point Restart	S O H	%	S	S T X	SS 0X1X2CPP
2.	CE Attention required, cannot continue	S O H	%	S	S T X	0X1X2DZ1Z2

OPTION	Optype	Point-to-Point		Multipoint
		Contention	Dial	
Read Initial	TI	X	X	X
Read Initial Inquiry	TIQ	X		
Read Connect	TIC		X	
Read Connect with Tone	TIW		X	
Read Continue	TT	X	X	X
Read Continue with Leading Graphics	TTL	X	X	X
Read Repeat	TP	X	X	X
Read Repeat with Leading Graphics	TPL	X	X	X
Read Inquiry	TQ	X	X	X
Read Interrupt	TRV	X	X	X
Write Initial	TI	X	X	X
Write Connect	TIC		X	
Write Initial Transparent Block[1]	TIE	X	X	X
Write Initial Transparent Text[1]	TIX	X	X	X
Write Continue	TT	X	X	X
Write Transparent Block	TE	X	X	X
Write Transparent Text	TX	X	X	X
Write Wait Before Transmit	TW	X	X	X
Write Disconnect	TD		X	
Write End-of-Transmission	TR	X	X	X
Write Inquiry	TQ	X	X	X
Control Disable	TD		X	
Control Mode	TM	X	X	X

[1]If WRITE TIX or WRITE TIE is used with the 2770 on point-to-point lines and the Job Control switch on the 2770 is set for multiple output devices, the data will go to the output device of the highest priority.

Figure 67. READ/WRITE/CONTROL Options for IBM 2770

```
                       S     S       SS
3.  Job Restart        O % S T 0X1X2MPP
                       H     X

                       S     S       SS
4.  Format Error       O % S T 0X1X2FPP
                       H     X
```

where

X1 terminal address.

X2 component address.

Z1 terminal address (same as X1 if device requires CE attention, but station is operational; terminal address of alternate terminal if station requires CE attention; SP if no alternate terminal is available or desired).

Z2

component address of alternate device on this station or at alternate station.

Note: The S character must be a capital letter. The other characters may be either upper or lower case.

These messages are transmitted by placing the Job Select switch in the On-Line Test position and pressing the Terminal Reset key. If the message is to be entered from the keyboard, the Keyboard Request key is pressed and the message keyed in. It will be transmitted when the Enter key is pressed. The messages can be prepunched in cards and transmitted from the card reader, if desired. When BTAM recognizes an Error Status message, bit 3 of byte 24 in the DECB is turned on.

END-TO-END CONTROL CHARACTERS

There are six characters in each code that are not data link control characters. They provide secondary end-to-end control functions. These are:

- EM - The 'end-of-media' character is used to indicate the end of data on paper tape or Magnetic Data Inscriber. It is transmitted as data and reproduced in paper tape at the receiving station. The Magnetic Data Inscriber is an input-only device.

- IRS - The record separator character is used for card definition.

  a. When a card is read into the control unit, the IRS character is inserted after the last data character. If the data is to be recorded on paper tape, the IRS character is also recorded to maintain card definition so that the data may be printed or punched at a later time.

  b. When the control unit recognizes the IRS character, it causes a punch eject if received by the punch, or the new line function to be performed if received by the printer.

- NL - The new line 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 recorded in the medium.

- DCx - On a point-to-point line the device control character is used for component selection (i.e., of output device 1,2, or 3). Component selection is discussed in the section Point-to-Point Communication. For multipoint lines, the technique of polling and addressing is used, and the DC4 forms part of the addressing sequence.

- ESC - On either point-to-point or multipoint lines, the escape sequence is used for vertical forms control on the printer or for control of the display. Vertical forms control is achieved by specifying one of the two-character escape sequences shown in Figure 68.

  The two-character sequence appears as the first two characters following the STX for the first print line of a block of data, and as the first two characters following the NL character for each subsequent print line of the block of data.

  Two two-character escape sequences are used to control the display. These sequences are transmitted in text immediately following the STX character.

EBCDIC Sequence	USASCII Sequence	Forms Motion After Print
ESC /	ESC Q	Single space
ESC S	ESC R	Double space
ESC T	ESC S	Triple space
ESC A	ESC A	Skip to channel 1
ESC B	ESC B	Skip to channel 2
ESC C	ESC C	Skip to channel 3
ESC D	ESC D	Skip to channel 4
ESC E	ESC E	Skip to channel 5
ESC F	ESC F	Skip to channel 6
ESC G	ESC G	Skip to channel 7
ESC H	ESC H	Skip to channel 8
ESC I	ESC I	Skip to channel 9
ESC J	ESC J	Skip to channel 10
ESC K	ESC K	Skip to channel 11
ESC L	ESC L	Skip to channel 12
ESC M	ESC M	Space suppress

Figure 68. IBM 2213 Escape Sequence

The escape sequences provided for use with the display are:

ESC U
    Erase/Write
ESC '
    Write Line Address

The ESC U sequence causes the display screen to be erased and the cursor to be positioned in the first available display position. If text is included in the message block, it is displayed on the screen.

The ESC ' sequence requires that the display be equipped with the line addressing feature. When this sequence is used it must be followed by a character that specifies the line address. Valid line addresses are shown in Figure 69.

The ESC ' sequence causes the cursor to be positioned in the first available display position of the line specified. If message text is included it is displayed beginning at the line indicated by the cursor.

- VT and FF - The vertical tab character causes a skip to channel 2. The form feed character causes a skip to channel 1. The VT and FF characters may appear in text. They differ from the two-character escape sequences in that forms motion is performed upon detection of the VT or FF, while for an escape sequence forms motion is done after printing. If an escape sequence is outstanding, the VT or FF is not performed.

Function	Escape Sequence	
Erase screen	ESC U	
Erase screen and display message	ESC U (text)	
Write at Line Address	ESC ' x (text)	
	Line Address Code	
Display Line Number	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 69. IBM 2265 Valid Line Addresses

BUFFER DESCRIPTION

The basic 2772 Control Unit has two 128-position buffers. As a special feature the 2772 can have an expanded buffer consisting of two 256-position buffers. The basic 2772 Control Unit has the capability of receiving and sending one block per transmission. Thus, in nontransparent 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 varies depending on the device. For paper tape, text consists of the data characters plus 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 will not cause a punch eject. The IRS, ETB, or ETX character will cause punch eject. For the printer, the NL, IRS, and escape sequences are not considered part of the text.

IBM 50 MAGNETIC DATA INSCRIBER (MDI) -- EDIT FUNCTION

GENERAL CHARACTERISTICS

Data received from the IBM 50 Magnetic Data Inscriber (MDI) attachment to the 2772 contains MDI control characters. The TPEDIT macro provides the user with the capability of editing this data.

The macro receives control from the user in his problem program, edits the data as specified, and returns control to the user. The user has the option of gaining temporary control (via a user-specified exit routine) to process error records.

The TPEDIT macro is written in reentrant code. If data is to be received from more than one 50 MDI at a time, a separate parameter list and work area must be provided by the user for each of them.

When a BTAM user issues a READ instruction, he receives one block of data. This block of data may contain none, one, or more than one IBM 50 MDI logical records. It is the macro's responsibility to extract one record from this block of data, edit it, and give it to the user with a return code indicating whether the input area is empty or not. If the input area is not empty, the user should reissue the TPEDIT macro to obtain another record. When control is returned to the user with an indication that the input area is empty, he can reuse the input area.

If the input area contains a partial record, the available portion is edited into the work area, and maintained there. A return code indicating that the input area is empty but a record is not available is given to the user. The user must obtain the rest of the record by using READ macros. When control is given back to the TPEDIT macro, the characters in the input area (until EOR is encountered) are treated as the remaining portion of the partial record.

TPEDIT Macro Description

The TPEDIT macro instruction activates a re-enterable editing routine and is used to specify the type of work to be done on the input received from the 50 MDI attachment to the 2772.

```
|Name   |Operation|Operand
|[name] |TPEDIT   |[EDIT={EDITD}]
|       |         |       {EDITR}
|       |         |[,RECFORM={VAR  }]
|       |         |          {UNDEF}
|       |         |[,ERROPT={ IGNORE                  }]
|       |         |         {(name,{NOCHK }) }
|       |         |         {      {VOKCHK}) }
|       |         |,MINLN=n
|       |         |[,REPLACE={X'19'}]
|       |         |          {X'XX'}
|       |         |[,BUFFER ={NO }]
|       |         |          {YES}
```

EDIT= EDITD
      EDITR

specifies the type of editing to be done.

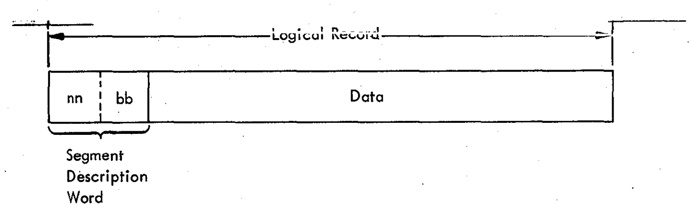
EDITD
edits the input and deletes start-of-record (SOR) and end-of-record (EOR) delimiters.

EDITR
edits the input, and the start-of-record and end-of-record delimiters are retained as part of the output.

The edit consists of the following functions: records are extracted one at a time from the input area by scanning for the record delimiting codes (SOR and EOR). DUP codes are replaced with the character from the corresponding location of the record in the work area when control was last returned to the TPEDIT logic. 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 will not appear in the output stream. Line control characters are deleted if found in the output buffer. However, in transparent mode only DLE STX, STX, ETX, and ETB are deleted; all others that are present remain in the data.

RECFORM={VAR }
        {UNDEF}

this operand specifies the format of the output from the macro. If RECFORM=VAR, a segment descriptor word is appended to each record as shown.



where nn (2 bytes) = length of logical record, and bb (2 bytes) = binary zeros reserved for system use.

This four-byte field is included in the record length returned to the user via a parameter list.

Note: This four-byte field must be allowed for by the user when determining the size of the work area (see the section on Input to the Macro).

If RECFORM=UNDEF, no sector descriptor word is appended.

ERROPT={IGNORE   }
       {(name,{NOCHK }) }
       {   {VOKCHK}) }

this operand specifies whether a user error exit routine is provided to handle error records.

IGNORE
specifies that 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.

name
specifies the name of the user error exit routine to be entered when the macro detects logical errors or replacement characters in the record.

Note: The user establishes addressability in the error exit routine.

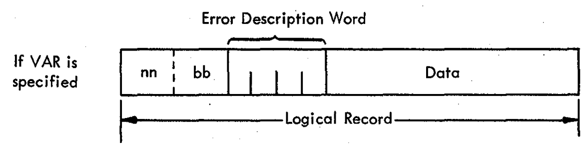
NOCHK
specifies that the records are not to be checked for verify OK (VOK) code.

VOKCHK
specifies that the records are to be checked for the VOK code. If a record does not contain a VOK code, it is considered an error record.

When an error record is encountered and control passes to this user-supplied routine, register 13 contains the address of a 72-byte register savearea aligned on a fullword boundary; register 1 contains the address of a 2-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 RECFORM=VAR, the logical record length (nn) includes these four bytes when passed to the error exit routine.



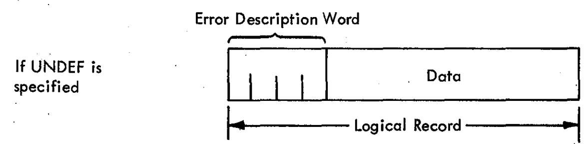


Figure 70 shows the contents of the error description word. Further information is contained under Error Record Identification.

Byte 0	Byte 1	Byte 2	Byte 3
Level Status	Type Status	Program Level	Record Status
(0)-for any error record which will not cause questionable data to be in the following records.	(0)-No identifiable error	(1)-P1	(U)-Unverified record
		(2)-P2	
	(1)-start-of-record (SOR) or end-of-record (EOR) in error.	(3)-P3	(V)-Verified Record
		(4)-P4	(E)-Neither of the above. End-of-record (EOR) in error.
		(5)-P5	
(1)-for any error which may cause questionable data in the following records.	(2)-length error	(6)-P6	
	(4)-field error	(7)-P7	
	(8)-data check error	(8)-P8	
(2)-for any error which contains questionable data due to error level or preceding records and may cause questionable data to be in the following records.	Note: This field may contain hexadecimal combinations of these error types; e.g., a C would indicate a data check error and a field error.	(E)-None of the above. Start-of-record (SOR) in error.	
The error description word in in EBCDIC format, e.g., a 2 is represented as X'F2'; a C is represented as X'C3'.			

Figure 70. Format of Error Description Word for the TPEDIT Macro

The error exit routine can be used to analyze and, if possible, correct the error record. When control returns to the edit routine via register 14, the user must set register 15 to zero if the error record is to be bypassed. To direct the edit routine to ignore an error in the record and thereby process the record in the normal manner, register 15 must be set to a non-zero value. Note that neither acceptance nor bypassing of the error record changes its effects on subsequent records. The error description word is removed by the edit routine when control returns from the error exit routine.

MINLN=n
this parameter specifies the minimum acceptable length of an input record.

For EDIT=EDITD, SOR and EOR codes are excluded from the length; for EDIT=EDITR, SOR and EOR codes are included.

REPLACE= <u>X'19'</u>
X'XX'
this operand specifies the code to be used as a replacement character whenever an EBCDIC substitute (X'3F') is detected in the input. (See Figure 65.)

1. X'19' is chosen as the assumed value because it is an end-of-data signal for an IBM 50 MDI cartridge. Therefore, it can never appear as a valid data byte.

2. For REPLACE=X'XX' the user can replace XX with hexadecimal characters of his own choosing. These choices may be made from the code chart shown in Figure 71, except for the special characters noted below.

3. BSC control characters should not be used as substitute characters if the data is to be transmitted via BSC facilities after editing.

Hexadecimal characters representing special-purpose Magnetic Data Inscriber codes that should not be used as replacement bytes are:

X'00'	LZ
X'11'	DUP
X'12'	LZS
X'18'	CAN
X'1D'	GS
X'1E'	VOK
X'3C'	RM
X'71'	P1
X'72'	P2
X'73'	P3
X'74'	P4
X'75'	P5
X'76'	P6
X'77'	P7
X'78'	P8

BUFFER={<u>NO</u> | YES}
this operand specifies whether the user's data is in BTAM buffers or not. If YES is specified, the TPEDIT macro edits all data in the buffer chain before indicating that the input area is empty.

<u>Input to the TPEDIT Macro</u>

Register 1 must point to a four-word parameter list aligned on a fullword boundary containing:

<u>Word</u>	<u>Description of Contents</u>
1	Input Address If the user is using BTAM buffering techniques, this address points to the first buffer in the chain. The TPEDIT macro edits all records in the buffer chain before indicating that the input area is empty. If buffers are not used, this word contains the address of the data to be edited.
2	Input Length If BTAM buffers are used, this is the length of one buffer. For an I/O area, this is the length of the data to be edited. <u>Note</u>: If buffers are being used, then the buffer length, specified in the DTFBT, must be a multiple of 8. This prevents unused space at the end of each buffer.
3	Edit Work Area Address The work area must be provided by the user and be large enough to contain an error description word (EDW), plus four bytes for the segment descriptor word for the V-format record (if specified), plus the length of the data record, plus the necessary work area to be used by the macro. The size (in bytes) of the work area may be determined from the formula: S = 84 + 76E + R + 4V where: S = work area size (in bytes). E = 0 if ERROR=IGNORE is coded. = 1 if ERROR=name is coded.

V = 0 if RECFORM=UNDEF is coded.
  = 1 if RECFORM=VAR is coded.
R = 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.

		Quad I				Quad II				Quad III				Quad IV			
Bits 0, 1		00				01				10				11			
Bits 2, 3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
1st Hex Digit		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	LZ				Space	&	Minus -							$\bar{0}$	082	0
0001	1		Dup					/	P1					A	J		1
0010	2		LZ Start						P2					B	K	S	2
0011	3								P3					C	L	T	3
0100	4								P4					D	M	U	4
0101	5								P5					E	N	V	5
0110	6								P6					F	O	W	6
0111	7								P7					G	P	X	7
1000	8		Can						P8					H	Q	Y	8
1001	9		ED											I	R	Z	9
1010	A					¢	!		:								
1011	B						$	,	#								
1100	C				RM	<	*	%	@								
1101	D		GS			(	)	Underline	' Prime								
1110	E		VOK			+	;	>	=								
1111	F					|	¬	?	"								

Bits 4, 5, 6, 7 — 2nd Hex Digit

IBM 50 Code Name	EBCDIC Code Name
LZ (LZ Fill)	NUL
Dup	DC 1
LZ Start	DC 2
Can (Cancel)	Can
ED (End Data)	EM
RM (Rec Mark)	DC 4
GS (Group Sep)	IGS
VOK (Ver OK)	IRS

IBM 50 Code Name (See Note)
P1 (Prog 1)
P2 (Prog 2)
P3 (Prog 3)
P4 (Prog 4)
P5 (Prog 5)
P6 (Prog 6)
P7 (Prog 7)
P8 (Prog 8)

Note: Codes are assigned for IBM 50 use only.

Figure 71. IBM 50 Magnetic Data Inscriber Control Codes

The value of the maximum record size should not include the four-byte segment descriptor word added to a variable-length record.

Records which exceed the maximum record size are considered error records.

Register 13 must contain the address of a 72-byte register savearea aligned on a fullword boundary.

Return Codes From TPEDIT Macro

When a record has been edited, a return code indicating record availability and status of the input area is placed in register 15 before returning control to the user. The return codes and their interpretation are as follows:

Hex Code	Interpretation
00	Record available, input is empty. The macro has edited the last logical record in the input area and is passing the record to the user.
04	Record available, input is not empty. The macro has edited one logical record and is passing the record to the user.
08	Record not available, input is empty. Either (1) the last logical record was a partial record or (2) an ETX was the last character in the buffer.
0C	Last record; i.e., end of transmission (EOT).

For return codes 00 and 04, the record address and the address of the record length are given to the user in a two-word parameter list aligned on a fullword boundary. The address of the parameter list is returned in register 1. The list has the following format:

Word	Contents
1	Pointer to the record.
2	Pointer to the record length.

Error Record Identification

This section describes what the TPEDIT macro considers to be error records. Once the TPEDIT macro has determined that a record is in error, the record is bypassed to the user error exit routine if ERROR=name is specified in the TPEDIT macro statement. If an error exit is not specified, the record is returned normally to the user.

The TPEDIT macro maintains information about each record as it is being edited. This information is summarized in the discussion of the error description word (EDW) below. When the EDW contains a non-zero value in either the level status (byte 0) or the type status (byte 1), the record is considered an error record, and the EDW is inserted between the four-byte record length field and the data portion if RECFORM=VAR is specified. Otherwise, the EDW is appended to the start of the record to aid the user in analyzing the error.

The format of the error description word is shown in Figure 70.

Level Status (Byte 0): The level status indicator identifies error records that result from inter-record dependency and that cannot be identified in the type status byte.

The level status is presented with each record and has a value of:

0 For any error record that will not cause questionable data in the following records.

1 For any error record that may cause questionable data in the following records, and the level status of the previous record was 0.

2 For any error record that has questionable data within is content due to the error level of preceding records; that may cause questionable data in the following records; and where the level status of the previous record was either 1 or 2.

A level status of other than 0 is presented with error 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 side by side.
- 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 this cartridge.
- The record has a data duplication dependency on a previous record with one of the above conditions.

The level status is set to 0 whenever the TPEDIT macro encounters a record without one of the above errors, a cancelled record, or the first record of a cartridge.

Type Status (Byte 1): The type status indicator identifies records in error because of SOR, EOR, length, field, or data check error conditions.

The type status is presented with each error record and has a value of:

0 For any record that has no identifiable errors in its content, but contains questionable data due to a level status other than 0.

1 For any record that has an SOR character of other than P1 through P8 or a GS code, or that has an EOR character of other than a VOK code for records when the user has specified VOKCHK, or that has an EOR character of other than a VOK code or RM code for records when the user has specified NOCHK.

2 For any record that has an incorrect length because it is:

   - Longer than the specified maximum.
   - Shorter than the specified minimum (MINLN).
   - Not equal to the length of the first valid record of the same program level encountered on this cartridge.

4 For any record that has a field error. A field error is a field where duplication or left zero justification functions did not occur due to an error condition.

8 For any record that has a data check error.

The type status indicator can also have values of 3, 5, 6, 7, 9, A, B, C, D, E, and F. These values indicate a combination 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 (2).

Note: A data check error is indicated by 2772 replacement 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.

Record Status (Byte 3): This byte contains an indication of the end-of-record (EOR) character associated with this record.

Sample Error Records

These records (Figure 72) show some of the errors that may occur during processing and their effect on the error description word. 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 1 was a valid record. It contained a program level 1 code, and thus established the valid length for all program level 1 records in this cartridge.

Record 2 has a data check in the SOR location. Level status is set to one 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. Because it contained no DUP codes, the level status is set to 0.

Record 5 is shorter than the first program level 1 record on this cartridge (length error). This record also contains an RM code rather than a VOK code in the EOR location (VOKCHK was specified). Because the macro cannot determine why the record is short, all data duplicated from this record is questionable, and the level status is 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 (m1). Note that is 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. Because it contained no DUP codes, the level status is set to 0.

Program Considerations

All cancelled records are bypassed and are not passed as error records.

All input records less than three bytes long (SOR location, one data byte, EOR location) are treated as cancelled records. An input record of this size may be the remaining portion of a record that was

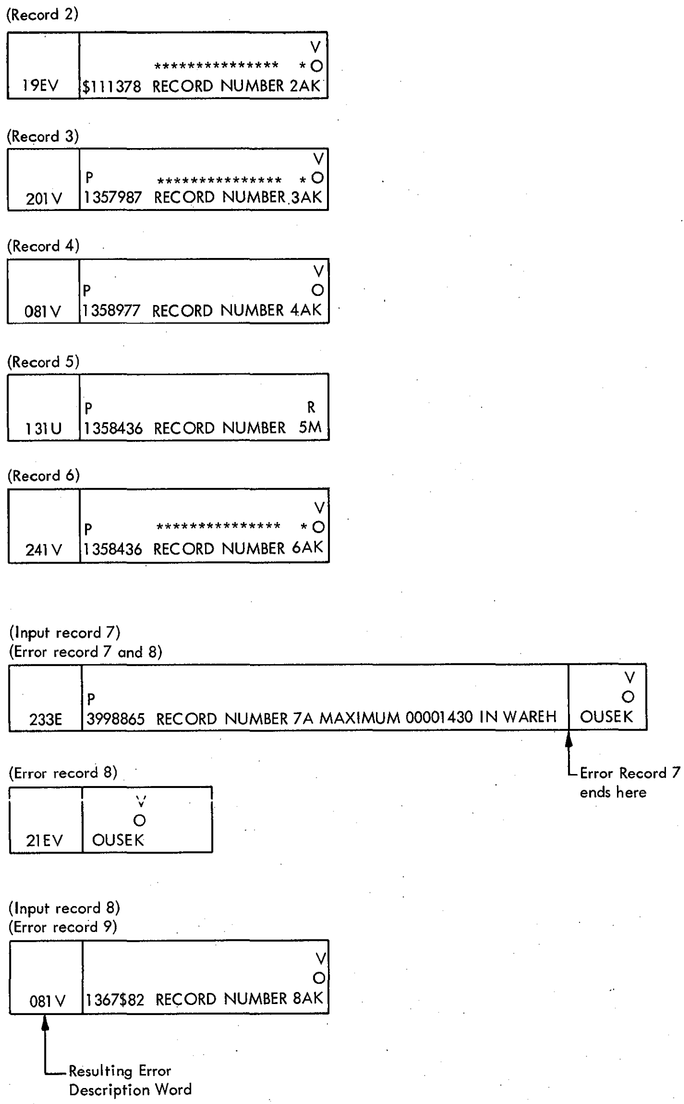


Figure 72. Sample Error Records

longer than the maximum user-specified record size.

Data duplication will occur with the DUP code being replaced with the character from the corresponding location of the previous record in the work area.

Data duplication will not occur and the DUP code will be replaced with the user-specified error replace character, and a field error will be indicated, for any of the following conditions:

1. The DUP code is encountered in the first record of a cartridge.

2. The DUP code is encountered in a record and the previous record was a cancelled record.

3. The DUP code is encountered in a record and its position would cause duplication of the previous record's end-of-record delimiter location or a position beyond the length of the previous record.

4. The DUP code is encountered in a record and its position would cause duplication of an error replace character.

Left zero justification will not occur, the left zero fill code will be replaced with the user-specified error replace character, and a field error will be indicated, for either of the following conditions:

1. The left zero fill code (LZ) is encountered without its corresponding left-zero start code (LZS).

2. The user-specified maximum record size is exceeded before encountering the valid end of a left-zero field.

If the BTAM buffers are being used, the BSC control characters ETB and ETX should not be entered as data on the IBM 50 MDI cartridges.

End-of-Cartridge Code

A unique code, writted 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 IBM 50 MDI cartridge, the end-of-cartridge code is the EM character (X'19').

After initiation of a READ operation, the IBM 50 MDI continues to read data from the tape until the EM (X'19') character is sensed. When the MDI sends this character to the 2772, the adapter signals the tape to rewind, and the 2772 transmits the data in its buffer to the CPU.

POINT-TO-POINT COMMUNICATION

Point-to-point communication with the 2770 may occur over switched or nonswitched networks. In point-to-point communication, the programmer 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 component selection is not required and the device control character is not present in the first block of data, the data will go to the highest-priority output device. If more than one output device is assigned by the Job Select switch and component selection is required for this setting, the device control character is mandatory. The device control character must be immediately preceded by an STX character and should be followed by the ETB character or text. Valid device control characters are:

DC1	Output Device 1
DC2	Output Device 2
DC3	Output Device 3

When operating in transparent mode, the user must either send the device control character in normal mode or transmit to the component selected by the Job Select Switch. The device control character will be recognized as a component selection character only when it is sent in the first message after an EOT has been received, or following the first STX in conversational mode.

Nonswitched Network

In a contention system, either BTAM or the 2770 may initiate transmission by sending the inquiry signal (ENQ) over the line. When a true contention situation occurs (both BTAM and the 2770 are bidding for control of the line) either the 2770 or the CPU may be declared the master station, depending upon the FEATURE operand in the DTFBT macro instruction (i.e., either SLV or MAS coded).

Identification: The identification feature provides the 2772 with the capabilities of: 1) initially transmitting a two-character identification sequence after a connection has been established on a switched line, and 2) accepting identification characters from the line.

An identification character is selected by the user at the time of system installation. The 2772 transmits the identification character twice. The character is transmitted preceding the line bid ENQ if it is the transmitter, or preceding the response to selection if it is the receiver.

Security Identification: The security identification feature may be selected for use on a switched network in place of the identification feature. This feature functions the same as the identification feature with the following exception. When a line bid is transmitted by the 2772, three additional identification characters are inserted between the repeated identification character and the ENQ character. These three characters are selected by the user and may be three distinct characters.

The responses to a line bid are the same as for the identification feature.

For a detailed description of identification procedures, see the section on ID Verification under System/370 to System/370.

For detailed descriptions of the identification and security identification features, see the IBM 2770 Data Communications System Components publication, GA27-3013.

MULTIPOINT COMMUNICATION

Transmission of messages to or from a particular 2770 on a multipoint line is initiated by the CPU, which places the line in control mode and then sends the appropriate polling or addressing sequence. The polling and addressing sequences are coded by the user in his polling and addressing lists using the DFTRMLST macro instruction. These sequences are four characters long.

The first character is an alphabetic terminal address character. This character may be any alphabetic character, and it is set at the time of the installation of the 2770. The uppercase letter is the polling address; the corresponding lowercase letter is the selection address.

The second character is the same as the first character.

The third character in the polling or addressing sequence is a numeric component selection character. Valid component selection characters are:

Input Device	Character
Any ready input device	0
Keyboard	5
Input 1	6
Input 2	7
Output Device	**Character**
Output 1	DC1
Output 2	DC2
Output 3	DC3

The fourth and final character in the sequence is the ENQ character. The general poll provides the ability to transmit a single polling sequence and to receive data from any ready input device on the polled 2770. To accomplish this, the third character in the polling list defined by the DFTRMLST macro instruction must be a 0 character.

IBM 3270 INFORMATION DISPLAY SYSTEM (REMOTE)

IN GENERAL

The 3270 Information Display System, which consists of an alphameric display screen (CRT) and keyboard and an associated printer, provides input/output for a wide range of applications. The 3270 used as a teleprocessing system and attached to the computer through a transmission control unit and telecommunications lines is referred to in this book as the "remote 3270." The 3270 attached directly to a multiplexer or selector channel is referred to as the "local 3270" and, because the local and remote 3270 require different programming considerations, is described separately in the general section, "Local Device-Dependent Considerations."

USING THIS SECTION WITH OTHER PUBLICATIONS

Before writing an application program using a remote 3270, it is necessary to understand the remote 3270's physical characteristics and capabilities, described in IBM 3270 Information Display System Component Description, GA27-2749. After reading the 3270 Component Description, the programmer can use this book, particularly this section, as a guide to the BTAM macro instructions that are used to define and control I/O for the remote 3270. He must, however, use the component description to construct data areas (called "data streams") to be sent to the terminal to display an image or print a line, and to interpret data streams received from the terminal. The formats for these data streams are shown in this section with applicable macro instructions; however, the 3270 Component Description must be used to understand terms used within these formats.

It is also necessary to understand binary synchronous telecommunications procedures, described in General Information - Binary Synchronous Communications, GA27-3004.

SAMPLE REMOTE 3270 APPLICATION PROGRAM

A sample BTAM application program designed to demonstrate a remote 3270 application is provided with the system. The sample program, named SAMP327R, is shown in Appendix L.

REMOTE 3270 CONFIGURATION

A remote 3270 can consist of either or both of two configurations. The first consists of:

- One or more 3271 Control Units, Models 1 or 2
- One or more 3277 Display Stations, Model 1 or 2, attached to each control unit
- Optionally, one or more 3284 or 3286 Printers, Models 1 or 2

Up to 32 terminal devices (display stations and printers) can be attached to a single control unit.

The second configuration consists of:

- One or more 3275 Display Stations, Models 1 or 2
- Optionally, a 3284 Printer, Model 3, attached to a 3275 Display Station.

The 3275 Display Station is a "stand-alone" device and does not require a control unit to communicate.

The remote 3270 is attached over telecommunications lines to any of the following transmission control units:

- A 2701 Data Adapter Unit
- A 2703 Transmission Control Unit
- An Integrated Communications Adapter.

The transmission control unit is attached to a computer channel.

A remote 3270 configuration is shown in Figure 73. For planning considerations, see An Introduction to the IBM 3270 Information Display System, GA27-2739.

Both local and remote 3270 configurations can be defined and controlled by a single application program.

PROGRAMMING CAPABILITIES

Using BTAM macro instructions defined in this section, an application programmer can:

- Read modified fields from a display station buffer after a terminal operator has completed his entry and caused an attention (for instance, by pressing the ENTER key).
- Read from a display station buffer fields modified by an operator without waiting for an attention indication.
- Read only those modified fields beginning at a specified buffer location.

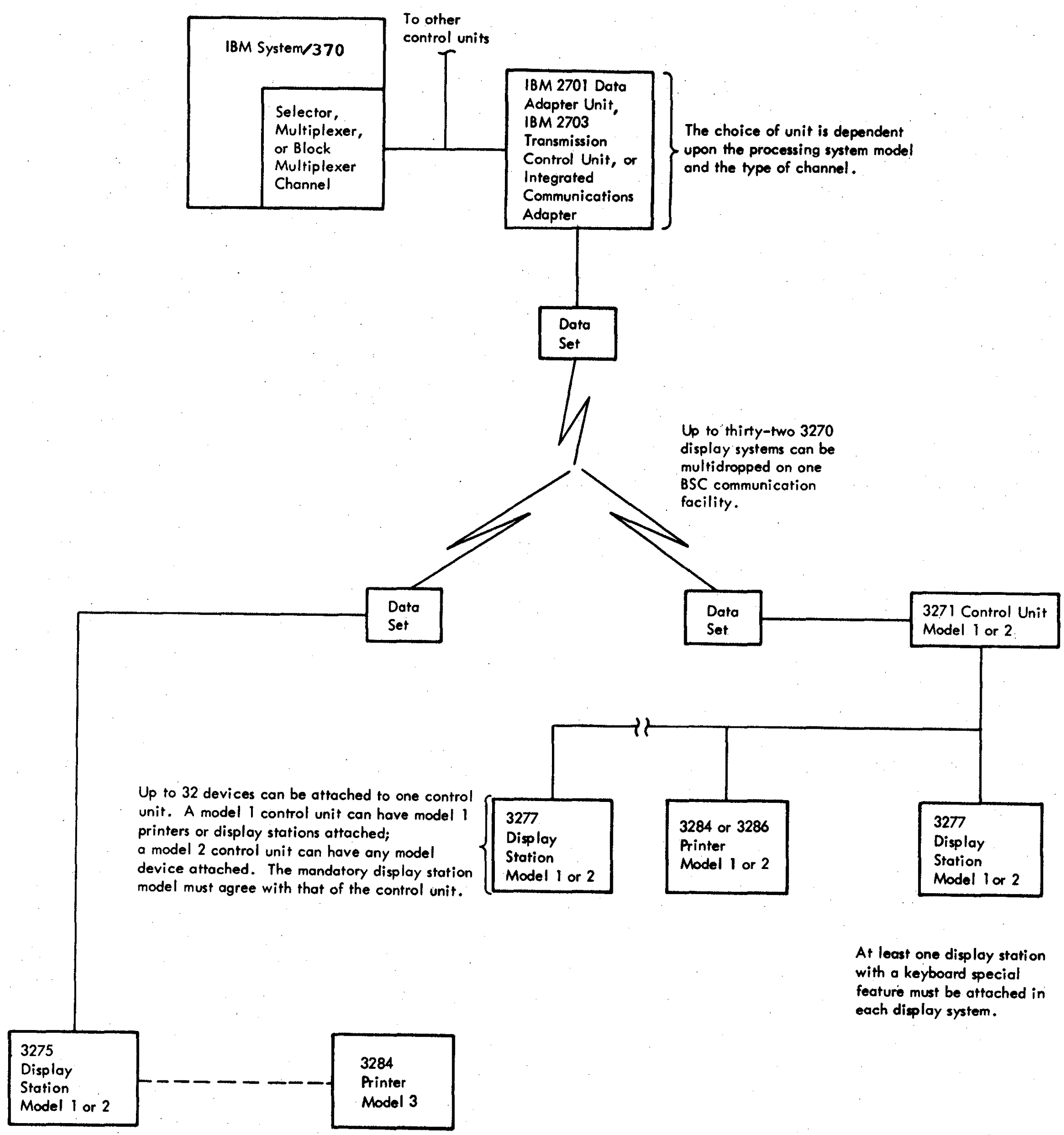


Figure 73. Remotely Attached 3270 Display System

- Read the entire buffer contents, both modified and unmodified data, including attribute characters.
- Read the buffer contents, both modified and unmodified data, including attribute characters, beginning at a specific buffer location.
- Write data to any desired position in a display station or printer buffer.
- Erase the data presently at the device buffer (on the screen or in the printer buffer) and write data to any buffer location.
- Erase all unprotected fields in the display or printer buffer.
- Copy data in one device buffer to the buffer of a device attached to the same control unit. (For instance, have the contents of a display station buffer printed at a nearby printer.)

How each of these capabilities can be used within the context of an application is discussed later in this section under "Read Operations" and "Write Operations."

Attribute characters, modified and unmodified data, and protected and unprotected data are described in the 3270 Component Description.

DEFINING A REMOTE 3270

To write an application program involving the remote 3270, three things must be defined:

- The BTAM logic needed by the application program, defined with the BTMOD macro instruction.
- The line group configuration of the remote 3270, defined with the DTFBT macro instruction.
- The terminal lists applicable to each line group, defined with the DFTRMLST macro instruction.

Defining BTAM: BTMOD

The BTMOD macro instruction is used to define the sections of BTAM logic needed by the application program; BTMOD is described fully in the general section "BTAM Macro Instruction Format Descriptions." BTMOD may be coded by itself and assembled separately from the application program or may be coded with the application program. To include support for the remote 3270 (that is, support for binary synchronous transmission), the following operands must be coded in the BTMOD macro instruction:

```
BSC=YES
BSCMPT=YES
```

Other operands applicable to the remote 3270 are: SEPASMB, BUFFER, ERLOGIC, CANCEL, TRANSL, BSCTEST, DECBEXT, RMSR, and RESETPL. Additional operands will be applicable if the application program performs I/O with other terminal systems (for example, a local 3270 or a remote 2260).

An example of a BTMOD macro instruction for a remote 3270 is:

```
BTMR3270    BTMOD SEPASMB=YES,BUFFER=YES,
                  ERLOGIC=NC,TERMTST=YES,
                  BSC=YES,BSCMPT=YES,
                  BSCTEST=YES,RESETPL=YES
```

A BTAM logic module named BTMR3270 will be assembled separately from the application program. It will include buffer logic, error counts in binary synchronous error recovery procedures, the facility to have online terminal tests performed, logic for multipoint BSC (binary synchronous) devices, and logic to reset polling (in addition to the basic BTAM logic always provided as a result of assembling BTMOD). The BTAM logic module in this example will not include logic applicable to other devices, logic for immediately canceling a job, code translation, the DECB extension, or Recovery Management Support Recording (RMSR).

Defining a Line Group: DTFBT

A line group is a group of telecommunications lines attached to a channel through the same type of transmission control unit. The line connection between the control unit and the remote devices must be of the same type, all lines must share the same buffer pool if one is defined, and all devices must have the same line control procedures, transmission code, and operating characteristics.

Where a remote 3270 configuration is being defined that uses only one type of transmission control unit (for instance, one or more 2701 Data Adapter Units) and the transmission code, EBCDIC or ASCII, desired is the same for all lines, it is possible that the remote 3270 can be defined by a single DTFBT macro instruction. Where more than one type of transmission control unit or transmission code is required, more than one DTFBT macro instruction must be coded. Since the 3271 Control Unit and the 3275 (stand-alone) Display Station use the same line control procedure, multipoint binary synchronous, they may be defined as part of the same line group in one DTFBT macro instruction.

The format of DTFBT is shown in the general section, "BTAM Macro Instruction Format Descriptions." Applicable operands are: LINELST, SWITCH, CU, DEVICE, BUFCB, BUFNO, BUFL, SEPASMB, MODNAME, TERMTST, LERBADR, MODELST, CTLCHAR, RETRY, CONFIG, and LCBNUM.

SWITCH must either be specified NO or omitted. CU may be either 2701 or 2703 (2703 is also specified for the Integrated Communications Adapter). DEVICE must be BSC3. CTLCHAR may be either EBCDIC or ASCII (TRANSCD is not applicable). CONFIG must be specified MPT.

An example of a DTFBT macro instruction is:

```
LINEG1   DTFBT LINELST=(004,008,010),
               CU=2703,DEVICE=BSC3,
               TERMTST=YES,LERBADR=ERRTAB,
               CTLCHAR=ASCII,CONFIG=MPT
```

The lines associated with SYS004, SYS008, and SYS010 are defined as part of the same line group. They are connected to either a 2703 Transmission Control Unit or an Integrated Communications Adapter. Devices attached to these lines are reached with the binary synchronous procedure for nonswitched multipoint lines, the BSC online terminal test capability is requested, error recording is specified for a table at ERRTAB (the statement name of either an LERB or RMSRTAB macro instruction), the transmission code will be ASCII, and the line configuration is defined as multipoint.

SYS004, SYS008, and SYS010 will have relative line numbers 0, 1, and 2, respectively. A line control block (LCB) will be generated for each of these lines in the DTFBT control block created as a result of assembling DTFBT.

Defining a Terminal List: DFTRMLST

A terminal list defines specific control units and terminals attached to a line and is used in polling and addressing operations resulting from READ and WRITE macro instructions.

There are two kinds of polling for remote 3270 devices: general polling, where a response is sought from any device attached to a particular control unit and the control unit has the responsibility of querying each device in turn for readiness to provide input; and specific polling, where a response is sought, one at a time, from particular devices attached to a particular control unit.

Selection or addressing, performed during write operations, is always to a specific device. (See the "Introduction" section for a general discussion of polling and addressing.)

A terminal list is defined in a DFTRMLST macro instruction and later referred to by the entry operands of READ and WRITE macro instructions. A terminal list consists of one or more entries, each containing either a control unit address and a general polling indicator in place of a device (for general polling) or a control unit and device address (for specific polling or addressing). Ordinarily, a terminal list for general polling is separate from a list for specific polling or one for addressing.

A terminal list for general polling contains entries in this format:

CU ADDRESS	CU ADDRESS	7F	7F	ENQ

A terminal list for specific polling and addressing has entries is this format:

CU ADDRESS	CU ADDRESS	DEVICE ADDRESS	DEVICE ADDRESS	ENQ

The control unit and device addresses are repeated because binary synchronous multipoint communications for the 3270 uses double addressing as a check against intermittent transmission line errors. (A discussion of double addressing can be found under "Multipoint" in this general section.) The hexadecimal values for defining a terminal list depend on which control unit and device are specified and whether the transmission is to be in EBCDIC or ASCII. These addresses are shown in Appendix J. The hexadecimal representation for the ENQ characters is either X'2D' (EBCDIC) or X'05' (ASCII).

The format in which the terminal list entry is written in the DFTRMLST macro instruction is different from the format in which it is generated in the program. The format of the terminal list for the remote 3270, the Auto Poll Terminal List (AUTOLST, AUTOWLST), is shown in Appendix A.

Writing the DFTRMLST Macro Instruction

For both general and specific polling, the DFTRMLST macro instruction for use with the remote 3270 has this format:

Name	Operation	Operation
symbol	DFTRMLST	{AUTOLST | AUTOWLST}, eotsyn,pollchars,...

AUTOLST|AUTOWLST
: specifies whether the terminal list is to be an open polling list (to be polled only once) or a wraparound polling list (to be polled continuously until a positive response is received). AUTOLST is written for an open polling list; AUTOWLST is written for a wraparound polling list.

eotsyn
: specifies the bit configuration for the EOT and SYN characters. 3732 is written for EBCDIC transmission code; 0416 is written for ASCII transmission code.

pollchars
: specifies in hexadecimal numbers a sequence of polling characters in the format described above under "Defining a Terminal List: DFTRMLST."

Here is an example of a DFTRMLST macro instruction to create a wraparound terminal list with entries for general polling:

```
LISTPL   DFTRMLST AUTOWLST,3732,40407F7F2D,
                  C4C47F7F2D,C2C27F7F2D
```

A wraparound terminal list is defined, with entries for control units 0, 4, and 2. General polling is to be performed by all three control units. The transmission is to be in EBCDIC.

Here is an example of a DFTRMLST macro instruction to create an open terminal list with entries for specific polling:

```
LISTSPEC  DFTRMLST  AUTOLST,0416,
                    2020202005,2020414105,
                    2020424205,4141434305
```

An open terminal list is defined with entries for devices 0, 1, and 2 on control unit 0 and for device 3 on control unit 1. The transmission is to be in ASCII.

For addressing (also called selection), the DFTRMLST macro instruction for use with the remote 3270 has this format:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(addrchars,...)

OPENLST
: specifies that the terminal list is to be used for addressing.

addrchars
: specifies an entry for each device that can be addressed. This entry is specified in the same format as for polling.

Here is an example of a DFTRMLST macro instruction for a terminal list for addressing:

```
LISTADR   DFTRMLST OPENLST,(C9C940402D,
                   C9C9C1C12D,C9C9C2C22D)
```

A terminal list for addressing devices 0, 1, and 2 on control unit 9 is defined. 2D in the ENQ byte of each entry indicates the transmission is to be in EBCDIC. In writing to one of the devices, the entry operand of the WRITE will contain the address of the entry within the terminal list (LISTADR) of the device to which a message is to be written.

READING AND WRITING WITH THE REMOTE 3270

Having defined the required BTAM logic with the BTMOD macro instruction, the remote 3270 line group configuration with one or more DTFBT macro instructions, and individual control units and devices on each line with one or more DFTRMLST macro instructions, an application is now ready to control input and output for terminal devices. Each remote 3270 programming capability (for example, reading input from the operator) is effected by a combination of READ and/or WRITE macro instructions, WAIT macro instructions, and, in some cases, by special characters in the data stream called an "escape command sequence." The requirements for effecting the remote 3270 programming capabilities are shown in Figure 74 and discussed in detail later in this section under "Read Operations" and "Write Operations."

The general format of the READ, WRITE, and WAIT macro instructions is explained in the general section "BTAM Macro Instruction Formats"; the operands applicable to the remote 3270 and their meaning are explained under the macro instruction descriptions in this section. The codes applicable in the optype operands for READ and WRITE macro instructions for the remote 3270 are shown in Figure 75. A general explanation of what each type of macro instruction does is discussed under "Multipoint BSC Channel Programs" in this general section.

Following execution of each READ and WRITE macro instruction, a return code appears in register 15; following execution of each WAIT, a completion code can be found in the first byte of the DECB. This information is discussed later in this section under "Error Recovery."

In the discussion that follows of read and write operations, the transmission control characters (STX, ETX, etc.), device control characters and field definition information are shown in data stream formats; these characters are described in IBM 3270 Information Display System Component Description, GA27-2749.

READ OPERATIONS

The application programmer reading from a remote 3270 can perform any of several functions. These functions are:

- Read modified fields after operator action
- Read modified fields
- Read modified fields from position
- Read buffer
- Read buffer from position

These functions are performed by combining one or more READ or WRITE macro instructions with an appropriate ESC command code in an output data stream. (This is shown in Figure 74.) Each function is described individually below.

To Do This	Use These Macro Instructions[1]	With This ESC Code[3]
Read Modified Fields After Operator Action	READ Initial (TI)	No ESC required
	Then READ Continue(s) (TT)	
Read Modified Fields	WRITE Initial Conversational (TIV) or WRITE Conversational (TV)	6
	Then READ Continue(s) (TT)	No ESC
Read Modified Fields from Position[2]	WRITE Initial (TI) and/or WRITE Continue (TT)	1
	Then WRITE Conversational (TV)	6
	Then READ Continue(s) (TT)	No ESC
Read Buffer	WRITE Initial Conversational (TIV) or WRITE Conversational (TV)	2
	Then READ Continue(s) (TT)	No ESC
Read Buffer from Position[2]	WRITE Initial (TI) and/or WRITE Continue (TT)	1
	WRITE Conversational (TV)	2
	Then READ Continue(s) (TT)	No ESC
Write	WRITE Initial (TI) and/or WRITE Continue(s)	1
	Then WRITE EOT (TR)	No ESC
Erase and Write	WRITE Initial (TI) and/or WRITE Continue(s) (TT)	5
	Then WRITE EOT (TR)	No ESC
Erase Unprotected Fields	WRITE Initial (TI) or WRITE Continue (TT)	?
	Then WRITE EOT (TR)	No ESC
Copy	WRITE Initial (TI) and/or WRITE Continue(s) (TT)	7
	Then WRITE EOT (TR)	No ESC

[1]A WAIT macro instruction will usually be required with each READ or WRITE macro instruction to determine I/O completion.
[2]In order to effect the Read Modified Fields from Position and Read Buffer from Position functions, a WRITE Initial or Continue must be issued first to establish the screen position by specifying an SBA address, and then a WRITE Conversational must be issued to send the escape command (it will also read up to 256 bytes of data).
[3]The ESC code, in character form, is preceded by the ESC character.

Figure 74. Remote 3270 Read and Write Functions

Read Modified Fields after Operator Action

This is the basic means of reading data entered at a terminal by an operator. This function is requested by issuing a READ Initial macro instruction and at least one READ Continue macro instructions. The READ Initial specifies a terminal list and initiates a general or specific polling operation; data is read when a terminal is encountered at which the terminal operator has done one of the following:

- Pressed one of these keys:

  ENTER
  PF (PROGRAM FUNCTION) keys 1-12
  PA (PROGRAM ATTENTION) keys 1-3
  TEST REQUEST
  CLEAR

Type of Macro Instruction	Optype Code
READ Initial	TI
READ Continue	TT
READ Repeat	TP
READ Inquiry	TQ
READ Interrupt	TRV
WRITE Initial	TI
WRITE Initial Conversational	TIV
WRITE Continue	TT
WRITE Conversational	TV
WRITE Inquire	TQ
WRITE EOT (End-of-Transmission)	TR
WRITE Wait-before-Transmitting	TW

Figure 75. Remote 3270 Optype Codes

- Selected a detectable field with the selector pen. (See the Component Description for how to establish a detectable field.)
- Inserted a card in the operator identification card reader and caused it to be read.

All modified fields are read from the terminal buffer into the input data area specified by the program. A maximum of 256 bytes of data are read as the result of one macro instruction. By monitoring the data area for an ETX character (meaning no more message blocks remain to be read), the program can determine whether all the data was read. After the READ Initial, at least one READ Continue must be issued; if all data was read on the READ Initial, the READ Continue must be issued to read the EOT character.

Applicable operands for the READ Initial and Continue macro instructions for the remote 3270 are: decb, optype, dtfbt, area, length, entry, rln, and MF. Although no more than 256 bytes can be read during a single transmission, the length should be specified as 300 if the largest RFT test pattern is to be accommodated. The entry operand specifies the entry within a terminal list with which polling (for READ Initial) is to begin. The rln operand specifies, for READ Initial, the line on which the polling is to occur; for READ Continue, it specifies the line on which read operations are to contine.

The message read by READ Initial will be in one of the formats shown below. The index byte at the beginning of the input message tells which entry in the terminal list was read. Either this byte or the control unit and device address bytes in the data stream may be used to determine which device responded positively to polling.

If the operator pressed the ENTER key, a PF (PROGRAM FUNCTION) key, or selected a detectable field with the selector pen, the message read will be, assuming the terminal buffer is formatted, in this format:

INDEX BYTE	STX	CU ADDR	DEVICE ADDR	AID	CURSOR ADDRESS	SBA	BUFFER ADDR	TEXT	SBA	BUFFER ADDR	...	{ETB / ETX}

If, in the above case, no fields have been modified by the operator (or already set to modified by the program), the format of the input message is:

INDEX BYTE	STX	CU ADDR	DEVICE ADDR	AID	CURSOR ADDRESS	ETX

If the terminal buffer is unformatted, the input message is:

INDEX BYTE	STX	CU ADDR	DEVICE ADDR	AID	CURSOR ADDRESS	TEXT	...	{ETB / ETX}

If the operator pressed the CLEAR key or pressed a PA (PROGRAM ATTENTION) key, the input message is:

INDEX BYTE	STX	CU ADDR	DEVICE ADDR	AID	ETX

If a card or cards were read by means of the identification card reader, the input message is:

INDEX BYTE	STX	CU ADDR	DEVICE ADDR	AID	0-37 CHARACTERS	{EOR / EOI}	LRC	ETX

If a test request message was entered, the input message is in this format (although the application is not normally aware of it):

INDEX BYTE	SOH	%	/	STX	TEXT	...	ETX

If a status message was read, the input message is in this format:

INDEX BYTE	SOH	%	R	STX	CU ADDR	DEVICE ADDR	SENSE/ STATUS 1	SENSE/ STATUS 2	ETX

In all of the above cases, at least one READ Continue would be issued to read successive blocks of the message if the message ended with an ETB or to read an EOT if the message ended in an ETX. (Note that the EOT can be discerned by checking the EOT-received flag in the DECB.)

A message block received as the result of a READ Continue has this format (unless it is unformatted):

STX	...	SBA	BUFFER ADDR	TEXT	SBA	BUFFER ADDR	...	{ETB / ETX}

To terminate a general poll (if, for example, there has been no response after a certain time from terminals in a wraparound terminal list), the RESETPL macro instruction can be used. To terminate reading from a device when the input buffers are too few and it is necessary to restart the operation, a READ Interrupt (TRV) can be issued.

See the Component Description publication for an explanation of the AID, cursor address, SBA, and other data stream characters illustrated in the message formats above.

Here is an example of reading modified fields after operator action:

```
LOOK1       READ     DECB1,TI,DTFBT1,AREA,300,(5),2,MF=E

            LTR      15,15
             .
             .
             .
            BZ       COMPLETE
             .
             .
             .
COMPLETE    TWAIT    DECB1,TERMTST
             .
             .       (move data from AREA)
             .
FINISH      READ     DECB1,TT,DTFBT1,AREA,300,2,,MF=E
             .
             .
             .
            TWAIT    DECB1,TERMTST
```

The READ Initial issued at LOOK1 initiates a poll of a terminal list beginning at the entry whose address is in register 5. The poll is conducted on relative line number 2 of the line group defined by DTFBT1. When a device ready to send a message is found, the first (or only) message block is read into AREA. The 300-byte input area length will accommodate the largest RFT test should a test request occur. The maximum of 256 bytes read includes both text and control data. Only modified fields are read. Where a modified field contains nulls (X'00'), the nulls are not read. DECB1, defined by a previous READ with MF=L specified, is used to record the results of the read operation.

On determining that the READ Initial was initiated successfully (a return code of X'00' in register 15), the application program performs other processing. When the input from the READ Initial at LOOK1 is required by the program, a WAIT macro instruction is issued, specifying the DECB associated with the READ Initial. (To accommodate the possibility of an online test request, a TWAIT with TERMTST specified can be coded instead of a WAIT.) If the READ Initial has already resulted in a message being read, control returns to the application program immediately; if not, the application is held in a wait state until a terminal is ready to be read and a read operation does occur.

When the program regains control following a WAIT or TWAIT, and determines that reading was successful, the program can now analyze the message to determine if more blocks are to be received. If this were a READ Continue and the EOT-received flag were on in the DECB, all data would have been received from the device, and reading input from the operator would be completed. Since it was a READ Initial that was issued, however, it is necessary to issue a READ Continue, even if all data was received in the first message block; this will be necessary in order to read an EOT.

In the READ Continue at FINISH, another message block is read from the device.

Skipping or Activating a Terminal Entry: CHGNTRY

The CHGNTRY macro instruction allows a particular entry in a terminal list to be ignored during subsequent READ Initial operations, or, if it is being ignored, for it once again to be polled. (See the format of CHGNTRY in the general section "BTAM Macro Instruction Formats"; note that the general format is applicable to the remote 3270 and not the special form applicable to the local 2260 and local 3270.)

Each READ Initial macro instruction causes, for specific polling, all devices for which there are entries in a terminal list to be polled for readiness to transmit data, beginning at the entry specified in the entry operand. For general polling, the control unit specified in each terminal list entry is polled, beginning with the control unit whose entry address is specified in the entry operand. To have a particular entry (device for specific polling entries, control unit for general polling entries) ignored during one or more READ Initial operations, CHGNTRY may be issued with the SKIP operand specified. The device or control unit is then ignored for any pending or subsequent READ Initial operations. To reactivate an entry, CHGNTRY is issued with the ACTIVATE operand specified. When the entry is reactivated, it is used during subsequent READ Initial operations.

Terminating A Poll: RESETPL

The RESETPL macro instruction allows the application program to terminate a poll that is under way as the result of a READ Initial macro instruction. This may be desirable if, after a certain amount of time, no response has been received from a wraparound polling list, or because the program wants to perform another read or write operation on the line. (The format of RESETPL is described in the general section "BTAM Macro Instruction Format Descriptions.")

RESETPL is different from the READ Interrupt macro instruction. RESETPL terminates a READ Initial operation before a read operation has begun; READ Interrupt terminates a read operation after it has begun but before the last message block has been received.

Following a RESETPL, a WAIT macro instruction must be issued. Following the WAIT, a completion code of X'54' in the DECB indicates that the READ Initial operation was terminated before reading took place. A completion code other than X'54' indicates that a reading had already begun at the time RESETPL was issued and was completed successfully or unsuccessfully (depending on the completion code). In either case, the line is now free for another READ or WRITE macro instruction.

Read Modified Fields

This function is similar to Read Modified Fields after Operator Action except that the operation is directed to a specific device and is performed immediately; it does not depend on an attention-causing action by the terminal operator. The purpose of this process is to read all modified fields in the device buffer.

The Read Modified Fields function may be desirable to use in order to minimize input area requirements during polling. A READ Initial may be issued specifying an area length of 5, and only five bytes reserved as an input area by the application program. When data is read as a result of the READ Initial (only the index byte, STX, control unit and device address, and AID will be read), a larger area may now be obtained and the complete message read by using the Read Modified Fields function.

The program requests the Read Modified Fields function by issuing:

- A WRITE Initial Conversational or a WRITE Conversational macro instruction
- One or more READ Continue macro instructions

The WRITE Initial Conversational or WRITE Conversational (the latter is used if contact is already established with the device) is issued with the area operand containing the address of an output data stream containing:

STX	ESC	6	ETX

The length operand is specified as 4. The auxarea operand specifies the area to receive the input data stream; the auxlength operand specifies its length.

The WRITE macro instruction reads the first (or only) message block to be read. Only modified fields are read. READ Continues to read all modified fields in the terminal buffer must be issued until the EOT character is read (detected by the EOT-received flag in the DECB).

Note that the entry operand is applicable to WRITE Initial Conversational; it is not applicable to WRITE Conversational.

If BTAM is providing buffering, on completion of the WRITE, the auxlength and the auxarea address are contained in the DECB extension (see "Posting in the DECB" in the general section "Assembly Considerations.")

The input data stream will be in one of the formats shown for "Read Modified Fields After Operator Action." If a test request message is read, it will not be handled by BTAM and will appear in the input area.

An example of the Read Modified Fields function (omitting the associated WAIT macro instructions) is:

```
SENDESC   WRITE   DECB1,TV,DTFBT1,(AREA1,AUXAREA1),(4,256),,3,MF=E
            .
            .
            .
READMOR   READ    DECB1,TT,DTFBT1,AREA2,256,,3,MF=E
            .
            .
            .
```

Assuming that the Read Modified Fields function is being used in conjunction with a previous Read Modified Fields after Operator Action function, a WRITE Conversational is issued (WRITE Initial Conversational is not required because contact with the device is already established). The write operation will occur to the device (on relative line number 3 of the line group defined by DTFBT1) with which the program is in contact. The ESC message described earlier is located at AREA1; on receiving this message, the device transmits a maximum of 256 bytes of its message to AUXAREA1. DECB1 has been defined by a previous operation in which MF=L was specified.

On determining that all modified fields had been sent in the input message received as a result of the WRITE Conversational, a READ Continue is sent to read an EOT (thus allowing the device to be placed in control mode and to become receptive to future I/O operations). If, in the example, all modified fields had not been received in the first message block (an ETB was received instead of an ETX), additional READ Continues would have been required.

Performing this operation requires that DECBEXT=YES be specified in the BTMOD macro instruction.

Read Modified Fields from Position

This function reads all modified fields beginning at a specified position in the device buffer. As with the Read Modified Fields function, no operator attention-causing action is required. The process is directed to a specific device. It can be used in a manner similar to that for reading modified fields, described above, except here, the program selects only a certain portion of the screen (terminal buffer) to read, even though the terminal operator may have modified other portions of the screen. (Note that this process has a less important role in controlling the remote 3270 than the READ Modified Fields from Position macro instruction has for the local 3270. With a local 3270, the READ Modified Fields from Position macro instruction must be used to read successive blocks of a message from a terminal buffer; with a remote 3270, the READ Continue macro instruction fulfills this function.)

Read Modified Fields from Position is accomplished by issuing:

- A WRITE Initial or WRITE Continue macro instruction
- A WRITE Conversational macro instruction
- One or more READ Continue macro instructions

The WRITE Initial macro instruction is issued with the entry operand specifying the address of the terminal list entry for the device to be read. If contact is already established with the device (as it would be if this process is being used in conjunction with a READ Initial), a WRITE Continue macro instruction can be issued, and the entry operand is not required. In either case, the area operand must specify an output area containing:

STX	ESC	1	WCC	SBA	BUFFER ADDRESS	ETX

The output data stream can also include, following the WCC, data to be written to the terminal. The WCC should be set to inhibit resetting of modified data tags, and the last buffer address should be the position from which the read modified operation is to start.

The WRITE Initial or Continue must be followed with a WRITE Conversational macro instruction with the area operand specifying an output area containing:

STX	ESC	6	ETX

The WRITE Conversational reads a maximum of 256 bytes (including end-to-end and control characters) from the buffer location established by the previous WRITE Initial or Continue. The input data stream will be in one of the formats shown for "Read Modified Fields after Operator Action." If a test request message is read, it will not be handled by BTAM and will appear in the input area.

To read additional message blocks, one or more READ Continue macro instructions are issued. At least one READ Continue is required to read a final EOT character.

An example of the READ Modified Fields from Position function (omitting associated WAIT macro instructions) is:

```
PICKFLD   WRITE   DECB1,TT,,AREA1,8,,,MF=E
            .
            .
            .
FIRSTRD   WRITE   DECB1,TV,,(AREA2,AUXAREA1),(4,256),,,MF=E
            .
            .
            .
FINISH    READ    DECB1,TT,,AREA3,256,,,MF=E
            .
            .
            .
```

Assuming that a READ Initial has established contact with a device and it is desired to read modified fields beginning at some distance into the buffer, a WRITE Continue is issued, sending the output message with ESC code 1 from AREA1, which causes the buffer address to be set at the position from which the read operation is to begin. The dtfbt, entry, and rln operands have been specified in a previous READ or WRITE in which DECB1 was created and MF=L specified. The WRITE Conversational at FIRSTRD sends ESC code 6 and receives a maximum 256-byte data stream containing, in addition to end-to-end and control characters, the first message block of modified fields beginning at the buffer location specified in the preceding WRITE Continue. Assuming that the required data is not contained in the message block just read, a READ Continue is issued to receive another block. As soon as the program detects an ETX in the input data stream, a final READ Continue (not shown) is issued to read an EOT.

The program could have issued a READ Interrupt if it determined that all the information it required was contained in the message read in as the result of the WRITE Conversational; no further READ Continues would be required.

Performing this operation requires that DECBEXT=YES be specified in the BTMOD macro instruction.

Read Buffer

This function reads the entire contents of a specified terminal buffer, including modified and unmodified fields, attribute characters, and nulls (X'00'). It is intended primarily for diagnostic uses. The Read Buffer function is accomplished by issuing:

- A WRITE Initial Conversational or WRITE Conversational macro instruction
- One or more READ Continue macro instructions

If contact is not already established with the device, a WRITE Initial Conversational is issued with the entry operand specifying the entry in the addressing terminal list for the device. If, as the result of previous I/O operations, contact has already been established with the device, a WRITE Conversational is issued, and the entry operand can be omitted. In either case, the area operand should specify the address of an output area containing:

STX	ESC	2	ETX

After this message has been written to the device, the WRITE Initial Conversational or WRITE Conversational reads the first message block from the terminal buffer (since only a maximum of 256 bytes can be transmitted by one READ macro instruction, more read operations will be required to read the entire buffer). READ Continues are then issued to read as many remaining blocks of the terminal buffer as the program requires.

All data beginning at location 0 in the terminal buffer is read. In addition, a special character (SF) is inserted by the hardware into the input data stream to indicate the beginning of each field. The input data stream for the first message block, if the terminal buffer is formatted, appears as:

STX	CU ADDR	DEVICE ADDR	AID	CURSOR ADDRESS			SF	ATTR CHAR	TEXT	SF	ATTR CHAR		{ETB / ETX}

Subsequent message blocks appear as:

STX			SF	ATTR CHAR	TEXT	SF	ATTR CHAR		{ETB / ETX}

If the terminal buffer is unformatted, no SF characters are inserted since there are no fields. The input following the cursor address would consist of all character locations in the buffer, including nulls.

An example of the Read Buffer function (omitting the associated WAIT macro instructions) is:

```
READBUF   WRITE   DECB1,TIV,DTFBT1,(AREA1,AUXAREA1),(4,256),ENTRY1,2,MF=E
            .
            .
            .
READMOR   READ    DECB1,TT,DTFBT1,AREA1,256,,2,MF=E
            .
            .
            .
```

To analyze the complete contents of a particular terminal buffer, an application program issues a WRITE Initial Conversational macro instruction specifying the address of the device's entry, ENTRY1, in a terminal list. The operation is to occur on relative line number 2 of the line group defined by DTFBT1. The output message containing ESC code 2 is located in AREA1. The read part of the WRITE Initial Conversational reads a maximum of 256 bytes of data into input area AUXAREA1. DECB1 has been defined previously by a READ or WRITE macro instruction with MF=L specified.

After issuing the WRITE Initial Conversational and upon analyzing the contents of the message block read, the program can issue READ Continues to read successive 256-byte blocks of the terminal buffer (in the example, only one READ Continue is shown). After reading the final block, a READ Continue is required to read an EOT.

Performing this operation requires that DECBEXT=YES be specified in the BTMOD macro instruction.

Read Buffer from Position

This function reads the contents of a specified terminal buffer beginning at a specified buffer position. All fields, modified and unmodified, attribute characters, and nulls (X'00') are read. As with the Read Buffer function, Read Buffer from Position is intended primarily for diagnostic uses. The Read Buffer from Position function is requested by issuing:

- A WRITE Initial or a WRITE Continue macro instruction
- A WRITE Conversational macro instruction
- One or more READ Continue macro instructions

If contact is not already established with the device, a WRITE Initial is first issued with the entry operand specifying the address of the entry in an addressing terminal list for the device to be read. If contact has been established, a WRITE Continue is issued, and the entry operand can be omitted. In both cases, the area operand specifies the address of an output area containing:

STX	ESC	1	WCC	SBA	BUFFER ADDRESS	ETX

The output data stream could also include, following the WCC, data to be written to the terminal. The WCC should be set to inhibit resetting of modified data tags, and the buffer address should be the position from which the Read Buffer from Position operation is to begin.

The WRITE Initial or Continue is followed with a WRITE Conversational macro instruction with the area operand specifying an output area containing:

STX	ESC	2	ETX

The WRITE Conversational reads the first message block from the terminal buffer beginning at the location specified in the preceding WRITE Initial or Continue. All data beginning at the specified location is read. In addition, a special character (SF) is inserted into the input data stream to indicate the beginning of each field. The input data stream for the first message block, if the terminal buffer is formatted, appears as:

STX	CU ADDR	DEVICE ADDR	AID	CURSOR ADDRESS			SF	ATTR CHAR	TEXT	SF	ATTR CHAR	{ETB / ETX}

Subsequent message blocks appear as:

STX		SF	ATTR CHAR	TEXT	SF	ATTR CHAR		{ETB / ETX}

If the terminal buffer from the specified beginning location is unformatted, no SF characters can be inserted, since there are no fields. The input following the cursor address would consist of all character locations in the buffer, including nulls.

An example of the Read Buffer from Position function is:

```
SLECTBF   WRITE   DECB1,TI,DTFBT1,AREA1,8,ENTRY1,3,MF=E
            .
            .
            .
STARTRD   WRITE   DECB1,TV,DTFBT1,(AREA2,AUXAREA1),(4,256),,3,MF=E
            .
            .
            .
LAST      READ    DECB1,TT,DTFBT1,AREA3,256,,3,MF=E
            .
            .
            .
```

A WRITE Initial is issued to write the message in AREA1 to the device specified by ENTRY1. The message sets the buffer address from which the buffer contents are to be read. The device is located on relative line number 3 in the line group defined by DTFBT1. DECB1 has been defined by a previous READ or WRITE macro instruction with MF=L specified.

After the buffer address is set, a WRITE Conversational is issued, sending the ESC message at AREA2 to the device, and receiving the first message block in AUXAREA1. This block will contain, if the screen is formatted, all fields, modified and unmodified, attribute characters, and SF characters inserted in the data stream to delineate fields (plus other characters in the data stream format shown above).

To read more of the buffer or to read a final EOT, a READ Continue is issued. In the example, the READ Continue reads up to 256 bytes into AREA3.

Performing this operation requires that DECBEXT=YES be specified in the BTMOD macro instruction.

WRITE OPERATIONS

The application programmer writing to the remote 3270 can:

- Write
- Erase and write
- Erase unprotected fields
- Copy

These functions are performed by combining one or more WRITE macro instructions with an appropriate ESC command code in the output data stream. (See Figure 74 at the beginning of this section.) Each function is described below.

Write

The Write function writes a message to a terminal (display station or printer) buffer. To accomplish the Write function, the application programmer issues:

- A WRITE Initial or a WRITE Continue macro instruction
- A WRITE End-of-Transmission macro instruction

If contact has not been previously established with the selected device, a WRITE Initial macro instruction 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 Continue is issued and the entry operand is not applicable and should be omitted. In both cases, the area operand specifies an output area containing:

STX	ESC	1	WCC	ORDERS and TEXT	...	{ETX / ETB}

An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs. See IBM 3270 Information Display System Component Description, GA27-2749, for how to write the WCC, SBA, and other data stream characters.

To send the message in blocks instead of in one data stream, additional WRITE Continues may be issued with the output area in the format described above. To terminate the process, a WRITE End-of-Transmission (EOT) is issued.

An example of the Write function is:

```
SENDMSG   WRITE   DECB1,TI,DTFBT1,AREA1,300,ENTRY5,2,MF=E
            .
            .
            .
ENDWRT    WRITE   DECB1,TR,DTFBT1,,,,MF=E
            .
            .
            .
```

The WRITE Initial causes 300 bytes of data (including various end-to-end and control characters) to be sent to the device whose address is at ENTRY5. The device is on relative line number 2 of the line group defined by DTFBT1. DECB1 has been previously created by a READ or WRITE macro instruction that specified MF=L.

Since the 300-byte output area was sufficient to hold the entire message that the program wanted to send, no WRITE Continues are necessary. The WRITE End-of-Transmission at ENDWRT is issued to send an EOT. Since BTAM provides the EOT, the area and length operands should be omitted. Since contact is already established, the rln and entry operands can also be omitted.

Programming Note: If a terminal operator has made an entry and pressed the ENTER key but no READ Initial has yet been issued, a WRITE Initial to the display may nullify the operator input. This situation may be avoided by reserving areas of the display for operator input only (nothing will be written to these areas) and then setting the RMD (Reset Modified Data) bit to zero (meaning "do not reset modified data tags") in the WCC (Write Control Character) of the WRITE Initial message. Setting the RMD bit off in the WCC is required because if modified data tags are reset as part of the WRITE Initial, the pending attention will not be honored since there will be an indication that no fields have been modified.

Erase and Write

The Erase and Write function erases the buffer of a selected terminal and then writes a message to the terminal buffer. The erasure consists of changing each character location in the buffer to X'00'. With the message omitted, the process can be used just to erase the buffer. Erase and Write is accomplished by issuing:

- A WRITE Initial or a WRITE Continue macro instruction
- A WRITE End-of-Transmission macro instruction

If contact has not been previously established with the selected device, a WRITE Initial macro instruction 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 Continue is issued, and the entry operand is not applicable and should be omitted. In both cases, the area operand specifies an output area containing:

STX	ESC	5	WCC	ORDERS and TEXT	ETX / ETB

An SBA order sequence should follow immediately after the WCC so that the write operation can be retried if an error occurs.

To send the message in blocks instead of in one data stream from one large output area, subsequent WRITE Continues may be issued with ESC code 1 and WCC specified (see the "Write" function).

To terminate the Erase and Write process, a WRITE End-of-Transmission is issued.

The example shown under the Write function is applicable to the Erase and Write function. To erase and write, the data stream placed in the output area, AREA1, would contain an ESC code of 5. To simply erase the buffer, orders and text would be omitted from the data stream and the ETX character would follow the WCC.

Erase Unprotected Fields

The Erase Unprotected Fields function sets all unprotected fields in a selected terminal buffer to nulls (X'00'). It also resets the MDT (modified data tag) bits in the attribute characters of unprotected data fields to 0, restores the keyboard, resets the AID, and repositions the cursor to the first character location in the first unprotected field in the buffer. If the buffer is completely protected, the keyboard is restored, the AID reset, the cursor moved to location 0, and no erasure takes place. (See the Component Description for a description of the attribute and AID characters.) The Erase Unprotected Fields function is accomplished by issuing:

- A WRITE Initial or a WRITE Continue macro instruction
- A WRITE End-of-Transmission macro instruction

If contact has not been established with the device, a WRITE Initial is issued with the entry operand specifying the address of the selected device's entry in an addressing terminal list. If contact is already established, a WRITE Continue is issued with the entry operand omitted. In either case, the area operand specifies an output area containing:

STX	ESC	?	ETX

A WRITE End-of-Transmission is then issued.

An example of the Erase Unprotected Fields function is:

```
WIPE      WRITE   DECB1,TT,DTFBT1,AREA1,4,,3,MF=E
            .
            .
            .
EOT       WRITE   DECB1,TR,DTFBT1,,,,MF=E
```

It is assumed that the program has just read input after operator action and is in contact with the device. In order to allow the terminal operator to make another entry in the same fields, a WRITE Continue is issued to erase the fields just modified by the operator. AREA1 contains a message containing the ESC code of ? (question-mark character). The device is attached to a control unit on relative line number 3 of the line group defined by DTFBT1. The entry operand is not required. DECB1 has been defined previously by a READ or WRITE with MF=L specified.

To terminate the operation, a WRITE End-of-Transmission is issued.

Copy

The Copy function selects a device and copies into its buffer the contents of the buffer of another device attached to the same 3271 control unit. It can be used to transfer the contents of a display station screen to a printer to get a printout of the screen or to copy the contents of one screen on to another. The Copy function is performed by issuing:

- A WRITE Initial or a WRITE Continue macro instruction
- A WRITE End-of-Transmission macro instruction

The WRITE Initial (if contact has not yet been established) is issued with the entry operand specifying the address of the terminal list entry of the device to which data is to be copied and the area operand specifying an output area containing:

STX	ESC	7	CCC	FROM DEVICE ADDR	ETX

The WRITE Continue is issued if contact is already established; the area, length, entry, and rln operands should be omitted.

See the Component Description for a description of the CCC. The "from device address" is the one-byte address of the device from which the data is to be copied.

Note: If the output data stream is to be translated from EBCDIC to ASCII, the "from device address" must be given in EBCDIC.

An example of the Copy function is:

```
COPY        WRITE    DECB1,TI,DTFBT1,AREAFROM,6,ENTRYTO,4,MF=E
              .
              .
              .
EOT         WRITE    DECB1,TR,DTFBT1,,,,MF=E
              .
              .
              .
```

Assume that the program has received an indication from a device (as part of the input message on a Read Modified Fields after Operator Action function) that the screen contents are to be printed. The WRITE Initial is issued with ENTRYTO specifying the printer to which data is to be copied and the message at AREAFROM containing the address of the device (a display attached to the same control unit) from which data is to be copied.

The WRITE End-of-Transmission is issued to terminate the operation.

COMPLETION ANALYSIS AND ERROR RECOVERY

To analyze the results of a READ or WRITE macro instruction, the application program interprets:

- A return code in register 15 immediately after the READ or WRITE macro instruction is issued
- A completion code and other information in the DECB following the WAIT macro instruction associated with the READ or WRITE
- A status/sense message that may be read from a device if the completion code is X'41'

A return code of X'00' in register 15 indicates the initiation of an I/O operation. A return code other than X'00' means that I/O did not begin and a WAIT must not be issued. Return codes that may be received are described in the section "Posted Error Information" in the general section "Assembly Considerations." The meanings of the return codes suggest possible recovery action.

If the return code in register 15 is X'00', a WAIT macro instruction is issued to await and determine completion of the I/O operation. As a result of WAIT, a completion code is posted in the first byte of the DECB. This, along with other information that may be posted in the DECB is used to determine the outcome of an I/O operation. Recovery actions that the application program may take based on information in the DECB are shown in Figures 76 and 77.

Under certain conditions (shown in Figures 76 and 77), more information is required to determine the nature of the error or condition. This information is obtained by reading the status and sense bytes from the device. A READ Initial must be issued with the entry operand specifying an entry in a specific polling terminal list that contains the device's address. Possible recovery actions for various status and sense settings are shown in Figures 78 and 79.

Error recovery for errors occurring during line transmission rather than at the device is automatically attempted by BTAM's error recovery procedures for binary synchronous communication. These procedures are included in the BTAM logic for all binary synchronous devices. The application program is not notified of the error if recovery is successful; the program is notified only if recovery procedures are not successful. In addition, a message is sent to the system operator. See "Binary Synchronous Error Recovery Procedures" in the general section "Assembly Considerations."

BUFFERING

The application programmer can, in controlling input and output with a remote 3270, either define his own I/O areas or use the buffering services of BTAM. See "Buffering" in the general section "Assembly Considerations."

SIX-BIT STRUCTURED DATA

To allow for 6-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 Appendix K). 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 <u>IBM 3270 Information Display System Component Description</u>, GA27-2749.

CODE TRANSLATION

BTAM supports remote devices transmitting characters in either EBCDIC or ASCII (also referred to as "USASCII") code. Since DOS processes information in 8-bit EBCDIC code, the 6-bit ASCII code must be translated into EBCDIC on input; it must be translated from EBCDIC to ASCII just prior to output.

The application programmer processing ASCII data must code TRANSL=YES in his BTMOD macro instruction and translate incoming and outgoing data by using the TRNSLATE macro instruction, specifying the standard BTAM binary synchronous translation tables, RASA (input) and SASA (output). In addition, so that polling and addressing sequences are recognized, the ASCII representation of the graphic characters forming the polling and addressing sequences for each device must be entered in the lists defined with the DFTRMLST macro instruction. The hexadecimal representation to be used in defining terminal list entries for each control unit and device for output operations and for identifying control units and devices on input operations is shown in Appendix J.

<u>Examples</u>: Line 1 of Figure 80 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 in transmission code.) Line 2 of Figure 80

indicates the characters contained in the message. Line 3 shows the message after 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 81 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.

ONLINE TESTING: (RFT)

See "Online Terminal Test for Binary Synchronous Communication" in the general section "Assembly Considerations." The following information is supplementary for the remote 3270:

Only requests for tests (RFTs) initiated from a terminal are serviced; requests for tests initiated by the application program are not serviced by BTAM for the remote 3270 (the ONLTST macro instruction is therefore not applicable).

FOLLOWING A WAIT FOR THIS MACRO INSTRUCTION...	IF THE DECB CONTAINS... Completion Code Byte 0	Response Info 8	TCU Sense 16	Flag Byte 24	TP Code 28	Error Info 29	THIS MEANS...	DO ACTION NUMBER...
READ Initial (TI)	7F						Normal completion - no error, text received	1 or 5
	7F	Bit 3					Normal completion - no error, status/sense message received	1 or 5
	41		Bit 7		03		No response to polling received	2 or 4
	41			Bit 1	11	Bit 6	Error or exceptional condition - Text ending with ENQ received	3
	42						Normal completion - RFT message received	6
	54						Normal completion - no error, polling ended without positive response	2 or none
READ Continue (TT)	7F						Normal completion - no error, text received	1 or 5
	7F			Bit 1			Normal completion - no error, device operation ended, EOT received	None
	7F	Bit 3					Normal completion - no error, sense/status message received	1 or 5
	42						Normal completion - RFT message received	6
	62						Acknowledgement not recognized by remote device, ENQ received	1

ACTION		
	1	Issue a READ Continue (TT) to acknowledge text received without error.
	2	Issue a READ Initial (TI) to retry the operation.
	3	Issue a READ Initial (TI) to receive the status/sense message, using the specific poll address for the device from which the message is to be received.
	4	Issue a WRITE EOT (TR) to reset the line.
	5	Issue a READ Interrupt (TRV) to stop general polling. Examine the status/sense byte, if required (see Figure 78).
	6	Issue a TWAIT with TERMTST specified to allow RFT processing to continue.

Figure 76. Suggested Actions Following Completion of Remote 3270 Read Operations

FOLLOWING A WAIT FOR THIS MACRO INSTRUCTION...	IF THE DECB CONTAINS... Completion Code Byte 0	Response Info 8	TCU Sense 16	Flag Byte 24	TP Code 28	Error Info 29	THIS MEANS...	DO ACTION NUMBER...
WRITE Initial (TI)	7F						Normal completion - no error	4 or 5
	41		Bit 7		06		No response to addressing received	2
	41		Bit 7		20		No response to text received	2
	41			Bit 1	20		EOT response to text received	1
	41				20	Bit 7	NAK response to text received (ERP retries, then EOT)	2
	61	Bit 6			06		RVI response to addressing received	4, then 1
	61				06		WACK response to addressing received	2 or 4
	61				20		WACK response to text received[1]	4
WRITE Continue (TT)	7F						Normal completion - no error	4 or 5
	41		Bit 7		20		No response to text received	5
	41			Bit 1	20		EOT response to text received	1
	41				20	Bit 7	NAK response to text received (ERP retries, then EOT)	2
	61				20		WACK response to text received[1]	4
WRITE Initial Conversational (TIV)	7F						Normal completion - no error	7
	41		Bit 7		06		No response to addressing received	3
	41		Bit 7		11		No response to text received	3
	41			Bit 1	11		EOT response to text received	1
	41				11	Bit 7	NAK response to text received (ERP retries, then EOT)	3
	41			Bit 1	11	Bit 6	Text ending with ENQ received	1
	61	Bit 6			06		RVI response to addressing received	4, then 1
	61				06		WACK response to addressing received	3 or 4
WRITE Conversational (TV)	7F						Normal Completion - no error	7
	41		Bit 7		11		No response to text received	6
	41			Bit 1	11		EOT response to text received	1
	41				11	Bit 7	NAK response to text received (ERP retries, EOT)	3
	41			Bit 1	11	Bit 6	Text ending with ENQ received	1

ACTION		
	1	Issue a READ Initial (TI) using a specific poll address for the device where completion was posted to receive the status/sense message.
	2	Issue a WRITE Initial (TI) to retry the operation.
	3	Issue a WRITE Initial Conversational (TIV) to retry the operation.
	4	Issue a WRITE EOT (TR) to reset the line.
	5	Issue a WRITE Continue (TT) to retry or to transmit more data.
	6	Issue a WRITE Conversational (TV) to retry.
	7	Issue a READ Continue (TT) to pick up data.

Figure 77. Suggested Actions Following Completion of Remote 3270 Write Operations

IF THE STATUS/SENSE BYTES CONTAIN...	MNEMONIC	THIS MEANS...	APPLICABLE TO...[1]	SEE THIS ACTION NUMBER IN FIGURE 79.
X'4050'	IR	Intervention Required for one of these reasons: • A command attempted to start a printer but found it not ready (out of paper, "hung," etc.). The printout is suppressed. • The power was off on the printer.	3271, 3275	3A
		• The control unit received a selection addressing sequence or specific poll sequence for a device that is "unavailable" or went "not ready" during a printout. (A general poll does not respond to an "unavailable" or "not ready" indication and proceeds to the next device.) • The control unit receives a command other than Diagnostic Read or Write, for a device that the CU has logged as "unavailable"/"not ready."	3271	
		• The printer went "not ready" during a printout.	3275	
X'4060'	CR	Command Reject. Receipt of an invalid or illegal 3270 channel command (for example, NOP, Sense, Select, or Copy if not installed).	3271, 3275	4
X'40C1'	OC	Operation Check. Any of the following: • An illegal buffer address or an incomplete order sequence received on a Write or Erase/Write command.	3271, 3275	4
		• CCC or "from" address not received on a Copy command.	3271	
		• Invalid command sequence (ESC is not received in second data character position). • An I/O interface "over-run" is detected. This occurs during a command when a data byte is presented to the control unit by the TCU before the operation required by the previous data byte has completed.	3271, 3275	
X'40C2'	CC	Control Check. Timeout. A device has failed to respond to control unit communications within a specified period of time.	3271	2A
X'40C3'	CC,OC	Control Check, Operation Check. The condition above was detected while the control unit was executing an operation with the "from" device during a Copy command.	3271	1B
X'40C4'	DC	Data Check. Either one of: • An "internal" parity check or a cursor check occurred in either the control unit or device buffer	3271, 3275	2A
		• A "transmit" parity check occurred on data sent between the device and the control unit.	3271	
X'40C6'	DC, OC	Data Check, Operation Check. A condition above occurred while the control unit was executing an operation with the "from" device during a Copy command.	3271	1B
X'40D1'	IR, OC	Intervention Required, Operation Check. Either of: • A Copy command contains a "from" address specifying an "unavailable" device. • An IR condition (see IR) is detected while the CU is executing an operation with the "from" device during a Copy command.	3271	3B
X'4C40'	DB, US	Device Busy, Unit Specify. The addressed device is presently busy executing an operation or a busy condition was detected previously by a command.	3271, 3275	10

Figure 78. Analyzing a Remote 3270 Status/Sense Message (Part 1 of 2)

IF THE STATUS/SENSE BYTES CONTAIN...	MNEMONIC	THIS MEANS...	APPLICABLE TO...[1]	SEE THIS ACTION NUMBER IN FIGURE 79.
X'4E40'	DB,US,DE	Device Busy, Unit Specify, Device End. A busy condition was defected. However, a Device End indication means the device is no longer busy and the operation should be retried.	3271,3275	2A
X'C140'	TC	Detection of a BSC error on the TCU transmission.	3275	12
X'C4C1'	OC,US	Operation Check, Unit Specify. A "from" address on a Copy Command specified a device with a "locked" buffer. (The device was not authorized to be copied from.)	3271	13
X'C240'	DE	Device End. Signals that a previously detected "busy" condition has gone "not busy," a "not ready" device has gone "ready," or a "not available" device is now "available."	3271, 3275	N/A
X'C250'	IR, DE	Same as X'4050'.		3A
X'C2C4'	DC, DE	Same as X'40C4'.		2A
X'C2C8'	EC, DE	Equipment Check, Device End. A mechanical "hang" or a character generator read out error on the printer.	3275	7
X'C2D8'	IR,EC,DE	Same as above.	3275	7
X'C4C4'	DC, US	Same as X'40C4'.		2A
X'C4C5'	DC,OC,US	Same as X'40C4' occurring while the CU was executing an operation with the "from" device on a COPY Command.	3271	2B
X'C6C4'	DC,US,DE	Same as X'40C4'.	3271	8
X'C6C8'	EC,US,DE	Same as X'C2C8'.	3271	7
X'C6D8'	IR,EC, US,DE	Same as X'C2C8'.	3271	7
X'C840'	DB	The addressed device is presently "busy" executing an operation or a "busy" condition was detected previously by a command. The device is or was busy executing a printout, accepting data from an identification card reader or performing keyboard functions. Set under either of these conditions: • A command is addressed to a busy device. • A specific poll sequence makes a status poll to a device and finds it busy.	3271, 3275	9
X'C8C1'	DB, OC	The same as X'C840' and the CU was executing an operation with the "from" device during a Copy Command.	3271	11

[1]In analyzing a status/sense message, the programmer may need to determine whether the device is a 3271 or a 3275. One way to do this is to compare the device specified in the status/sense message with a list of all 3275s; if the device is not found in this list, a 3271 can be assumed.

Figure 78. Analyzing a Remote 3270 Status/Sense Message (Part 2 of 2)

The application programmer who anticipates that the RFT online test facility will be used must ensure that his input area (or, if buffering is used, the first buffer if there is more than one) will accommodate both the RFT message and the largest test pattern. The largest test pattern with the RFT message requires a 300-byte area; the minimum input area size in this case is 300 bytes (if buffering is used, the BUFL operand of DTFBT must specify at least 300).

An RFT message (a request from a terminal operator to receive an online test) can be received when a READ Initial macro instruction is issued using either a general or specific poll or, when a general poll is underway and a new device is encountered, when a READ Continue is issued. Following the WAIT macro instruction associated with the READ, a completion code of X'42' in the DECB indicates that an RFT message has been received. The program must issue a TWAIT macro instruction with the TERMTST operand specified. (More simply, the program can use the TWAIT with TERMTST specified where it would normally use a WAIT.) As a result of the TWAIT, the selected test is written to the device the requested number of times. If the RFT message is received on a READ Initial, after the test has been written, polling will be

Action Number	Action
1A	Execute a new address selection sequence and retransmit the message starting with the command sequence which was being executed when the error occurred. If the operation is not successful after two retries, consider this an unrecoverable error and follow procedure 5A.
1B	Same as 1A except follow procedure 5B after two retries.
1C	Same as 1A except a new address selection is not performed and this message is retransmitted as part of a present device selection.
1D	Same as 1A except retransmit the entire failing chain of commands.
2A	It is suggested that the user reconstruct the entire screen buffer image if this is possible and retry the failing chain of commands (within the BSC sequence of operations). If the information in the screen buffer is such that it cannot or need not be reconstructed, the operation may still be retried. If the operation is not successful after 3 retries, consider this an unrecoverable error and follow procedure 5A.
2B	The error occurred during the execution of a Copy command. Execute procedure 2A except that it is the buffer of the "from" device specified by the Copy command that should be reconstructed. After three retries, execute procedure 5B.
3A	The error indicates that the printer is out of paper, has the cover open, or has the print mechanism "hung"; or the device is unavailable. Wait for the display operator or system operator to intervene and mechanically ready the printer. Then retry the printout by issuing a WRITE with the WCC and no data stream. (There is no data error and the data is still intact in the device buffer and can be sensed.) Or execute procedure 2A.
3B	The error indicates that the "from" device specified in a Copy command is "unavailable." The device address associated with the error status/sense information is not the one requiring readying. The device requiring corrective action is the "from" device specified in the Copy command. This "from" device should be determined and made ready. Then execute procedure 1B.
4	An unrecoverable programming error has occurred. Examine the data stream to locate the problem.
5A	Request maintenance on the device giving the trouble. After repair, attempt to reconstruct the screen buffer image if possible, starting with an Erase/Write command in order to correct a missing or multiple cursor situation in the device buffer. Retry the failing CCWs as done in the procedure previous to 5A.
5B	The "from" device specified by the Copy command in the failing chain of CCWs is malfunctioning. The "from" device should be determined from the data stream information, and maintenance should be requested on the device. After repair, reconstruct the screen buffer image if possible. The sequence of commands used to reconstruct the image should start with an Erase/Write command to correct a missing or multiple cursor situation in the device buffer. Retry failing CCW s as done in the procedure previous to 5B.
6	Retransmit the last block of the message.
7	The error has occurred during a printing and indicates either a character generator readout error or a print mechanism hang. There is no data error. The proper error recovery procedure is application dependent, since the user may or may not want a new printout. If a new printout is required, follow procedure 3A.

Figure 79. Suggested Actions Based on Remote Status/Sense Messages (Part 1 of 2)

Action Number	Action
8	A data error occurred in the device buffer during printing; follow procedure 2A.
9	A specific poll detected that the addressed device is busy. Periodically issue a specific poll to pick up the device end status/sense bit which is sent by the device to the TCU when the device becomes not busy (unless this status change is detected on a selection addressing sequence).
10	A command was erroneously addressed to a busy device. Periodically issue a READ Initial with a specific poll to pick up the device end status/sense bit, which is sent by the device to the TCU when the device becomes not busy. Then follow procedure 2A.
11	This indicates that in attempting to execute a Copy command the "from" device was found to be busy. Execute procedure 1A when the "from" device is not busy. (A specific poll read picks up the device end status/sense bit.) The device address associated with the status/sense message is the address of the "to" device and not that of the busy "from" device.
12	A BSC error was detected during a text transmission from the TCU. Follow procedure 2A if the failing command is a Write command which has a data stream of more than 1 byte or if it is in a chain of commands and one of the previous commands in the chain is a Write command without an SBA order immediately following the WCC character. In all other cases, follow procedure 1D. If, after following the above retry procedure, the problem is not corrected, follow procedure 5A.
13	An unauthorized attempt was made to read data. An effort was made to execute a Copy command but access to the "from" device data was not authorized. The device address associated with the error status/sense bits is that of the "to" device.

Figure 79. Suggested Actions Based on Remote Status/Sense Messages (Part 2 of 2)

reestablished. Following the TWAIT, the completion code will be other than X'42', and any data read from some device as a result of the polling operation will be in the input area specified by the READ Initial. If the RFT message is received on a READ Continue, after the test has been performed, another read operation does not occur, and the DECB is posted to indicate that an EOT response was received to the READ Continue. If the program wishes the general poll to continue, a READ Initial must be issued to reestablish contact.

HOW TO REQUEST AN ONLINE TEST (RFT) FROM A REMOTE 3270

To initiate an online test, a display station operator must:

1. Assure himself that the screen is unformatted (one way to do this is to press the CLEAR key, then the RESET key).
2. With the cursor now at location 0, type in a message with the format:

X	X	Y	Y	N	ADDR			

where XX is a number from 23 through 34 specifying the desired test; YY is a number from 01 through 99 specifying the number of times the test is to be written to the device (if the test is a printer, the test can only be sent one time); N is the number 4, indicating the length of ADDR; where ADDR is a sequence of four alphameric characters specifying the control unit and device to which the test is to be sent. Alphabetic characters must be typed in uppercase. To obtain the correct control unit and device characters, see Appendix J. Because double addressing is used, each control unit and device character must be repeated. For example, to send a test message to control unit 0, device 1, in EBCDIC transmission, the operator would press the hyphen key twice and type two A's.

3. Press the TEST REQUEST key.

The test should now appear at the selected display station or printer.

ONLINE TESTING: OLTEP

Remote 3270 devices can also be tested from the system console by invoking the Online Test Executive Program (OLTEP). Unlike the local 3270 where OLTEP and the application can run concurrently, a BTAM application program using remote 3270 devices must be terminated in order to test the devices with OLTEP. (See "Online Testing: Concurrent Use of OLTEP" in the section on the local 3270 in the general section "Local Device-Dependent Considerations.")

2260 TO 3270 CONVERSION

A special facility is provided in BTAM to allow BTAM application programs written for local or remote 2260s to be modified for use with the local or remote 3270. Although not all features of the 3270 can be utilized, this facility does have the advantage of allowing quick conversion from the 2260 to the 3270 without requiring the 2260 processing portions of the application program to be rewritten.

Essentially, this facility provides a macro instruction, SCANREQ, that will translate the 3270 data stream into the 2260 format on input and translate the 2260 data stream format to that of the 3270 before output. This facility is described fully in <u>IBM 2260 BTAM and 2260 GAM to IBM 3270 BTAM Conversion Guide</u>, GC27-6975.

<u>IBM 3735 PROGRAMMABLE BUFFERED TERMINAL</u>

See the <u>IBM 3735 Programmer's Guide</u>, GC30-3001, and <u>IBM 3735 Concepts and Applications</u>, GC27-3043.

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 80. Example of 3270 Message Translation from ASCII to EBCDIC

1 Message content	Start of text	Escape command: 1		WCC	SBA order	Buffer address: 0064		SF order	Attr Char	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	{IGS / 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 81. Example of 3270 Message Translation from EBCDIC to ASCII

BSC CHANNEL PROGRAMS

POINT-TO-POINT CONTENTION

Read Initial (TI)

This macro instruction is issued to monitor the line for the Inquiry (ENQ) signal from the remote device. When this signal is received and acknowledged, the first message from the remote device is read into the specified input area.

The channel program generated and executed is:

1. Prepare
2. Read ENQ
3. Write positive acknowledgment -- ACK-0
4. Read message

Read Initial Inquiry (TIQ)

This macro instruction permits the problem program (at initial contact time) to monitor the line for a request to transmit from the remote device. When the ENQ character with which the remote device is bidding for the line is detected, the operation is posted as complete. If the user does not wish to read data, he may issue a WRITE Wait-Before-Transmitting macro instruction, or if he wishes to receive data he may issue a READ Continue macro instruction.

The channel program generated and executed is:

1. Prepare
2. Read ENQ

Note: The READ TIQ and READ TI macro instruction reinitialize the alternating acknowledgment information used by BTAM to check that the appropriate alternating acknowledgments are sent and received.

Read Inquiry (TQ)

This macro instruction may be used when the CPU, as the receiving station, times out on a text read command in a READ macro channel program. As the receiving station, the CPU does not initiate recovery from such a time-out, but generally issues the READ Inquiry macro instruction awaiting an ENQ from the remote device requesting the CPU to transmit its last response. For this reason, READ Inquiry does not cause BTAM to reinitialize the information on alternating acknowledgments. When the ENQ is received, the appropriate macro instruction can then be executed to retransmit the last response of the CPU.

The channel program generated and executed is:

1. Read ENQ

Note: The user should issue this macro only if he desires more retries than are provided automatically by BTAM ERP.

Read Continue (TT)

This macro instruction is issued to read messages from the remote device after initial contact has been successfully established.

The channel program generated and executed is:

1. Write positive acknowledgment
2. Read message -- data or an EOT

Read Continue with Leading Graphics (TTL)

This macro instruction is used to perform the same basic function as READ Continue, that is, acknowledge the last message received and read the next message into main storage. In addition, this macro instruction permits the sending of up to seven graphic characters (i.e., non-control characters) preceding the BTAM-supplied positive acknowledgment sequence (either ACK-0 or ACK-1).

If, in the BTMOD macro instruction, DECBEXT is coded as 'NO', or DECBEXT is omitted, the leading graphics must be supplied by the user in an output area of the following format:

- The first byte (count byte) of the output area must contain a binary count of the graphics to be sent. This count must not be zero. A count of zero results in a channel program check due to an invalid count.
- The bytes following the count byte must contain the graphics in transmission code.

The user must point to the count byte by the entry operand in the macro instruction. While the user may place the count byte and graphics in a buffer, the main storage address specified by the entry operand must be that of the count byte itself.

If DECBEXT is coded as 'YES' in the BTMOD macro instruction, the address of the leading graphics must be supplied by using the DECB extension.

Note: Leading graphics can serve any function required by the application program. The only restriction is that the bit configuration of the graphics not coincide with the bit configuration of defined data link control characters (e.g., ENQ, STX).

The channel program generated and executed is:

1. Write graphics
2. Write positive acknowledgment
3. Read message

Read Repeat (TP)

This macro instruction is issued following an unsuccessful READ Initial or READ Continue to request retransmission of the last block of data from the remote device.

The channel program generated and executed is:

1. Write negative acknowledgment -- NAK
2. Read message

Note: This macro instruction should be issued only if the user desires more retries than BTAM ERP has already attempted automatically.

Read Repeat with Leading Graphics (TPL)

This macro instruction is issued following an unsuccessful READ Initial or READ Continue to request retransmission of the last block of data from the remote device. In addition, the user is allowed to send leading graphics with the BTAM-supplied negative acknowledgment (NAK). The leading graphics must be supplied by the user as explained for READ Continue with Leading Graphics. For further information, refer to the discussion of the MODE Operand of the CONTROL macro instruction.

The channel program generated and executed is:

1. Write graphics
2. Write NAK
3. Read message

Read Interrupt (TRV)

This macro instrcution is issued to allow the CPU, as the receiving station, to request the transmitting device not to send any additional message blocks, so that a message can be sent to the remote device by the CPU. The RVI character is sent to the transmitting device, indicating that the CPU has correctly received the previous message, but does not wish to receive any additional message blocks. The transmitting device will transmit any buffered message blocks that could be lost when a message is received from the CPU. If the CPU receives an EOT, it can then transmit data without data loss. If an EOT is not received, the CPU should read text by issuing READ Continue macro instructions until an EOT is received. When the EOT is recieved, the CPU may transmit. The RVI character must not be transmitted by the CPU twice in succession without transmitting an intervening normal acknowledgment (ACK-0, ACK-1, NAK) unless ENQ is received for retry purposes.

The channel program generated and executed is:

1. Write RVI sequence
2. Read message or response

Write Initial (TI)

This macro instruction is used to bid for the line and to establish the direction of transmission. The CPU transmits the first message to the remote device.

The channel program generated and executed is:

1. Write ENQ
2. Read response to ENQ
3. Write message
4. Read response to message

Write Initial Transparent Block (TIE) and Write Initial Transparent Text (TIX)

These two macro instructions are used when the user desires to gain control of the line and send data over the line in transparent mode. BTAM provides either the 2-character end-of-transparent-text sequence (DLE,ETX) for the TIX optype. This means the user is responsible only for the 2-character start-of-transparent-text sequence (DLE,STX) appearing in his message. The length operand specifies only the number of bytes in the user's output area and does not include the end characters provided by BTAM.

The channel program generated and executed is;

1. Write ENQ
2. Read response to ENQ

Word 1		Address	of input	area	(right adjusted)
Word 2		Length	of input	area	(right adjusted)

Figure 82. Parameter List for Conversational WRITEs

3. Write message
4. Write termination characters -- DEL, ETB or DLE,ETX
5. Read response to checking

Note: Transparent mode permits sending in text any character or bit configuration without regard to coincidence of the bit configuration with a defined data link control character. Therefore, such characters as EOT or ETB can be transmitted in transparent mode and will not cause any control function to be performed by either the transmitting or receiving binary synchronous control unit.

Write Initial Conversational (TIV)

This macro instruction is used when the user desires to bid for the line, send a message to the remote device, and receive a message in response. Since the CPU controls the direction of transmission by successfully bidding for the line, line discipline permits the remote device to send a message only if the CPU is prepared to receive that message. The CPU controlling the line can only receive the first message from the remote device by means of one of the conversational WRITEs. WRITE Initial Conversational provides this facility at initial contact time.

It is the user's responsibility to coordinate the mutual exchange of messages (conversational) between the CPUs within the problem programs for the two devices. This is properly a function of message processing and is not performed by BTAM.

In coding any one of the conversational WRITEs, the user must specify his output message by the area and length operands, and, if the inlist technique is being used, define the equivalent parameters for the expected input message by specifying the address of a parameter list by the entry operand. The parameter list contains two full words with the format shown in Figure 82.

If the address specified in Word 1 is zero, it is assumed that the user is requesting BTAM to read the input message into buffers. Word 2, in this case, must specify the maximum length of the expected message. BTAM performs the buffering as it does for the READ macro instruction when area is coded 'S' in that macro instruction. The only exception is that the address of the first buffer (into which the message is read) is placed in Word 1 of the parameter list rather than into the area field of the DECB.

If the DECB extension is being used, it is required that the user specify his input parameters by the auxarea and auxlength operands. (See the section on DECB extension for WRITE.) Auxarea may be coded as 'S'. Hence, BTAM will perform buffering in the same manner as any other READ. The address of the first buffer in the chain will be placed in the auxarea field of the DECB extension.

The channel program generated and executed is:

1. Write ENQ
2. Read response -- ACK-0
3. Write message
4. Read response -- alternating acknowledgment or message

There are certain restrictions on the use of this and other conversational WRITEs.

- The first conversational WRITE issued by the CPU controlling the line permits the remote device to send its first message. Coordination of the conversation between the CPUs is a function of message processing and not of BTAM.
- The user is cautioned that a conversational WRITE should be followed by a READ Continue in normal operation (no errors) and never by another WRITE if a message (rather than ACK-0 or ACK-1) has been received on the conversational WRITE. This restriction must be followed in order to safeguard against losing a message without being able to recover.
- The message sent via the conversational WRITE must end with ETX and never ETB, since ETB implies that more blocks are to be sent. This is a rule of the line control procedures.
- An additional use of the conversational WRITEs is to receive a positive or negative acknowledgment with leading

graphics in response to the message transmitted. Such a response can only be received into a user-specified input area since the normal response field (in the DECB) is only two bytes. If leading graphics with NAK are received, ERP will not automatically retransmit the previously-sent message. It is not the function of BTAM to analyze leading graphics. Instead this function is part of the user's application.

Write Initial Transparent Conversational (TIXV)

This macro instruction is issued to establish initial contact, transmit a message in transparent mode, and receive a (conversational) text reply.

The channel program generated and executed is:

1. Write ENQ
2. Read response -- ACK-0
3. Write message
4. Write termination characters -- DLE ETX
5. Read response -- positive acknowledgment or message

Note: If the inlist technique is used, see WRITE Initial Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of the DECB extension for WRITE.

Write Continue (TT)

This macro instruction is issued to transmit a message after initial contact has already been established.

The channel program generated and executed is:

1. Write message
2. Read response -- ACK-0 or ACK-1

Write Transparent Block (TE) and Write Transparent Text (TX)

These two macro instructions are issued to transmit text in transparent mode after initial contact has been established. The channel programs for the two macros are identical with the exception of the end characters transmitted by command (2).

The channel program generated and executed is:

1. Write message
2. Write termination characters -- DLE, ETB or DLE,ETX
3. Read response -- ACK-0 or ACK-1

Write Conversational (TV)

This macro instruction is issued after initial contact has been established. It enables the CPU to receive a message in response to a message sent.

The channel program generated and executed is:

1. Write message
2. Read response or message

If an acknowledgment is received instead of a message, it is checked by BTAM for correctness, and a completion code 7F is posted.

Note: If the inlist technique is used, see WRITE Initial Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of DECB extension for WRITE.

Write Transparent Conversational (TXV)

This macro instruction is issued to transmit a message in transparent mode and to receive a message in reply from the remote device. Initial contact must have been established previously.

The channel program generated and executed is:

1. Write message
2. Write termination characters -- DLE,ETX
3. Read response or message

Note: If the inlist technique is used, see WRITE Initial Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of DECB extension for WRITE.

Write Inquiry (TQ)

This macro instruction is issued when no response or an invalid response is received from the remote device in reply to a message. The remote device is requested to transmit its last response (ACK-0, ACK-1, NAK, or a conversational text reply).

The channel program generated and exectued is:

1. Write ENQ

2. Read response

The response is read into the user-supplied input area. The user must specify both the input area and length in a parameter list (as described for WRITE Initial Conversational using the inlist technique) or by the auxarea and auxlength operands (as described in the section on DECB extension for WRITE).

Note: This macro instruction should be issued only when the user desires more retries than BTAM ERP automatically provides (the number of retries is equal to that specified in the RETRY operand in DTFBT).

Write Wait-Before-Transmitting (TW)

This macro instruction is issued to inform the remote device that a temporary condition exists, requiring a delay before the remote device continues transmission. This may result, for example, from a temporary lack of buffers in the CPU for receiving messages.

This macro instruction may be issued following completion of a READ Initial Inquiry that had been issued by the slave CPU in recovering from the contention situation.

This macro instruction may also be issued after a message has been read, to defer sending a negative or positive response. Again, this could be for the reason that some temporary condition exists making immediate reading of a message undesirable.

The expected response from the remote device is the ENQ character, which, if received, causes BTAM to post normal completion. Another response that can occur is the EOT character that causes BTAM to post completion code 41 and turn ON the 1-bit of the DECB flag byte. Text messages are illegal responses to WACK.

After completion, the user may reissue the WRITE WACK macro instruction, if still not ready, or he may issue the appropriate macro instruction to send his deferred response or message.

The channel program generated and executed is:

1. Write WACK

2. Prepare

3. Read ENQ

Write End-of-Transmission (TR)

This macro instruction is issued to inform the remote device that transmission is ended on the line. It may also be used to end transmission when irrecoverable errors exist. Further transmission on the line requires that the READ Initial or Write Initial be used to reestablish contact between the CPU and the remote device.

The channel program generated and executed is:

1. Write EOT

Control Mode (TM)

This macro instruction is issued to set the Error Information Byte mode of the 2701/2703.

The channel program generated and executed is:

1. Set mode

POINT-TO-POINT DIAL

Read Initial (TI)

The READ Initial macro instruction is used to establish initial contact with the remote device and to read a message from that device.

There are eight separate channel programs for READ Initial: four for the case in which an answering list is specified by the entry operand, and four for the case in which a calling list is specified.

Programming Note: The printer must be ready when a READ Initial with DIALST is used for the 2780.

Answering - No ID Verification: The channel program generated and executed is:

1. Enable

2. Read ENQ

3. Write ACK-0

4. Read message

Answering - IAM Procedure: The code RIX must be coded in the FEATURE sublist of the DFTBT macro instruction for the line group. The channel program generated and executed is:

1. Enable

2. Read ID, ENQ into the terminal list

3. Write ACK-0

4. Read message

Answering - WRU Procedure: The code RXW must be coded in the FEATURE sublist of the DTFBT macro instruction for the line group.

The channel program generated and executed is:

1. Enable

2. Read ENQ

3. Write ID sequence

4. Write ACK-0

5. Read message

Answering - IAM/WRU Procedure: The code RIW must be coded in the FEATURE sublist of the DTFBT macro instruction for the line group.

The channel program generated and executed is:

1. Enable

2. READ ID,ENQ into the terminal list

3. Write ID sequence

4. Write ACK-0

5. Read message

Calling - no ID Verification: When performing the calling function, the READ Initial dials the telephone number of the remote device, establishes valid line connection, indicates to the remote device that the call was made to receive messages, and then receives the first message (if any) that the remote device transmits.

The channel program generated and executed is:

1. Dial -- user-specified number

2. Write ENQ

3. Read response -- ACK-0

4. Write EOT

5. Read response -- ENQ,EOT, or DLE EOT

6. Write ACK-0

7. Read message

The response to EOT is read into the DECB response field. Under normal circumstances, this will be the ENQ character that indicates that the remote device wants to control the line in order to transmit to the CPU. In this case, BTAM restarts the channel program at command (6). Two other responses -- EOT and DLE,EOT -- result in BTAM posting the operation as complete with completion code 7F and the appropriate bit in the DECB flag byte ON. EOT indicates that the remote device has nothing to transmit and is yielding the right-to-transmit to the CPU. DLE,EOT means the remote device has nothing to send and is disconnecting the line.

Calling - IAM Procedure: The code SIX must be coded in the FEATURE sublist of the DTFBT macro instruction defining the line group.

The channel program generated and executed is:

1. Dial -- user-specified number

2. Write ID from terminal list

3. Write ENQ

4. Read response -- ACK-0

5. Write EOT

6. Read response -- ENQ,EOT or DLE EOT

7. Write ACK-0

8. Read message

The response to EOT is handled exactly as it is for calling with no ID verification.

Calling - WRU Procedure: The code SXW must be coded in the FEATURE sublist of the DTFBT macro instruction defining the line group.

The channel program generated and executed is:

1. Dial -- user-specified number

2. Write ENQ

3. Read ID, ACK-0 into terminal list

4. Write EOT

5. Read response -- ENQ, EOT, or DLE EOT

6. Write ACK-0

7. Read message

The response to EOT is handled exactly as it is for calling with no ID verification.

alling - IAM/WRU Procedure: The code SIW ust be coded in the FEATURE sublist of the TFBT macro instruction defining the line roup.

The channel program generated and xecuted is:

1. Dial -- user-specified number
2. Write ID sequence from terminal list
3. Write ENQ
4. Read ID,ACK-0 into terminal list
5. Write EOT
6. Read response -- ENQ, EOT, or DLE EOT
7. Write ACK-0
8. Read message

The response to EOT is handled exactly as it is for calling with no ID verification.

Read Connect (TIC)

This macro allows initial contact to be established with a remote BSC station and performs 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 in the first fullword of the list, and 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 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'). If the received sequence does not match any of the authorized [ID] ENQ sequences, BTAM determines whether ENQ alone, an invalid sequence, or DLE EOT was received. If ENQ alone was received, BTAM posts a completion code of X'62'. If an invalid sequence was received, BTAM retries the Read ID ENQ command the number of times specified by the user, and, if all the retries are unsuccessful, posts a completion code of X'44' and disables the line. If DLE EOT was received, BTAM turns on the DLE EOT bit in the DECB flag byte and posts a completion code of X'41'.

This macro is for use 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 and executed 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 (TIW)

This macro instruction provides the IBM World Trade user with the ability to use expanded ID verification in Binary Synchronous Communications. The macro will transmit data to create an answer tone.

This macro allows initial contact to be established with a remote BSC station and performs 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. See the discussion under READ Connect for a description of how BTAM analyzes the received sequence.

This macro instruction will not work if used with data sets that generate a tone. The DECB extension must be used for defining the area and length of the data that is to be transmitted. The data defined by the user for tone generation must be all mark characters (X'FF'). Enough data should be transmitted so that the tone will last at least three seconds. The optype that must be coded in the READ macro instruction is READ TIW.

The macro is for use 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 and executed is:

1. Enable

2. Write Tone

3. Read [ID] ENQ

4. Write [ID] ACK-0

5. Read Text

6. Write DLE EOT

7. Disable

8. TIC to Enable command

Read Initial with Tone (TIZ)

This macro instruction provides the IBM World Trade user with the ability to use Manual Answer or Auto Answer. The macro instruction will transmit data to create an answer tone. This macro instruction will not work if used with data sets that generate a tone. The user must use the DECB extension to define the area and length of the data that is to be transmitted. The data defined by the user for tone generation must be all Mark characters. Enough data should be transmitted so that the tone will last at least three seconds. The optype that must be coded in the READ macro instruction is TIZ.

The channel program generated and executed is:

1. Enable

2. Write tone

3. Read [ID] ENQ

4. Write [ID] ACK-0

5. Read text

Read Inquiry (TQ)

This macro instruction is used for the CPU, when it is the receiving station, to recover from a text-read time-out occurring in a previously-issued READ macro instruction. The initiative to attempt recovery belongs to the remote device (the transmitting station), which sends the ENQ character when its own Read response times-out. When this ENQ is received, the READ Inquiry is completed and the user may issue the appropriate macro instruction to allow data exchange to continue.

The channel program generated and executed is:

1. Read ENQ

Note: The user should issue this macro instruction only if he desires more retrie: of a text-read time-out than are provided by BTAM ERP.

Read Continue (TT)

This macro instruction is issued to read messages from the remote device after initial contact has been successfully established. When entry is coded 'S,' an ID sequence is retried.

The channel program generated and executed is:

1. Write positive acknowledgment

2. Read message -- data or EOT

Read Continue with Leading Graphics (TTL)

This macro instruction is used to perform the same basic function as READ Continue, that is, acknowledge the last message received and read the next message into main storage. In addition, this macro instruction permits the sending of a maximum of 7 graphic characters (i.e., non-control characters) preceding the BTAM-supplied positive acknowledgment sequence (either ACK-0 or ACK-1).

If, in the BTMOD macro instruction, DECBEXT is coded a 'NO' or DECBEXT is omitted, the leading graphics must be supplied by the user in an output area of the following format:

- The first byte (count byte) of the output area must contain a binary count of the graphics to be sent. This count must not be zero. A count of zero results in a channel program check due to an invalid count.

- The bytes following the count byte must contain the graphics in transmission code.

The user must point to the count byte by the entry operand in the macro instruction. While the user may place the count byte and graphics in a buffer, the main storage address specified by the entry operand must be that of the count byte itself.

If, in the BTMOD macro instruction, DECBEXT is coded as 'YES,' the address of the leading graphics must be supplied by using the DECB extension.

The channel program generated and executed is:

1. Write graphics

2. Write positive acknowledgment

3. Read message

Read Repeat (TP)

This macro instruction is used to request retransmission of a message received in error.

The channel program generated and executed is:

1. Write negative acknowledgment -- NAK

2. Read message

Read Repeat with Leading Graphics (TPL)

This macro instruction is issued following an unsuccessful READ Initial or READ Continue to request retransmission of the last block of data from the remote device. In addition, the user is allowed to send leading graphics with the BTAM-supplied negative acknowledgment (NAK). For further information, refer to the discussion of the MODE operand of the CONTROL macro instruction. The leading graphics must be supplied by the user as explained for READ Continue with Leading Graphics.

The channel program generated and executed is:

1. Write graphics

2. Write NAK

3. Read message

Read Interrupt (TRV)

This macro instruction is issued to allow the CPU, as the receiving station, to request the transmitting device not to send any additional message blocks, so that a message can be sent to the device by the CPU. The RVI character is sent to the transmitting device, indicating that the CPU has correctly received the previous message, but does not wish to receive any additional message blocks. The transmitting device will transmit any buffered message blocks that could be lost when a message is received from the remote. If there is no buffered data that could be lost, the transmitting device will send an EOT. If the CPU received an EOT, it can then transmit data without data loss. If an EOT is not received, the CPU should read text by issuing READ Continue macro instructions until a EOT is received. When the EOT is received, the CPU may transmit. The RVI character must not be transmitted by the CPU twice in succession without transmitting an intervening normal acknowledgement (ACK-0, ACK-1, NAK) unless ENQ is received for retry purposes.

The channel program generated and executed is:

1. Write RVI character

2. Read message or response

Write Initial (TI)

This macro instruction is issued to perform the calling operation and to transmit the first message to the remote (answering) device. There are four separate channel programs for WRITE Initial, depending upon which code appears in the FEATURE sublist in the DTFBT macro defining the line group. There is also a channel program for manual dial.

No ID Verification: The channel program generated and executed is:

1. Dial user-specified phone number

2. Write ENQ

3. Read ACK-0

4. Write message

5. Read response -- ACK-1

IAM Procedure: The code SIX must be coded in the FEATURE sublist of the DTFBT macro instruction for the line group. The channel program generated and executed is:

1. Dial user-specified phone number

2. Write ID from terminal list

3. Write ENQ

4. Read ACK-0

5. Write message

6. Read response -- ACK-1

WRU Procedure: The code SXW must be coded in the FEATURE sublist of the DTFBT macro instruction for the line group. The channel program generated and executed is:

1. Dial user-specified phone number

2. Write ENQ

3. Read ID,ACK-0

4. Write message

5. Read response -- ACK-1

IAM/WRU Procedure: The code SIW must be coded in the FEATURE sublist of the DTFBT macro instruction for the line group.

The channel program generated and executed is:

1. Dial user-specified phone number
2. Write ID from terminal list
3. Write ENQ
4. Read ID,ACK-0
5. Write message
6. Read response -- ACK-1

Manual Dial: The channel program generated and executed is:

1. Enable
2. Write [ID] ENQ
3. Read [ID] ACK-0
4. Write message
5. Read ACK-1

The Enable allows the operator to call from the control unit to a station that is using either manual answer or automatic answer and establish line connection.

Write Connect (TIC)

This macro is used to originate a call to a remote BSC station, whether 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 a 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 with an ID that 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 posts a completion code of X'44'. If the control byte is 1, BTAM retries the Write [ID] ENQ command. If all retries are unsuccessful, BTAM posts a completion code of X'44'.

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 command. If all retries are unsuccessful, BTAM posts a completion code of X'44', transmits the DLE EOT sequence, and disables the line.

When BTAM retries the Write ID ENQ (or ENQ alone) command (as in the two preceding cases), the maximum number of retries is that specified by the user in the RETRY operand of the DTFBT macro for the line. If all retries are unsuccessful, BTAM posts a completion code of X'44'.

If the response from the remote station is ACK-0 (with no preceding ID), NAK (with no preceding ID), or WACK, BTAM posts a completion code of X'7F', X'44', or X'61', respectively. If the response is DLE EOT, BTAM turns on the DLE EOT bit in the DECB flag byte and posts normal completion (X'7F').

If a valid ID NAK sequence or NAK alone is received, the NAK bit in the DECB (bit 7 of the error information byte) is turned on. If a completion code of X'44' is posted on a WRITE Connect macro and the DECB NAK bit is not on, BTAM has disabled the line connection for the user because an invalid ID sequence was received.

If no response at all is received from the remote station, BTAM retries the Write ID ENQ (or ENQ alone) command up to the number of times specified in the RETRY operand of the DTFBT macro; if all retries are unsuccessful, BTAM breaks the line connection after transmitting the DLE EOT sequence and posts a completion code of X'41'.

The channel program generated and executed 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 generated and executed 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 the user reissues 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 Initial Transparent Block (TIE) and Write Initial Transparent Text (TIX)

These macro instructions are similar to WRITE Initial, except that an extra Write CCW is included in the channel programs to transmit the end characters terminating the transparent block. For optype TIE, the DLE,ETB sequence is transmitted; for optype TIX, the DLE,ETX sequence is sent. The format of the additional Write CCW is shown in the channel program for no ID verification (IAM, WRU, IAM/WRU, and manual-dial channel programs are also possible as discussed for WRITE Initial).

The channel program generated and executed for no ID verification is:

1. Dial user-specified phone number

2. Write ENQ

3. Read ACK-0

4. Write message

5. Write termination characters -- DLE, ETB or DLE,ETX

6. Read response -- ACK-1

Write Initial Conversational (TIV)

The channel programs for WRITE TIV are similar to those described for WRITE Initial. The difference is that the Read response CCW in the channel programs for WRITE Initial is replaced by a Read text. This is shown in the following channel program for no ID verification. (IAM,WRU, IAM/WRU, and manual-dial channel programs are also available as discussed for WRITE Initial.)

The channel program generated and executed for no ID verification is:

1. Dial user-specified phone number

2. Write ENQ

3. Read ACK-0

4. Write message

5. Read message

In the DFTRMLST macro instruction defining the DIALST (no ID verification) or the IDLST (IAM,WRU, or IAM/WRU), the user must also code the inlist positional operand when using the inlist technique. This oeprand specifies the symbolic address of a two-word parameter list:

- Word 1 contains the address of the input area provided for the message response from the remote device. If Word 1 contains all zeros, BTAM provides buffers for the expected message.

- Word 2 contains the length of the input area.

When the DECB extension is used, the inlist operand is not needed.

Write Initial Transparent Conversational (TIXV)

There are four channel programs available for this macro instruction, depending upon the ID verification procedure specified in the DTFBT macro instruction for the line group, and one channel program for manual dial. The difference between the channel programs for this macro and WRITE Initial is the insertion of an extra Write CCW to transmit the DLE,ETX end characters, and the replacement of the Read response CCW by a Read text CCW.

The channel program generated and executed for no ID verification is:

1. Dial user-specified phone number

2. Write ENQ

3. Read ACK-0

4. Write message

5. Write DLE,ETX

6. Read message

The user must specify the address and length of the input area as explained for WRITE Initial Conversational. The inlist technique or the DECB extension may be used.

Write Continue (TT)

This macro instruction is issued to transmit a message after initial contact has already been established.

The channel program generated and executed is:

1. Write message

2. Read response -- ACK-0 or ACK-1

Write Transparent Block (TE) and Write Transparent Text (TX)

These two macro instructions are issued to transmit text in transparent mode after initial contact has been established. The channel programs for the two macros are identical with the exception of the end characters transmitted by command (2).

The channel program generated and executed is:

1. Write message

2. Write termination characters -- DLE, ETB or DLE,ETX

3. Read response -- ACK-0 or ACK-1

Write Conversational (TV)

This macro instruction is issued after initial contact has been established. It enables the CPU to receive a message in response to a message sent.

The channel program generated and executed is:

1. Write message

2. Read response or message

If an acknowledgment is received instead of a message, it is checked by BTAM for correctness and completion code 7F is posted.

Note: If the inlist technique is used, see WRITE Initial Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of DECB extension for WRITE.

Write Transparent Conversational (TXV)

This macro instruction is issued to transmit a message in transparent mode and to receive a message in reply from the remote device. Initial contact must have been established previously.

The channel program generated and executed is:

1. Write message

2. Write termination characters -- DEL, ETB or DLE,ETX

3. Read response or message

Note: If the inlist technique is used, see WRITE Initial Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of DECB extension for WRITE.

Write End-of-Transmission (TR)

This macro instruction is used to relinquish the right of the CPU to transmit. A logical use of this macro instruciton is in place of the conversational WRITEs.

The channel program generated and executed is:

1. Write EOT

2. Read response

The response is read into the response field of the DECB. On a time-out of this commmand (no response is received), it is retried until a response is received or 'n' retries have occurred following the initial attempt. Since the Read command has a three-second time-out, a total of 3n seconds of line time may be expended in waiting for a response from the remote device.

Note: 'n' is the number specified by RETRY in the DTFBT.

The normal responses from the remote device are:

- ENQ indicating the remote device wishes to become the transmitting station.
- EOT indicating the remote device also has nothing to transmit and relinquishes its right to transmit.
- DLE,EOT indicating the disconnect signal.

This macro instruction causes BTAM to reinitialize the information on alternating acknowledgments. This information is used by BTAM to check that the appropriate alternating acknowledgments are sent and received.

Write Wait-Before-Transmitting (TW)

This macro instruction is issued to send the WACK sequence to the remote device and to await the response.

The channel program generated and executed is:

1. Write WACK

2. Prepare

3. Read ENQ

If ENQ is received, completion code 7F is posted. If EOT or DLE,EOT is received, completion code 41 (along with the appropriate bit in the FLAG byte) is posted. If the remote device does not respond (no completion is posted), the user must issue a RESETPL to cancel the incomplete Prepare command. The user could then issue a CONTROL Disable to disconnect the line.

Write Inquiry (TQ)

This macro may be issued for the following reasons:

1. To send the ENQ character when no response, an invalid response, or a WACK sequence has been received from the remote device in reply to a message. This means that the user desires more retries than BTAM ERP has already attempted.

2. To send the ENQ character as a bid for the line. It is used in this manner, for example, after EOT has been received from the remote device indicating the remote device has yielded its right to transmit.

3. To recover from certain error conditions which can occur on READ Initial (calling list) or on Initial WRITEs. (See Figures 38 and 39. As stated above, issuance of WRITE Inquiry for this purpose indicates the user desires more retries than BTAM ERP has already attempted.

In the case of 1 or 2, the user must specify the input area address and length in one of two ways: using the inlist technique, the user specifies the address of a parameter list by the entry operand (see Figure 51), or, using the DECB extension, the user specifies the input area address by the auxarea operand and the length by the auxlength operand.

If the correct acknowledgment is received, BTAM posts normal completion. Invalid responses are retried the number of times specified by the user (see RETRY in DTFBT), and are appropriately posted if irrecoverable. A NAK response is posted immediately, indicating for case 1 that the previously-sent message should be retransmitted, or for cases 2 and 3 that the remote device is not ready. If a message is received, normal completion is posted.

In case 3 the user may code entry as 'S' (see Figures 44 and 45). In this case BTAM restarts the previously-generated channel program (for the READ or WRITE Initial) at the command following the Dial command. When entry is not coded as 'S,' the channel program generated and executed is:

1. Write ENQ

2. Read response

Write Disconnect (TD)

This macro instruction is used on a switched-line data link to inform the remote device that disconnection is occurring. This macro instruction may be issued at normal end of communication or when irrecoverable errors occur on the line.

The channel program generated and executed is:

1. Write DLE,EOT

2. Disable

Control Disable (TD)

This macro instruction is used to disable the switched connection when the disconnect signal (DLE,EOT) is received from the remote device. The user is alerted to this disconnect signal being received when bit 2 of the Flag byte in the DECB is on and a completion code of 7F to a READ or 41 to a WRITE is set. There are two possible channel programs for this macro instruction, depending upon whether the MODE operand is present or omitted.

MODE present: The channel program generated and executed is:

1. Disable

2. Set mode

MODE omitted: The channel program generated and executed is:

1. Disable

Control Mode (TM)

This macro instruction is used to set the Error Information Byte mode of the adapter.

The channel program generated and executed is:

1. Set mode

MULTIPOINT

Double Addressing

Transient conditions such as lightning impulses or switching pulses can introduce errors in data transmitted over a communications 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 whereby one valid polling or addressing character is changed to another can result in polling or addressing the wrong system or component.

To avoid such an occurrence, double addressing is used for certain BSC stations. In this technique, a remote station (or a station component) is represented by two identical characters (rather than one, 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 or component 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.

If a station whose address is K were 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 instead to station K.[1]

This technique may be used for either station addresses, component addresses, or both (where a separate address is used for a station and its components). Figure 83 illustrates single versus double addressing. Each example shows sample polling and addressing characters for one station. In example 1, there is no separate component address, so two different characters, A and B, are used as station addresses. In examples 2 and 3, the station address is A, 1 is the address of a component to be polled, and 2 is the address of a component to be addressed. Example 3 shows double addressing applied to both station and component addresses.

Applicability: For System/360 Model 20, 1800, 2715, 2770, 2972, and 3270 stations in a multipoint network, double addressing must be used. It is not supported for the 1130 or 2780.

Because all list entries in terminal lists must have the same length, when 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.

----

[1]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.

Example	Function	Single Addressing		Double Addressing	
		Station Address	Component Address	Station Address	Component Address
1	Polling Addressing	A B	- -	AA BB	- -
2	Polling Addressing	A A	1 2	AA AA	1 2
3	Polling Addressing	A A	1 2	AA AA	11 22

Figure 83. Single Versus Double Addressing

Read Initial (TI)

This macro instruction sets the line to control mode and starts the Auto Poll operation with the polling list entry given by the entry operand. When a terminal responds with data or does not respond at all, the first byte of the input area contains the index byte of the polling list entry representing the terminal. The text, if any, begins in the second byte of the input area. If a terminal responds negatively (with an EOT), the next terminal is polled, and so on until the last terminal in the list is polled. If all polled terminals respond negatively, the Poll command ends and chains to the I/O No-op command, which causes an interrupt. The index byte of the responding terminal will be in the input area. By inspecting the index byte, the program can determine which terminal, if any, responded.

Depending upon whether the terminal list used is the AUTOLST or AUTOWLST type, one of two separate channel programs is generated and executed. When the terminal list specified via the entry operand is an AUTOLST, the channel program is:

1. Write EOT
2. Poll using the AUTOLST
3. I/O No-op
4. Read index byte and message

For AUTOWLST, upon encountering the end of the list, the Auto Poll operation continues to poll, starting at the beginning of the list and continuing indefinitely until data is received or some terminal does not respond at all. The channel program for an AUTOWLST is:

1. Write EOT
2. Poll using the AUTOWLST
3. TIC to (5)
4. Read index byte and message
5. Poll from beginning of AUTOWLST
6. TIC to (5)
7. Read index byte and message

In issuing the READ Initial, the user may specify (via the entry operand) that the Auto Poll start with any entry in the list. The symbol in the name field of the DFTRMLST macro instruction used to generate the polling list is the symbolic address of the first entry in the list, and not of the three-byte header preceding the list. In computing the address of an entry in an AUTOLST or AUTOWLST list, the user should realize that each entry consists of the number of characters in the polling sequence (including ENQ), plus the index byte.

Read Continue (TT)

This macro instruction is issued to read messages from the remote device after initial contact has been successfully established.

The channel program generated and executed is:

1. Write positive acknowledgment
2. Read message -- data or an EOT

Read Continue with Leading Graphics (TTL)

This macro instruction is used to perform the same basic function as READ Continue: that is, acknowledge the last message received and read the next message into main storage. In addition, this macro instruction permits the sending of a maximum of seven graphic characters (i.e., non-control characters) preceding the BTAM-supplied positive acknowledgment sequence (either ACK-0 or ACK-1).

If, in the BTMOD macro instruction, DECBEXT is coded as 'NO' or DECBEXT is omitted, the leading graphics must be supplied by the user in an output area of the following format:

- The first byte (count byte) of the output area must contain a binary count of the graphics to be sent. This count must not be zero. A count of zero results in a channel program check due to an invalid count.
- The bytes following the count byte must contain the graphics in transmission code.

The user must point to the count byte by the entry operand in the macro instruction. While the user may place the count byte and graphics in a buffer, the main storage address specified by the entry operand must be that of the count byte itself.

If, in the BTMOD macro instruction, DECBEXT is coded as 'YES,' the address of the leading graphics must be supplied by using the DECB extension.

Note: Leading graphics can serve any function required by the application program. The only restriction is that the bit configuration of the graphics not coincide with the bit configuration of defined data link control characters (e.g., ENQ, STX).

The channel program generated and executed is:

1. Write graphics
2. Write positive acknowledgment
3. Read message

Read Repeat (TP)

This macro instruction is issued following an unsuccessful READ Initial or READ Continue to request retransmission of the last block of data from the remote device.

The channel program generated and executed is:

1. Write negative acknowledgment - NAK
2. Read message

Note: This macro instruction should be issued only if the user desires more retries than BTAM ERP has already attempted automatically.

Read Repeat with Leading Graphics (TPL)

This macro instruction is issued following an unsuccessful READ Initial or READ Continue to request retransmission of the last block of data from the remote device. In addition, the user is allowed to send leading graphics with the BTAM-supplied negative acknowledgment (NAK). For further information, refer to the discussion of the MODE operand of the CONTROL macro instruction. The leading graphics must be supplied by the user as explained for READ Continue with Leading Graphics.

The channel program generated and executed is:

1. Write graphics
2. Write NAK
3. Read message

Read Inquiry (TQ)

This macro instruction may be used when the CPU, as the receiving station, times out on a text read command in a READ macro channel program. As the receiving station, the CPU does not initiate recovery from such a time-out, but generally issues the READ Inquiry macro instruction, awaiting an ENQ from the remote device requesting the CPU to transmit its last response. For this reason, READ Inquiry does not cause BTAM to reinitialize the information on alternating acknowledgments. When the ENQ is received, the appropriate macro instruction can then be executed to retransmit the last response of the CPU.

The channel program generated and executed is:

1. Read ENQ

Note: The user should issue this macro only if he desires more retries than are provided automatically by BTAM ERP.

Read Interrupt (TRV)

This macro instruction is issued to allow the CPU, as the receiving station, to request the transmitting device not to send any additional message blocks, so that a message can be sent to the device by the CPU. The RVI character is sent to the transmitting device, indicating that the CPU has correctly received the previous message, but does not wish to receive any additional message blocks. The transmitting device will transmit any buffered message blocks that could be lost when a message is received from the remote device. If there is no buffered data that could be lost, the transmitting device will send an EOT. If the CPU receives an EOT, it can then transmit data without data loss. If an EOT is not received, the CPU should read text by issuing READ Continue macro instructions until an EOT is received. When the EOT is received, the CPU may transmit. The RVI character must not be transmitted by the CPU twice in succession without transmitting an intervening normal acknowledgment (ACK-0, ACK-1, NAK), unless ENQ is received for retry purposes.

The channel program generated and executed is:

1. Write RVI character
2. Read message or response

Write Initial (TI)

This macro instruction is issued to select a specific multidropped remote device before transmitting the first message.

The channel program generated and executed is:

1. Write EOT
2. Write user-specified addressing sequence
3. Read response to addressing -- ACK-0
4. Write message

5. Read response to message

Write Initial Transparent Block (TIE) and Write Initial Transparent Text (TIX)

These macro instructions are issued to select a remote device and to transmit the first message in transparent mode.

The channel program generated and executed is:

1. Write EOT
2. Write user-specified addressing sequence
3. Read response to addressing -- ACK-0
4. Write message
5. Write termination characters -- DLE ETB or DLE ETX
6. Read response to message

Write Initial Conversational (TIV)

This macro instruction is used when the user desires to select a remote device, send a message to the remote device, and receive a message in response. In coding a conversational WRITE, the user must specify his output message by the area and length operands, and his input parameters by the auxarea and auxlength operands (see the section on DECB Extension for Write). Auxarea may be coded as 'S'. Hence, BTAM will perform buffering in the same manner as with any other READ. The address of the first buffer in the chain will be placed in the auxarea field of the DECB extension.

The channel program generated and executed is:

1. Write EOT
2. Write addressing characters
3. Read response to addressing
4. Write message
5. Read response -- positive acknowledgment or message

Write Initial Transparent Conversational (TIXV)

This macro instruction is issued to establish initial contact, transmit a message in transparent mode, and read a conversational reply. The DECB extension must be used to define the input area (see the DECB extension for WRITE).

The channel program generated and executed is:

1. Write EOT
2. Write addressing characters
3. Read response to addressing
4. Write message
5. Write termination characters -- DLE ETX
6. Read response -- positive acknowledgment or message

Write Continue (TT)

This macro instruction is issued to transmit a message after initial contact has already been established.

The channel program generated and executed is:

1. Write message
2. Read response -- ACK-0 or ACK-1

Write Transparent Block (TE) and Write Transparent Text (TX)

These two macro instructions are issued to transmit text in transparent mode after initial contact has been established. The channel programs for the two macros are identical with the exception of the end characters transmitted by command (2).

The channel program generated and executed is:

1. Write message
2. Write termination characters -- DLE ETB or DLE ETX
3. Read response -- ACK-0 or ACK-1

Write Conversational (TV)

This macro instruction is issued after initial contact has been established. It enables the CPU to receive a message in response to a message sent.

The user must specify his output message by the area and length operands, and if the inlist technique is being used, define the equivalent parameters for the expected input message by specifying the address of a parameter list by the entry operand. The parameter list contains two full words with the format shown in Figure 84.

Word 1		Address	of input	area	(right adjusted)
Word 2		Length	of input	area	(right adjusted)

Figure 84. Format of Parameter List

If the address specified in Word 1 is zero, it is assumed that the user is requesting BTAM to read the input message into buffers. Word 2, in this case, must specify the maximum length of the expected message. BTAM performs the buffering as it does for the READ macro instruction when area is coded 'S' in that macro instruction. The only exception is that the address of the first buffer (into which the message is read) is placed in Word 1 of the parameter list, rather than into the area field of the DECB.

If the DECB extension is being used, it is required that the user specify his input parameters by the auxarea and auxlength operands (see the section on DECB extension for WRITE). Auxarea may be coded as 'S'. Hence, BTAM will perform buffering in the same manner as any other READ. The address of the first buffer in the chain will be placed in the auxarea field of the DECB extension.

The channel program generated and executed is:

1. Write message
2. Read response or message

If an acknowledgment is received instead of a message, it is checked by BTAM for correctness, and completion code 7F is posted.

Write Transparent Conversational (TXV)

This macro instruction is issued to transmit a message in transparent mode and to receive a message in reply from the remote device. Initial contact must have been established previously.

The channel program generated and executed is:

1. Write message
2. Write termination characters -- DLE,ETX
3. Read response or message.

Note: If the inlist technique is used, see WRITE Conversational for the format of the parameter list pointed to by the entry operand. If the DECB extension is used, see the description of DECB extension for WRITE.

Write Inquiry (TQ)

This macro instruction is issued when no response or an invalid response is received from the remote device in reply to a message. The remote device is requested to transmit its last response (ACK-0, ACK-1, NAK, or a conversational text reply).

The channel program generated and executed is:

1. Write ENQ
2. Read response

The response is read into the user-supplied input area. The user must specify both the input area and length in a parameter list (as described for WRITE Conversational using the inlist technique), or by the auxarea and auxlength operands (as described in the section on DECB extension for WRITE).

Note: This macro instruction should be issued only when the user desires more retries than BTAM ERP automatically provides (the number of retries being equal to that specified in the RETRY operand in DTFBT).

Write Wait-Before-Transmitting (TW)

This macro instruction is issued to inform the remote device that a temporary condition exists, requiring a delay before the remote device continues transmission. This may result, for example, from a temporary lack, in the CPU, of buffers for receiving messages.

This macro instruction may also be issued after a message has been read, to defer sending a positive response. This could be because some temporary condition exists making immediate reading of a message undesirable.

The expected response from the remote device is the ENQ character, which, if received, causes BTAM to post normal completion. Another response that can occur is the EOT character that causes BTAM to post completion code 41 and turn ON the 1-bit of the DECB flag byte. Text messages are illegal responses to WACK.

After completion, the user may reissue the WRITE WACK macro instruction, if still not ready, or he may issue the appropriate macro instruction to send his deferred response or message.

The channel program generated and executed is:

1. Write WACK

2. Prepare

3. Read ENQ

Write End-of-Transmission (TR)

This macro instruction is issued to inform the remote device that transmission is ended on the line. It may also be used to end transmission when irrecoverable errors exist. Further transmission on the line requires that the READ Initial or WRITE Initial be used to reestablish contact between the CPU and the remote device.

The channel program generated and executed is:

1. Write EOT

Control Mode (TM)

This macro instruction is issued to set the Error Information Byte mode of the 2701/2703.

The channel program generated and executed is:

1. Set mode

START-STOP LISTS

All of the following terminal-list descriptions include a control byte for each terminal or component entry (Figure 85). The individual bit positions within the byte are used for:

Bit Position	Meaning
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. Turned on and off with CHGNTRY macro.
2	If on, indicates that the list is a wrap-around list.
3-7	List entry number. Each entry numbered successively starting with 0. This field limits the maximum number of terminal or component entries for lists that are created by the DFTRMLST macro to 32.

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. Example for the 1050 and 1030 is shown in Figure 86.

WRAPLST Format
wraparound lists differ from open lists in that the format bit (bit 2 in the control byte) is ON for the last entry in the list. (Examples are shown in Figure 87.)

DIALST Format
terminal lists for 1050s on switched networks are illustrated in Figure 88.

IDLST Format
terminal lists for TWX 33/35s are illustrated in Figure 89.

SSALST and SSAWLST Format
terminal lists for start-stop Auto Poll are listed in Figure 90.

WTTALST Format
terminal lists for World Trade terminals are illustrated in Figure 91.

The two-byte field that follows the last entry in the terminal list is an offset used by the polling restart routine to get back to the start of the list.

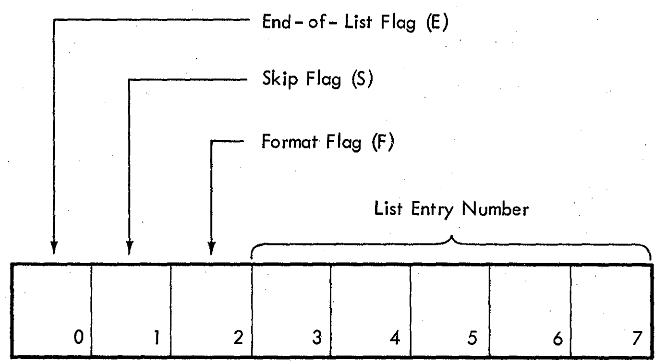


Figure 85. Format of Control Byte

Format

Polling Characters (in Transmission Code)	E	S	F	Entry No.	Offset
1 Byte (1030) 2 Bytes (1050, 1060, 83B3, 115A, 2260)	Control Byte 1 Byte				2 Bytes

Example IBM 1050

A	5	0	0	0	0	0	0	0	0	
A	6	0	1	0	0	0	0	0	1	
B	5	0	0	0	0	0	0	1	0	
B	6	0	0	0	0	0	0	1	1	
C	5	0	1	0	0	0	1	0	0	
E	0	1	0	0	0	0	1	0	1	18 2 Bytes

Example IBM 1030

D	0	0	0	0	0	0	0	0	
E	0	0	0	0	0	0	0	1	
F	1	0	0	0	0	0	1	0	6 2 Bytes

Figure 86. Open Polling or Addressing List (OPENLST): Format and Examples

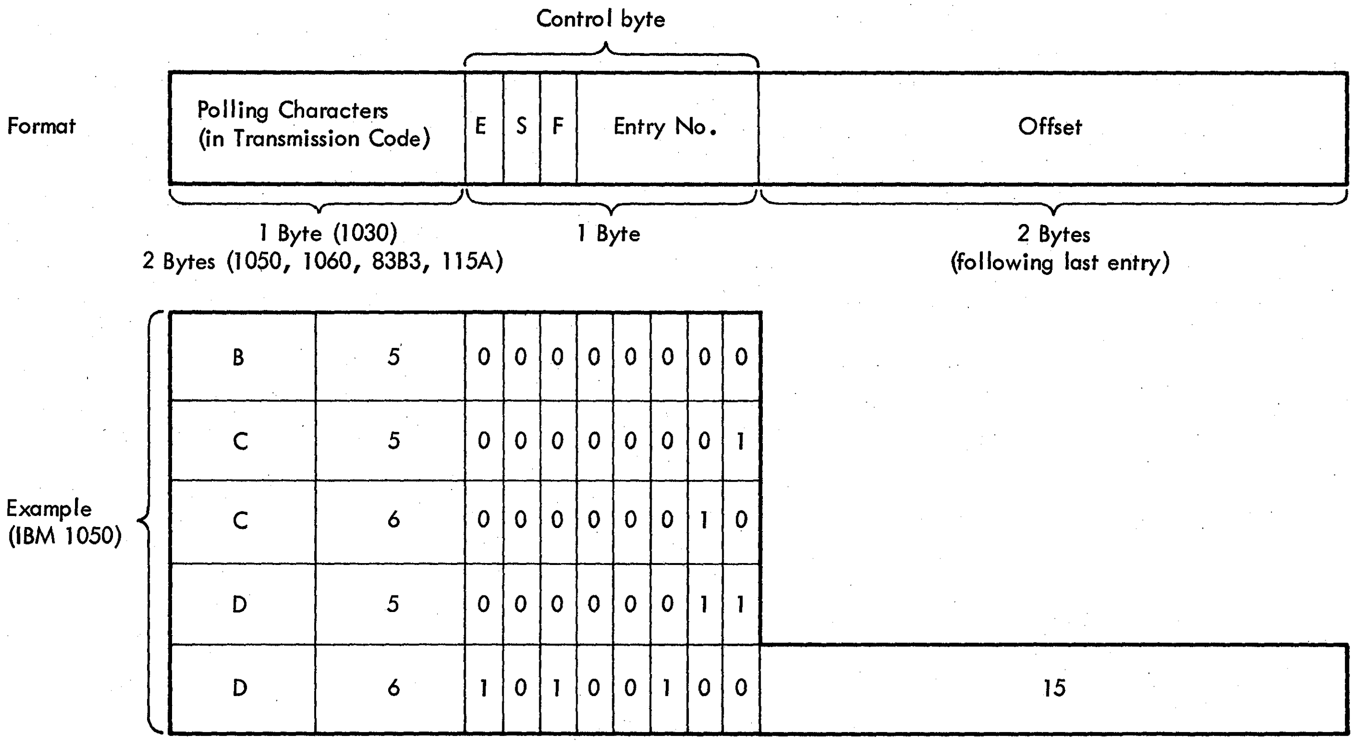


Figure 87. Wraparound Polling List (WRAPLST): Format and Example

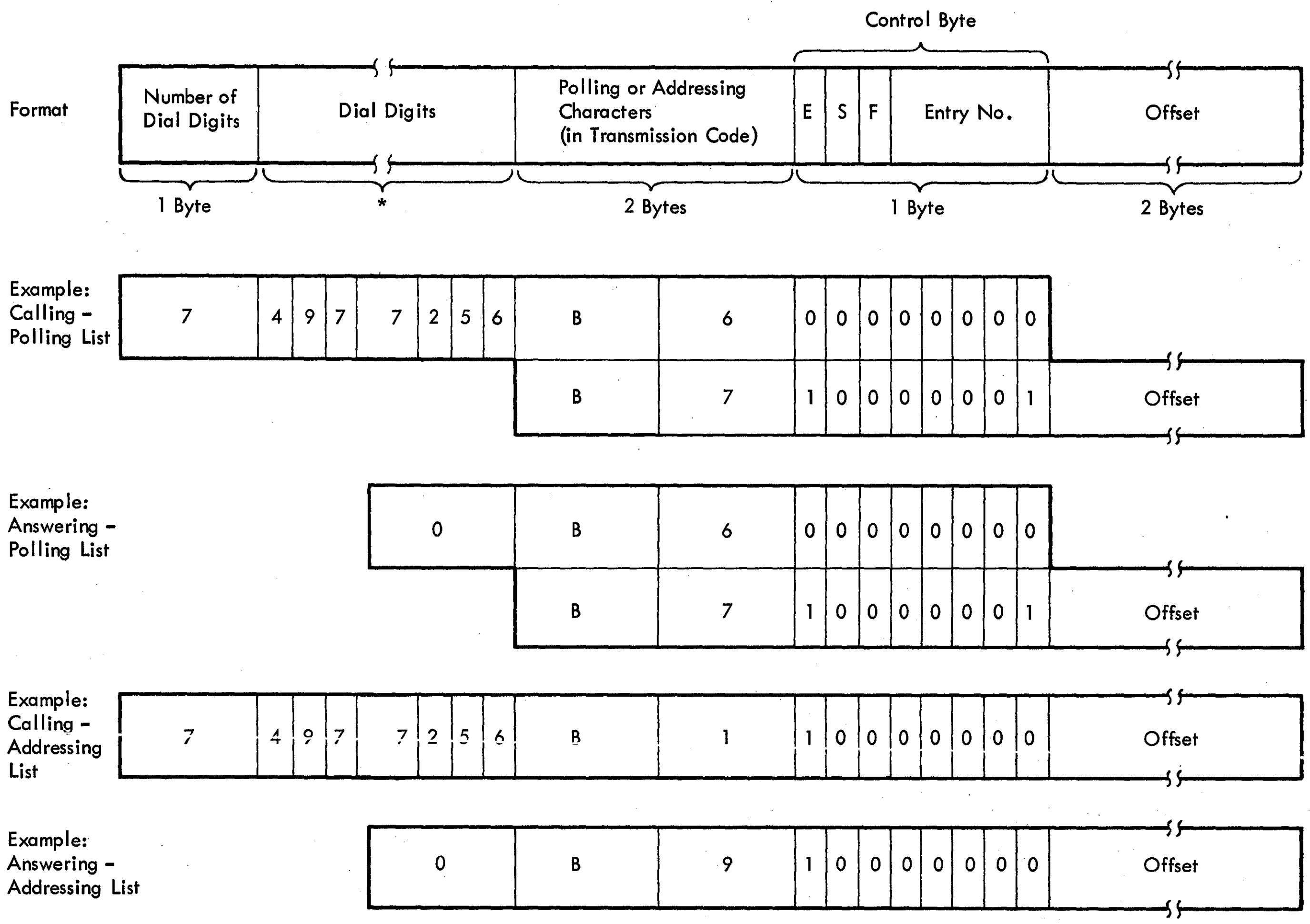
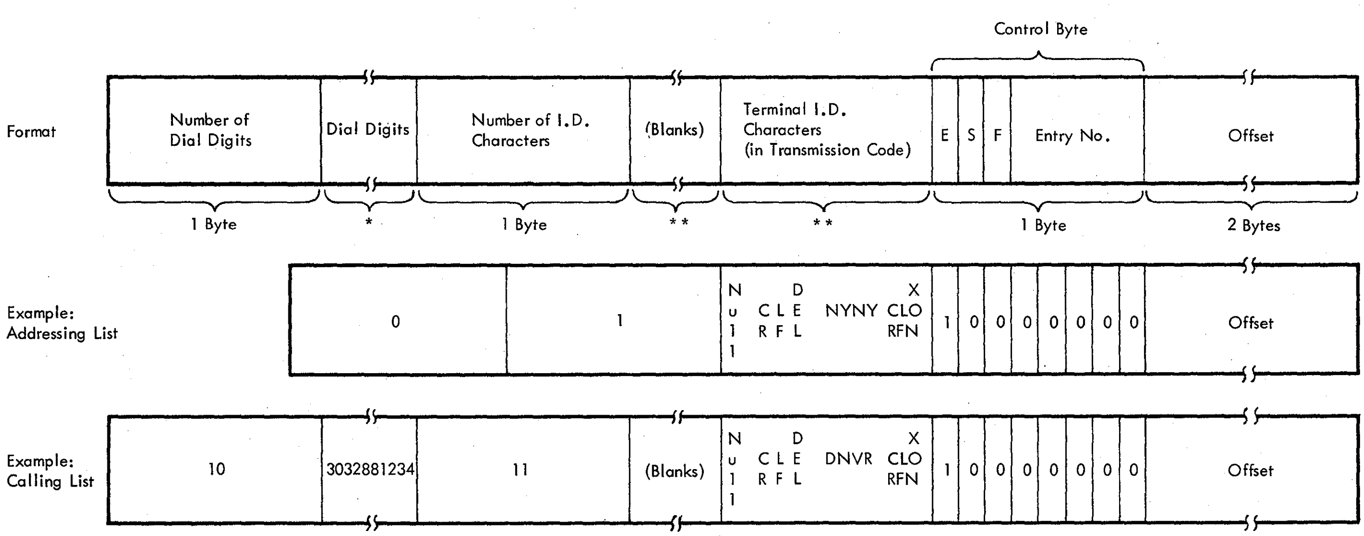


Figure 88. 1050 Terminal List (DIALST): Format and Examples



* Length in bytes equals number of dial digits
** Length in bytes equals number of terminal I.D. characters

Figure 89. TWX-Terminal List (IDLST): Format and Example

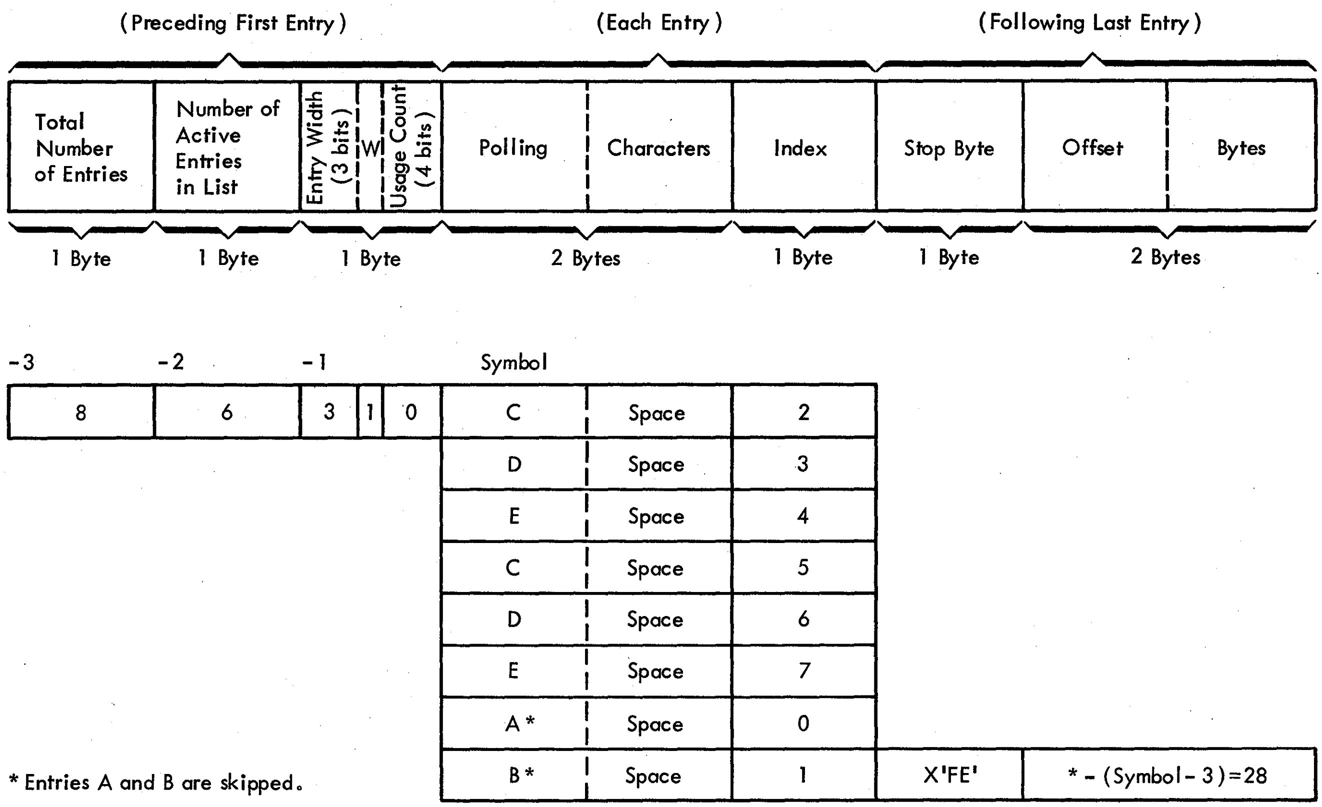


Figure 90. 2740SC Terminal List (SSAWLST): Format and Example

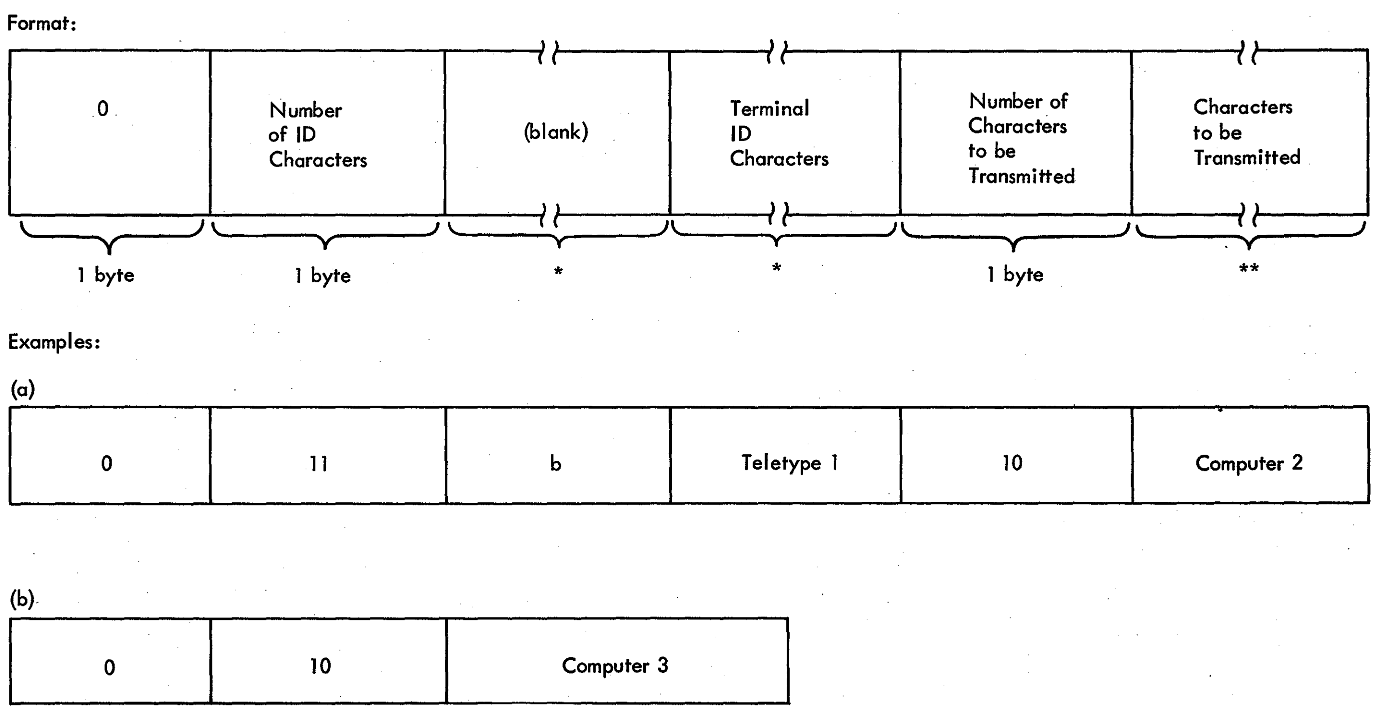


Figure 91. WTTA Terminal List (WTTALST): Format and Examples

BSC LISTS

FORMAT OF DIALST

(No ID Verification)

This list is used as a calling list only.

n_d	dial-chars	inlist address
1 byte	n_d bytes	3 bytes

n_d
    specifies the number (in binary) of dial characters in the list.

dial-chars
    specifies the actual characters (in binary) to be used in dialing the remote device.

inlist address
    used only if the WRITE Intial Conversational (TIV) or WRITE Initial Transparent Conversational (TIXV) is issued using the inlist technique. It specifies the address of the parameter list containing the:

- Address of the input area (to receive the conversation). An address of all zeros means buffering is to be used.
- Length of the input area.

For an answering operation, the DIALST list is a single byte containing all zeros.

FORMAT OF IDLST

(For ID Verification)

IDLST

call list:

n_d	dialchars	n_r	0----0	recv-chars	n_s	send-chars	inlist address
1 byte	n_d bytes	1 byte	n_r bytes	n_r bytes	1 byte	n_s bytes	3 bytes

answer list:

0	n_r	0----0	recv-char	n_s	send-chars
1 byte	1 byte	n_r bytes	n_r bytes	1 byte	n_s bytes

n_d
    the number (in binary format) of dialing characters to be used in calling the remote device.

dialchars
    the actual dialing characters (in binary format) used.

n_r
the number (in binary) of identification characters the remote device is expected to send. (This byte may be 0.)

0----0
the received ID characters will be read into these bytes.

recv-chars
the (user-supplied) ID characters expected to be received from the remote device.

n_s
the number (in binary) of ID characters that the CPU will send to the remote device.

inlist address
the address of a user-maintained parameter list to be used if the CPU is calling and wishes to issue a conversation-type WRITE Initial using the inlist technique. The parameter list pointed to must contain 2,fullwords:

Word 1 - address of the input I/O area. (All zeros means buffering is to be used.)

Word 2 - length of the expected message.

Note: Both recv-chars and send-chars are in transmission code representation.

FORMAT OF AUTO POLL TERMINAL LIST

Figure 92 shows the format of the Auto Poll Terminal list.

FORMAT OF SWLIST

Figure 93 shows the format of the calling and answering terminal lists of the SWLIST form.

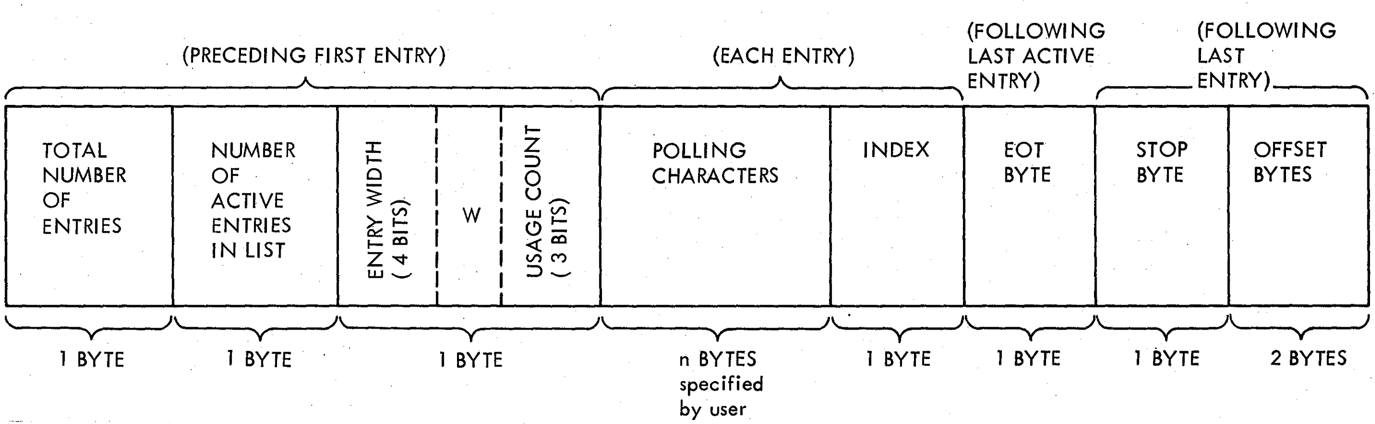


Figure 92. Format of Auto Poll Terminal List (AUTOLST,AUTOWLST)

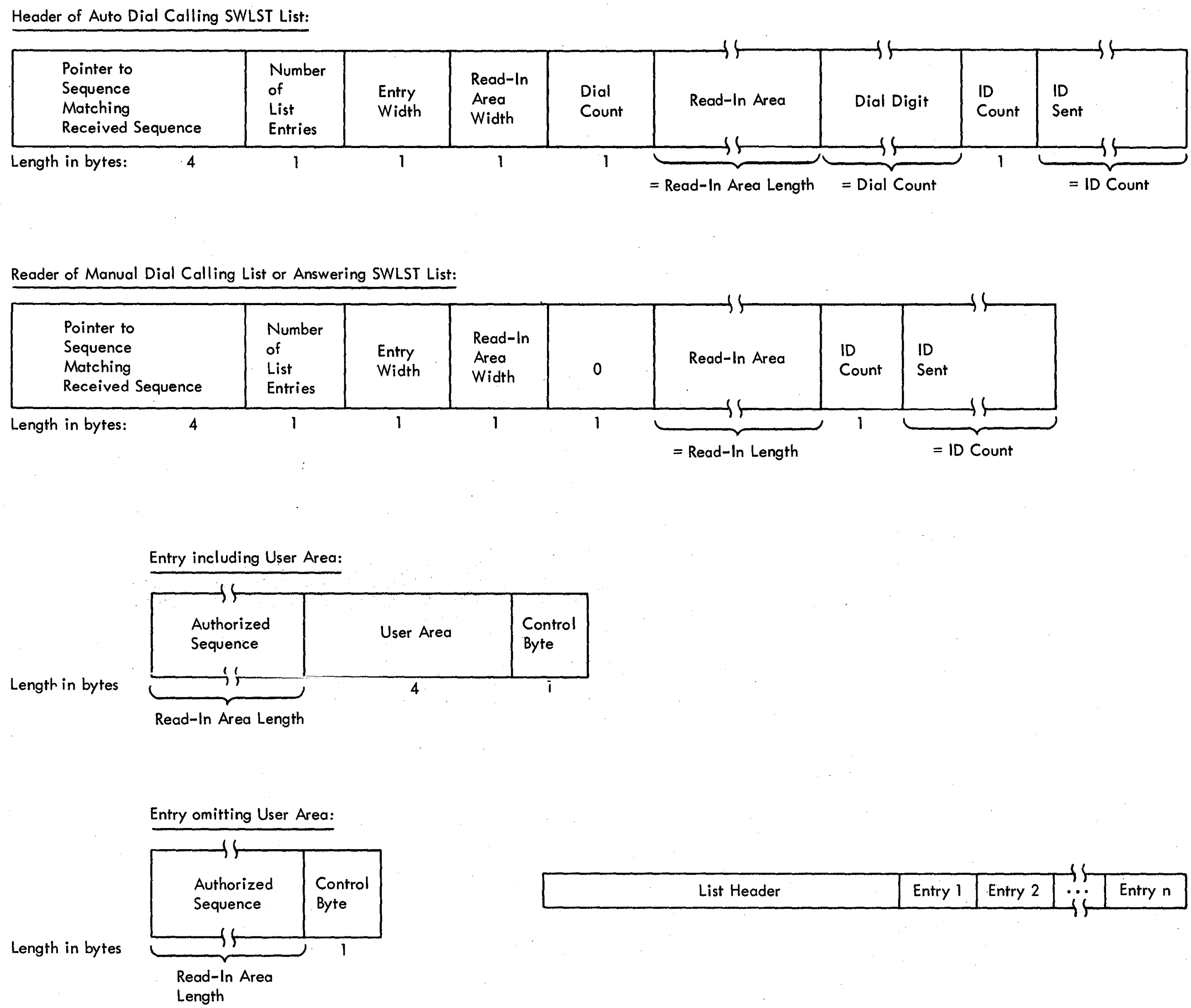


Figure 93. Format of Calling and Answering Lists of the SWLST Form

Field	Contents
Pointer to Sequence Matching Received Sequence	Address (right-adjusted) of the last authorized ID sequence that was received before completion of the READ Connect or WRITE Connect macro.
Number of List Entries	Number of entries (binary) in the list (i.e., number of authorized ID sequences that will be honored).
Entry Width	Number of bytes (binary) in each entry in the list. This number is specified by the entrywidth operand of the DFTRMLST macro, and should equal the number of bytes required to accommodate the longest expected ID sequence, plus the userdata field (0 or 4), plus one (for the control byte).
Read-In Area Width	Number of characters (binary) 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.

Dialcount	For an automatic dialing list: number of dialing digits (binary) to be used in calling the remote station. For a manual dialing list or an answering list: 0.
Read-In Area	The 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 2 bytes.
Dialchars	The dialing digits (binary) for an automatic-dial calling list. For an answering list or a manual-dial calling list, this field is omitted.
Idcount	The number of characters (binary) in the sequence defined in the idsent field.
Idsent	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 transmitted 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.
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 the Read-In Area length will never be less than 2, the size of this field will be 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 the userlength operand of the DFTRMLST macro is specified 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 the description of the DFTRMLST macro for the significance of these values.)

The following table contains the hexadecimal representation of all alphabetic and numeric characters as they appear in the various transmission codes. These hexadecimal values can be used in the DFTRMLST macro instruction.

Example: To send A over a line, the user must code

62 for a 1030, 1050, 1060, or 2740
18 for a 115A or a 83B3
83 for a Model 33 or 35 TWX Station
A1 for a 2260-2848

Character	1030, 1050, 1060, & 2740	83B3 &115A*	TWX
A	62	18	83
B	64	13	43
C	67	OE	C3
D	68	12	23
E	6B	10	A3
F	6D	16	63
G	6E	0B	E3
H	70	05	13
I	73	0C	93
J	43	1A	53
K	45	1E	D3
L	46	09	33
M	49	07	B3
N	4A	06	73
O	4C	03	F3
P	4F	0D	0B
Q	51	1D	8B
R	52	0A	4B
S	25	14	CB
T	26	01	2B
U	29	1C	AB
V	2A	0F	6B
W	2C	19	EB
X	2F	17	1B
Y	31	15	9B
Z	32	11	5B
EOT			21
LF			50
CR			B1
DLE			09
DC1			88
DC2			48
DC3			C9
DC4			28
NUL			00
DEL			FF
XON			88
XOFF			C9

*Keyboards A and C for 83B3 and 115A

Character	1030, 1050, 1060, & 2740	83B3 & 115A*	TWX
1	02	3D	8D
2	04	39	4D
3	07	30	CD
4	08	2A	2D
5	0B	21	AD
6	0D	35	6D
7	0E	3C	ED
8	10	2C	1D
9	13	23	9D
0	15	2D	0D
Space	01	04	05
/	23	-	-

*Keyboards A and C for 83B3 and 115A

Note: The 1050 polling and addressing codes can be sent in either upper or lower case letters but only the lower case is shown.

To poll or address the 2848 Display Control, the hexadecimal values used are:

40 through 5F
A0 through BF
E0 through FF

The valid hexadecimal values for selecting the 2260 Display Station are:

A0 through B8

Macro	Operand	Values Allowed	
ASMTRTAB	tablename	code	RC30,RC50,RF50,RASA,RSCI,RCT1,RCT2,RC80,RC60, RC40,RF40,RCTW,RCT3 SD30,SD50,SASA,SSCI,SCT1,SCT2,SD80,SD40,SD60,SCTW, SCT3
BTMOD	SEPASMB	code	NO,YES
	BSCTEST	code	NO,YES
	BUFFER	code	NO,YES,REQREL
	ERLOGIC	code	E,N,C,NC
	TERMTST	code	NO,YES
	RESETPL	code	NO, YES
	SWITCH	code	NO,YES,NEWID
	L2260	code	NO,YES
	L3277	code	NO,YES
	TST3277	code	NO,YES
	AUDIO	code	NO,YES
	CANCEL	code	NO,YES
	TRANSL	code	NO,YES
	BSCS	code	NO,YES
	BSCMPT	code	NO,YES
	SSAPL	code	NO,YES
	WTTA	code	NO,YES
	DECBEXT	code	NO,YES
	OBRSDR	(see RMSR)	
	RMSR	code	NO,YES
BTRD	dtfbt	symbol (r)	Specifying the symbolic name of the DTFBT whose counters are to be written to the SDR disk file.
CHGNTRY	list	symbol (r)	Specifying address of the list.
	listype	code	OPENLST,WRAPLST,DIALST,AUTOLST,AUTOWLST, SSALST,SSAWLST

Macro	Operand	Values Allowed	
CHGNTRY (cont'd)	position	absexp (r)	0-31 for OPENLST,WRAPLST,DIALST 0-97 for AUTOLST,AUTOWLST 0-98 for SSALST,SSAWLST
	numchars	absexp (r)	Specifying the number of polling/ addressing characters per entry.
	action	code	SKIP,ACTIVATE
CHGNTRY (Local 2260 and local 3270)	dtfbt	symbol (r)	Specifying address of the DTFBT.
	listype	code	ATTLST
	position	absexp (r)	0-198 possible, must not exceed number of lines in DTFBT, less one.
	action	code	SKIP,ACTIVATE
CHGNTRY (SWLST type only)	list	symbol (r)	Specifying name of DFTRMLST macro that defined the list containing entry to be changed
	listype	code	SWLST
	position	absexp (r)	Specifying relative position of entry in list
	action	code	ACTIVATE, DISC, POST
CLOSE	dtfname	symbol (r)	1 to 16 addresses may be specified.
CONTROL*	decb	symbol (r) (1)	Specifying the address of a DECB.
	OPTYPE	code	TI,TD,TM,
	dtfbt	symbol (r)	Specifying the address of a DTFBT.
	area	relexp (r)	Specifying the address of the I/O area.
	length	absexp (r)	Range allowed: 2-32,767 inclusive
	entry	relexp (r)	Specifying the address of an entry in a terminal list.
	rln	absexp (r)	The range allowed is 0-198 inclusive (default: X'FF')
	MODE	0 1 (r)	Specifying the checking mode to be used.
	MF	code	L,E

Macro	Operand	Values Allowed	
DFTRMLST	listype	code	OPENLST,WRAPLST,DIALST,IDLST,AUTOLST,ATUOWLST, SSALST,SSAWLST,WTTALST,SWLST
	device dependent operands	Note: These operands are defined in the sections describing the individual devices. Some commonly-shared values are: • number of dial characters - not to exceed 127. • actual dial characters - can be any decimal digits 0 to 9 inclusive. • number of entries - up to 32 for OPENLST,WRAPLST,DIALST. - up to 98 for AUTOLST,AUTOWLST. - up to 99 for SSALST,SSAWLST.	
DFTBT (Maximum of 255 per assembly)	LINELST	integer	Allowable range: 000-244. (Maximum of 31 may be specified)
	SWITCH	code	NO,YES
	CU	code	2701,2702,2703,7770,2848,3272
	DEVICE	code	1030,1050,1060,2260,83B3,115A,TW33,S360,TW35, 2740,2848,2780,3277,1130,2020,WTTA,2972,BSC1,BSC2, BSC3
	FEATURE	code	STC,CHK,TRC,BSC,SIX,SXW,SIW,RIX RWX,RIW,MAS,SLV,APL,KBL,OIU,IAM,WRU,MON
	BUFCB	symbol	Specifying the address of a buffer control block.
	BUFNO	absexp	Allowable range: 2-255 inclusive.
	BUFL	absexp	Allowable range: 16-32,760 inclusive. 22-32,760 for remote 3270.
	SEPASMB	code	NO,YES
	MONDLY	absexp	0-20 inclusive
	EOM	code	WRU,X'hh',X'hhlF'
	EOT	code	2EOM,X'hhlF'
	MODNAME	symbol	Specifying the BTAM name (address) of the module used. The default name is IJLBTM.
	LERBADR	symbol	Line error block or RMSR table address.
	MODELST	code	0,1,2,3,4,5,6,7
	MSGL	absexp	Allowable range: 2-32,767 inclusive.
	TERMTST	code	NO,YES
	ERROPT	code	E,N,R,W,RW
	CTLCHAR	code	EBCDIC,ASCII,TRNSCD
	RETRY	absexp	Allowable range: 0-15
	CONFIG	code	PPT,MPT
	LCBNUM	integer	Allowable range: 1-199
DTFBTND	SEPASMB	code	NO,YES

Macro	Operand	Values Allowed	
LOPEN	decb	symbol (r)	1 address may be specified.
OPEN	dtfname	symbol (r)	1 to 16 addresses may be specified.
LERB	nlines	absexp	1 to 99 lines, Asmblb O: 1 to 199 lines, Asmblr F.
	transmct	absexp	1 to 255, default 255.
	datack	absexp	1 to 255, default 10 or 'transmct' if less than 10.
	intreq	absexp	1 to 255, default 5 or 'transmct' if less than 5.
	nontto	absexp	1 to 255, default 5 or 'transmct' if less than 5.
LERPRT	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	rln	absexp (r) (0)	Range allowed: 0 to 30 inclusive. If omitted, all lines having at least one non-zero accumulator will be printed.
	CLEAR	code	NO,YES
READ*	decb ~~dtfbt~~ length rln MF	Same as for CONTROL. See the special use of rln for the local 3270 under "Local Device-Dependent Considerations."	
	optype	code	TI,TIR,TIL,TT,TTR,TV,TVR,TP,TPR,TB,TBL,TBP,TO,TOL, TS,TQ,TIQ,TTL,TPL,TE,TRV,TIC,TIZ,TIW,TM,TMP
	area	relexp (r) 'S'	Specifying the address of the input area.
	entry	relexp (r) 'S'	Specifying the address of an entry in a terminal list. See the special use for the local 3270 under "Local Device-Dependent Considerations."
RELBUF	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	(r)		Specifying the register (2-12) in which the address of the first buffer in the chain of buffers has been loaded by the user.
REQBUF	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	(r)		Specifying the register (2-12) into which the address of the first buffer in a chain is to be loaded by BTAM.
	count	absexp (r) (0)	Range allowed: 1-255 inclusive. Default=1

Macro	Operand	Values Allowed	
RESETPL	decb	symbol (r) (1)	Specifying the address of a DECB.
RMSRTAB	nlines	absexp	Specifying the number of lines associated with the line group (DTFBT). Range allowed: 1-99/D Assembler; 1-199/F Assembler.
	pchars	absexp	Specifying the hexadecimal representation of the polling or addressing characters for each terminal on which statistics are to be kept.
SDRTAB	(see RMSRTAB)		
TRNSLATE	dtfbt	symbol (r) (1)	Specifying the address of a DTFBT.
	table**	code (r)	RC30,RC50,RF50,RASA,RSCI,RCT1,RCT2,RC80.RF40, RC40,RC60,RCTW,RCT3 SD30,SD50,SASA,SSCI,SCT1,SCT2,SD80,SD40,SD60,SCTW SCT3
	area	relexp (r)	Specifying the address of the I/O area containing the data to be translated.
	length	absexp (r) (0) 'S'	Range allowed: 1-32,767.
TWAIT	(r)		Specifying the register (2-12) into which BTAM is to load the address of the ECB which was stopped complete.
	TERMTST		Coded as shown
	ECBLIST	relexp (r) (1)	Specifying the address of a parameter list.
WAIT	count	absexp (r) (0)	Range allowed: 0-4095 inclusive. Default=1
	ECB	symbol (r) (1)	Specifying the address of an ECB.
	ECBLIST	relexp (r) (1)	Specifying the address of a parameter list.
WRITE*	decb dtfbt area rln MF		Same as for CONTROL. See the special use of rln for the local 3270 under "Local Device-Dependent Considerations."
	optype	code	TI,TIR,TIL,TIO,TC,TCO,TT,TTR,TV,TVR,TL,TLR,TLL,TS, TSR,TSL,TB,TA,TN,TIE,TIX,TIXV,TIV,TE,TX,TXV,TD,TR,TW, TQ,TIC,TUS

Macro	Operand	Values Allowed	
WRITE* (Cont'd)	length	absexp (r) 'S'	Range allowed: 2-32,767
	entry	relexp (r) 'S'	Specifying the address of a terminal list or an entry in a terminal list or the address of a parameter list (BSC), or the address of a frame-change control character string (2760 OIU)

*For MF=L: No register notation is permitted; MF operand omitted: register notation not permitted for decb operand.
**A symbol may also be used to specify the address of a user-defined translation table.

```
************************************************************************
*                                                                      *
*  THIS SAMPLE PROGRAM IS DESIGNED TO POLL TWO IBM 1050S ON A          *
*  NONSWITCHED LINE, DETERMINE WHICH TERMINAL HAS A MESSAGE TO         *
*  BE READ, AND WRITE THIS MESSAGE TO THE OTHER TERMINAL.  THE         *
*  BTAM MACRO INSTRUCTIONS USED ARE                                    *
*                                                                      *
*                BTMOD                                                 *
*                DTFBT                                                 *
*                DFTRMLST                                              *
*                OPEN                                                  *
*                CLOSE                                                 *
*                READ                                                  *
*                WRITE                                                 *
*                WAIT                                                  *
*                REQBUF                                                *
*                RELBUF                                                *
*                                                                      *
************************************************************************

         START X'2800'
BEGIN    BALR  BASERG,0
         USING *,BASERG                  ESTABLISH ADDRESSABILITY
OPEN1    OPEN  DTFBT1                    OPEN LINE GROUP FOR USE
REQB1    REQBUF DTFBT1,(9),5             REQUEST 5 BUFFERS
         BAL   10,TBUFR                  CHECK TO SEE IF BUFFERS
***                                      WERE AVAILABLE
         LA    9,12(9)                   POINT REGISTER 9 TO DATA
***                                      PORTION OF BUFFER
***        POLL BOTH TERMINALS BY USING A WRAPAROUND POLLING LIST.
***        READ DATA INTO DATA FIELD OF FIRST BUFFER.
READ1    READ  DECB1,TI,DTFBT1,(9),24,POLL2,MF=E
         BC    7,DUMP1                   CHECK FOR GOOD SIO
REQB2    REQBUF DTFBT1,(8),6             REQUEST 6 MORE BUFFERS
         BAL   10,TBUFR                  CHECK TO SEE IF BUFFERS
***                                      WERE AVAILABLE
         LA    8,12(8)                   POINT REGISTER 8 TO DATA
***                                      PORTION OF BUFFER
         LA    DECBRG,DECB1              LOAD DECB ADDRESS INTO
***                                      REGISTER
         WAIT  1,ECB=DECB1               WAIT FOR COMPLETION OF READ
         CLI   0(DECBRG),X'7F'           CHECK COMPLETION CODE IN
***                                      DECB FOR NORMAL I/O
         BE    READC                     IF GOOD READ, CONTINUE
***                                      TO READ
READR    READ  DECB1,TP,MF=E             IF ERROR, TRY READING AGAIN
         BC    7,DUMP1                   CHECK FOR GOOD SIO
         WAIT  1,ECB=DECB1               WAIT FOR COMPLETION OF READ
         CLI   0(DECBRG),X'7F'           CHECK COMPLETION CODE IN
***                                      DECB FOR NORMAL COMPLETION
         BNE   DUMP1                     DUMP IF COMPLETION CODE
***                                      INDICATES ABNORMAL I/O
***
***        READ DATA INTO FIRST BUFFER OF SECOND SET
***
READC    READ  DECB1,TT,DTFBT1,(8),24,MF=E   CONTINUE TO READ
         BC    7,DUMP1                   CHECK FOR GOOD SIO
         WAIT  1,ECB=DECB1               WAIT FOR COMPLETION OF READ
         CLI   0(DECBRG),X'7F'           CHECK COMPLETION CODE IN
***                                      DECB FOR NORMAL COMPLETION
```

```
         BNE    READ2                        IF ABNORMAL, TRY READING AGAIN
         B      WRITEA                       BRANCH TO WRITE ACK FOR
***                                          NORMAL COMPLETION
READ2    READ   DECB1,TP,MF=E                IF ERROR, TRY READING AGAIN
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF READ
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BNE    DUMP1                        DUMP IF COMPLETION CODE
***                                          INDICATES ABNORMAL I/O
WRITEA   WRITE  DECB1,TA,MF=E                WRITE POSITIVE ACKNOWLEDGEMENT
***                                          FOR READ
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         LA     DTFREG,ADDRB1                LOAD TERMINAL B1 ADDRESS
***                                          INTO DTF REGISTER
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF WRITE
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BNE    WRITTEN                      WRITE NAK IF COMPLETION
         L      LOCREG,DECB1+20              LOAD ADDRESS OF DEVICE
***                                          COMMAND CODE FROM DECB
         CLI    0(LOCREG),X'E2'              DID TERMINAL A1 SUPPLY INPUT
         BE     WRITE1                       IF TRUE, THEN WRITE TO
***                                          TERMINAL B1
         LA     DTFREG,ADDRA1                OTHERWISE, WRITE TO TERMINAL A1
WRITE1   WRITE  DECB1,TI,DTFBT1,(9),24,(DTFREG),MF=E  WRITE MESSAGE
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF WRITE
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BNE    DUMP1                        DUMP IF COMPLETION CODE
***                                          INDICATES ABNORMAL I/O
WRITEC   WRITE  DECB1,TT,DTFBT1,(8),24,MF=E  CONTINUE WRITING
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         SH     9,=H'12'                     POINT REGISTER 9 TO
***                                          BEGINNING OF BUFFER
         RELBUF DTFBT1,(9)                   RELEASE FIRST GROUP OF
***                                          INPUT BUFFERS
         BAL    10,TBUFR                     CHECK TO SEE IF BUFFERS
***                                          WERE RELEASED
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF WRITE
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BE     WRITEN                       IF WRITE IS GOOD, GO TO
***                                          WRITE NAK
         WRITE  DECB1,TT,DTFBT1,MF=E         TRY TO WRITE AGAIN
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF WRITE
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BNE    DUMP1                        DUMP IF COMPLETION CODE
***                                          INDICATES ABNORMAL I/O
WRITEN   WRITE  DECB1,TN,MF=E                WRITE NAK
         BC     7,DUMP1                      CHECK FOR GOOD SIO
         SH     8,=H'12'                     POINT REGISTER TO
***                                          BEGINNING OF BUFFER
         RELBUF  DTFBT1,(8)                  RELEASE LAST GROUP OF
***                                          INPUT BUFFERS
         BAL    10,TBUFR                     CHECK TO SEE IF BUFFERS
***                                          WERE RELEASED
         WAIT   1,ECB=DECB1                  WAIT FOR COMPLETION OF WRITE
         CLI    0(DECBRG),X'7F'              CHECK COMPLETION CODE IN
***                                          DECB FOR NORMAL I/O
         BNE    DUMP1                        DUMP IF COMPLETION CODE
***                                          INDICATES ABNORMAL I/O
CLOSE1   CLOSE  DTFBT1                       CLOSE LINE GROUP FROM
***                                          FURTHER USE
         EOJ
```

```
TBUFR    LTR   15,15
         BZ    0(10)
DUMP1    PDUMP BEGIN,PRGEND
         B     CLOSE1                   GO TO CLOSE DTFBT AND
***                                     CANCEL JOB
***
***      REGISTER EQUATES
***
DECBRG   EQU   2
DTFREG   EQU   3
BASERG   EQU   5
LOCREG   EQU   6
***
***      DTFBT SYSTEM CONFIGURATION
***
DTFBT1   DTFBT LINELST=(004),SWITCH=NO,CU=2701,DEVICE=1050,              X
               BUFCB=BFPOOL,BUFNO=12,BUFL=36,SEPASMB=NO,                 X
               MODNAME=IJLBTMAX
***
***      TERMINAL LISTS
***
POLL2    DFTRMLST  WRAPLST,(E20B,E40B)  POLL A1 AND B1
ADDRA1   DFTRMLST  OPENLST,(E202)
ADDRB1   DFTRMLST  OPENLST,(E402)
***
***      READ TO CREATE A DECB
***
         READ  DECB1,TI,DTFBT1,,,,0,MF=L
         LTORG
ENDSECT  CSECT
PRGEND   DC    F'0'
         END   BEGIN
/*
```

```
***                                                                       ***
*****************************************************************************
***                                                                       ***
***            THIS BINARY SYNCHRONOUS PROGRAM IS DESIGNED                ***
***            TO ILLUSTRATE COMMUNICATION BETWEEN TWO                    ***
***            SYSTEM/360 MACHINES.A 2701 DATA ADAPTER UNIT               ***
***            IS USED IN THIS EXAMPLE AS THE CONTROL UNIT                ***
***            FOR THE TERMINAL CPU.THIS PROGRAM IS DESIGNED              ***
***            SO THAT ANY NUMBER OF MESSAGES CAN BE SENT                 ***
***            BETWEEN THE TWO CPU'S.ALL GOOD MESSAGES                    ***
***            RECEIVED BY EACH SYSTEM/360 ARE PRINTED OUT                ***
***            ON A PRINTER                                               ***
*****************************************************************************
***                                                                       ***
         START X'2800'
         BALR  5,0
         USING *,5              ESTABLISH ADDRESSIBILITY
STARTA   OPEN  DTFBTA           OPEN BTAM LINE GROUP FOR USE
         OPEN  MSPRINT          OPEN PRINTER LINE GROUP FOR USE
         SR    CNTRG,CNTRG      CLEAR COUNTER REGISTER
         BAL   RETRG,UPDATE     UPDATE MESSAGE NUMBER
***           READ MESSAGE FROM OTHER TERMINAL                            ***
READTI   READ  DECBA,TI,DTFBTA,INPUTA,LENGTHA,,0,MF=E
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR READ COMPLETION
         CLI   DECBA,X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                               NORMAL COMPLETION                       ***
         BNE   DUMPA            DUMP IF ABNORMAL COMPLETION
         CLC   INPUTA(4),ENDMSG   CHECK TO SEE IF MESSAGE READ IS
***                                     LAST MESSAGE                      ***
         BE    END        IF YES,PRINT MESSAGE ON CONSOLE AND DUMP
         MVC   MSGOUT(47),INPUTA+1   SET UP MESSAGE FOR PRINTER
         PUT MSPRINT,MSGOUT    PRINT INPUT MESSAGE
         CNTRL MSPRINT,SP,2       SPACE PRINTER
         PRTOV MSPRINT,12         CHECK FOR PRINTER OVERFLOW
         WRITE DECBA,TT,,MSG1A,49,MF=E
***        WRITE THE MESSAGE TO THE OTHER TERMINAL AND READ RESPONSE
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'   CHECK COMPLETION CODE IN DECB FOR
                                  NORMAL COMPLETION
         BNE   DUMPA            DUMP IF ABNORMAL COMPLETION
         WRITE DECBA,TR,MF=E    SEND EOT
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                               NORMAL COMPLETION                       ***
         BNE   DUMPA            DUMP IF ABNORMAL COMPLETION
         BAL   RETRG,UPDATE     UPDATE MESSAGE NUMBER
         WRITE DECBA,TIV,MSG1A,49,INLISTA,0,MF=E
***      WRITE MESSAGE AND THEN READ MESSAGE FROM OTHER TERMINAL          ***
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'      CHECK COMPLETION CODE IN DECB FOR
***                               NORMAL COMPLETION                       ***
         BNE   DUMPA            DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUT(47),INPUTA+1  SET UP MESSAGE FOR PRINTER
         PUT   MSPRINT,MSGOUT PRINT OUTPUT MESSAGE
         CNTRL MSPRINT,SP,2     SPACE PRINTER
         PRTOV MSPRINT,12       CHECK FOR PRINTER OVERFLOW
         READ  DECBA,TT,INPUTA,55,,0,MF=E    WRITE POSITIVE
***      ACKNOWLEDGEMENT FOR PREVIOUS MESSAGE AND READ NEXT MESSAGE ***
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR READ COMPLETION
         CLI   DECBA,X'7F'      CHECK COMPLETION CODE IN DECB FOR
***                               NORMAL COMPLETION                       ***
         BNE   DUMPA            DUMP IF ABNORMAL COMPLETION
         BAL   RETRG,UPDATE     UPDATE MESSAGE NUMBER
         B     READTI           CONTINUE READING
END      EQU   *
         READ DECB,TT,,INPUTA,,,MF=E      READ RESPONSE
         BC      7,DUMPA CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA       WAIT FOR READ COMPLETION
         EXCP  CCBWRITE EXECUTE CHANNEL PROGRAM TO WRITE OUT MESSAGE
***                      ON CONSOLE
         WAIT  CCBWRITE  WAIT FOR COMPLETION OF CHANNEL PROGRAM
DUMPA    EQU   *
         CLOSE DTFBTA           CLOSE BTAM LINE GROUP
         CLOSE MSPRINT   CLOSE PRINTER LINE GROUP
         PDUMP STARTA,FINISHA DUMP
         EOJ                    SPECIFY END OF JOB
```

```
***                                                                    ***
**************************************************************************
***                                                                    ***
***            THE UPDATE ROUTINE IS DESIGNED TO UPDATE THE            ***
***            MESSAGE NUMBER THAT APPEARS IN EACH MESSAGE
***            FORMAT ON THE PRINTER. THIS MESSAGE NUMBER IS
***            THE EXACT NUMBER OF MESSAGES THAT HAVE BEEN
***            READ FROM THE OTHER TERMINAL.
***
**************************************************************************
***                                                                    ***
UPDATE   LA    CNTRG,1(CNTRG)
         CVD   CNTRG,WORKA      CONVERT TO DECIMAL
         UNPK  NUMOUT(4),WORKA(8)   UNPACK
         MVI   NUMOUT,X'40'     MOVE SPACE INTO FIRST POSITION
         OI    NUMOUT+3,X'F0' REMOVE SIGN FROM MESSAGE FORMAT
         BR    RETRG            RETURN TO CALLER
DTFBTA   DTFBT DEVICE=S360,LINELST=(004),CU=2701,MODELST=(2),          X
               CTLCHAR=EBCDIC,FEATURE=(BSC),MODNAME=IJLBTMAX
***            READ TO INITIALIZE DECB
         READ  DECBA,TI,DTFBTA,INPUTA,30,,0,MF=L
***            DOS MACRO TO GENERATE PRINTER MODULE LOGIC
         PRMOD CONTROL=YES,PRINTOV=YES,RECFORM=FIXUNB,WORKA=YES
MSPRINT DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRT,PRINTOV=YES,   -
               WORKA=YES
MSG1A    EQU   *
         DC    X'02'
         DC    C'BINARY SYNCHRONOUS COMMUNICATION -- MESSAGE'
NUMOUT   DC    XL4'00'
         DC    X'03'
INPUTA   DS    80X
LENGTHA  EQU   55
CNTRG    EQU   6
RETRG    EQU   8
ENDMSG   DC    X'02'
         DC    C'END'
WORKA    DS    1D
CCBWRITE CCB   SYSLOG,CCWRITE
CCWRITE  CCW    X'09',MESSAGE,X'20',20
MESSAGE  DC    C'END MESSAGE RECEIVED'
INLISTA  DC    A(INPUTA,55)
MSGOUT   DC    CL132' '
MSGPRT   DC    CL132' '
FINISHA  DS    1F
         END
         START X'2800'
         BALR  4,0
         USING *,4              ESTABLISH ADDRESSABILITY
STARTB   EQU   *
         OPEN  DTFBTB           OPEN BTAM LINE GROUP FOR USE
         OPEN  MSPRNTB          OPEN PRINTER LINE GROUP FOR USE
         LA    5,12
         SR    6,6              CLEAR COUNTER REGISTER
         BAL   RETREG,UPDATEB UPDATE MESSAGE NUMBER
***      WRITE MESSAGE TO AND READ MESSAGE FROM TERMINAL A             ***
WRITIV   WRITE DECBB,TIV,DTFBTB,MSG1B,49,INLISTB,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB        WAIT FOR COMPLETION OF WRITE
         CLI   DECBB,X'7F'      CHECK COMPLETION CODE IN DECB FOR
***                              NORMAL COMPLETION                     ***
         BNE   DUMPB            DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUTB(47),INPUTB+1  SET UP MESSAGE FOR PRINTER
         PUT    MSPRNTB,MSGOUTB  PRINT RECEIVED MESSAGE
         CNTRL MSPRNTB,SP,2  SPACE PRINTER
         PRTOV MSPRNTB,12     CHECK FOR PRINTER OVERFLOW
***       WRITE POSITIVE ACKNOWLEDGEMENT AND READ MESSAGE              ***
         READ  DECBB,TT,INPUTB,55,,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB        WAIT FOR READ COMPLETION
         CLI   DECBB,X'7F'      CHECK COMPLETION CODE IN DECB FOR
***                              NORMAL COMPLETION                     ***
         BNE   DUMPB            DUMP IF ABNORMAL COMPLETION
         BAL   RETREG,UPDATEB  UPDATE MESSAGE NUMBER
***                             PREPARE AND READ MESSAGE               ***
         READ  DECBB,TI,,INPUTB,55,,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB        WAIT FOR READ COMPLETION
         CLI   DECBB,X'7F'      CHECK COMPLETION CODE IN DECB FOR
***                              NORMAL COMPLETION                     ***
         BNE   DUMPB            DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUTB(47),INPUTB+1  SET UP MESSAGE FOR PRINTER
         PUT   MSPRNTB,MSGOUTB  PRINT RECEIVED MESSAGE
         CNTRL MSPRNTB,SP,2      SPACE PRINTER
         PRTOV MSPRNTB,12     CHECK FOR PRINTER OVERFLOW
         WRITE DECBB,TT,,MSG1B,49,MF=E  WRITE MESSAGE TO TERMINAL A
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB        WAIT FOR WRITE COMPLETION
         CLI   DECBB,X'7F'      CHECK COMPLETION CODE IN DECB FOR
```

```
***                               NORMAL COMPLETION
         BNE   DUMPB              DUMP IF ABNORMAL COMPLETION
         WRITE DECBB,TR,MF=E      SEND EOT
         BC    7,DUMPB   CHECK    FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB          WAIT FOR WRITE COMPLETION
         CLI   DECBB,X'7F'        CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION                    ***
         BNE   DUMPB              DUMP IF ABNORMAL COMPLETION
         BAL   RETREG,UPDATEB UPDATE MESSAGE NUMBER
         BCT   5,WRITIV
         LA    5,12
EXCPWRIT EXCP  CCBWRIT1           EXECUTE CHANNEL PROGRAM TO WRITE OUT
***                                 MESSAGE ON CONSOLE                   ***
         WAIT  CCBWRIT1  WAIT FOR COMPLETION OF CHANNEL PROGRAM
         CLI   MSINPUT,X'C3'      CHECK TO SEE IF THERE ARE MORE
***                                 MESSAGES TO WRITE                    ***
         BE    WRITIV             IF SO,CONTINUE WRITING
         CLI   MSINPUT,X'83'      ARE THERE MORE MESSAGES TO WRITE
         BE    WRITIV             IF SO,CONTINUE WRITING
         CLI   MSINPUT,X'E2'      IS STOP INDICATED
         BE    DUMPB1             IF YES,CONCLUDE CHANNEL PROGRAM AND DUMP
         CLI   MSINPUT,X'A2'      IS STOP INDICATED
         BNE   EXCPWRIT  IF NO,BRANCH TO EXECUTE CHANNEL PROGRAM AGAIN
DUMPB1   EQU   *        DUMP TO CONCLUDE PROGRAM
         WRITE DECBB,TI,,ENDMSG,5,MF=E  WRITE END MESSAGE
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB          WAIT FOR COMPLETION OF WRITE
         WRITE DECBB,TR,MF=E          SEND EOT
         BC     7,DUMP           CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB         WAIT FOR WRITE COMPLETION
DUMPB    EQU   *                  DUMP TO CONCLUDE PROGRAM
         CLOSE DTFBTB             CLOSE BTAM LINE GROUP
         CLOSE MSPRNTB   CLOSE PRINTER LINE GROUP
         PDUMP STARTB,FINISHB  DUMP FROM STARTB TO FINISHB
         EOJ
***                                                                      ***
****************************************************************************
***                                                                      ***
***            THE UPDATE ROUTINE IS DESIGNED TO UPDATE THE              ***
***            MESSAGE NUMBER THAT APPEARS IN EACH MESSAGE
***            FORMAT ON THE PRINTER. THIS MESSAGE NUMBER IS
***            THE EXACT NUMBER OF MESSAGES THAT HAVE BEEN
***            READ FROM THE OTHER TERMINAL.
***
****************************************************************************
***                                                                      ***
UPDATEB  LA    6,1(6)
         CVD   6,WORKB            CONVERT COUNTER TO DECIMAL
         UNPK  OUTB(4),WORKB(8)  UNPACK
         MVI   OUTB,X'40'         MOVE SPACE INTO FIRST POSITION
         OI    OUTB+3,X'F0'       REMOVE SIGN FROM MESSAGE FORMAT
         BR    RETREG             RETURN TO CALLER
DTFBTB   DTFBT DEVICE=S360,LINELST=(005),CU=2701,MODELST=(2)               X
               CTLCHAR=EBCDIC,FEATURE=(BSC),MODNAME=IJLBTMAX
MSPRNTB  DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRTB,                 X
               PRINTOV=YES,WORKA=YES
***            READ TO INITIALIZE DECB
         READ  DECBB,TI,DTFBTB,INPUTB,30,,0,MF=L
***            DOS MACRO TO GENERATE PRINTER MODULE LOGIC
         PRMOD CONTROL=YES,PRINTOV=YES,RECFORM=FIXUNB,WORKA=YES
MSPRNTB  DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRTB,PRINTOV=YES,     -
               WORKA=YES
MSG1B    EQU   *
         DC    X'02'
         DC    C'BINARY SYNCHRONOUS COMMUNICATION -- MESSAGE'
OUTB     DC    XL4'00'
         DC    X'03'
ENDMSG   DC    X'02'
         DC    C'END'
         DC    X'03'
RETREG   EQU   10
MSINPUT  DS    CL4'00'
         SPACE 2
INPUTB   DS    80X
INLISTB  DC    A(INPUTB,55)
WORKB    DS    1D
CCBWRIT1 CCB   SYSLOG,CCWRITE
CCWRITE  CCW   X'09',MESSU,X'60',26
         CCW   X'0A',MSINPUT,0,2
MESSU    DC    C'REPLY C TO WRITE,S TO STOP'
MSGOUTB  DC    CL132'  '
MSGPRTB  DC    CL132'  '
ENDSECT  CSECT
FINISHB  DS    1F
         END
```

The following illustration shows the relationship between the two CPU's of the above binary synchronous program. SIDEA and SIDEB represent the two CPU's.

SIDEA	SIDEB
READ TI	WRITE TIV
Prepare to read	Write ENQ
Read ENQ	Read ACK-0
Write ACK-0	Write message
Read message	Read message
WRITE TT	READ TT
Write message	Write ACK-1
Read response	Read message
WRITE TR	READ TI
Write EOT	Prepare
	Read ENQ
WRITE TIV	Write ACK-0
Write ENQ	Read message
Read ACK-0	
Write message	WRITE TT
Read message	Write message
	Read response
READ TT	
Write ACK-1	WRITE TR
Read message	Write EOT

This appendix includes code charts for EBCDIC, USASCII, and 6-bit TRANSCODE.

SIX BIT TRANSCODE

Code Positions → 0 1 / 2 3 4 5	0 0	0 1	1 0	1 1
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	¤ 12-8-4	* 11-8-4	% 0-8-4	@ 8-4
1 1 0 1	BEL 0-9-8-7	US 11-9-8-7	ENQ 0-9-8-5	NAK 9-8-5
1 1 1 0	SUB 9-8-7	EOT 9-7	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: USASCII	EBCDIC	Transcode
ACK-0	DLE,0	DLE,X'70'	DLE, - (hyphen)
ACK-1	DLE,1	DLE,X'61'	DLE,T
WACK	DLE,;	DLE,X'6B'	DLE,Ƶ
RVI	DLE,<	DLE,X'7C'	DLE,2

Bit Positions 4, 5, 6, and 7	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	Digit Punches
0000	(1) NUL	(2) DLE	(3) DS	(4)	SP (5)	& (6)	- (7)	(8)	
0001	SOH	DC1	SOS				/ (13)		1
0010	STX	DC2	FS	SYN					2
0011	ETX	DC3							3
0100	PF	RES	BYP	PN					4
0101	HT	NL	LF	RS					5
0110	LC	BS	EOB	UC					6
0111	DEL	IL	PRE	EOT					7
1000		CAN							8
Zone Punches	9 12	9 11	9 0	9	9 12 0	9 12 11	9 11 0	9 12 11 0	

(Bit Positions 0 and 1 → 00, 01; Bit Positions 2 and 3 → 00, 01, 10, 11)

Bit Positions 4, 5, 6 and 7	10 00	10 01	10 10	10 11	11 00	11 01	11 10	11 11	Digit Punches
0000					(9)	(10)	(11)	0 (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
Zone Punches	12 0	12 11	11 0	12 11 0	12	11	0		

(Bit Positions 0 and 1 → 10, 11; Bit Positions 2 and 3 → 00, 01, 10, 11)

Bit Positions 4, 5, 6, and 7	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	Digit Punches
1001		EM							8-1
1010	SMM	CC	SM		¢	!	(15)	:	8-2
1011		CU1	CU2	CU3	.	$	,	#	8-3
1100	FF	IFS		DC4	<	*	%	@	8-4
	CR	IGS	ENQ	NAK	(	)	_	'	8-5
1110	SO	IRS	ACK		+	;	>	=	8-6
1111	SI	IUS	BEL	SUB	|	¬	?		8-7
Zone Punches	9 12	9 11	9 0	9	12	11	0		

(Bit Positions 0 and 1 → 00, 01; Bit Positions 2 and 3 → 00, 01, 10, 11)

Bit Positions 4, 5, 6, and 7	10 00	10 01	10 10	10 11	11 00	11 01	11 10	11 11	Digit Punches
1010									8-2
1011									8-3
1100									8-4
1101									8-5
1110									8-6
1111									8-7
Zone Punches	12 0	12 11	11 0	12 11 0	9 12 0	9 12 11	9 11 0	9 12 11 0	

(Bit Positions 0 and 1 → 10, 11; Bit Positions 2 and 3 → 00, 01, 10, 11)

(1) 12-0-9-8-1
(2) 12-11-9-8-1
(3) 11-0-9-8-1
(4) 12-11-0-9-8-1
(5) No Punches
(6) 12
(7) 11
(8) 12-11-0
(9) 12-0
(10) 11-0
(11) 0-8-2
(12) 0
(13) 0-1
(14) 11-0-9-1
(15) 12-11

USASCII	EBCDIC Equivalents
RS	IRS
ETB	EOB
ESC	PRE
FS	IFS

S/370		2980 Graphics or C. C.					
		Num. (L.C.)			Alph. (U.C.)		
Code	Graphic or C.C.	1	2	4	1	2	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	DC1						
13	TM						
14	RES			tr.pg light			tr.pg light
15	NL	NL	NL	NL	NL	NL	NL
16	BS						
17	IL	msg. light		msg. light	msg. light		msg. light
18	CAN						
19	EM						
1A	CC						
1B	CU1						

S/370		2980 Graphics or C.C.					
		Num. (L.C.)			Alph. (U.C.)		
Code	Graphic or C.C.	1	2	4	1	2	4
1C	IFS						
1D	IGS						
1E	IRS						
1F	IUS						
20	DS						
21	SOS						
22	FS						
23							
24	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

*Also used as a Terminal Selection Character.

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C.) 1	2	4	Alpha. (U.C.) 1	2	4
38							
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	←						
4B	.	3	.		.	¬	
4C	<						
4D	(						
4E	+						
4F	1						
50	&	val. I.D. char	&	val. I.D. char	&	+	&
51							
52							
53							

*Also used as a Terminal Selection Character

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C.) 1	2	4	Alph. (U.C.) 1	2	4
54							
55							
56							
57							
58							
59							
5A	!						
5B	$	-	$		$	!	
5C	*	$	@		*	¢	
5D	)						
5E	;						
5F	¬						
60	-	F	-		-	_	
61	/	T	/		/	?	
62							
63							
64							
65							
66							
67							
68							
69							
6A							
6B	,	2	,	,	,	|	,
6C	%						
6D	_						
6E	>						
6F	?						
70							
71							

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C.) 1	2	4	Alph. (U.C.) 1	2	4
72							
73							
74							
75							
76							
77							
78							
79							
7A	:						
7B	#	$	#	ǂ	#	"	⧻
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							

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C.) 1	2	4	Alph. (U.C.) 1	2	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						
48	y						
A9	z						
AA							
AB							

S/370		2980 Graphics or C.C.					
	Graphic	Num. (L.C.)			Alph. (U.C.)		
Code	or C.C.	1	2	4	1	2	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	‡	c	C	C	C	C
C4	D	N B	d	?	D	D	D
C5	E	X	e	E	E	E	E
C6	f	O B	f	F	F	F	F
C7	G	$\overline{s}$	g	G	G	G	G

S/370		2980 Graphics or C.C.					
	Graphic	Num. (L.C.)			Alph. (U.C.)		
Code	or C.C.	1	2	4	1	2	4
C8	H	$\underline{00}$	h	#	H	H	6
C9	I	8	i	$\overline{C}$	I	I	0
CA							
CB*				M			2
CC							
CD							
CE							
CF							
DO*							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	O B	Q	Q	5
D9	R	$\overline{A}$	r	C ∇	R	R	_
DA							
DB							
DC							
DD							
DE							
DF							
E0*				Δ $			4
E1				/			Y
E2	S	T F	s	$	S	S	S

*A non-EBCDIC code.

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C) 1	2	4	Alph. (U.C.) 1	2	4
E3	T	$\overline{B}$	t	Δ	T	T	T
E4	U	7	u	$\overline{M}$	U	U	U
E5	V	S P	v		V	V	V
E6	W	Q	w	*	W	W	W
E7	X	M ∇	x	N B	X	X	X
E8	Y	$\overline{D}$	y	↧	Y	Y	3
E9	Z	V	z		z	z	z
EA							
EB*				I			1
EC	┌┤						
ED							
EE							
EF							
F0	0	U	0	0	0	)	.
F1	1	↕	1	1	1	=	L

*A non-EBCDIC code.

S/370		2980 Graphics or C.C.					
Code	Graphic or C.C.	Num. (L.C.) 1	2	4	Alph. (U.C.) 1	2	4
F2	2	H	2	2	2	<	$
F3	3	C F	3	3	3	;	#
F4	4	$\overline{M}$	4	4	4	:	O
F5	5	$\overline{L}$	5	5	5	%	P
F6	6	$\overline{C}$	6	6	6	'	*
F7	7	M O	7	7	7	>	7
F8	8	M ∇	8	8	8	*	8
F9	9	C ∇	9	9	9	(	9
FA	LVM						
FB							
FC							
FD							
FE							
FF	E.O.						

WHAT RMSR IS

RMSR (Recovery Management Support Recording) is a DOS facility that allows records to be kept in the system of errors and other events that may aid in the analysis of problems. The RMSR recorder file is created during IPL (Initial Program Load) if requested by the BTAM user; records are written to the file automatically each time certain conditions occur. The file can be retrieved and printed by executing the DOS EREP program. RMSR replaces the DOS OBR/SDR facility.

HOW BTAM PROVIDES RMSR

If requested, BTAM will automatically maintain certain counts in an RMSR table and write records to the RMSR recorder file. The RMSR table is generated in the application program by assembling the RMSRTAB macro instruction. A count is kept by BTAM in the RMSR table of transmissions and recoverable errors. Operands in the RMSRTAB macro instruction allow these counts to be kept either by line or device. A record is written to the recorder file whenever:

- An unrecoverable error occurs on a line or device.
- Either the transmission count exceeds a count of 255 or the recoverable error count exceeds 15.
- A BTRD macro instruction is issued to have the counts in the RMSR table written prior to terminating the program.

Whenever a record is written, whether because of an unrecoverable error, a count exceeding a threshold, or a BTRD macro instruction, both the transmission count and error count are written and then both are reset to zero.

The recoverable error count is incremented each time any of the following conditions occur:

- Unit exception on WRITE
- Time out on Prepare and nontext Read
- Time out on Dial, Disable, and Enable
- Intervention required
- Overrun
- Bus out check on Write and Dial
- Bus out check at initial selection
- Data check on Write
- Data check on Read
- Data check on Poll
- Parity error for 2740-M2
- VRC error for 2740-M2
- Electronic failure for 2740-M2
- Selectric failure for 2740-M2

Note: RMSR in BTAM does not provide for processing in CE mode.

HOW TO HAVE RMSR FOR A BTAM APPLICATION

- A supervisor must be assembled with ERRLOG=YES in the SPVSR macro instruction.
- A SYSREC (Recorder) file must be created via the Job Control statement SET RF=CREATE and DLBL and EXTENT cards. This procedure is described in the DOS System Control and System Service Programs publication, GC24-5036.
- A BTMOD must be used with ERLOGIC=E and RMSR=YES specified. (OBRSDR=YES will be accepted; however, an MNOTE will be generated.) BTAM does not support RMSR counts and LERB counts simultaneously.
- RMSR tables must be generated in the user's program with the RMSRTAB macro instruction for each line and/or terminal on which the user wants statistics kept. The counts are kept optionally by line or by terminal for all nonswitched multipoint devices. The address of this macro instruction must be specified in the LERBADR field of the DTFBT macro instruction.

Note: Programs using the SDRTAB macro instruction will not have to be rewritten; the SDRTAB macro instruction will assemble RMSR tables and RMSR counts will be maintained by BTAM. An MNOTE will be generated.

Offset				
0	Relative line number	Channel and unit	Channel and unit	Terminal ID
4	Terminal ID	Terminal ID	Terminal ID	Device type
8	Characteristics	Transmission counter	Error counter	Reserved
0C	Reserved	Reserved	Reserved	Reserved
10	Reserved	Length of terminal ID	Length of terminal ID	Guard byte

Figure 94. RMSR Table

FORMAT OF THE RMSR TABLE

Figure 94 shows the format of the RMSR table. The meanings of the fields are:

Relative line number
: the relative number within the line group.

Channel and unit
: the physical line number.

Terminal ID
: the polling or addressing characters of a terminal or component on which counts are to be kept (not used if counts are being kept only on a line).

Device type
: the type of device as specified in the DTFBT macro instruction in the DEVICE operand.

Characteristics
: reserved for DOS use.

Transmission counter
: the number of READ, WRITE, and CONTROL macro instructions issued on the line or to the device.

Error counter
: total number of recoverable errors recorded since the last record was written to the RMSR recorder file.

Length of terminal ID
: reserved for system use.

Guard byte
: reserved for system use.

Examples of Using the RMSRTAB Macro Instruction

Example 1: Assume a configuration of two lines associated with a line group defined by the DTFBT macro instruction with statement name LINEGR1 shown below.

```
LINEGR1    DTFBT  LINELST=(001,002),CU=2701,
                  DEVICE=2780
```

If each of these lines had two terminals (with terminal IDs 6204 and 6203 on the first line and terminal IDs A1A2 and A2A4 on the second line) and counts were to be kept on all four terminals, RMSRTAB would be coded like this:

```
KEEPTRK1    RMSRTAB  2,(6204,6203),
                     (A1A2,A2A4)
```

During assembly, one 20-byte RMSR table would be generated for each terminal. During execution, error and transmission counts would be kept in these tables.

Note: Only one RMSRTAB macro instruction can be associated with a line group.

Example 2: Assume that there are four lines associated with a line group defined by the DTFBT macro instruction with statement name LINEGR2 shown below:

```
LINEGR2    DTFBT  LINELST=(002,004,006,008),
                  CU=2703,DEVICE=BSC3
```

If it was desired to keep counts on the first and fourth lines by terminal (with terminals 6204 and 6203 attached to the first line and terminals E1E2 and E3E4 attached to the fourth) and to keep counts on the second and third lines by line only, RMSRTAB would be coded like this:

```
KEEPRTK2    RMSRTAB  4,(6204,6203),,,
                     (E1E2,E3E4)
```

During assembly, one 20-byte RMSR table would be generated for each terminal on the first and fourth lines and one for each of the second and third lines. During execution, counts would be kept by terminal on the first and fourth lines and by line on the second and third lines.

APPENDIX I: WORLD TRADE TELEGRAPH CODES ITA2 AND ZSC3

Code Combination No.	Elements 12 345	Lettershift Hex Code	Lettershift Character	Figureshift Hex Code	Figureshift Character ITA2	Figureshift Character 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	0B	G	2B	N/A	0
8	00 101	05	H	25	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	1B		3B	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
LTPS = Letters shift
FIGS = Figures shift

The three figures in this appendix show the control unit and device addresses (Figures 95, 96, and 97) to use when defining polling or selection terminal lists with the DFTRMLST macro instruction and for identifying control unit and device addresses on receiving input. On input, the addresses shown must be identified before any code translation from ASCII to EBCDIC.

The character forms shown are those used in the U.S.A. interface code. Certain characters must be replaced by equivalent hexadecimal values for use in the United Kingdom, Belgium, France, Austria, and Germany.

Control Unit	Address in Character Form	Address in Hexadecimal Form (EBCDIC)	Address in Hexadecimal Form (ASCII)
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

Figure 95. Control Unit Addresses for Polling and Identification

Control Unit	Address in Character Form	Address in Hexadecimal Form (EBCDIC)	Address in Hexadecimal Form (ASCII)
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	5A
10	¦ (\)	6A	5C
11	,	6B	2C
12	%	6C	25
13	_	6D	5F
14	>	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 96. Control Unit Addresses for Selection

Device	Address in Character Form	Address in Hexadecimal Form (EBCDIC)	Address in Hexadecimal Form (ASCII)
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 97. Device Addresses for Polling, Selection, and Identification

In creating and interpreting certain characters in a 3270 Information Display System data stream, information that consists initially of six bits must be formed into an eight-bit configuration. The characters that have this six-bit structure that requires conversion to an eight-bit structure are: the WCC, the CCC, the 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 Component Description_, GA27-2749). Figure 98 should be used to convert six-bit structured data to an eight-bit structure for output and to interpret the eight-bit structure that is received on input.

EBCDIC Bits 23 4567	EBCDIC Hex	Graphic Character	EBCDIC Bits 23 4567	EBCDIC Hex	Graphic Character	EBCDIC Bits 23 4567	EBCDIC Hex	Graphic Character	EBCDIC Bits 23 4567	EBCDIC Hex	Graphic Character
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	¢	01 1010	5A	!	10 1010	6A		11 1010	7A	:
00 1011	4B	.	01 1011	5B	$	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	|	01 1111	5F	¬	10 1111	6F	?	11 1111	7F	"

Figure 98. I/O Interface Code for Six-Bit Structured Data

```
SAMP327L CSECT
         SPACE 5
*        THIS IS A SAMPLE PROBLEM FOR A LOCAL 3270 INFORMATION DISPLAY
*  SYSTEM.  PLEASE NOTE THAT THIS PROGRAM HAS BEEN WRITTEN FOR TWO
*  3270 DEVICES, AT LEAST ONE OF WHICH MUST BE A 3277 DISPLAY.
*
*        TO ASSEMBLE THIS SAMPLE PROBLEM, FIRST REMOVE ALL CATALS AND
*  BKEND CARDS, AND THEN PUNCH AND INSERT THE NECESSARY JOB CONTROL
*  CARDS FOR AN ASSEMBLY.
*
*        NOW THE OUTPUT FROM THE ASSEMBLER (THE OBJECT DECK) MUST BE
*  LINKAGE-EDITED TO THE CORE IMAGE LIBRARY IN THE BG, F1, OR F2
*  PARTITION.
*
*        THE FOLLOWING I/O ASSIGNMENTS MUST BE MADE TO ASSOCIATE THE
*  3270 DEVICES WITH THE PROGRAM:
*                          // ASSGN SYS005,X'CUU'
*                          // ASSGN SYS006,X'CUU'
*
*        TO EXECUTE THIS SAMPLE PROGRAM, ENTER FROM THE OPERATORS
*  CONSOLE OR THROUGH THE CARD READER THE FOLLOWING:
*                          // EXEC SAMP327L
*
*        IT MAY BE NECESSARY TO ALTER THIS SAMPLE PROBLEM SO THAT IT
*  MAY FUNCTION WITH MORE THAN TWO DEVICES.  TO DO SO, THE FOLLOWING
*  CARDS MUST BE CHANGED IN THE SOURCE DECK:
*                          1. DTFBT MACRO (LINELST OPERAND)
*                          2. SDRTAB MACRO
*
*        THIS SAMPLE PROGRAM IS RESTRICTED TO A MAXIMIM OF 32 DEVICES,
*  ALL OF WHICH MUST BE ATTACHED TO ONE LOCAL 3272 CONTROL UNIT.
         EJECT
*        REGISTER EQUATES
REGZERO  EQU   0                  REGISTER 0
REG2     EQU   2                  WORK REGISTER
LCBREG   EQU   3                  LCB ADDRESS
WORKREG  EQU   3                  WORK REGISTER
MSGADDR  EQU   4                  ADDR OF OUTPUT MESSAGE
MSGLEN   EQU   5                  LENGTH OF OUTPUT MESSAGE
RLNREG   EQU   6                  RELATIVE LINE NUMBER
DTFREG   EQU   7                  DTFBT ADDRESS
FMTREG   EQU   8                  FORMAT IDENTIFIER
DSPTABRG EQU   9                  ADDRESS OF DISPLAY TABLE
LNKREG   EQU   10                 LINKAGE REGISTER
BASEREG  EQU   11                 FIRST BASE REG
BASEREG2 EQU   12                 SENOND BASE REG
PTRTAB   EQU   13                 PRINTER RLN TABLE ADDRESS
RTNCDRG  EQU   15                 RETURN CODE REGISTER
REG15    EQU   15                 REGISTER 15
         SPACE 5
*        EQUATES
ZERO     EQU   0                  LENGTH OF 0
PTR3270  EQU   1                  LCB PRINTER INDICATOR FLAG
ONE      EQU   1                  LENGTH OF 1
TWO      EQU   2                  LENGTH OF 2
FOUR     EQU   4                  FORMAT 1 IDENTIFIER
EIGHT    EQU   8                  FORMAT 2 IDENTIFIER
PTRFLAG  EQU   16                 DISP TO PRINTER FLAG IN LCB
EIGTEEN  EQU   18                 DISP INTO DECB FOR RESIDUAL COUNT
```

```
THRTY9   EQU   39                     DISP INTO DECB FOR POLLING POINTER
DTFLEN   EQU   40                     DTFBT LENGTH
LCBLEN   EQU   64                     LCB LENGTH
FOURTY2  EQU   X'42'                  RFT RECEIVED COMP CODE
SIXTY4   EQU   X'64'                  DEVICE BUFFER CLOBERED COMP CODE
PA1      EQU   X'6C'                  ATTENTION ID FOR PA1 KEY
CLEAR    EQU   X'6D'                  ATTENTION ID FOR CLEAR KEY
PA2      EQU   X'6E'                  ATTENTION ID FOR PA2 (CNCL) KEY
ENTER    EQU   X'7D'                  ATTENTION ID FOR ENTER KEY
SEVENF   EQU   X'7F'                  NORMAL COMPLETION CODE
LASTLCB  EQU   X'8C'                  LAST LCB INDICATOR
PRINTER  EQU   X'8C'                  PRINTER IS DEFINED
PRNTWCC  EQU   X'D8'                  WCC WITH START PRINT BIT ON
HEXFF    EQU   X'FF'                  HEX CONSTANT OF 255
         EJECT
         BALR  BASEREG,0              ESTABLISH
         USING *,BASEREG,BASEREG2              ADDRESSABILITY
         LR    BASEREG2,BASEREG       INITIALIZE
         AH    BASEREG2,H4C96                  SECOND BASE
*        OPEN THE LINE GROUP
         OPEN  DTFBTL
         SR    RLNREG,RLNREG          CLEAR RLN REG
         L     DTFREG,VDTF            ADDRESS THE DTFBT
         LA    LCBREG,DTFLEN(DTFREG) ADDRESS FIRST LCB
         LA    PTRTAB,PTRRLNTB        ADDRESS THE PRINTER TABLE
         LA    DSPTABRG,DSPTAB        ADDRESS THE DISPLAY TABLE
CHECKPTR EQU   *
         TM    PTRFLAG(LCBREG),PTR3270  DEVICE A PRINTER
         BNO   INITIAL                NO, WRITE INITIAL MESSAGE
         OI    FLAGS,PRINTER          SET PRINTER DEFINED FLAG
         ST    PTRTAB,CURRPTR         SAVE POINTER TO CURRENT PTR RLN
         STC   RLNREG,ZERO(PTRTAB)    STORE PRINTER RLN
         LA    PTRTAB,ONE(PTRTAB)     ADDR NEXT ENTRY IN PTR TABLE
LCBUPDT  EQU   *
         LA    RLNREG,ONE(RLNREG)     GET NEXT RLN
         TM    PTRFLAG(LCBREG),LASTLCB  LAST LCB
         BO    INITIAL                YES, GO WRITE WITH INVALID RLN
         LA    LCBREG,LCBLEN(LCBREG) ADDR NEXT LCB
         B     CHECKPTR               CHECK NEXT DEVICE
INITIAL   EQU  *
         LA    MSGADDR,FORMAT0        ADDR OF FORMAT0 MESSAGE
         LA    MSGLEN,FMT0SZ          LENGTH OF MESSAGE
         BAL   LNKREG,WRITETS         GO WRITE FORMAT 0
         BAL   LNKREG,RETCODE         CHECK RETURN CODE
         BAL   LNKREG,WAITD           WAIT FOR COMPLETION
         B     LCBUPDT                UPDATE LCB POINTER
READ     EQU   *
         XC    INAREA(255),INAREA     CLEAR INPUT
         XC    INAREA(43),INAREA               AREA
         IC    RLNREG,RLNREAD         GET RLN OF LAST DEVICE READ
         LA    RLNREG,ONE(RLNREG)     GET NEXT RLN
         CH    RLNREG,MAXRLN          HAVE WE EXCEEDED MAX RLN
         BNL   RLNZERO                YES, USE RLN OF 0
         B     GOREAD                 NO, GO ISSUE THE READ
RLNZERO  EQU   *
         SR    RLNREG,RLNREG          GET RLN OF 0
GOREAD   EQU   *
         BAL   LNKREG,READTI          GO READ A DISPLAY
         BAL   LNKREG,RETCODE         CHECK RETURN CODE
         BAL   LNKREG,WAITD           WAIT FOR COMPLETION
         IC    RLNREG,DECBD+THRTY9 GET RLN OF DEVICE JUST READ
```

```
         STC    RLNREG,RLNREAD      SAVE FOR NEXT READ
         SR     FMTREG,FMTREG       CLEAR FORMAT REG
         LA     DSPTABRG,DSPTAB     ADDR OF DISPLAY TABLE
         IC     FMTREG,ZERO(DSPTABRG,RLNREG)  GET FORMAT ID
         B      FORMATER(FMTREG)
FORMATER EQU    *
         B      FMT0                FORMAT 0 ON SCREEN
         B      FMT1                FORMAT 1 ON SCREEN
         B      FMT2                FORMAT 2 ON SCREEN
FMT0     EQU    *
*   VERIFY THE NAME AND SOCIAL SECURITY NUMBER. ASSUMING THAT THEY
* ARE VALID, WE SHALL CONTINUE PROCESSING.
FMT01    EQU    *
         LA     FMTREG,FCUR         GET FORMAT 1 ID
         STC    FMTREG,ZERO(DSPTABRG,RLNREG)  STORE IN DISPLAY TABLE
         LA     MSGADDR,FORMAT1     ADDR OF FORMAT1 MESSAGE
         LA     MSGLEN,FMT1SZ       LENGTH OF MESSAGE
         BAL    LNKREG,WRITETS      GO WRITE FORMAT 1
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
         B      READ                GO READ ANOTHER DISPLAY
FMT1     EQU    *
         CLI    INAREA,ENTER        ENTER KEY INTERRUPT
         BE     ENTERINT            YES, GO UPDATE RECORDS
         CLI    INAREA,PA1          PA1 KEY INTERRUPT
         BE     PA1INT              YES, GO MAKE HARD COPY
         CLI    INAREA,PA2          PA2 OR CNCL KEY INTERRUPT
         BE     PA2INT              YES, GO DEACTIVATE TERMINAL
         CLI    INAREA,CLEAR        CLEAR KEY INTERRUPT
         BE     CLEARINT            YES, GO WRITE FORMAT 2
         B      READ                IGNORE THE INTERRUPT AND GO READ
ENTERINT EQU    *
*   CREATE A NEW OR UPDATE AN EXISTING ENTRY IN YOUR PERMANENT
* DATA SET.
ENTERIN1 EQU    *
         BAL    LNKREG,WRITETUS     GO ERASE ALL UNPROTECTED DATA
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
         B      READ                GO READ ANOTHER DISPLAY
PA1INT   EQU    *
         TM     FLAGS,PRINTER       IS A PRINTER DEFINED
         BNO    NOPRINT             NO, CANNOT HONOR PRINT REQUEST
         L      PTRTAB,CURRPTR      ADDR PRINTER RLN TABLE
         CLI    ZERO(PTRTAB),HEXFF  CHECK FOR END OF TABLE
         BNE    PA1INT1             NO, GO READ BUFFER
         LA     PTRTAB,PTRRLNTB     ADDR BEGINNING OF PTR TABLE
         ST     PTRTAB,CURRPTR      SAVE CURRENT PTR TABLE ENTRY
PA1INT1  EQU    *
         BAL    LNKREG,READTB       GO READ ENTINE BUFFER
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
         IC     RLNREG,ZERO(PTRTAB) GET RLN OF PRINTER
         MVI    INAREA+TWO,PRNTWCC  GET WCC WITH START PRINT BIT ON
         LA     MSGADDR,INAREA+TWO  ADDR OF PRINTER OUTPUT MSG
         LH     WORKREG,DECBD+EIGTEEN GET RESIDUAL COUNT
         L      MSGLEN,TWOTHO       GET ORIGINAL COUNT
         SR     MSGLEN,WORKREG      GET NUMBER OF BYTES READ
         S      MSGLEN,TWO2         DECREASE LENGTH BY TWO
         BAL    LNKREG,WRITETS      GO WRITE TO PRINTER
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
```

```
         LA    PTRTAB,ONE(PTRTAB)  POINT TO NEXT PRINTER ENTRY
         ST    PTRTAB,CURRPTR      SAVE IT FOR NEXT TIME
         IC    RLNREG,RLNREAD      GET RLN THAT REQUESTED PRINT
         B     ENTERIN1            GO ERASE UNPROTECTED DATA
PA2INT   EQU   *
*   DETERMINE IF ANY DATA WAS ENTERED. IF SO, CREATE A NEW OR UPDATE
* AN EXISTING ENTRY IN YOUR PERMANENT DATA SET. NOW DEACTIVATE THE
* TERMINAL.
PA2INT1  EQU   *
         CHGNTFY DTFBTL,ATTLST,(RLNREG),SKIP
         LA    MSGADDR,CLOSEMG     ADDR OF CLOSE MSG
         LA    MSGLEN,CLOSEMGL     LENGTH OF MSG
         BAL   LNKREG,WRITETS      GO WRITE ENDING MSG
         BAL   LNKREG,RETCODE      CHECK RETURN CODE
         BAL   LNKREG,WAITD        WAIT FOR COMPLETION
         LA    FMTREG,HEXFF        GET DEACTIVATED TERMINAL ID
         STC   FMTREG,ZERO(DSPTABRG,RLNREG)  STORE IN DISPLAY TABLE
         B     READ                GO READ ANOTHER DISPLAY
CLEARINT EQU   *
         LA    MSGADDR,FORMAT2     ADDR OF FORMAT 2 MSG
         LA    MSGLEN,FMT2SZ       LENGTH OF MSG
         BAL   LNKREG,WRITETS      GO WRITE FORMAT 2
         BAL   LNKREG,RETCODE      CHECK RETURN CODE
         BAL   LNKREG,WAITD        WAIT FOR COMPLETION
         LA    FMTREG,EIGHT        GET FORMAT 2 ID
         STC   FMTREG,ZERO(DSPTABRG,RLNREG)  STORE IN DISPLAY TABLE
         B     READ                GO READ ANOTHER DISPLAY
FMT2     EQU   *
         CLI   INAREA,ENTER        ENTER KEY INTERRUPT
         BE    FMTO1               YES, GO WRITE FORMAT 1
         CLI   INAREA,PA1          PA1 KEY INTERRUPT
         BE    PA1INT              YES, GO MAKE HARD COPY
         CLI   INAREA,PA2          PA2 OR CNCL KEY INTERRUPT
         BE    PA2INT1             YES, GO DEACTIVATE TERMINAL
         CLI   INAREA,CLEAR        CLEAR KEY INTERRUPT
         BE    CLEARINT            GO WRITE FORMAT 2
         B     READ                GO READ ANOTHER DISPLAY
NOPRINT  EQU   *
         LA    MSGADDR,NOPTR       ADDR OF NO PRINTER MSG
         LA    MSGLEN,NOPTRL       LENGTH OF MSG
         BAL   LNKREG,WRITETI      GO WRITE MSG
         BAL   LNKREG,RETCODE      CHECK RETURN CODE
         BAL   LNKREG,WAITD        WAIT FOR COMPLETION
         B     READ                GO READ ANOTHER DISPLAY
         EJECT
RETCODE  EQU   *
         B     RTNCDTAB(RTNCDRG)   BRANCH TO CORRESPONDING ENYRY
RTNCDTAB EQU   *
         B     RTNCD0              I/O SUCCESSFULLY INITIATED
         B     RTNCD4              DTFBT BUSY
         B     RTNCD8              INVALID RLN
         B     RTNCDC              INVALID TYPE CODE
         B     RTNCD10             ALL SKIP BITS ON
         B     RTNCD14             LINE ERROR AT OPEN
         B     RTNCD18             NO BUFFERS
         B     RTNCD1C             NO BUFFER POOL
         B     RTNCD20             NO BUFFER MANAGEMENT
         B     RTNCD24             BSC USAGE COUNT EXCEEDED
         B     RTNCD28             3270 LOCAL PRINTER BUSY
         B     RTNCD2C             3270 NO SBA ORDER IN READ FROM POS
         B     RTNCD30             DEVICE BUFFER UNRELIABLE
```

```
         B     RTNCD34              OLTEP HAS THE LOCAL 3270
RTNCD0   EQU   *
         BR    LNKREG               RETURN
RTNCD4   EQU   *
         S     LNKREG,EIGHT8        SUBTRACT 8 FROM RETURN ADDR
         BR    LNKREG                  TO RETRY THE OPERATION
RTNCD8   EQU   *
         STH   RLNREG,MAXRLN        SAVE MAXIMUM + 1 RLN
         SR    RLNREG,RLNREG        CLEAR RLN TO 0
         B     READ                 GO READ
RTNCDC   EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD10  EQU   *
         B     CLOSE                ALL TERMINALS, TERMINATE
RTNCD14  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD18  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD1C  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD20  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD24  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
RTNCD28  EQU   *
         S     LNKREG,EIGHT8        SUBTRACT 8 FROM RETURN ADDR
         BR    LNKREG                  TO RETRY THE OPERATION
RTNCD2C  EQU   *
         B     ABNORMAL             THIS CONDITION SHOULD NOT OCCUR
*                              SINCE NO READS FROM POSITIONS ARE USED
RTNCD30  EQU   *
         S     LNKREG,EIGHT8        SUBTRACT 8 FROM RETURN ADDR
         BR    LNKREG               TO RETRY THE OPERATION
RTNCD34  EQU   *
         B     READ
         EJECT
WAITD    EQU   *
         WAIT  ECB=DECBD
         SPACE 5
CHK7FCC  EQU   *
         CLI   DECBD,SEVENF         NORMAL COMPLETION CODE
         BNE   CHK64CC              NO CHECK OTHERS
         BR    LNKREG               RETURN
CHK64CC  EQU   *
         CLI   DECBD,SIXTY4         DEVICE BUFFER CLOBBERED FROM RFT
         BNE   CHK42CC              NO, CHECK SOME MORE
         IC    RLNREG,DECBD+THRTY9 GET RLN OF CLOBBERED DEVICE
         IC    FMTREG,C(RLNREG,DSPTABRG) GET FORMAT ID
         B     WTFMTX(FMTREG)
WTFMTX   EQU   *
         B     WTFMT0               RE-INITIALIZE FORMAT 0
         B     WTFMT1               RE-INITIALIZE FORMAT 1
         B     WTFMT2               RE-INITIALIZE FORMAT 2
         B     ABNORMAL             DUMP, SHOULD BE INACTIVE
WTFMT0   EQU   *
         LA    MSGADDR,FORMAT0      ADDR OF OUTPUT MESSAGE
         LA    MSGLEN,FMT0SZ        LENGTH OF MESSAGE
WTDVUNRL EQU   *
         WRITE DECBD,TS,DTFBTL,(MSGADDR),(MSGLEN),,(RLNREG),MF=E
         BNZ   ABNORMAL             DUMP, ABNORMAL RETURN CODE
         WAIT  ECB=DECBD
```

```
         B     CHK7ECC               CHECK COMPLETION CODE
WTFMT1   EQU   *
         LA    MSGADDR,FORMAT1       ADDR OF FORMAT 1 MSG
         LA    MSGLEN,FMT1SZ         LENGTH OF MSG
         B     WTDVUNBL              GO RESTORE SCREEN
WTFMT2   EQU   *
         LA    MSGADDR,FORMAT2       ADDR OF FORMAT 2 MSG
         LA    MSGLEN,FMT2SZ         LENGTH OF MSG
         B     WTDVUNBL              GO RESTORE SCREEN
CHK42CC  EQU   *
         CLI   DECBD,FOURTY2         RFT RECEIVED COMP CODE
         BNE   ABNORMAL
         TWAIT (REG2),TERMTST,ECBLIST=DECBADDR
         B     CHK7ECC               CHECK COMPLETION CODE
         SPACE 5
CLOSE    EQU   *
         BTRD  DTFBTI
         CLOSE DTFB1L
         EOJ
         SPACE 5
ABNORMAL EQU   *
         STM   REGZERO,REG15,REGSAVE      SAVE REGS
         SR    REG2,REG2                  GET BEGINNING ADDR
         L     WORKREG,PEND               GET ENDING ADDR
         PDUMP (REG2),(WORKREG)
         EOJ
H4096    DC    H'4096'
FORMAT0  EQU   *
         DC    X'C71DC811C150'       WCC, SF = PROT, SBA = 80
         DC    C'GOOD MORNING.'
         DC    X'1D6011C15F'         SF = PROT, SBA = 94
         DC    C'THIS BEGINS THE DEMONSTRATION '
         DC    C'OF THE DOS/BTAM '
         DC    X'1DC8'               SF = PROT
         DC    C'3270 '
         DC    X'1D60'               SF = PROT
         DC    C'LOCAL SAMPLE PROGRAM.'
         DC    X'1D6011C3F0'         SF = PROT, SBA = 240
         DC    C'ENTER THE FOLLOWING:'
         DC    X'1D6011C4D8'         SF = PROT, SBA = 280
         DC    C'NAME:'
         DC    X'1D4013'             SF = UNPROT, IC
         DC    X'11C5401D60'         SBA = 320,SF = PROT
         DC    C'SOC SEC NUM:'
         DC    X'1D40'               SF = UNPROT
         DC    X'11C5E81D60'         SBA = 360,SF = PROT
FMT0SZ   EQU   *-FORMAT0
         SPACE 5
FORMAT1  EQU   *
         DC    X'C71D60114040'       WCC, SF = PROT, SBA = 0
         DC    C'ENTER DATA REQUESTED BELOW:'
         DC    X'11C150'             SBA = 80
         DC    C'NAME:'
         DC    X'1D401311C1F81D60'   SF = UNPROT, IC, SBA = 120,
*                                     SF = PROT
         DC    C'ADDR:'
         DC    X'1D4011C2601D60'     SF = UNPROT, SBA = 160, SF = PROT
         DC    C'CITY:'
         DC    X'1D4011C3C81D60'     SF = UNPROT, SBA = 200, SF = PROT
         DC    C'STATE:'
         DC    X'1D4011C3E41D60'     SF = UNPROT, SBA = 228, SF = PROT
```

```
         DC     C'ZIP:'
         DC     X'1D5C11C3F01D6011C4D8'  SF = UNPROT, SBA = 240,
*                                        SF = PROT, SBA = 280
         DC     C'ENTER KEY: ENTER DATA;'
         DC     X'11C540'                SBA = 320
         DC     C'PA1 KEY: PRINT DATA;'
         DC     X'11C5E8'                SBA = 360
         DC     C'PA2 (CNCL) KEY: DEACTIVATE TERMINAL;'
         DC     X'11C650'                SBA = 400
         DC     C'CLEAR KEY: CONTROL OPTIONS;'
FMT1SZ   EQU    *-FORMAT1
         SPACE 5
FORMAT2  EQU    *
         DC     X'C711404013'            WCC, SBA = 0, IC
         DC     C'XXYY3CUU'
         DC     X'114CE8'                SBA = 40
         DC     C'TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESS'
         DC     C'AGE OVER SAMPLE FORMAT ABOVE:  XX=TEST NO. (23-28)'
         DC     C'  YY=REPEATS (01-99)  CUU=ADDRESS OF TARGET DEVICE'
         DC     X'11C3C8'                SBA = 200
         DC     C'THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KE'
         DC     C'Y TO RESUME AFTER TEST.'
FMT2SZ   EQU    *-FORMAT2
         SPACE 5
NOPTR    EQU    *
         DC     X'C61D4011C6F8'          WCC, SF = PROT, SBA = 440
         DC     C'NO PRINTER DEFINED FOR THIS PROGRAM'
NOPTRL   EQU    *-NOPTR
         SPACE 5
CLOSEMG  EQU    *
         DC     X'C711404D1DC8'          WCC, SBA = 0, SF = PROT
         DC     C'THIS TERMINAL IS NOW INACTIVE. WHEN ALL REMAINING '
         DC     C'TERMINALS BECOME INACTIVE, THE LOCAL 3270 SAMPLE '
         DC     C'PROGRAM WILL BE OVER.'
CLOSEMGL EQU    *-CLOSEMG
         EJECT
*        READ AND WRITE MACROS
         DS     0F
WRITETS  EQU    *
         WRITE DECBD,TS,DTFBTL,(MSGADDR),(MSGLEN),,(RLNREG),MF=E
         BR     LNKREG                   RETURN
         SPACE 5
READTI   EQU    *
         READ   DECBD,TI,DTFBTL,INAREA,300,,(RLNREG),MF=E
         BR     LNKREG                   RETURN
         EJECT
READTB   EQU    *
         READ   DECBD,TB,DTFBTL,INAREA,2000,,(RLNREG),MF=E
         BR     LNKREG                   RETURN
         SPACE 5
WRITETI  EQU    *
         WRITE DECBD,TI,DTFBTL,(MSGADDR),(MSGLEN),,(RLNREG),MF=E
         BR     LNKREG
         EJECT
WRITETUS EQU    *
         WRITE DECBD,TUS,DTFBTL,FLAGS,1,,(RLNREG),MF=E
         BR     LNKREG
         SPACE 5
*        CONSTANTS
REGSAVE  DC     16F'0'                   REGISTER SAVE AREA
CURRPTR  DC     F'0'                     POINTER TO CURRENT PTR RLN
```

```
TWO2     DC    F'2'                    CONSTANT OF TWO
EIGHT8   DC    F'8'                    CONSTANT OF 8
TWOTHO   DC    F'2000'
INAREA   DC    500F'0'                 INPUT AREA
DECBADDR DC    X'80'
         DC    AL3(DECBD)
VDTF     DC    V(DTFBTL)
         DS    0F
PEND     DC    X'00CC9FFF'
DSPTAB   DC    XL32'0'                 DISPLAY TABLE
RLNREAD  DC    X'00'                   RLN OF DEVICE JUST READ
FLAGS    DC    X'00'                   BYTE OF FLAGS
PTRRLNTB DC    X'FFFFFFFFFFFFFFFF'     PRINTER RLN TABLE
         DC    X'FFFFFFFFFFFFFFFF'
         DC    X'FFFFFFFFFFFFFFFF'
         DC    X'FFFFFFFFFFFFFFFF'
MAXRLN   DC    H'0'                    MAXIMUM RLN SAVE
         EJECT
* THE FIRST OPERAND IN THE SDRTAB MACRO MUST BE CHANGED TO REFLECT
* THE NUMBER OF 3270'S IN YOUR SYSTEM. PRESENTLY THE 2 REFLECTS THE
* LINELST OPERAND ON THE DTFBT MACRO.
SDRTABLE SDRTAB 2,,
         SPACE 5
* THE LINELST OPERAND SHOULD BE MODIFIED TO REFLECT THE NUMBER OF
* 3270 DISPLAYS AND PRINTERS IN YOUR SYSTEM. THE PRESENT LINELST
* OPERAND MEANS THAT THIS SAMPLE PROGRAM WILL USE TWO 3270'S.
DTFBTL   DTFBT LINELST=(005,006),                                      X
               CU=3272,DEVICE=3277,TERMTST=YES,MODNAME=L3270MOD,       X
               LERBADR=SDRTABLE
         SPACE 5
         READ  DECBD,TI,DTFBTL,MF=L
         LTORG
         EJECT
L3270MOD BTMOD L3277=YES,TST3277=YES,OBRSDR=YES
         END
```

```
SAMP327R CSECT
         SPACE 5
*        THIS IS A SAMPLE PROGRAM FOR A REMOTE 3270 INFORMATION DISPLAY
*  SYSTEM.  PLEASE NOTE THAT THIS PROGRAM HAS BEEN WRITTEN FOR TWO
*  3270 DEVICES, BOTH OF WHICH MUST BE 3277 DISPLAYS.
*
*        TO ASSEMBLE THIS SAMPLE PROBLEM, FIRST REMOVE ALL CATALS AND
*  BKEND CARDS, AND THEN PUNCH AND INSERT THE NECESSARY JOB CONTROL
*  CARDS FOR AN ASSEMBLY.
*
*        NOW THE OUTPUT FROM THE ASSEMBLER (THE OBJECT DECK) MUST BE
*  LINKAGE-EDITED TO THE CORE IMAGE LIBRARY IN THE BG, F1, OR F2
*  PARTITION.
*
*        THE FOLLOWING I/O ASSIGNMENT  MUST BE MADE TO ASSOCIATE THE
*  3270 DEVICES WITH THE PROGRAM:
*                              // ASSGN SYS005,X'CUU'
*
*        TO EXECUTE THIS SAMPLE PROGRAM, ENTER FROM THE OPERATORS
*  CONSOLE OR THROUGH THE CARD READER THE FOLLOWING:
*                             // EXEC SAMP327R
*
*        IT MAY BE NECESSARY TO ALTER THIS SAMPLE PROBLEM SO THAT IT
*  MAY FUNCTION WITH MORE THAN TWO DEVICES.  TO DO SO, THE FOLLOWING
*  CARDS MUST BE CHANGED IN THE SOURCE DECK:
*                             1. DFTRMLST MACROS   (SELDSPLY)
*                                                  (SPECPOL)
*
*        THIS SAMPLE PROGRAM IS RESTRICTED TO A MAXIMIM OF 32 DEVICES,
*  ALL OF WHICH MUST BE ATTACHED TO ONE REMOTE 3271 CONTROL UNIT.
         EJECT
*        REGISTER EQUATES
REGZERO  EQU   0                   REGISTER 0
REG2     EQU   2                   WORK REGISTER
WORKREG  EQU   3                   WORK REGISTER
MSGADDR  EQU   4                   ADDR OF OUTPUT MESSAGE
MSGLEN   EQU   5                   LENGTH OF OUTPUT MESSAGE
SELREG   EQU   6              SPEC POLL ADDR OF 3270 DISPLAY
FMTREG   EQU   8                   FORMAT IDENTIFIER
DSPTABRG EQU   9                   ADDRESS OF DISPLAY TABLE
LNKREG   EQU   10                  LINKAGE REGISTER
BASEREG  EQU   11                  FIRST BASE REG
BASEREG2 EQU   12                  SENOND BASE REG
PTRTAB   EQU   13                  PRINTER RLN TABLE ADDRESS
RTNCDRG  EQU   15                  RETURN CODE REGISTER
REG15    EQU   15                  REGISTER 15
         SPACE 5
*        EQUATES
ZERO     EQU   0                   LENGTH OF 0
ONE      EQU   1                   LENGTH OF 1
TWO      EQU   2                   LENGTH OF 2
THREE    EQU   3                   LENGTH OF 3
FOUR     EQU   4                   FORMAT 1 IDENTIFIER
FIVE     EQU   5              LENGTH OF TERMINAL LIST ENTRIES
SIX      EQU   6                   LENGTH OF 6
TP06     EQU   6                   TP CODE OF 6
SEVEN    EQU   7                   LENGTH OF 7
EIGHT    EQU   8                   FORMAT 2 IDENTIFIER
```

```
TP11     EQU   X'11'                  TP CODE OF 11
SIXTN    EQU   16                     LENGTH IF 16
TP20     EQU   X'20'
TWENTY4  EQU   24                LENGTH OF 24
TWENTY8  EQU   28                     LENGTH OF 28
FIFTY4   EQU   54                     NON OPERATION COMP CODE
TWO55    EQU   255               LENGTH OF 255
TIMEOUT  EQU   X'01'                  DECB TIME OUT FLAG
RVIRSP   EQU   X'02'                  RVI RESPONSE TO ADDR FLAG
SSMSG    EQU   X'10'                  SENSE/STATUS RECEIVED FLAG
EOTRSPTX EQU   X'40'                  EOT RESPONSE TO TEXT
EOTRCVD  EQU   X'40'                  EOT RECEIVED FLAG
FOURTY1  EQU   X'41'                  I/O ERROR COMP CODE
FOURTY2  EQU   X'42'                  RFT RECEIVED COMP CODE
SIXTY1   EQU   X'61'                  RVI RESPONSE TO ADDR COMP CODE
PA1      EQU   X'6C'                  ATTENTION ID FOR PA1 KEY
CLEAR    EQU   X'6D'                  ATTENTION ID FOR CLEAR KEY
PA2      EQU   X'6E'                  ATTENTION ID FOR PA2 (CNCL) KEY
ENTER    EQU   X'7D'                  ATTENTION ID FOR ENTER KEY
SEVENF   EQU   X'7F'                  NORMAL COMPLETION CODE
LAST     EQU   X'8C'             SIGNIFIES END OF POLLING LIST
TWO53    EQU   X'FD'
         EJECT
         BALR  BASEREG,0              ESTABLISH
         USING *,BASEREG,BASEREG2               ADDRESSABILITY
         LR    BASEREG2,BASEREG       INITIALIZE
         AH    BASEREG2,H4096                     SECOND BASE
*        OPEN THE LINE GROUP
         OPEN  DTFBTR
         LA    DSPTABRG,DSPTAB        ADDRESS THE DISPLAY TABLE
         LA    SELREG,SELDSPLY        GET SELECTION ADDRESS OF
*                                               FIRST 3270 DISPLAY
         SPACE 5
INITIAL   EQU  *
         LA    MSGADDR,FORMAT0        ADDR OF FORMAT0 MESSAGE
         LA    MSGLEN,FMT0SZ          LENGTH OF MESSAGE
         BAL   LNKREG,WRITETI         GO WRITE FORMAT 0
         BAL   LNKREG,RETCODE         CHECK RETURN CODE
         BAL   LNKREG,WAITD           WAIT FOR COMPLETION
         LA    SELREG,FIVE(SELREG)    ADDR NEXT SPECIFIC POLL ENTRY
         TM    ZERO(SELREG),LAST      END OF SELECTION LIST
         BO    READ                   YES, GO ISSUE A READ
         LA    SELREG,ONE(SELREG)     ADDR OF NEXT ENTRY
         B     INITIAL                NO, WRITE TO REMAINING DISPLAYS
         SPACE 5
READ     EQU   *
         LA    MSGADDR,INAREA         ADDR OF INPUT AREA
         XC    INAREA(255),INAREA     CLEAR INPUT
         XC    INAREA(43),INAREA                    AREA
         BAL   LNKREG,READTI          GO READ A DISPLAY
         BAL   LNKREG,RETCODE         CHECK RETURN CODE
         BAL   LNKREG,WAITD           WAIT FOR COMPLETION
         MVC   CUDVSAVE(TWO),INAREA+TWO SAVE CU,DV
         TM    DECBI+EIGHT,SSMSG      SENSE/STATUS MESSAGE RECEIVED
         BO    SSCHECK
         LA    SELREG,SPECPOL         ADDR OD SPEC POLL TABLE
         SR    REG2,REG2              CLEAR REGISTER 2
CHKIT    EQU   *
         CLC    ONE(TWO,SELREG),INAREA+TWO CHECK FOR CU,DV
         BE    FNDSEL                 YES, GET SELECTION ADDR
         LA    REG2,ONE(REG2)         ADD ONE TO INDEX
```

```
         LA     SELREG,SIX(SELREG)  POINT TO NEXT ENTRY
         TM     ZERO(SELREG),LAST   END OF LIST
         BO     ABNORMAL
         B      CHKIT               NO KEEP CHECKING
FNDSEL   EQU    *
         STC    REG2,INDEX          SAVE INDEX BYTE
         LR     WORKREG,REG2        GET INDEX INTO ODD REGISTER
         M      REG2,SIXL           MULTIPLY INDEX BY 6
         LR     REG2,WORKREG        RE-ESTABLISH INDEX REG
         LA     SELREG,SELDSPLY     GET SELECTION ADDRES
         AR     SELREG,REG2         ADDR OUTPUT ENTRY IN TABLE
         ST     SELREG,SELSAVE      SAVE SELECTION ADDR
         SR     FMTREG,FMTREG       CLEAR FORMAT REG
         LA     DSPTABRG,DSPTAB     ADDR OF DISPLAY TABLE
         IC      FMTREG,ZERO(DSPTABRG,REG2) GET FORMAT ID
         B      FORMATER(FMTREG)
         SPACE 5
FORMATER EQU    *
         B      FMT0                FORMAT 0 ON SCREEN
         B      FMT1                FORMAT 1 ON SCREEN
         B      FMT2                FORMAT 2 ON SCREEN
         SPACE 5
FMT0     EQU    *
*   VERIFY THE NAME AND SOCIAL SECURITY NUMBER. ASSUMING THAT THEY
* ARE VALID, WE SHALL CONTINUE PROCESSING.
FMT01    EQU    *
         LA     FMTREG,FOUR         GET FORMAT 1 ID
         STC     FMTREG,ZERO(DSPTABRG,REG2) STORE IN DISPLAY TABLE
         LA     MSGADDR,FORMAT1     ADDR OF FORMAT1 MESSAGE
         LA     MSGLEN,FMT1SZ       LENGTH OF MESSAGE
         BAL     LNKREG,WRITETI     GO WRITE FORMAT 1
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
         B      READ                GO READ ANOTHER DISPLAY
         SPACE 5
FMT1     EQU    *
         CLI    INAREA+FOUR,ENTER   ENTER KEY INTERRUPT
         BE     ENTERINT            YES, GO UPDATE RECORDS
         CLI    INAREA+FOUR,PA1     PA1 KEY INTERRUPT
         BE     PA1INT              YES, GO MAKE HARD COPY
         CLI    INAREA+FOUR,PA2     PA2 OR CNCL KEY INTERRUPT
         BE     PA2INT              YES, GO DEACTIVATE TERMINAL
         CLI    INAREA+FOUR,CLEAR   CLEAR KEY INTERRUPT
         BE     CLEARINT            YES, GO WRITE FORMAT 2
         B      READ                IGNORE THE INTERRUPT AND GO READ
         SPACE 5
ENTERINT EQU    *
*   CREATE A NEW OR UPDATE AN EXISTING ENTRY IN YOUR PERMANENT
* DATA SET.
ENTERIN1 EQU    *
         LA     MSGADDR,ERALUNP     ADDR OF MESSAGE
         LA     MSGLEN,ERALUNPL     LENGTH OF MESSAGE
         BAL    LNKREG,WRITETI      GO ERASE ALL UNPROTECTED DATA
         BAL    LNKREG,RETCODE      CHECK RETURN CODE
         BAL    LNKREG,WAITD        WAIT FOR COMPLETION
         B      READ                GO READ ANOTHER DISPLAY
PA1INT   EQU    *
         B      NOPRINT             NO PRINTER DEFINED
PA2INT   EQU    *
*   DETERMINE IF ANY DATA WAS ENTERED. IF SO, CREATE A NEW OR UPDATE
* AN EXISTING ENTRY IN YOUR PERMANENT DATA SET. NOW DEACTIVATE THE
```

```
* TERMINAL.
PA2INT1  EQU    *
         LA     MSGADDR,CLOSEMG       ADDR OF CLOSE MSG
         LA     MSGLEN,CLOSEMGL       LENGTH OF MSG
         LA     SELREG,SELDSPLY       ADDR OF SELECTION TABLE
ONCEMORE EQU    *
         BAL    LNKREG,WRITETI        GO WRITE ENDING MSG
         BAL    LNKREG,RETCODE        CHECK RETURN CODE
         BAL    LNKREG,WAITD          WAIT FOR COMPLETION
         LA     SELREG,FIVE(SELREG)   POINT TO INDICATOR BYTE
         TM     ZERO(SELREG),LAST     END OF SELECTION LIST
         BO     CLOSE                 YES, TERMINATE PROGRAM
         LA     SELREG,ONE(SELREG)    POINT TO NEXT ADDRESS
         B      ONCEMORE              NO, WRITE ANOTHER MESSAGE
CLEARINT EQU    *
         LA     MSGADDR,FORMAT2       ADDR OF FORMAT 2 MSG
         LA     MSGLEN,FMT2SZ         LENGTH OF MSG
         BAL    LNKREG,WRITETI        GO WRITE FORMAT 2
         BAL    LNKREG,RETCODE        CHECK RETURN CODE
         BAL    LNKREG,WAITD          WAIT FOR COMPLETION
         LA     FMTREG,EIGHT          GET FORMAT 2 ID
         STC    FMTREG,ZERO(DSPTABRG,REG2) STORE IN DISPLAY TABLE
         B      READ                  GO READ ANOTHER DISPLAY
FMT2     EQU    *
         CLI    INAREA+FOUR,ENTER     ENTER KEY INTERRUPT
         BE     FMT01                 YES, GO WRITE FORMAT 1
         CLI    INAREA+FOUR,PA1       PA1 KEY INTERRUPT
         BE     PA1INT                YES, GO MAKE HARD COPY
         CLI    INAREA+FOUR,PA2       PA2 OR CNCL KEY INTERRUPT
         BE     PA2INT1               YES, GO DEACTIVATE TERMINAL
         CLI    INAREA+FOUR,CLEAR     CLEAR KEY INTERRUPT
         BE     CLEARINT              GO WRITE FORMAT 2
         B      READ                  GO READ ANOTHER DISPLAY
NOPRINT  EQU    *
         LA     MSGADDR,NOPTR         ADDR OF NO PRINTER MSG
         LA     MSGLEN,NOPTRL         LENGTH OF MSG
         BAL    LNKREG,WRITETI        GO WRITE MSG
         BAL    LNKREG,RETCODE        CHECK RETURN CODE
         BAL    LNKREG,WAITD          WAIT FOR COMPLETION
         B      READ                  GO READ ANOTHER DISPLAY
         EJECT
RETCODE  EQU    *
         B      RTNCDTAB(RTNCDRG)     BRANCH TO CORRESPONDING ENYRY
RTNCDTAB EQU    *
         B      RTNCD0                I/O SUCCESSFULLY INITIATED
         B      RTNCD4                DTFBT BUSY
         B      RTNCD8                INVALID RLN
         B      RTNCDC                INVALID TYPE CODE
         B      RTNCD10               ALL SKIP BITS ON
         B      RTNCD14               LINE ERROR AT OPEN
         B      RTNCD18               NO BUFFERS
         B      RTNCD1C               NO BUFFER POOL
         B      RTNCD20               NO BUFFER MANAGEMENT
         B      RTNCD24               BSC USAGE COUNT EXCEEDED
RTNCD0   EQU    *
         BR     LNKREG                RETURN
RTNCD4   EQU    *
         S      LNKREG,EIGHT8         SUBTRACT 8 FROM RETURN ADDR
         BR     LNKREG                   TO RETRY THE OPERATION
RTNCD8   EQU    *
RTNCDC   EQU    *
```

```
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
RTNCD10  EQU    *
         B      CLOSE                 ALL TERMINALS, TERMINATE
RTNCD14  EQU    *
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
RTNCD18  EQU    *
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
RTNCD1C  EQU    *
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
RTNCD20  EQU    *
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
RTNCD24  EQU    *
         B      ABNORMAL              THIS CONDITION SHOULD NOT OCCUR
         EJECT
WAITD    EQU    *
         WAIT   ECB=DECBD
CHK7FCC  EQU    *
         CLI    DECBD,SEVENF          NORMAL COMPLETION CODE
         BNE    CHK42CC               NO, CHECK OTHERS
         BR     LNKREG                RETURN
CHK42CC  EQU    *
         CLI    DECBD,FOURTY2         RFT RECEIVED COMP CODE
         BNE    CHK54CC               NO, CHECK OTHERS
         TWAIT  (REG2),TERMTST,ECBLIST=DECBADDR
         B      CHK7FCC               CHECK COMPLETION CODE
CHK54CC  EQU    *
         CLI    DECBD,FIFTY4          NON OPERATION COMP CODE
         BNE    CHK41CC               NO, KEEP CHECKING
         B      READ                  GO ISSUE READ
CHK41CC  EQU    *
         CLI    DECBD,FOURTY1         I/O ERROR COMP CODE
         BNE    CHK61CC               NO, KEEP CHECKING
         CLI    DECBD+TWENTY8,TP06    TP CODE OF 6
         BNE    CKTP20                NO, CHECK OTHER TP CODES
         TM     DECBD+SIXTN,TIMEOUT   DID DEVICE TIME OUT
         BNO    ABNORMAL              NO, TERMINATE
         S      LNKREG,TWELVE         SUBTRACR 12 FROM RETURN ADDR
         BR     LNKREG                TO RETRY THE OPERATION
CKTP20   EQU    *
         CLI    DECBD+TWENTY8,TP20    TP CODE OF 20
         BNE    CKTP11                NO, CHECK OTHERS
CHKEOT   EQU    *
         TM     DECBD+TWENTY4,EOTRSPTX EOT RESPONSE TO TEXT
         BNO    ABNORMAL              NO, TERMINATE
         B      READ                  YES, GO GET SENSE STATUS MESSAGE
CKTP11   EQU    *
         CLI    DECBD+TWENTY8,TP11    TP CODE OF 11
         BNE    ABNORMAL              NO, TERMINATE
         B      CHKEOT                YES, CHECK EOT RESPONSE TO TEXT
CHK61CC  EQU    *
         CLI    DECBD,SIXTY1          RVI RESPONSE TO ADDRESSING
         BNE    ABNORMAL              NO, TERMINATE
         CLI    DECBD+TWENTY8,TP06    TP CODE OF 6
         BNE    CHKTP20               NO, CHECK TP CODE OF 20
         TM     DECBD+EIGHT,RVIRSP    RVI RESPONSE FLAG ON
         BNO    ABNORMAL              NO, TERMINATE
         B      READ                  YES, GO GET S/S MESSAGE
CHKTP20  EQU    *
         CLI    DECBD+TWENTY8,TP20    TP CODE OF 20
         BNE    ABNORMAL              NO, TERMINATE
         WRITE  DECBD,TR,DTFBTR,INAREA,,,0,MF=E
```

```
         B     CHK7ECC             CHECK COMPLETION CODE
         EJECT
CLOSE    EQU   *
         BTRD  DTFBTR
         CLOSE DTFBTR
         EOJ
         SPACE 5
SSCHECK  EQU   *
*     INVESTIGATE THE SENSE/STATUS BYTES SENT BY THE REMOTE DEVICE.
*  IF RECOVERY IS POSSIBLE, ATTEMPT TO DO SO.  WE SHALL ASSUME THAT
*  THE ERROR IS UNRECOVERABLE AND TERMINATE.
         SPACE 5
ABNORMAL EQU   *
         STM   REGZERO,REG15,REGSAVE    SAVE REGS
         SR    REG2,REG2                GET BEGINNING ADDR
         L     WORKREG,PEND             GET ENDING ADDR
         PDUMP (REG2),(WORKREG)
         EOJ
         EJECT
FORMAT0  EQU   *
         DC    X'0227F5'           STX,ESC,E/W
         DC    X'C71DC811C150'     WCC, SF = PROT, SBA = 80
         DC    C'GOOD MORNING.'
         DC    X'1D6011C15F'       SF = PROT, SBA = 94
         DC    C'THIS BEGINS THE DEMONSTRATION '
         DC    C'OF THE DOS/BTAM '
         DC    X'1DC8'             SF = PROT
         DC    C'3270 '
         DC    X'1D60'             SF = PROT
         DC    C'REMOTE SAMPLE PROGRAM.'
         DC    X'1D6011C3F0'       SF = PROT, SBA = 240
         DC    C'ENTER THE FOLLOWING:'
         DC    X'1D6011C4D8'       SF = PROT, SBA = 280
         DC    C'NAME:'
         DC    X'1D4C13'           SF = UNPROT, IC
         DC    X'11C54C1D6C'       SBA = 320,SF = PROT
         DC    C'SOC SEC NUM:'
         DC    X'1D4C'             SF = UNPROT
         DC    X'11C5E81D60'       SBA = 360,SF = PROT
         DC    X'C3'               ETX
FMT0SZ   EQU   *-FORMAT0
         SPACE 5
FORMAT1  EQU   *
         DC    X'0227F5'           STX,ESC,E/W
         DC    X'C71D6C114040'     WCC, SF = PROT, SBA = 0
         DC    C'ENTER DATA REQUESTED BELOW:'
         DC    X'11C150'           SBA = 80
         DC    C'NAME:'
         DC    X'1D401311C1F81D60' SF = UNPROT, IC, SBA = 120,
*                                   SF = PROT
         DC    C'ADDR:'
         DC    X'1D4011C2601D60'   SF = UNPROT, SBA = 160, SF = PROT
         DC    C'CITY:'
         DC    X'1D4011C3C81D60'   SF = UNPROT, SBA = 200, SF = PROT
         DC    C'STATE:'
         DC    X'1D4011C3E41D60'   SF = UNPROT, SBA = 228, SF = PROT
         DC    C'ZIP:'
         DC    X'1D5011C3F01D6011C4D8'  SF = UNPROT, SBA = 240,
*                                  SF = PROT, SBA = 280
         DC    C'ENTER KEY: ENTER DATA;'
         DC    X'11C540'           SBA = 320
```

```
         DC     C'PA1 KEY: PRINT DATA;'
         DC     X'11C5E8'            SBA = 360
         DC     C'PA2 (CNCL) KEY: DEACTIVATE TERMINAL;'
         DC     X'11C65C'            SBA = 400
         DC     C'CLEAR KEY: CONTROL OPTIONS;'
         DC     X'C3'                ETX
FMT1SZ   EQU    *-FORMAT1
         EJECT
FORMAT2  EQU    *
         DC     X'0227F5'            STX,ESC,E/W
         DC     X'C711404013'        WCC, SBA = 0, IC
         DC     C'XXYY4CCDD'
         DC     X'1140E8'            SBA = 40
         DC     C'TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESS'
         DC     C'AGE OVER SAMPLE FORMAT ABOVE:  XX=TEST NO. (23-28)'
         DC     C'  YY=REPEATS (01-99) CCDD=ADDRESS OF TARGET DEVICE'
         DC     C'THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KE'
         DC     C'Y TO RESUME AFTER TEST.'
         DC     X'11C3C8'            SBA = 200
         DC     X'03'                ETX
FMT2SZ   EQU    *-FORMAT2
         SPACE 5
NOPTR    EQU    *
         DC     X'0227F1'            STX,ESC,WRITE
         DC     X'C61DC811C6F8'      WCC, SF = PROT, SBA = 440
         DC     C'NO PRINTER DEFINED FOR THIS PROGRAM'
         DC     X'03'                ETX
NOPTRL   EQU    *-NOPTR
         SPACE 5
CLOSEMG  EQU    *
         DC     X'0227F5'            STX,ESC,E/W
         DC     X'C71140401DC8'      WCC, SBA = 0, SF = PROT
         DC     C'      THE REMOTE 3270 SAMPLE PROGRAM HAS CONCLUDED.'
         DC     X'C3'                ETX
CLOSEMGL EQU    *-CLOSEMG
         SPACE 5
         SPACE 5
READBUF  EQU    *
         DC     X'0227F203'          STX,ESC,RD BUF,ETX
READBUFL EQU    *-READBUF
         SPACE 5
ERALUNP  EQU    *
         DC     X'02276F03'          STX,ESC,EAU,ETX
ERALUNPL EQU    *-ERALUNP
         EJECT
*        READ AND WRITE MACROS
         DS     0F
WRITETI  EQU    *
         WRITE DECBD,TI,DTFBTR,(MSGADDR),(MSGLEN),(SELREG),0,MF=E
         BR     LNKREG
         SPACE 5
READTI   EQU    *
         READ   DECBD,TI,DTFBTR,INAREA,256,POLDSPLY,0,MF=E
         BR     LNKREG
         EJECT
READTRV  EQU    *
         READ   DECBD,TRV,DTFBTR,INAREA,256,,0,MF=E
         BR     LNKREG               RETURN
         SPACE 5
WRITETIV EQU    *
         WRITE DECBD,TIV,DTFBTR,((MSGADDR),INAREA),((MSGLEN),256),(SELRX
```

```
                EG),0,MF=E
         BR     LNKREG                RETURN
         EJECT
READTT   EQU    *
         READ   DECBD,TT,DTFBTR,(MSGADDR),256,,0,MF=E
         BR     LNKREG                RETURN
         EJECT
*        DISPLAY SELECTION ADDRESSES
*   THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS;
*                           1. 0TH CU, 0TH DEV (60604040)
*                           2. 0TH CU, 1ST DEV (6060C1C1)
SELDSPLY DFTRMLST OPENLST,(606040402D,6060C1C12D)
         SPACE 5
*    DISPLAY SPECIFIC POLLING ADDRESSES
*   THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS:
*                           1. 0TH CU, 0TH DEV (40404040)
*                           2. 0TH CU, 1ST DEV (4040C1C1)
SPECPOL  DFTRMLST OPENLST,(404040402D,4040C1C12D)
         EJECT
*        DISPLAY GENERAL POLLING ADDRESS
POLDSPLY DFTRMLST AUTOWLST,3732,40407F7F2D
         SPACE 5
         SPACE 5
*        CONSTANTS
H4096    DC     H'4096'
EIGHT8   DC     F'8'                  CONSTANT OF 8
DSPTAB   DC     XL32'0'               DISPLAY TABLE
TWELVE   DC     F'12'                 LENGTH OF 12
INDEX    DC     X'00'                 INDEX BYTE SAVE AREA
CUDVSAVE DC     X'0000'               CU,DV SAVEAREA
SELSAVE  DC     F'0'
NOPENTR  DC     X'0000000000'
SAVER    DC     D'0'
SIXL     DC     F'6'
INAREA   DC     500F'0'               INPUT AREA
INAREAL  EQU    *-INAREA
REGSAVE  DC     16F'0'                REGISTER SAVE AREA
         DS     0F
PEND     DC     X'0000FFF'
DECBADDR DC     X'80'
         DC     AL3(DECBD)
SDRTABLE SDRTAB 1,
DTFBTR   DTFBT LINELST=(005),SWITCH=NO,CU=2703,DEVICE=BSC3,CONFIG=MPT, X
                MODELST=(0),CTLCHAR=EBCDIC,MODNAME=R3270MOD,TERMTST=YES
         READ   DECBD,TT,DTFBTR,MF=L
         LTORG
         EJECT
R3270MOD BTMOD BSCS=YES,BSCMPT=YES,BSCTEST=YES,DECBEXT=YES,OBRSDR=YES
         END
```

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
AS	ID=								x				
	ASGROUP	x											
	DEGROUP=												
	tgroupname	x											
	deunumber								x				
ASCTR	ID=								x				
	HIGHCTR=								x				
	ROUTE=												x
	LOG												As shown
	ASLOG												As shown
	EXTALRM												As shown
	NEXTAS=								x				
ASLIST	device												x
	NORM=								x				
	LENGTH=												
	data length								x				
	gdlight2								x				
	DIGIT=												
	entrypos								x				
	compvalue								x				
	gdlight3								x				
	ENTRY=												x
	MSG=									x			
	INQDIST=												x
	MODULUS=												
	entrypos								x				
	data length								x				
	gdlight4								x				
	SELTRAN=												x
CONFIGUR	CORE=												x
	PC=												x

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
CONFIGUR (Cont'd)	GDU=												x
	FUNCERR=								x				
	ENDERR=								x				
	MONERR=								x				
	GETID=								x				
	STORID=								x				
	IDCOUNT=								x				
	INQDISP=												x
CTRGROUP	ctrno								x				
	sro								x				
	cttest								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								x				
	compvalue								x				
	MSG=									x			
	MODULUS=												
	entrypos								x				
	data length								x				
	DIGIT2=												
	entrypos								x				
	compvalue								x				
DISPGUID	DISPMSG=									x			
	SUPPRES=												x

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
GDUAS	ID=								x				
	GDUNUMB=								x				
GDULIST	PARAMNO=								x				
	NORGUID=								x				
	DISPMSG=	x											
	IDENT=								x				
	MSG=									x			
	ENTRY=												x
GDUTRANS	TRCODE=								x				
	TRLIST=	x											
PARAMNUM	PLN=								x				
	PARMLST=	x											
PARMLIST	CKLNGTH=												
	data length								x				
	gdlight								x				
	CKMONKY=												x
	CKMOD11=												
	data length								x				
	entrypos								x				
	gdlight								x				
	CKRANGE=												
	firstpos								x				
	lastpos								x				
	compvalue								x				
	LOWGUID=								x				
	HIGUID=								x				
	RNGETST=												x
	CKMOD10=												
	data length								x				
	entrypos								x				
	gdlight								x				
	CKOR=												
	data pos								x				
	checkchar											x	
	ORGUID=								x				
	CKAND=												
	startpos								x				
	endpos								x				
	checkchar											x	

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(71)	(0)							
PARMLIST (Cont'd)	ANDGUID=								x				
	CKNONUM=												
	startpos								x				
	endpos								x				
	gdlight								x				
	CKNUM=												
	startpos								x				
	endpos								x				
	gdlight								x				
	TRANSL=												x
	IDENT=												x
STEND	no operands												
TGROUP	TCn=												
	tcode	x											
	E												As shown
TRANSLAT	TRANSCH=											x	
	TRANTXT=									x			
TRLIST	ROUTE=												x
	LOG=												As shown
	NULL=												As shown
	asaddr								x				
	TEXT=												x
	TRID=								x				
	INQDISP=								x				
	DEMOD10=												x
	DEMOD11=												x
	GDU=												x

*See macro description for allowable values.

```
 CONFIGUR CORE=32,PC=YES,INQDISP=YES,GDU=YES,GETID=F0,STORID=F5,        X
               IDCOUNT=8,MCNERR=(4,5),FUNCERR=(6,7),ENDERR=(8,9)
*                       THE CORE=32 OPERAND OF THIS CONFIGUR MACRO
*                       INDICATES THAT THE USER'S 2715 HAS 32K OF
*                       STORAGE AVAILABLE.BY CODING PC=YES,THE USER
*                       INDICATES THAT PULSE COUNTERS EXIST ON HIS
*                       2790 SYSTEM.INQDISP=YES INDICATES THAT
*                       INQUIRY DISPLAY WILL BE USED ON THE 2790
*                       SYSTEM, WHILE 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 INDICATE THAT THE
*                       USER IS DEFINING 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 DIFINING 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=18,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 THE
*                          USER HAS FOUR AREA STATIONS WITH 2798 GUIDANCE
*                          DISPLAY UNITS ATTACHED.THE GDUNUMB OPERAND
*                          INDICATES THE NUMBER OF 2798'S ON THAT
*                          PARTICULAR AREA STATION.THERE ARE A TOTAL OF 48
*                          2798 GDU'S ON THIS USER'S 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,OR59.
*                   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 STATIONS 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 CODE AS INDICATED BY THE
*                   FOLLOWING EXPAND TGROUP MACRO.TRANSACTION
*                   CODES 91,92,93,94,95,AND 96 EACH 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=RDRSTEPM,                    X
               TC8=(EXPAND1,E),TC9=(EXPAND2,E)
EXPAND1  TGROUP  TC1=RDSTIPM,TC2=RDSTEPM,TC3=RDSTEPB,TC4=RDSIDEPM,      X
               TC5=RDSIDGPM,TC6=RDSIDGPB,TC7=WRIPB,TC8=WREPM,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 THE 2715.NEXTAS=42
*                          INDICATES THE 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 CTRSCHED 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
*                          CTRSCHED 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.14 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 AND 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 PARMLIST 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  CKMCNKY=NO,TRANSL=YES
*                       THIS MACRO INDICATES THE 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  CKLNGTH=(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 THE FIELD 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 THAT 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 IS AN ! CHARACTER.IF NOT,
*                    OPERATIONAL GUIDANCE LIGHT 9 WILL BE TURNED
*                    ON AT THE 2798 TO INDICATE AN ERROR.ALSO,AN AND
*                    CHECK IS PERFORMED SUCH THAT 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,
*                    THEN 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,THEN 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,THEN
*                   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),HIGUID=9,LOWGUID=13
*                   THIS MACRO INDICATES THAT A RANGE CHECK WILL
*                   BE PERFORMED ON POSITIONS 2 THROUGH 17.THE
*                   CHARACTER IN POSITION 2 MUST BE GREATER THAN
*                  3 AND LESS THAN 7.THE DIGITS IN POSITIONS 3
*                   THROUGH 17 MUST BE 0
PAR25    PARMLIST  CKMONKY=YES,CKLNGTH=(8,11),CKNUM=(7,8,12),          *
               CKAND=(2,6,5B,00,00,00,4E),ANDGUID=10
PAR26    PARMLIST  CKMONKY=YES,CKLNGTH=(17,11),CKMOD11=(6,11,13),      *
               IDENT=YES,CKMOD10=(5,2,9),CKOR=(8,5C,60,F0,4E,40),     *
               ORGUID=10
PAR27    PARMLIST CKLNGTH=(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  CKLNGTH=(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  CKLNGTH=(6,11),CKRANGE=(2,6,10,09,55,00,00),        *
               HIGUID=9,LOWGUID=13
PAR32    PARMLIST
PAR33    PARMLIST  CKLNGTH=(7,11),CKNUM=(2,7,12),IDENT=YES
PAR34    PARMLIST  CKLNGTH=(5,11),CKOR=(2,4E,60),ORGUID=10,            *
               CKRANGE=(3,5,10,50,00),                                *
               HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR35    PARMLIST  CKLNGTH=(11,11),CKOR=(4,C4,E3),ORGUID=10,CKAND=(2,3,*
               E2,D3),ANDGUID=10,CKRANGE=(5,11,55,88,00,73,80,39,78), *
               HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR36    PARMLIST  CKLNGTH=(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.THE CHARACTER
*                   IN POSITION 2 MUST BE A DIGIT GREATER THAN OR
*                   EQUAL TO 1 AND LESS THAN OR EQUAL TO 5.THE
*                   CHARACTER IN POSITION 3 MUST BE A DIGIT GREATER
*                   THAN OR EQUAL TO 2 AND LESS THAN OR EQUAL TO 8.
*                   IF THE DIGIT IN POSITION 2 IS LESS THAN 1 OR
*                   OR THE DIGIT IN POSITION 3 IS LESS THAN 2,THEN
*                   OPERATIONAL GUIDANCE LIGHT 13 IS TURNED ON AT
*                   THE 2798.IF THE DIGIT IN POSITION 2 IS GREATER
*                   THAN 5 OR THE DIGIT IN POSITION 3 IS GREATER
*                   THAN 8,THEN OPERATIONAL GUIDANCE LIGHT 9 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  CKLNGTH=(11,11),CKNUM=(8,11,12),                    *
               CKAND=(2,7,F2,F3,F9,F5,F1,F1),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
*                            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
*                            TRANSACTION STEPS AS DEFINED BY THE FOLLOWING
*                            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   E,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,DEMOD10=YES,INQDISP=YES
         ASLIST   E,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 STSTION 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 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,DEMOD11=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,DEMOD10=YES,INQDISP=YES
         ASLIST  B,NORM=16,MODULUS=(2,9,4),INQDISP=7
EXP2     TRLIST  ROUTE=(CPU,LOG),TRID=8,DEMOD11=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
*                      9 AND WILL BE CHECKED WITH THE SELF-CHECK
*                      CHARACTER IN POSITION 10.IF THE MODULUS 11
*                      CHECK FAILS,GUIDANCE LIGHT 5 WILL BE TURNED ON.
EXP3     TRLIST  ROUTE=(CPU,LOG),TRID=9,DEMOD10=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=(LCG),TRID=11
         ASLIST  B,NORM=16
DEU1     TRLIST  ROUTE=LCG,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=(LCG),TRID=13,DEMOD10=YES
         DEULIST DIGIT2=(2,1),MODULUS=(3,10)
DEU3     TRLIST  ROUTE=(LCG),TRID=14,DEMOD11=YES
         DEULIST DIGIT2=(2,5),MODULUS=(3,10),LENGTH=13
DEU4     TRLIST  ROUTE=(LCG),TRID=15
         DEULIST DIGIT=(2,6),DIGIT2=(3,9) LENGTH=11
DEUEXP   TRLIST  ROUTE=(LCG,NULL),DEMOD10=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 ERROR BUTTON WILL BE
*                        INDICATED AT THE DEU.
DEU6     TRLIST  ROUTE=(LCG),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 ALONG 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 DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,          X
               CTROP=READ
RDIPSG   TRLIST ROUTE=LOG,TRID=21
         DEULIST
         CTRLIST DEVCOD=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 OF THE
*                        DEVICE ADDRESS OF THE DEU INITIATING THE
*                        REQUEST TO BE READ.THERE WILL BE NO CHANGE
*                        IN THE PRESENT COUNT TEST CONDITION OF THE
*                        COUNTER.
RDEPSGM  TRLIST  ROUTE=LCG,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 ASSRESS 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=LCG,TRID=23
         DEULIST
         CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,                 X
               CTROP=READ
RDEPGPM  TRLIST  ROUTE=LCG,TRID=24
         DEULIST
         CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,                  X
               CTROP=READ
RDEPGPB  TRLIST  ROUTE=LCG,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=LCG,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=LCG,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=(LCG,42),TRID=28
         DEULIST
         CTRLIST DEVCOD=M,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,                 X
               CTROP=READSET
RDSTEPM  TRLIST  ROUTE=(LCG,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=RIADSET
RDSIDEPM 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        X
               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        X
               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         X
               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    X
               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='  '
ALRMTX   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.WHENTHE 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 SECOND 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 TRANSACTTION 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
CARDCRD  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 2798S 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
QUADEQN  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
```

ACRONYMS

ACK	Positive Acknowledgment Character
ARU	Audio Response Unit
BCD	Binary Coded Decimal
BSC	Binary Synchronous Communications
BTAM	Basic Telecommunications Access Method
BTMOD	BTAM Module
CC	Chain Command
CCB	Command Control Block
CCW	Channel Command Word
CD	Chain Data
CDC	Call Directing Code
CE	Channel End
CR	Carriage Return
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube
DAU	Data Adapter Unit
DCV	Digitally Coded Voice
DE	Device End
DECB	Data Event Control Block
DOS	Disk Operating System
DTFBT	Define the File for BTAM
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECB	Event Control Block
EIB	Error Information Byte
EM	End-of-Medium
ENQ	Inquiry Character
EOA	End-of-Addressing Character
EOB	End-of-Block Character
EOM	End-of-Message Character
EOR	End-of-Record (2826)
EOT	End-of-Transmission
ERP	Error Recovery Procedure
ESC	Escape Character
ETB	End-of-Block
ETX	End-of-Text
HT	Horizontal Tab
LCB	Line Control Block
LERB	Line Error Block
LF	Line Feed
LRC	Logitudinal Redundancy Check
LUB	Logical Unit Block
Modem	Modulator/Demodulator Device
NAK	Negative Acknowledgment Character
OIU	Optical Image Unit
PCI	Program-Controlled Interrupt
PUB	Physical Unit Block
RFT	Request-for-Test
RLN	Relative Line Number
SDA-II	Synchronous Data Adapter - Type II
SOH	Start-of-Header
STX	Start-of-Text
TCU	Transmission Control Unit
TIC	Transfer in Channel
TP	Teleprocessing
TSC	Transmitter Start Code
TWX	Teletypewriter Exchange
US	Unit Separator
USASCII	USA Standard Code for Information Interchange
VCT	Voice Code Translator
VRC	Vertical Redundancy Check
WTTA	World Trade Telegraph Adapter

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, 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 signaling 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 stations, 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 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 line 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 marker 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 compatability between business machines and communications facilities.

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 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; or

2. 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 insuring 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 tramsmission 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 the 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 mulitpoint 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.

INDEX

Where more than one page reference is given, the first page number indicates the major reference.

absexp (absolute expression) 12
access method 1,363
accumulators, LERB 28
ACK 6,362
acknowledgement
negative 6
positive 6
acknowledgements, alternating 6
acronyms 362
actions, error 60-64
ACTIVATE operand 17-18
for IBM 3270 (local) 17
activation code, test 25
address pointer 41
addressing
characters 3,297
codes 297
double 282
for IBM 3270 (remote)
control unit addresses 321
device addresses 322
list 289,295
terminal 120
alternating acknowledgements
ACK0 6
ACK1 6
ANDGUID operand (PARMLIST macro) 207
answer terminal list
addressing 120,289,295
polling 289
answering 6
AREA operand 30,32,38
area station
address, IBM 2715 201
format 194
sequence table 194
table 194
AS macro instruction, IBM 2715
explanation 201
format 201
operands 201
ASCTR macro (IBM 2715) 203
ASGROUP operand 201
ASLIST macro instruction, IBM 2715
explanation 214
format 215
operands 215
ASLOG operand (ASCTR macro) 203
ASMTRTAB macro instruction (obsolete) 298
assembly considerations 45
assurance, message 6
attention interruptions 86
ATTLST operand 18
AUDIO operand 15
audio response unit (ARU) 98,15
standard ERP for 67
auto poll 24
start-stop 15
AUTOLST operand 17,21
AUTOWLST operand 17,21
auxarea operand 32,38
auxlength operand 32,39
awareness, messages 60

background 7
basic access method 363
Basic Telecommunications Access Method 363
Baud 363
Baudot code 363
binary synchronous communication (BSC) 363
channel programs (general BSC) 269
multipoint 282
point-to-point contention 269
point-to-point dial 273
channel programs (System/370) 162
contention 162
dial 162
code charts 297,311,313,320,323
control procedures
ACK0 6
ACK1 6
intermix 363
on-line terminal tests 69-80
summary 74-75
stations supported 161
with IBM 1130 182
with IBM 2770 231
with IBM 2780 173
with IBM 2790 189
with IBM 2972-8 188
with IBM 3270 243
with System 3 185
with System/360 Model 20 185
blank cards (transmission) 231
block 363
braces 12
brackets 12
BSCMPT operand 15
BSCS operand 15
BSCTEST operand 16
BTAM
cancellations 68
configurations 4-5,83,84,244
definition 363
devices supported, BSC 80,161
general functions 1-7
logic module 1
machine and device requirements 1
macro instruction summary 9
macro operands values 298
restarting channel programs 1
translation tables, defining 35
BTMOD macro instruction

explanation 13
format 13
operands 13,298
BTRD macro instruction
explanation 16
format 16
operands 16,298
return codes 16,57
RMSR tables 16
BUFCB operand 25
buffer
definition 363
description 52
for IBM 3270 (local) 92
for IBM 3270 (remote) 261
for READ operations 53
for WRITE operations 53
format 52
pool 52
size 25
buffer management
format 52
logic 52
macro instructions 33
pool 52
routines 14
updating technique 54
BUFFER operand 14
buffer pool 52
control block 52-53,25
definition 52,363
format 53
organization 52
reservation 25,53
buffering 52
buffers
READ 53
WRITE 53
BUFL operand 25
BUFNO operand 25

call directing code (CDC) 45,101
calling 3,363
calling function, READ Initial 168
calling terminal list
addressing 46,21
polling 46,21
CANCEL operand 15
cancelation, BTAM program 68
CATALR statement 45
central station 363
centralized communication system 3,363
chaining 363
channel 2
channel programs
BSC
CPU to CPU contention 162
CPU to CPU dial 162
IBM 1030 108-111
CPU to terminal 109
terminal to CPU 108
IBM 1050, nonswitched 111-115
CPU to terminal 112
line control characters 114
programming notes 113
terminal to CPU 111
IBM 1050, switched 115-121
CPU to terminal 117
programming notes 119
terminal to CPU 115
IBM 1060 121-124
CPU to terminal 122
terminal to CPU 121
IBM 1130 182-185
multipoint 182
point-to-point 185
IBM 2260-2848 local 81-83
CPU to terminal 82
terminal to CPU 82
IBM 2260-2848 remote 152-157
CPU to terminal 154
terminal to CPU 152
IBM 2265-2845 remote 152-157
CPU to terminal 154
terminal to CPU 152
IBM 2740 124-151
2740 basic 124
CPU to terminal 125
terminal to CPU 124
2740 w/checking 125-127
CPU to terminal 126
terminal to CPU 125
2740 w/checking and OIU 127-131
CPU to terminal 129
terminal to CPU 127
2740 w/dial 131-133
CPU to terminal 132
terminal to CPU 131
2740 w/dial and checking 133-135
CPU to terminal 134
terminal to CPU 133
2740 w/dial, checking and OIU 135-141
CPU to terminal 137
terminal to CPU 135
2740 w/dial and transmit control 141-143
CPU to terminal 142
terminal to CPU 141
2740 w/dial, transmit control and checking 143-146
CPU to terminal 145
terminal to CPU 143
2740 w/station control 146-148
CPU to terminal 148
terminal to CPU 146
2740 w/station control and checking 148-152
CPU to terminal 150
terminal to CPU 148
IBM 2770 241-242
multipoint 242
point-to-point 241
IBM 2780 177-181
contention 177
dial 178
multipoint 180
IBM 2790 189
contention 189
local 189
multipoint 189
switched 189
IBM 2972 188
multipoint 188
IBM 3270 (local) 87-91

IBM 3270 (remote) (see channel programs, general BSC, multipoint) 282
IBM 7770 99
CPU to terminal 99
programming information 98
terminal to CPU 99
IBM System 3 185
conversational mode 185
multipoint 185
point-to-point 185
IBM System/360 Model 20 185-186
multipoint 186
point-to-point 185
World Trade Telegraph 158-159
CPU to terminal 159
terminal to CPU 158
World Trade Telex 160
CPU to terminal 160
terminal to CPU 160
WU TWX 33/35 105-106
CPU to terminal 106
terminal to CPU 105
WU 115A 101-102
CPU to terminal 102
terminal to CPU 101
terminal to terminal 102
channel programs, general BSC 269-287
multipoint 282
point-to-point contention 269
point-to-point dial 273
character test 71
characters, polling 3
CHGNTRY macro instruction
answering list 18
explanation 16
for IBM 3270 (local) 88,17
for IBM 3270 (remote) 251,16
format 17
operands 16-17
return codes 17,57
special form 17
2260 local 17
circuit call 101
CKAND operand (PARMLIST macro) 207
CKLNGTH operand (PARMLIST macro) 207
CKMONKY operand (PARMLIST macro) 207
CKMOD10 operand (PARMLIST macro) 207
CKMOD11 operand (PARMLIST macro) 207
CKNONNUM operand (PARMLIST macro) 207
CKNUM operand (PARMLIST macro) 207
CKOR operand (PARMLIST macro) 207
CKRANGE operand (PARMLIST macro) 207
CLOSE macro instruction
explanation 18
format 18
operands 18
code
charts 311,313,320,323
IBM 2780 and 2972 313
end-of-cartridge (MDI) 241
feature, dual 26
IBM 2770 231
translation
for IBM 3270 (remote) 261
translation table 34
coding format, macro instructions 12-13
command control block (CCB) 7
common-carrier 2
communication interface, dual 26
communication line group 22,46
communication lines 2,362
configuration 2,4,5
enabling 6
leased (see nonswitched line) 2
private 2
communication network
configuration 2,4,5
binary synchronous 5
start-stop 4
communication system (see communication network)
communication terminal (see terminal)
completion codes 57
components, BTAM 9
concepts, BTAM 1-8
conditioning, line 31
CONFIG operand 26
CONFIGUR macro instruction, IBM 2715
explanation 199
format 199
operands 199
configuration, network (see network)
contention 3,161,269,363
control, line 3
control characters
data link 6
end-to-end 6
line 114
CONTROL macro instruction
creation of DECB 47
explanation 19
format 19
operands 19
options 19
parameters 19
return codes 20,54
use of 19
control mode 363
control station 2,364
control transactions
IBM 2715 223
IBM 2740 223
control unit address, IBM 3270 321
conversion from 2260 to 3270 268
core operand 199
counter control byte 228
counter readout (see IBM 2715 scheduled counter readout)
counter testing (see IBM 2715 counter testing)
CPU operand 204
creation of DECB 47
CTINIT operand (CTRGROUP macro) 205
CTLCHAR operand 26
ASCII 26
EBCDIC 26
TRNSCD 26
CTRADR operand (CTRLIST macro) 218
CTRGROUP macro (IBM 2715) 205
CTRLIST macro (IBM 2715) 218
CTROP operand (CTRLIST macro) 218
CTRRD operand (CTRLIST macro) 218
CTRSCHED operand (CTRGROUP macro) 205
CTTEST operand (CTRLIST macro) 218
CU operand (control unit) 23
cursor 364

data adapter unit 364
data collection system 364
data communication system 364
IBM 2770 231
data entry unit 192
format 194
IBM 2795 192
IBM 2796 192
index table 196
sequence table 196
table 194
data event control block (DECB) 49
creation of 47
defining 46
event control block (ECB) 48
fields 48
posted information 48
use of 47
data link 2,364
data link characters 2
data path 2
data set (hardware) 2
datack operand 28
DECB
extension
READ 47
use of 47
WRITE 47
fields
explanation 48
table of 49
operand 19,31,37
sense bytes 49
status bytes 49
DECBEXT operand 16
decchars 12
decimal digits 12
declarative macro instructions 9
CONTROL 9
DFTBT 9
DFTRMLST 9
LERB 9
READ 9
RMSRTAB 9
SDRTAB 9
WRITE 9
defining line groups 46
DEGROUP operand 201
DEMOD10 operand (TRLIST macro) 213
DEMOD11 operand (TRLIST macro) 213
DEULIST macro instruction, IBM 2715
explanation 216
format 216
operands 216
DEVDOD operand (CTRLIST macro) 219
device address
IBM 2715 203
IBM 3270 321
device dependent considerations
audio response unit 98-100
BSC 161-287
local 81-97
start-stop 101-161
(see also channel programs)
DEVICE operand 23
DFTRMLST macro instruction 21
explanation 21
format 21
parameters 21
dialing 3
DIALST operand 17,293
DIGIT2 operand (DEULIST macro) 216
DISC operand 18
DISK operand 204,213
DISPGUID macro (IBM 2715) 212
DISPMSG operand
of DISPGUID macro 212
of GDULIST macro 217
double addressing 282
DTFBT macro instruction
explanation 22
format 22
parameters 22-27
DTFBTND macro instruction 300
dtfname operand 18,31
dual code feature 26
dual communication interface 26

EBCDIC code 311
ECB operand 37
ECBLIST operand 36
edit function, IBM 50MDI 234
EIB mode
error information byte 25
non-error information byte 25
enabling a line 6
end of block 364
end of cartridge code 241
end of inquiry 364
end of transmission 364
end-to-end control characters 233
ENDERR operand (CONFIGUR macro) 199
entry field and
address pointer table 41
polling pointer table 41
ENTRY operand
of ASLIST macro 215
of GDULIST macro 217
of ONLTST macro 30
EOM operand (end of message) 27
EOT operand (end of transmission) 27
ERLOGIC operand 14
ERP (error recovery procedures) 59-68
ERROPT operand 25
error actions
for IBM 3270 (local) 94
for IBM 3270 (remote) 262-266
error codes, IBM 2715 time stamp 191
error counts 34,65
error handling routine 14
error information byte 25
error information posting 54
return codes 54
CHGNTRY 57
ONLTST 57
READ, WRITE, CONTROL 54-56
TRNSLATE 57
TWAIT 57
error messages 68
error identification record 239
error record samples 240
error recovery procedures
audio response units (ARU) 68
binary synchronous communication
actions 66

purpose 65
retransmission 66
counts 65
for IBM 3270 (local) 92
for IBM 3270 (remote) 261
messages 68-69
posted information 54
start-stop 59
actions 60
counts 65
messages 59
suggestions 64
error statistics, RMSR 318,34
error status messages
IBM 2770 231
IBM 3270 (remote) 261
event 47,364
event control block 48
exchanges, common-carrier 3
executing READ, WRITE, CONTROL macros 40
expanded ID verification 166
explicit counter addressing (IBM 2715) 191
EXTALRM operand (ASCTR macro) 203
extension, DECB 47
external alarm 227

facilities, BTAM 1
feature 23
feature, dual code 26
FEATURE operand 23
foreground 7
foreign terminals (see BTAM supported devices and channel programs)
format description, macros 12
formats of terminal lists 288
BSC lists 293
start-stop lists 288
frame change test 72
FUNCERR operand (CONFIGUR macro) 199
functions
answering 5
calling 5
IBM 3270 (local) 86
IBM 3270 (remote) 243,248
functions, BTAM general 8

GDU operand (TRLIST macro) 213
GDUAS macro (IBM 2715) 202
GDUNUMB operand (GDUAS macro) 202
GDULIST macro (IBM 2715) 217
GDUTRANS macro (IBM 2715) 206
GETID operand (CONFIGUR macro) 199
glossary 363

hexadecimal characters 12
hexadecimal equivalents for polling and addressing characters 297
HIGHCTR operand (ASCTR macro) 203
HIGUID operand (PARMLIST macro) 207
horizontal format record 173

IAM (I am) 163
IBM 50 Magnetic Data Inscriber (MDI) 234
IBM 1030 108-110
IBM 1050 115-120
IBM 1060 121-123
IBM 1130 182-184
IBM 2260-2848 152-156
IBM 2265-2845 152-156
IBM 2715
control transactions 192
data entry 192
error status messages 229
external alarm 227
functions 189
message formats 221
message header
System/370 to 2715 221
2715 to System/370 222
programming notes 220
pulse count 189
counter testing 191
scheduled counter readout 191
set counter 192
tables 193
transaction header 222
IBM 2740 124-151
control transactions (w/2715) 229
IBM 2770 231-242
multipoint communication 242
point-to-point communication 241
IBM 2780 173-181
programming notes 176
IBM 2790 189-230
IBM 2795 190
IBM 2796 190
IBM 2798 225
IBM 2972 188
IBM 3270 Display System 83,243
capabilities supported 86,243
conversion to 268
devices supported 84,243
local 3270 display system 83
attention interruptions 86
buffer management 92
CHGNTRY macro 88
code translation 85,92
entry operand 86
erase and write buffer 91
erase unprotected fields 91
error recovery procedures 92
interruptions, attention 86
online testing 92-97
read buffer 89
from position 90
read modified fields 88
after operator action 87
from position 89
read operations 87
read options 86
READ TB 89
READ TBP 90
READ TI 87
READ TM 88
READ TMP 89
RESETPL macro 88
retry options 92
return codes for RESETPL 57
rln operand 87
sample program 324
SKIP operand 88
write 91

buffer 91
operations 90
options 90
WRITE TI 91
WRITE TS 91
WRITE TUS 91
online testing 92
remote 3270 display system
buffer management 261
CHGNTRY macro 251
code translation 261
control unit addresses
for polling 321
for selection 321
copy 260
device addresses 321
DFTRMLST 246
entry operand 248
erase and write buffer 258
erase unprotected fields 259
error recovery procedures 261
error status message 261
exceptional conditions 261
I/O interface code 261
line control 245
line groups 245
online test options 262
online testing 262
read buffer 255
from position 257
read modified fields 252
after operator action 248
from position 253
read operations 247
READ TI 248
READ TT 248
retry options 262-263
terminal lists 246
write buffer 257
write operations 257
WRITE TI 257
WRITE TIV 255
WRITE TR 258
WRITE TT 258
WRITE TTV 257
IBM 3735 Programmable Buffered Terminal 268
IBM 3944 (w/audio response) 98
IBM 3977 modem 29
IBM 7770 98
IBM System 3 185
IBM System/360 Model 20 185
ID operand
of AS macro 201
of ASCTR macro 203
of CTRGROUP macro 205
of GDUAS macro 202
ID verification 163
binary synchronous communication 163
codes 164
TWX 157
ID verification, expanded 166
IDCOUNT operand (CONFIGUR macro) 201
IDENT operand
of GDULIST macro 218
of PARMLIST macro 212
identification card reader, 3270 87,248
identification feature, IBM 2770 242
identification security, IBM 2770 242
IDLST operand 21
imperative macro instructions 9
BTMOD 13
BTRD 16
CHGNTRY 16
CLOSE 18
CONTROL 19
LERPRT 28
LOPEN 29
ONLTST 29
OPEN 31
READ 31
RELBUF 33
REQBUF 33
RESETPL 34
TRNSLATE 35
TWAIT 35
WAIT 37
WRITE 37
implicit counter addressing (IBM 2715) 190
index byte 283
in IBM 3270 (remote) 249
index table, DEU 196
INLINE operand 34
inlist technique 279
INQDIST operand
of ASLIST macro 215
of CONFIGUR macro 201
of TRLIST macro 214
inquiry 364
integer 12
interface, dual communication 26
intermediate block checking 175
intermix, BSC 80
intermixed terminals, BSC 80
interrupt
overflow 227
power 227
interruption handler, BTAM 60
interruptions, handling of 7,60
intreq operand 28
invitation-to-send 1,101

keyword operand 12

L2260 operand 14
L3277 operand 15
leading graphics, READ continue 167
leased line (see nonswitched line)
LERB (line error block) 27
LERB macro instruction
explanation 27
format 28
parameters 28
LERBADR operand 25
LERPRT macro instruction
explanation 28
format 29
parameters 29
librarian-control statement 45
line conditioning 29
line connection 364
line control 3
line control block (LCB) 7
line control characters 364
line, enabling 6

line error block 27
line groups, defining 46
LINELST operand 22
lines, communications
   nonswitched 2
   switched 3
local device 81
local terminal 81
local 3270 83-97
lock option 83
LOG operand
   ASCTR macro 203
   TRLIST macro 214
logic module, BTAM 1
logical unit block (LUB) 46
LOPEN macro instruction
   explanation 29
   format 29
   parameters 29
LOWGUIDE operand (PARMLIST macro) 208

machine and device requirements 1,22
macro format description 12
macro instructions, BTAM (see individual macro instructions)
   buffer management 33
   coding format 12-13
   declarative
      CONTROL 19
      DTFBT 22
      DFTRMLST 21
      LERB 27
      READ 31
      RMSRTAB 34
      SDRTAB 35
      WRITE 37
   imperative
      BTMOD 13
      BTRD 16
      CHGNTRY 16
      CLOSE 18
      CONTROL 19
      LERPRT 28
      LOPEN 29
      ONLTST 29
      OPEN 31
      READ 31
      RELBUF 33
      REQBUF 33
      RESETPL 34
      TPEDIT 235
      TRNSLATE 35
      TWAIT 35
      WAIT 37
      WRITE 37
   summary 9
macro operands, allowable values 298
macro restrictions 11
message 364
message assurance 6
message, contention 3
message control byte 221
message formats
   IBM 2715 221
   IBM 3270 (local) 87-91
   IBM 3270 (remote) 247-260
message, interruption handler 68
message segment 364
message-switching 2
messages, operator awareness 68
MODE operand 19
MODELST operand 25
modem 2
MODNAME 25
modulation 364
modulator/demodulator MODULUS operand
   of ASLIST macro 216
   of DEULIST macro 217
MONDLY operand 26
MONERR operand (CONFIGUR macro) 200
motor-on feature 26
MSG operand
   of ASLIST macro 216
   of CTRLIST macro 219
   of DEULIST macro 217
   of GDULIST macro 218
MSGL operand 25
multicomponent addressing 364
multiple entries in polling list 46
multipoint communication, IBM 2770 242
multipoint line 282
multitasking 7
multiterminal addressing 365

NAK (negative acknowledgement) 270
negative response 6
network 365
   configurations 4-5,83,84,243
   nonswitched 2
   switched 3
networks, telecommunications (see also network) 2
nlines operand 28
noncentralized communication system 3
nonswitched lines 2
NORGUID operand (GDULIST macro) 218
NORM operand 215
nontto operand 28
numchars operand 17

OBR/SDR (see RMSR)
OLTEP for IBM 3270 (local) 95
online terminal tests
   BSC 73
   requesting 69,76-77,95,267
   restrictions 72
   start-stop 69
   test types 69,77
   (see also BSC online tests)
ONLTST macro instruction
   explanation 29
   format 29
   parameters 30
OPEN macro instructions
   explanation 31
   format 31
   parameters 31
OPENLST operand 21
operands
   keyword 12
   positional 12
operator awareness messages 68

operation, line control 3
centralized 3
noncentralized 3
options, READ, WRITE, CONTROL (see also channel programs) 43-44
optype operand 19,32,38
ORGUID operand (PARMLIST macro) 209
overflow interrupt, IBM 2715 227

PARAMNO operand (GDULIST macro) 218
PARAMNUM macro (IBM 2715) 207
PARMLIST macro (IBM 2715) 207
PARMLST operand (PARAMNUM macro) 207
PC operand (CONFIGUR macro) 199
physical unit block 46
PLN operand (PARAMNUM macro) 207
point-to-point communication, IBM 2770 241
point-to-point line 365
polling 3,6,46,365
characters 297,365
codes 46
for IBM 3270 (remote)
control unit address 321
device addresses 321
pointer 52
terminal list 46
polling list types, nonswitched
AUTO poll 46
open list 46
wraparound list 46
polling lists, multiple entries 46
pool, buffer (see also buffer pool) 52
positional operand 12
positive response 6
POST operand 18
posted error information 54
posting in the DECB 48
prepare command, terminate 34
private lines 2
processing time, sharing of 6
program addressability, preserving 45
program
cancelations 68
considerations, MDI 240
example, IBM 2715 344
programming notes, IBM 2780 176
PUB (physical unit block) 46

(r), register notation 12
READ Connect macro instruction 167,170
with Tone macro instruction 167,170
READ Connect 167,170
READ macro instruction
creation of DECB 47
explanation 31
format 31
operation types (see channel programs and options)
options 44-45
parameters 31
return codes 33,54
READ/WRITE/CONTROL options 44-45
IBM 2770 232
IBM 3270 (local) 87
IBM 3270 (remote) 249
tables 44-45
terminal error status message, IBM 2715 229
record 365
record length, IBM 2772 231
register notation 12
register usage 45
RELBUF macro instruction
explanation 33
format 33
parameters 33
return codes 33,56
relexp 12
relocatable expression 12
remote device 365
remote station 365
remote terminal 365
remote 3270 243-268
REQBUF macro instruction
explanation 33
format 33
parameters 33
return codes 34,56
reset function 112
reset poll 34
RESETPL macro instruction
explanation 34
format 34
IBM 3270 (local) 88
parameters 34
return codes 34,57
response
negative 3
positive 3
restarting channel programs 1
restrictions, macro 11
retransmission of messages 20
RETRY operand 26
retry option
CPU to CPU contention 163
CPU to CPU dial 164-166
return codes 54-57
rln (relative line number) 19,31,37
RMSR
facilities 34,318
operand of BTMOD 16
RMSRTAB macro instruction 34
RNGETST operand (PARMLIST macro) 209
ROUTE operand
of ASCTR macro 204
of TRLIST macro 213
RVI (reverse interrupt) 49,270,277

sample programs
BSC 307
IBM 2715 344
IBM 3270 324
start-stop 304
scan point test 72
SDRTAB macro instruction (see RMSRTAB)
security identification 242
selection characters 3
SELECTRIC test 72
self defining term 12
SELTRAN operand (ASLIST macro) 216
sense bytes, DECB 49
SEPASMB operand 14
use of 14,45

serviceability facilities 57
set counter (see IBM 2715) 192
sharing processing time 6
six-bit TRANSCODE 311
SKIP operand 17,18
SROENAB operand (CTRGROUP macro) 206
SSALST operand 17,21
SSAPL operand 15
SSAWLST operand 17,21
standard ERP for audio response units 68
start-of-text (STX) 6
start-stop communications
  device dependent considerations 101-160
  error recovery procedures 59
  online terminal tests 69
  terminal lists 46
start-stop transmissions 365
station 365,2
station control 3
stations supported (BSC1, BSC2, BSC3) 161
statistics, error 318
status bytes, DECB 49
STEND macro instruction
  explanation 219
  format 219
STORID operand (CONFIGUR macro) 200
STX (start-of-text) 6
supported devices
  BTAM 11
  start-stop summary 101
SUPPRES operand (DISPGUID macro) 212
SWITCH operand 23
switched lines 3
switched network 3
SWLST operand 18
symbol 12
System/370 to System/370 communication 162

table definition block 194
table operand 35
tables, IBM 2715 193
telecommunication control unit 365
telecommunication network 2
  concepts 2
  configurations 4,5
telecommunications 365
teleprocessing 365
terminal
  addressing 46
  component 2,365
  error status message, IBM 2715 229
  list 46
  local 2
  polling 46
  remote 2
  types 2
terminal list formats 288-296
  defining 46
  online 46,76
terminal test, online 69
terminology, telecommunications 1-7,363
TERMTST operand 14,25,36
test activation code 25
text mode 365
TEXT operand 30
TGROUP macro instruction
  explanation 203
  format 203
  operands 203
TGROUP table 194
threshhold value 65,318
time sharing 6
time stamp, IBM 2715 191
TP codes 50
TPEDIT macro instruction
  error record identification 239
  explanation 235
  input to 237
  return codes 239
transaction
  control byte, IBM 2715 222,223-224
  group table (TGROUP) 194
  header, IBM 2715 222
  length, IBM 2715 221
  list tables (TRLIST) 194
TRANSCODE 311,173
transient macro instructions
  CLOSE 18
  LOPEN 29
  OPEN 31
TRANSL operand 15
translation tables 36
transmct operand 28
transmission codes, IBM 2770 231
transmission control unit (TCU) 23,365
transparent data 30
TRANTXT operand (TRANSLAT macro) 212
TRCODE operand (GDUTRANS macro) 206
tributary station 365
TRID operand 213
TRLIST macro instruction, IBM 2715
  explanation 213
  format 213
  operands 213
TRNSLATE macro instruction
  explanation 35
  format 35
  parameters 35
TRSRCT3 macro instruction
  explanation 42
  format 42
TRSRCTW macro instruction
  explanation 42
  format 42
TRSSCT macro instruction
  explanation 42
  format 42
TRSSCTW macro instruction
  explanation 42
  format 42
TWAIT macro instruction
  explanation 35
  format 35
  parameters 35-36
TWX 105

unit 365
updating, buffer pools 54
US (unit separator) 25
USASCII 311

verification (ID) 163
vertical forms control 173

WAIT macro instruction
   explanation 37
   format 37
   parameters 37
Western Union Plan 115A 101
Western Union TWX 33/35 terminals 105
work day number, IBM 2715 222
World Trade Telegraph Codes (ITA2 and ZSC3) 321
World Trade Telegraph Terminals 157-159
World Trade Telex Terminals 160
World Trade translation tables 42
WRAPLST operand 17,21
WRITE Connect macro instruction 166,169
WRITE macro instruction
   creation of DECB 37
   explanation 37
   format 38
   operation types (see channel programs and options) 43-44
   options 43-44
   parameters 38
   return codes 40,54
WRU (who are you) 163
WTTA operand 16
WU terminals 101

83B3 selective calling stations 103

GC27-6978-0

DOS V/4 (BTAM) Printed in U.S.A. GC27-6978-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

DOS Version 4
Basic Telecommunications
Access Method (BTAM)

READER'S COMMENT FORM

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?

- ☐ As an introduction
- ☐ As a text (student)
- ☐ As a reference manual
- ☐ As a text (instructor)
- ☐ 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):

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.)

Cut or Fold Along Line

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Cut or Fold Along Line

Fold Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department 636
Neighborhood Road
Kingston, New York 12401

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)