LY21-0051-0

File No. S34-36

IBM System/34
System Support Program
Logic Manual: Data Communications

Program Number 5726-SS1

SSP LOGIC: DATA COMMUNICATIONS

LY21-0051-0

File No. S34-36

Licensed Material
Property of IBM

IBM System/34
System Support Program
Logic Manual: Data Communications

Program Number 5726-SS1

SSP LOGIC: DATA COMMUNICATIONS

Preface

This publication is designed to aid personnel in supporting the IBM System/34 data communications programs by serving as a recall mechanism and as a guide to the program listings and microfiche. This publication does not contain directions for operating or programming System/34.

Note: The IBM System/34 System Support Program Product is documented by two manuals: this manual and *System Support Program Logic Manual: System.*

RELATED PUBLICATIONS

- *IBM System/34 Data Communications Reference Manual,* SC21-7703
- *IBM System/34 System Support Reference Manual,* SC21-5155
- *IBM System/34 System Data Areas and Diagnostic Aids Handbook,* LY21-0049
- *IBM System/34 System Support Program Logic Manual: System,* LY21-0050

First Edition (December 1977)

This edition applies to version 01 of the IBM System/34 System Support Program Product (Program Number 5726-SS1) and to all subsequent versions until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the specifications herein; before using this manual in connection with the operation of IBM systems, be sure you have the current edition.

Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

A Reader's Comment Form is at the back of this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.

Contents

This page intentionally left blank.

How This Manual is Organized

This manual is divided into two parts: Binary Synchronous Communications (BSC) and MULTI-LEAVING Remote Job Entry (MRJE) Utility. Each part consists of the following sections:

- *Introduction* contains general information about the functions and characteristics of the data communications component.
- *Method of Operation* contains functional diagrams that illustrate the operation of the data communications component. The section also contains references to the modules and routines that are called to perform the functions.
- *Program Organization* describes the organization of the data communications component through the use of storage maps and functional flow diagrams.
- *Directory* lists the modules, their descriptions, and their associated diagram numbers.
- *Data Areas* discusses the data areas that are unique to the system component. Other data areas are described in the *Data Areas Handbook.*

Note: Diagnostic Aids is not included. Information concerning diagnostic aids is in the *Data Areas Handbook.*

This page intentionally left blank.

Diagram numbers in this manual are assigned by part. For example, all diagrams in Part 1 are numbered 1.x. Any diagram with a zero preceding the first period (Diagram 0.1, for example) is a table-of-contents diagram. Each entry in a table-of-contents diagram identifies a particular input-process-output diagram. The input-process-output diagrams describe functions and are found in the *Method of Operation* section of each part of this manual.

Figure 1 is the overview table of contents for IBM System/34 programming. Following Figure 1 is the table of contents for all programming described in this manual (see Diagrams 0.1 and 0.2).

DIAGRAM TECHNIQUES

Diagrams in this manual graphically explain the functions of the program. These diagrams are structured from general to detailed levels; namely, the table-of-contents diagrams, one or more overview diagrams, and one or more lower-level diagrams for each overview diagram.

Table-of-Contents Diagrams

The table-of-contents diagrams (diagrams 0.x) show:

- The structural relationship of all the diagrams
- The descriptive title of each diagram
- The assigned number for each diagram

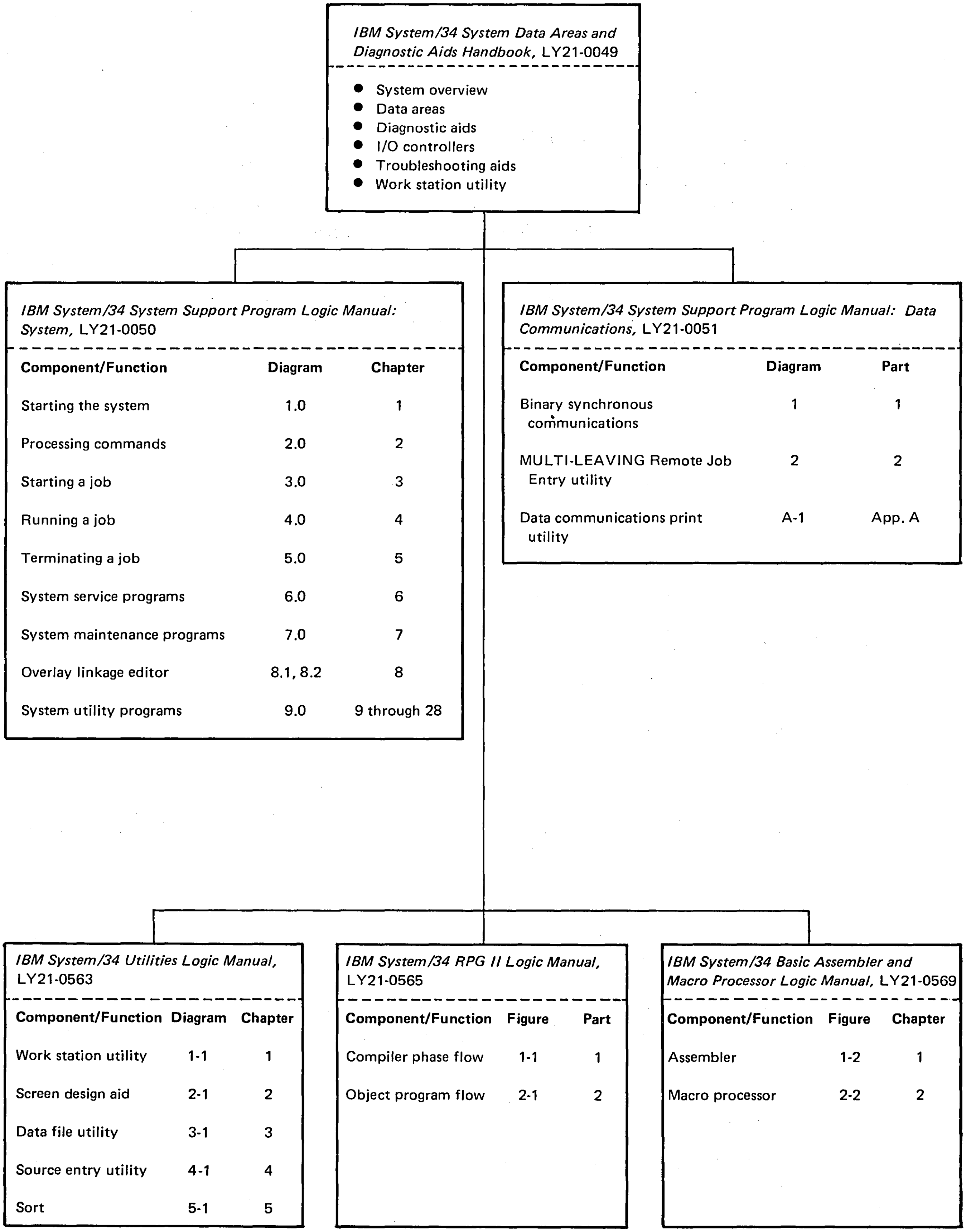
Note: The table-of-contents diagrams also show how the functional diagrams within this manual relate to the diagrams for the rest of System/34.

Overview Diagrams

The input-process-output format is used for both the overview and lower-level diagrams.

An overview diagram summarizes the functions of a group of related lower-level diagrams and is included in the appropriate *Method of Operation* section. Overview diagrams show:

- The main input items
- The main output items
- A brief description of functions (including references to lower-level diagrams for each function)



IBM System/34 System Data Areas and Diagnostic Aids Handbook, LY21-0049

- System overview
- Data areas
- Diagnostic aids
- I/O controllers
- Troubleshooting aids
- Work station utility

IBM System/34 System Support Program Logic Manual: System, LY21-0050

Component/Function	Diagram	Chapter
Starting the system	1.0	1
Processing commands	2.0	2
Starting a job	3.0	3
Running a job	4.0	4
Terminating a job	5.0	5
System service programs	6.0	6
System maintenance programs	7.0	7
Overlay linkage editor	8.1, 8.2	8
System utility programs	9.0	9 through 28

IBM System/34 System Support Program Logic Manual: Data Communications, LY21-0051

Component/Function	Diagram	Part
Binary synchronous communications	1	1
MULTI-LEAVING Remote Job Entry utility	2	2
Data communications print utility	A-1	App. A

IBM System/34 Utilities Logic Manual, LY21-0563

Component/Function	Diagram	Chapter
Work station utility	1-1	1
Screen design aid	2-1	2
Data file utility	3-1	3
Source entry utility	4-1	4
Sort	5-1	5

IBM System/34 RPG II Logic Manual, LY21-0565

Component/Function	Figure	Part
Compiler phase flow	1-1	1
Object program flow	2-1	2

IBM System/34 Basic Assembler and Macro Processor Logic Manual, LY21-0569

Component/Function	Figure	Chapter
Assembler	1-2	1
Macro processor	2-2	2

Figure 1. System/34 Logic Documentation Overview

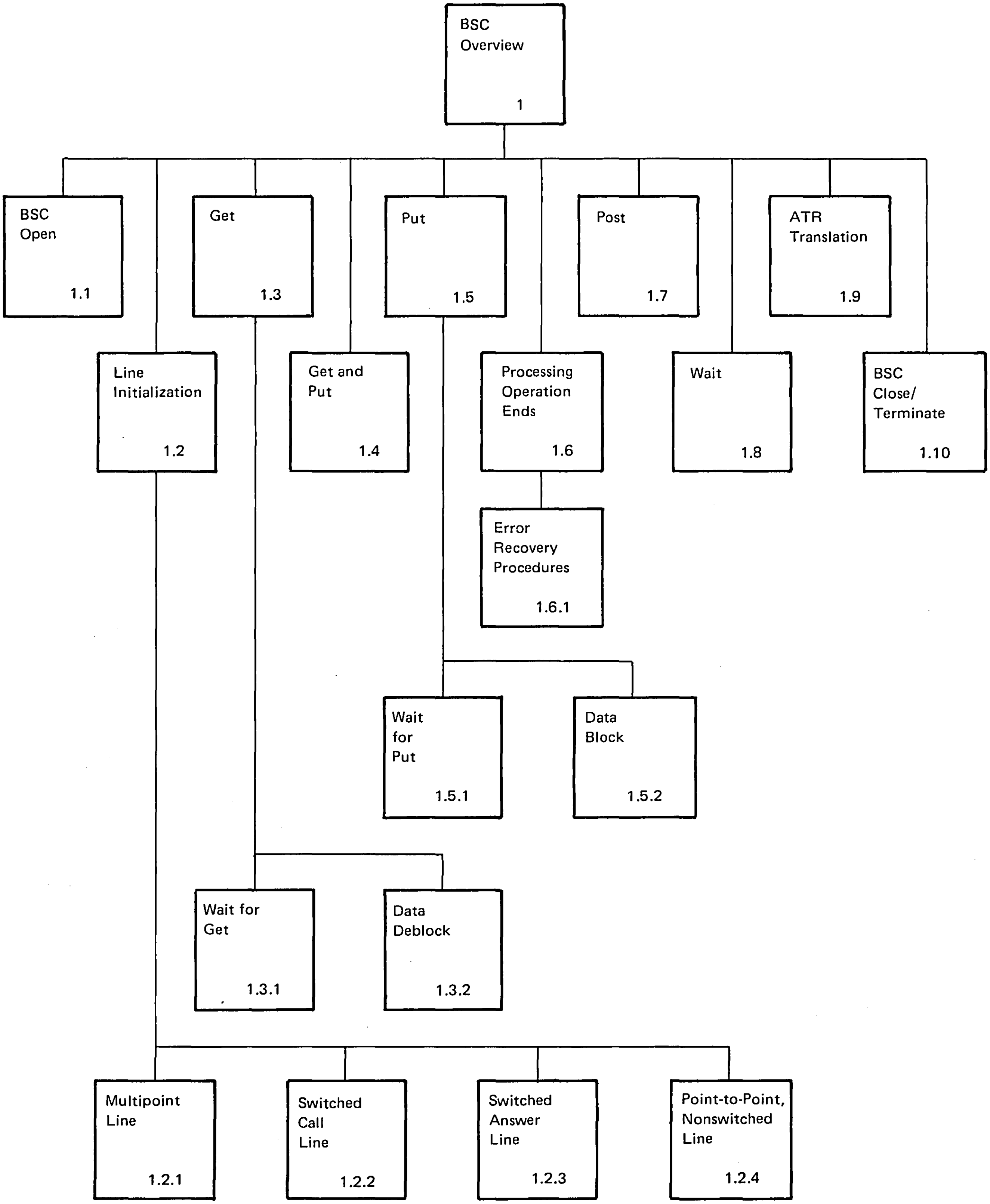


Diagram 0.1. BSC Overview Table of Contents

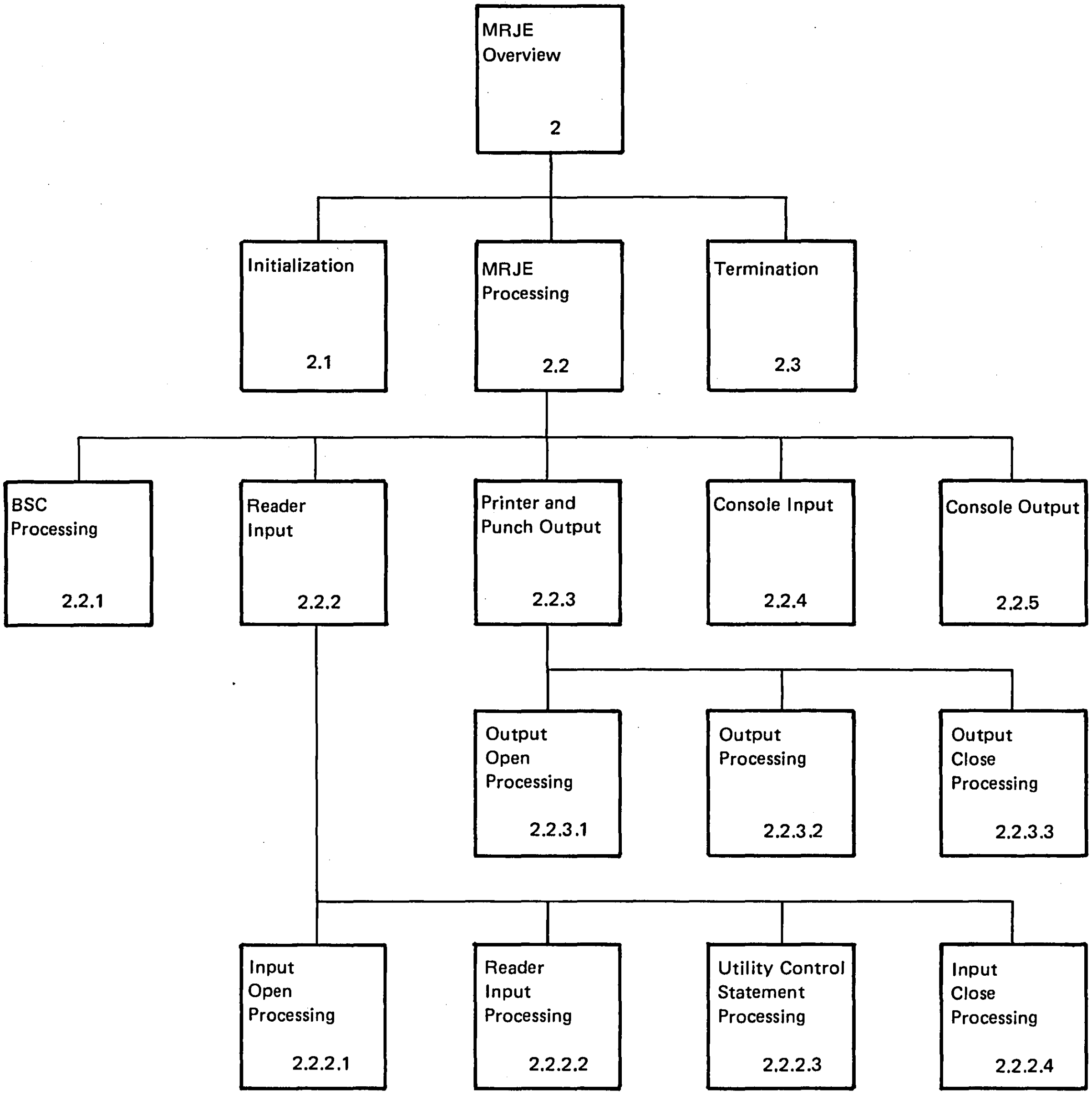

MRJE Overview 2
Initialization 2.1
MRJE Processing 2.2
Termination 2.3
BSC Processing 2.2.1
Reader Input 2.2.2
Printer and Punch Output 2.2.3
Console Input 2.2.4
Console Output 2.2.5
Output Open Processing 2.2.3.1
Output Processing 2.2.3.2
Output Close Processing 2.2.3.3
Input Open Processing 2.2.2.1
Reader Input Processing 2.2.2.2
Utility Control Statement Processing 2.2.2.3
Input Close Processing 2.2.2.4

Diagram 0.2. MRJE Overview Table of Contents

In the following example, arrows and boxes show relationships by their positions:

❶ All items in INPUT apply to all numbers in PROCESS.

❷ Item 🅑 applies only to number 3.

❸ Item 🅒 applies to all numbers in PROCESS.

❹ Items 🅒 and 🅓 apply only to number 4.

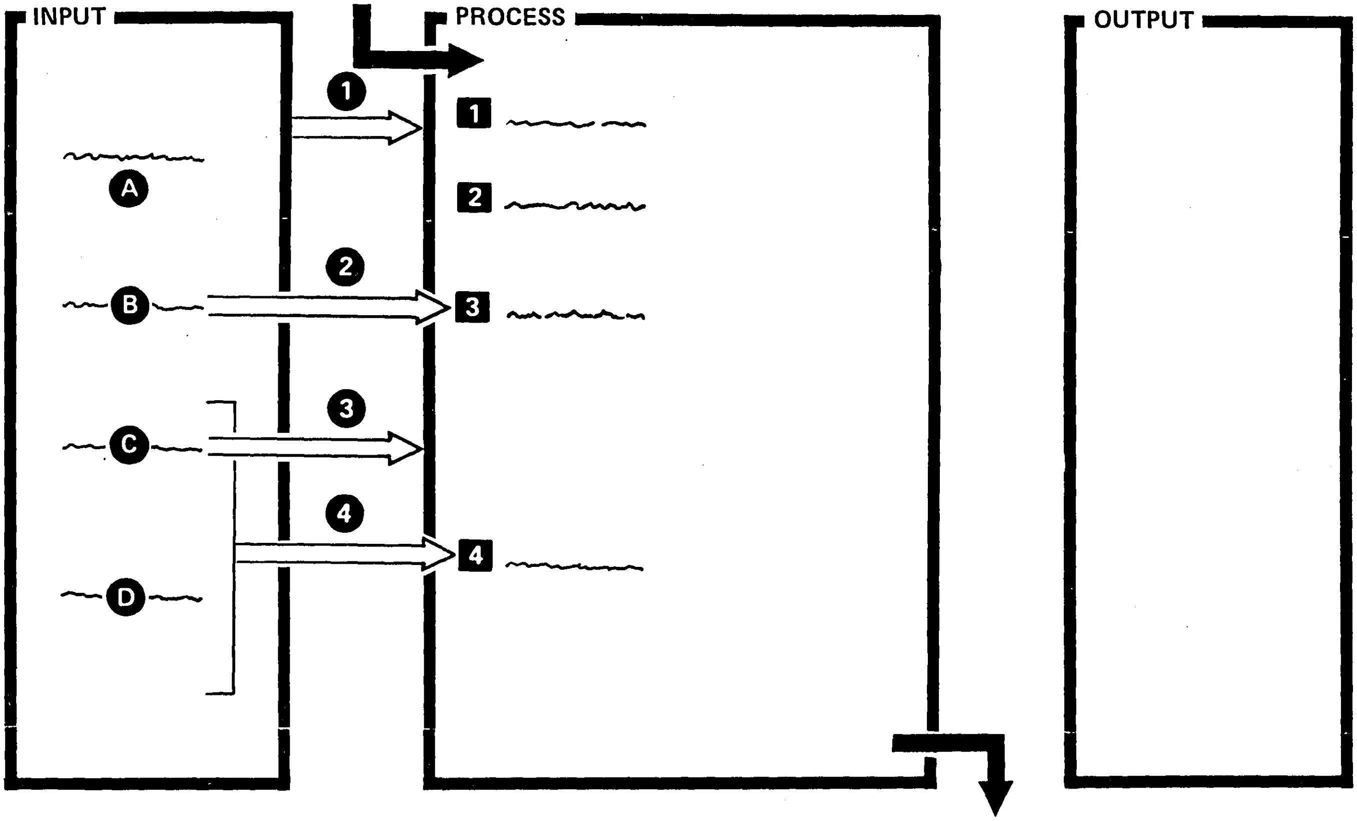
Legend

➡ Control flow (solid arrow)

⇒ Data flow (open arrow)

3 Reference number

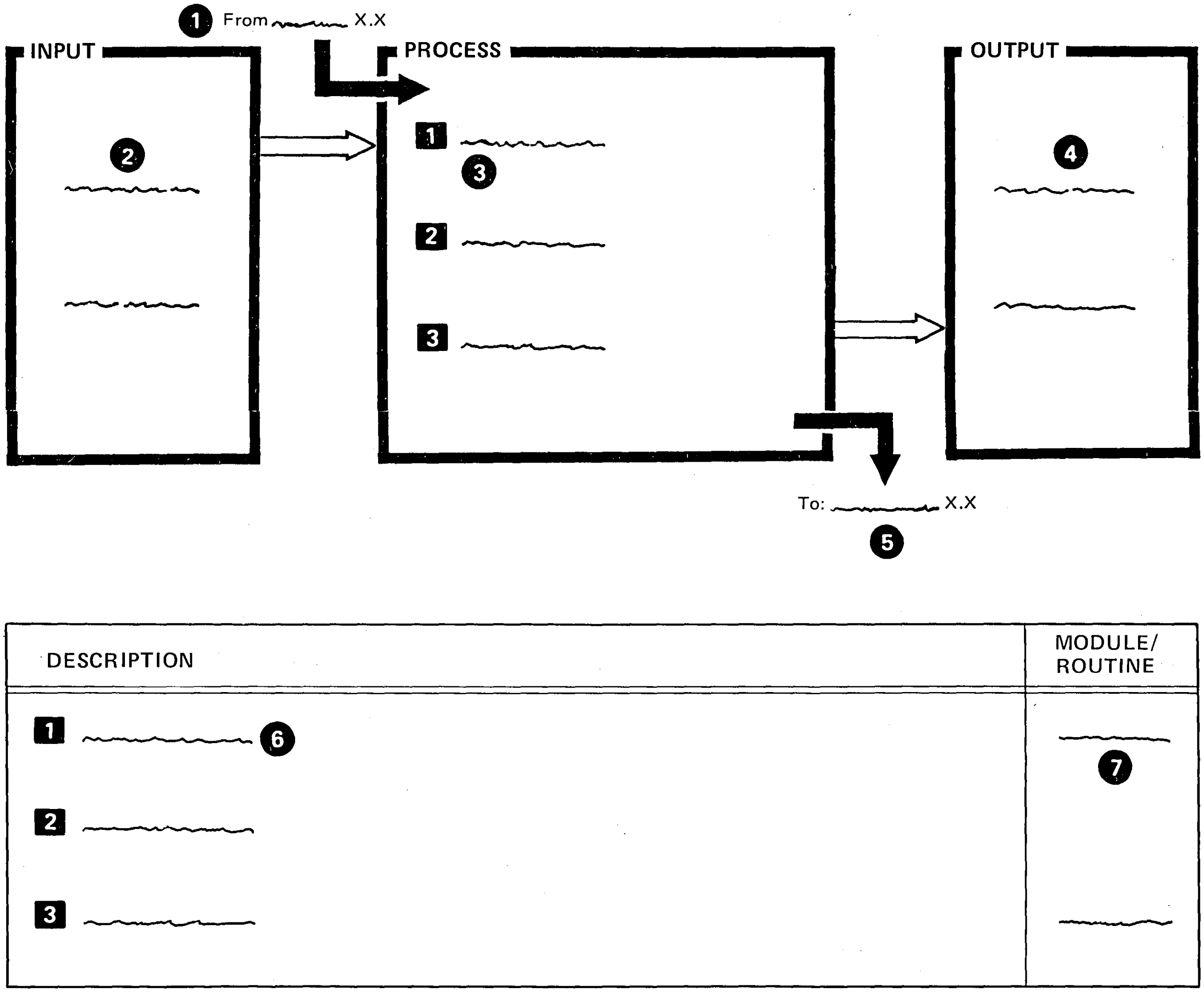
⌒→ Address pointer



Lower-Level Diagrams

Lower-level diagrams are included in the appropriate *Method of Operation* section and are arranged to show:

1. The name of each function that passes control to this diagram
2. The requirements for processing (INPUT)
3. A sequential listing of PROCESS steps
4. The results of processing (OUTPUT)
5. The name of each function (diagram) that receives control
6. An extended description (boxed numbers match those in the PROCESS area)
7. The module or routine where each step takes place



Introduction

The IBM System/34 binary synchronous communications (BSC) programming input/output control system supports the data communications capability of the IBM communications adapter.

System/34 can transmit and receive EBCDIC data in either the transparent or nontransparent text mode.

System/34 can transmit and receive ASCII data only in the nontransparent text mode.

Each mode can use intermediate block checking. Transparent block checking can be used only for receive files.

System/34 BSC supports the following line types:

- Point-to-point, nonswitched
- Point-to-point, switched
- Multipoint tributary

BSC can perform these functions:

- Receive only (get): Receives input data from a remote station.
- Transmit only (put): Transmits data to a remote station.
- Receive and transmit (get and put): Receives a file and then transmits a different file, or transmits a file and then receives a different file.

FUNCTIONAL ORGANIZATION

This section describes the operating sequence for the following BSC functions:

- Opening BSC files
- Receiving/transmitting data, which includes:
  - Initializing BSC lines
  - Receiving data (get), transmitting data (put), or receiving and transmitting data (get and put)
  - Processing BSC operation ends
  - Logging BSC errors
- Closing BSC files
- Terminating the BSC task

Figure 1-1 shows the functional control and data flow of BSC.

Functionally, System/34 BSC has three steps:

1. BSC establishes line control between the transmitting station and the receiving station.
2. BSC incorporates data to be transmitted into a format compatible with line control conventions.
3. BSC recovers data received in the form intended by the sender.

The user's program calls BSC and passes the address of a postopen DTF, which serves as the communication area between the two programs. BSC formats the IOB area into input/output blocks (IOBs) and then establishes line control between the sender and the receiver.

BSC moves data to be transmitted into an available I/O buffer, one record at a time, inserts appropriate control characters, and transmits the data, one block at a time, to the receiving station.

The receiving BSC accepts transmitted data into available I/O buffers. The receiving BSC then moves the data, excluding control characters, to the user's logical buffer.

Note: The assembler user can move the entire physical data block, including control characters, to the user's logical buffer. See the *Data Communications Reference Manual* for detailed information.

BSC disables the line (1) when all data has been transmitted, or (2) if an unrecoverable (permanent) error occurs when the user requests a close of BSC.

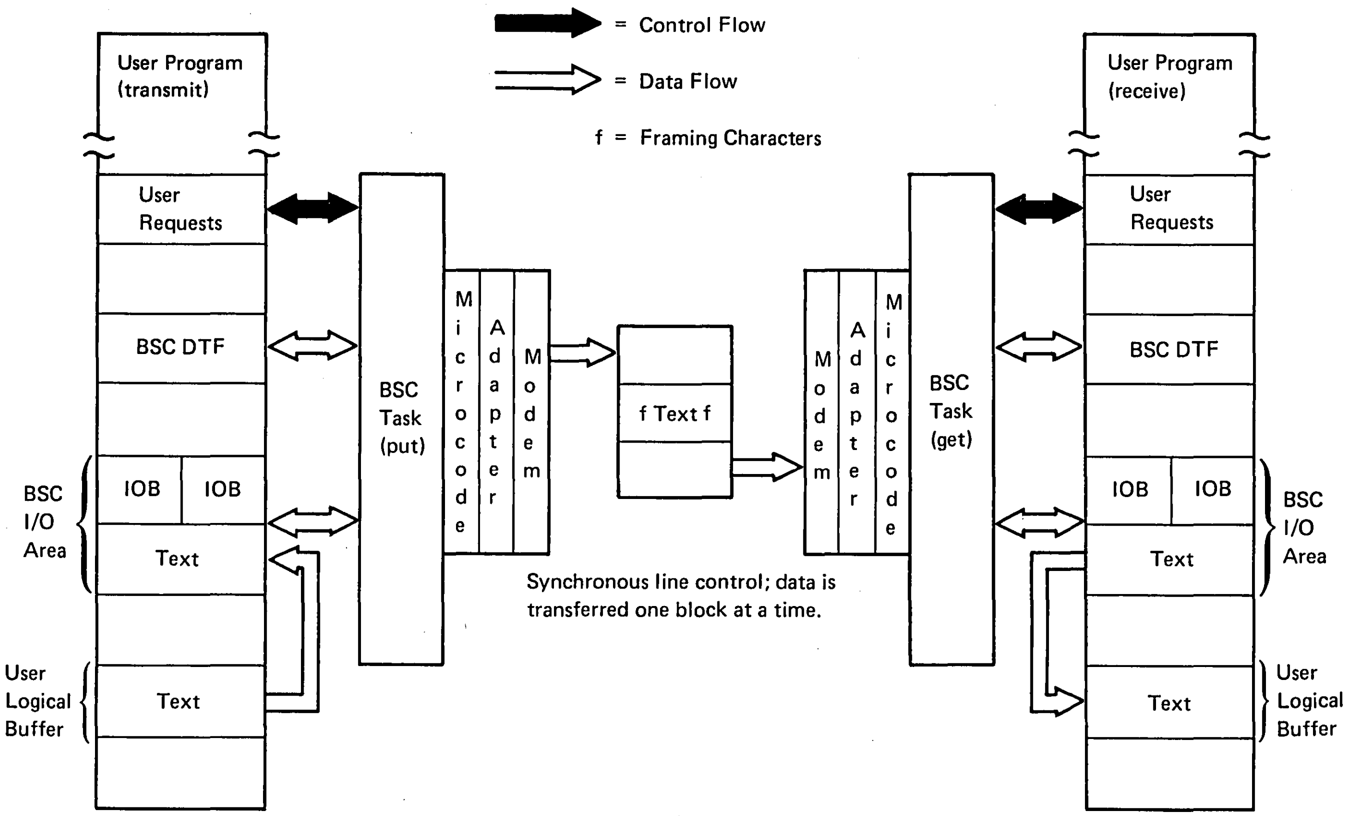


Figure 1-1. BSC Functional Control and Data Flow

Opening BSC Files

Before BSC can process any data, the user must open the BSC files. The common open routine (#DMOF) calls the BSC open transient (#BSOB) to open the BSC file for the DTF pointed to by index register 2 (XR2). BSC updates the DTF with any information specified previously by an ALTERBSC or OVERRIDE procedure and opens the DTF.

BSC open looks at the forward chaining pointer in the DTF and, if there are any other BSC DTFs in the chain, opens them also.

BSC open is a transient and functions completely within the transient area of the main storage nucleus.

Receiving/Transmitting Data

When System/34 receives and/or transmits data, BSC:

- Initializes the BSC lines
- Performs a get operation, a put operation, or a get-and-put operation
- Processes BSC operation ends
- Logs any BSC errors

Initializing BSC Lines

If the current request by the user's program is the first request for the file, the BSC data management mainline routine (#BSDB) calls the line initialization transient (#BSL0) to initialize the BSC line.

#BSL0 sets the file-active and file-used indicators in the DTF, formats the IOBs, loads BSC control storage, and enables the communications adapter (if the line is inactive and the communications adapter is not enabled). For switched lines, if the user specifies IDs in the DTF, #BSL0 allocates an area from the system assign/free area and moves the IDs from the user program into this area. Except for multipoint lines, #BSL0 then calls the appropriate line initialization transient to initialize the line: #BSL2 initializes switched call lines; #BSL3 initializes switched answer lines; #BSL4 initializes point-to-point, nonswitched lines. The line initialization transients call the BSC IOS (input/output supervisor) routine (#BSIO) to start all BSC operations. The line initialization transients then exit to #BSDB. For multipoint lines, #BSL0 returns to #BSDB.

While the initialization is in process, #BSDB calls the BSC wait routine (#BSIW) to ensure that BSC data transfer is complete. If an error occurs, #BSIW sets the user's permanent-error indicator (RPG) or posts an error-completion code in the DTF (assembler) and logs the error in the BSC error history table.

Performing a Get, Put, or Get-and-Put Operation

The data management mainline routine (#BSDB) controls the logic flow for BSC data management. After the line has been initialized, #BSDB calls #BSMG to deblock the data for a get operation or calls #BSMP to block the data for a put operation.

In addition, if blank compression is being used, #BSMG calls #BSXP to add the blanks removed by the remote station, and #BSMP calls #BSCP to take out the blanks. These four routines use the data management move routine (#BSMO) to move the data from the physical I/O buffer to the user's logical buffer (get operation) or from the user's logical buffer to the physical I/O buffer (put operation). The BSC IOS routine (#BSIO) is called to start BSC data transfer when a physical buffer is ready.

Processing BSC Operation Ends

The BSC operation end routine (#BSHB) analyzes all BSC operation ends for errors. If an error occurs, the appropriate error recovery procedure is executed. If no error occurs and the next I/O buffer is ready for data transfer, the BSC IOS routine (#BSIO) is called to initiate that transfer. If the next buffer is not ready for data transfer, a 2-second time-out is started.

#BSHB also traces every operation end if the trace has previously been activated by the TRACE service procedure.

Logging BSC Errors

Each time a permanent BSC error occurs, the BSC wait routine (#BSIW) makes an entry into the system error history table by calling the system logging utility (#CMWO).

At close time, the BSC close routine (#BSCL) logs text sent, text received, and error counters accumulated during the job by calling the system logging routine. This routine logs the counters for this job and adds these counters to the cumulative BSC counters.

Closing BSC Files

When BSC processing is complete, the user should close the BSC DTFs. The common close routine (#DMCL) gives control to the BSC close routine (#BSCL) by setting a special operation code in the DTF and requesting the BSC task. #BSCL completes the processing of any data in the I/O buffer (put files only), disables the BSC line if no new files are pending, logs all BSC error and text counters, frees the BSC I/O area, and closes the BSC file. #BSCL also closes any other BSC files that it finds in the DTF forward chain before returning to common close.

#BSCL is also called by the line initialization transient (#BSLO) to close any active files when a new file is pending.

BSC close is a main storage transient and is brought in only when needed to close BSC files or to terminate the BSC task.

Terminating the BSC Task

Abnormal Termination

The system termination routine (#CTECM) requests the BSC task whenever an abnormal condition occurs. By setting a special operation code in the DTF (actually a pseudo DTF), system termination causes the BSC close transient (#BSCL) to be given control.

#BSCL disables the BSC line, if it is active, and frees the BSC I/O area.

Normal Termination

When the BSC task is no longer required by a user program, the system termination routine (#CTECM) requests the BSC task to terminate, thereby freeing the 6,144 bytes of the main storage user area for other tasks. System termination requests this function by setting up a special operation code in a pseudo DTF in the same manner as for an abnormal termination. This again gives control to the BSC close routine (#BSCL), which performs the same functions as done for an abnormal termination in addition to terminating the task.

INPUT/OUTPUT

Input required by System/34 BSC consists of the following:

- A 68-byte DTF
- A logical buffer (addressed from the DTF)
- A communications configuration record
- A communication specification block (CSB)

Output depends on the program being run.

SYSTEM INTERFACES

System/34 BSC programming support consists of seven main storage transients and one main storage data management and IOS routine that reside in the system library. BSC interfaces with the SSP, BSC control storage, hardware, and the user's RPG II or basic assembler program.

SSP Interfaces

BSC operates under control of the SSP. The SSP controls BSC file allocation, termination, open, and close. It also controls BSC usage of the system log and trace functions.

BSC operates as a separate task from the user program. It is loaded into the user portion of main storage and attached as a task by the allocate function. By existing as a separate task, BSC runs independent of the user program, thus allowing the user program to be swapped to disk. BSC main storage transients (except BSC open) run under control of the BSC task, and are loaded into the main storage transient area of the nucleus as they are needed. Index register 2 (XR2) points to the BSC DTF whenever the common open routine (#DMOF) calls the BSC open transient (#BSOB). The common close routine (#DMCL) and the system termination routines invoke the BSC close and terminate functions by setting special operation codes in the DTF (pseudo DTF for terminate) and interfacing to the BSC task in a similar fashion to a user program. In the latter cases, XR2 also points to the DTF.

Control Storage Interface

The BSC line initialization transient (#BSLO) loads the BSC microcode into control storage via an SVC instruction. Communication between the BSC task and the BSC microcode is done via SVCs and an IOB.

Hardware Interface

The BSC task communicates with the hardware through the BSC control storage routine.

User Program Interface

A System/34 user can interface to BSC by coding a source program in either RPG II or basic assembler language (making use of macroinstructions). The source program is then processed by the RPG II compiler or the assembler, and a load module is created and stored in the system library. The scheduler program loads the load module from the library into the user portion of main storage for execution.

The RPG II compiler or the assembler generates a BSC DTF for each BSC file in the source program. The DTF is used to pass information between the user's program and the BSC task.

OPERATIONAL CONSIDERATIONS

System/34 BSC executes according to the user's job requirements and according to the user's system configuration. Specific considerations affecting BSC are: (1) the function required to run the job, (2) the mode in which the data is to be sent, (3) the line type used, and (4) the control statement and procedure command specifications.

COMMANDS AND CONTROL STATEMENTS

The OVERRIDE and ALTERBSC procedures (or the SETB and SETR utility control statements for the $SETCF utility) alter BSC configuration information.

The ALTERBSC procedure alters hardware and modem information.

The OVERRIDE procedure overrides certain parameters specified by the user program or specifies certain parameters that cannot be specified by the user program.

ERROR RECOVERY

Types of Errors

BSC errors can be caused by:

- System hardware
- Communications adapter
- Operator or programmer

Errors Caused by System Hardware

In general, nonrecoverable errors on non-BSC devices that occur during execution of a BSC program terminate BSC program execution. Recoverable errors on a non-BSC device may cause a communications adapter time-out if extensive time is required to recover from the error. BSC error recovery procedures may be required to reestablish line connections and synchronization.

Permanent Errors Caused by the Communications Adapter

Recovery from a BSC hardware error is handled by BSC programming support. The permanent-error condition is displayed on the system log device, and the error is recorded in the system error history table.

For RPG II users who have specified a permanent-error indicator, the error condition is displayed as an informational message and control is returned to the RPG II program. If no permanent-error indicator has been specified, a message requiring operator response is displayed either by the BSC task or the RPG II program.

For the basic assembler user, all permanent-error messages, except some open and close messages, are displayed as informational messages and control is returned to the user program with a completion code in the DTF.

Errors Caused by an Operator or Programmer

Error messages are displayed on the system log device for errors caused by invalid ALTERBSC and OVERRIDE procedure commands or invalid SETB and SETR utility control statements.

Error Recovery Diagnostics and Aids

System/34 BSC error recovery diagnostics and aids include:

- Automatic wrap test for unrecoverable errors
- Error logging in the BSC error history table
- Error retry count
- Message display
- Interrupt trace if specified in the TRACE command
- Permanent-error indicator (RPG II only)
- Exact completion code in DTF (basic assembler only)

Automatic Wrap Test

The BSC close routine (#BSCL) automatically calls (via a supervisor request) a control storage wrap test routine when an unrecoverable BSC hardware error occurs. The wrap test routine checks the BSC control storage, communications adapter, and modem (if an IBM internal modem is used). Based on the results of this check, BSC displays an operator message indicating the cause of the error.

Error Logging

BSC logs all BSC errors in a BSC error history table. This table is a push-down stack that shows the last 21 temporary or permanent BSC errors that have occurred.

Error Retry Count

By using the SETB utility control statement or the ALTERBSC procedure, the BSC user can specify from 1 through 255 error retries to be performed before the BSC line is aborted when an error condition occurs. If an error retry count is not specified by the user, the error retry count defaults to 7. (The assembler user can also specify this parameter in the $DTFB macroinstruction.)

Message Display

BSC uses the SSP system log function to display information and error messages from the system library. There are two types of BSC messages: (1) messages that require no response, and (2) messages that require an operator response to continue processing.

Trace Option

The BSC user can request a trace of the BSC operation ends. Trace output can be used to diagnose problems and errors.

Permanent-Error Indicator (RPG II Only)

When a BSC permanent error occurs, a permanent-error indicator is set and the normal completion code (hex 40) is posted in the BSC DTF. The BSC user may then code an error recovery routine within the RPG II program to allow the program to continue processing without operator intervention.

DTF Completion Code (Basic Assembler Only)

The basic assembler user is notified of a permanent error by a completion code (other than hex 40) in the DTF ($BSCMP). The user can use this completion code to do error recovery.

SYSTEM REQUIREMENTS

System/34 BSC runs on any model of System/34 that has the communications adapter installed.

BSC requires 6,144 bytes of main storage. This does not include the transient routines, which are brought into the 2,048-byte system main storage transient area as needed. Space is also required in the system queue space for the BSC I/O area, which consists of the following:

- IOBs (maximum of 4, each with a length of 40 bytes)
- Switched IDs (maximum of 8 bytes)
- Physical I/O buffers (1 or 2 buffers, each with a maximum size of 4,096 bytes)

If switched IDs are specified via the OVERRIDE procedure (or via the SETR utility control statement), the IDs are moved into the BSC I/O area. If switched IDs are specified in the user program, the specified IDs are moved into a 32-byte area that #BSLO allocates within the system assign/free area.

Method of Operation

This section contains the logical functions of BSC using an overview (see Diagram 1) and an input-process-output diagram for each function.

For detailed information on input-process-output diagrams, see *Diagram Techniques* in *How to Use This Manual* at the front of this publication.

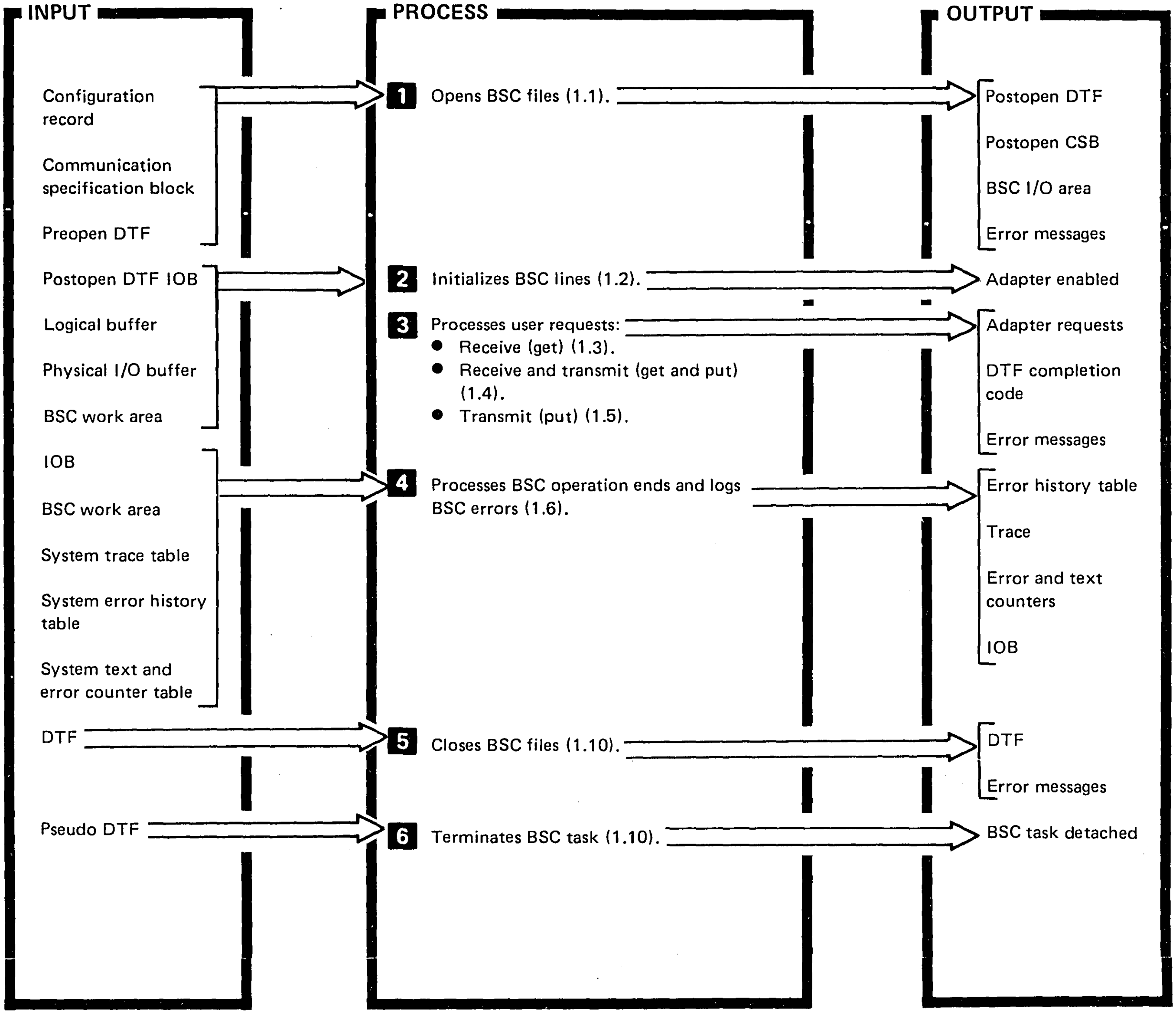
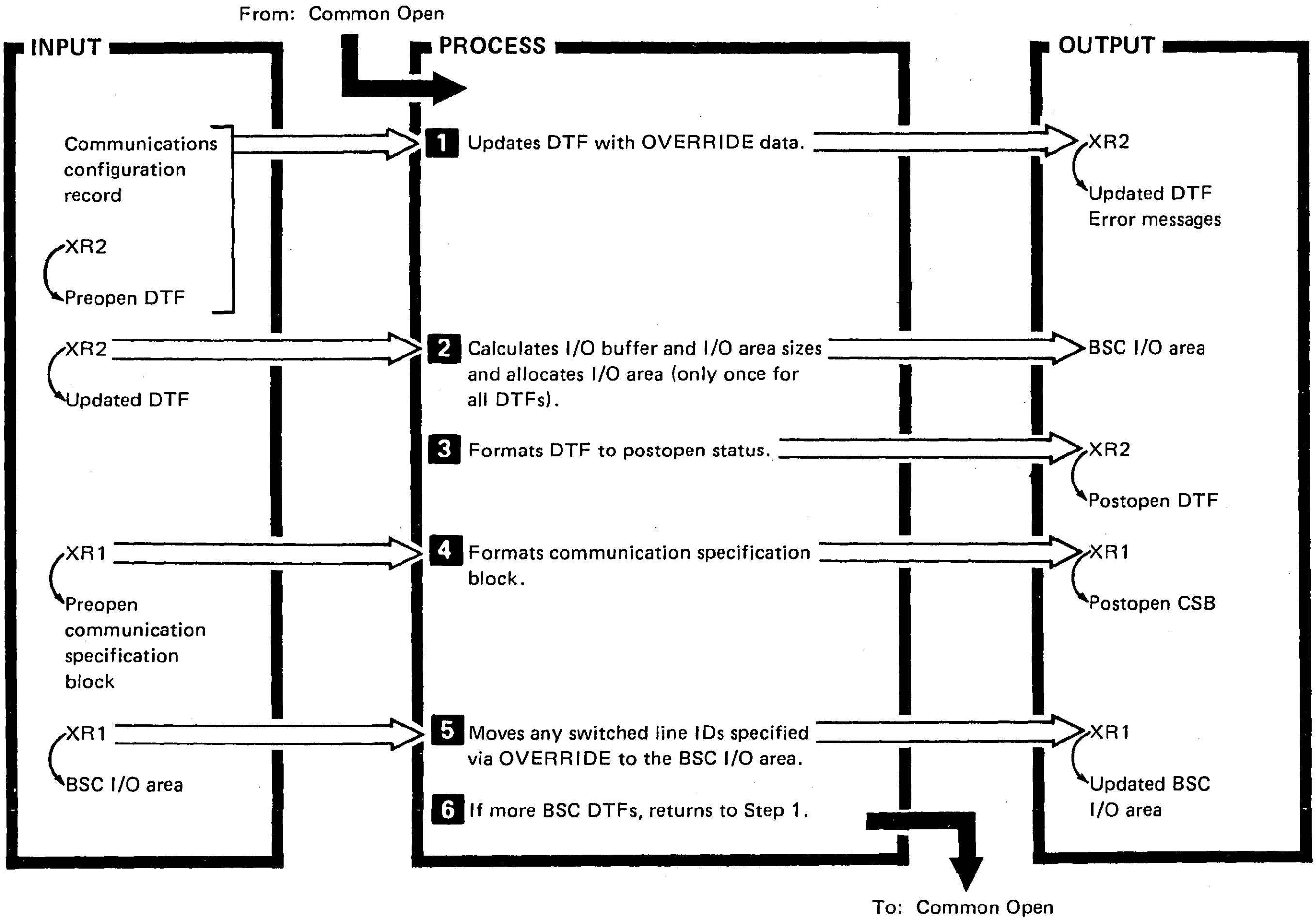


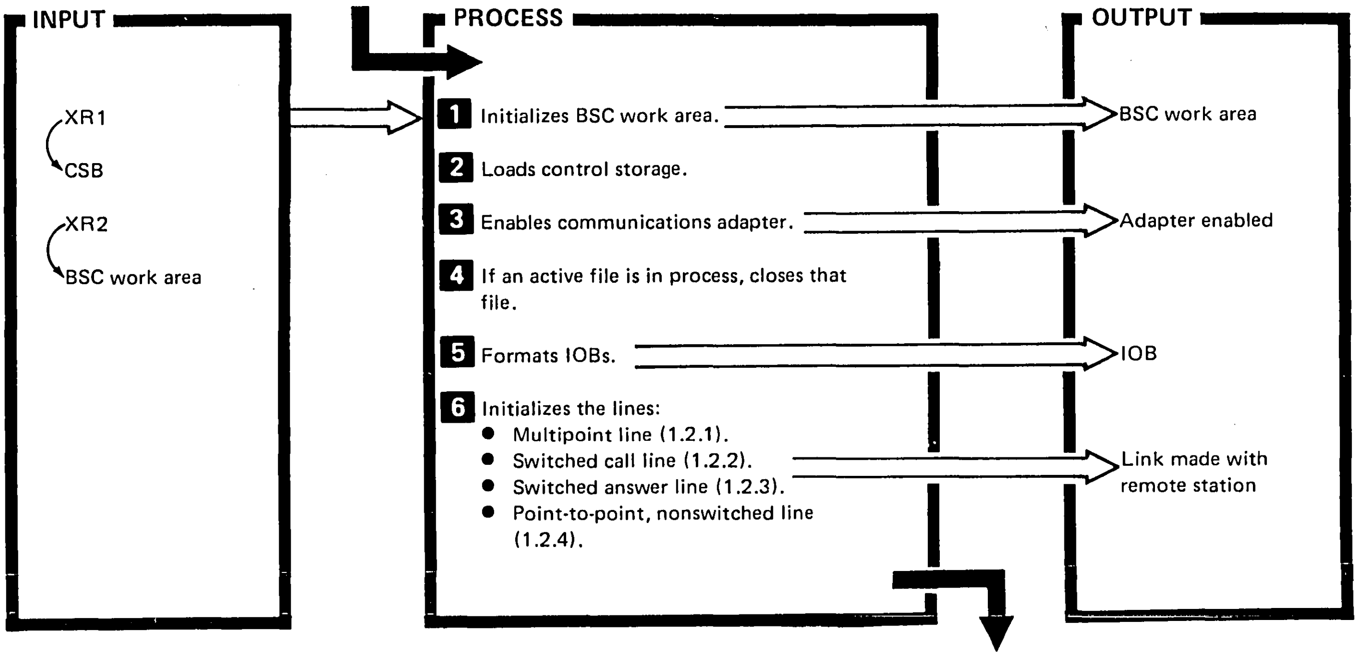
Diagram 1. BSC Overview


From: Common Open
INPUT
PROCESS
OUTPUT
Communications configuration record
XR2
Preopen DTF
1 Updates DTF with OVERRIDE data.
XR2
Updated DTF
Error messages
XR2
Updated DTF
2 Calculates I/O buffer and I/O area sizes and allocates I/O area (only once for all DTFs).
BSC I/O area
3 Formats DTF to postopen status.
XR2
Postopen DTF
XR1
Preopen communication specification block
4 Formats communication specification block.
XR1
Postopen CSB
XR1
BSC I/O area
5 Moves any switched line IDs specified via OVERRIDE to the BSC I/O area.
XR1
Updated BSC I/O area
6 If more BSC DTFs, returns to Step 1.
To: Common Open

DESCRIPTION	MODULE/ ROUTINE
1 Reads the communications configuration from disk.	#BSOB
2 • Determines the I/O buffer size from block size, record length, and ITB mode (if used) or record separator mode (if used). If the I/O buffer size is greater than 4,096 bytes, posts a 3301 error. (The size is saved in $BSDL.) • The BSC I/O area size is the total sizes of I/O buffer, IOBs, and switched IDs. Area is allocated from system queue space. If area cannot be allocated from system queue space, posts a 3288 error.	
3 • Sets normal completion code (hex 40) in $BSCMP. • Sets on file-opened flag ($BSFILOP) in $BSAT2.	
4 • Moves disk address on #BSL0 to CSBDLCS. • Sets CSBDLNDX to 0. • Sets I/O-area-allocated flag. • Sets CSB-opened flag. • Moves BSC I/O area address to CSB.	
5 If switched line IDs were specified via the OVERRIDE procedure (or the SETR utility control statement), moves them from the display station communications configuration record to the BSC I/O area.	
6 Chains forward through DTFs until a device code ($BSDEV) of hex 80 (BSC) is found and returns to Step 1, or exits if forward chain pointer ($BSCHB) equals hex FFFF.	

Diagram 1.1. BSC Open

From: Get and Put, 1.4, Step 1

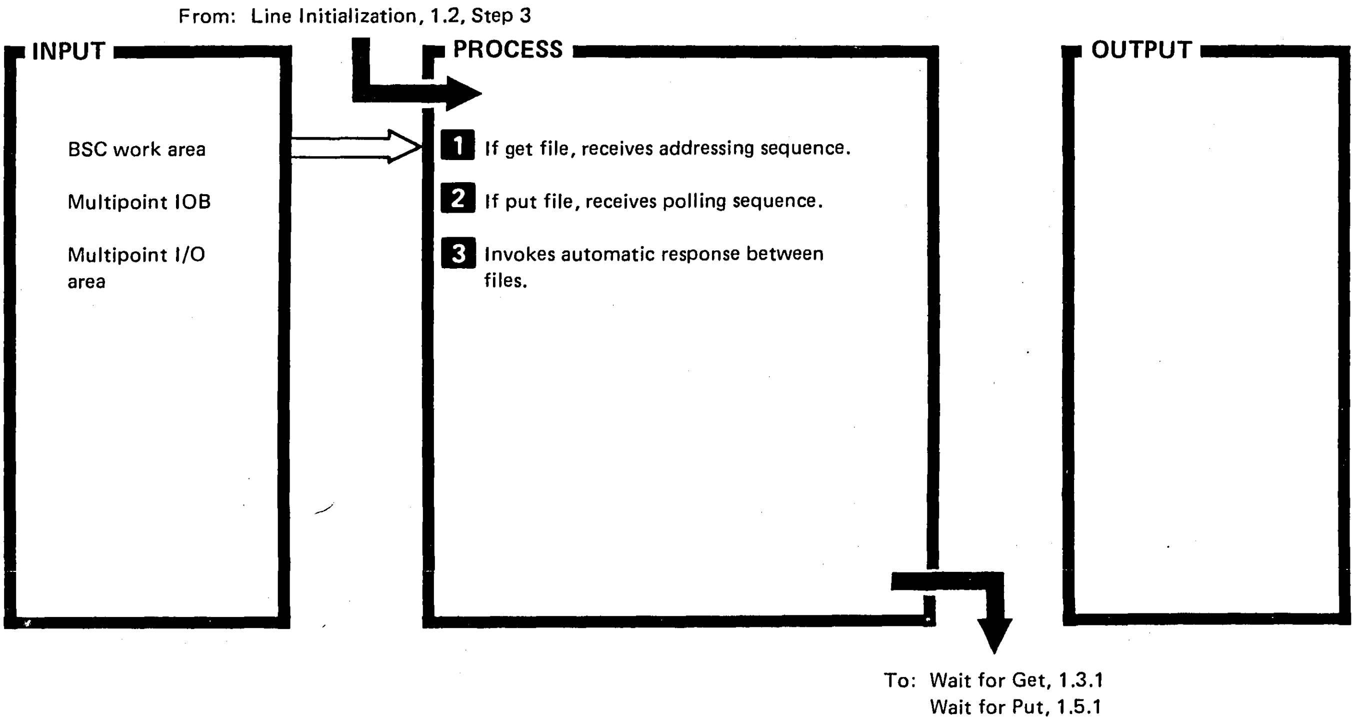


INPUT
XR1
CSB
XR2
BSC work area
PROCESS
1 Initializes BSC work area.
2 Loads control storage.
3 Enables communications adapter.
4 If an active file is in process, closes that file.
5 Formats IOBs.
6 Initializes the lines:
● Multipoint line (1.2.1).
● Switched call line (1.2.2).
● Switched answer line (1.2.3).
● Point-to-point, nonswitched line (1.2.4).
OUTPUT
BSC work area
Adapter enabled
IOB
Link made with remote station

To: Get and Put, 1.4, Step 2

DESCRIPTION	MODULE/ ROUTINE
1 Initializes all bytes of the BSC work area. Initializes the line control bytes to ASCII or EBCDIC.	#BSLO
2 Loads ASCII or EBCDIC BSC microcode into control storage.	
3 Issues enable SVC.	
4 Before a new file can be processed, closes the old active file and purges any data (if the file is a put file).	#BSCL
5 –	#BSLO
6 –	

Diagram 1.2. Line Initialization



From: Line Initialization, 1.2, Step 3

INPUT

BSC work area

Multipoint IOB

Multipoint I/O area

PROCESS

1 If get file, receives addressing sequence.

2 If put file, receives polling sequence.

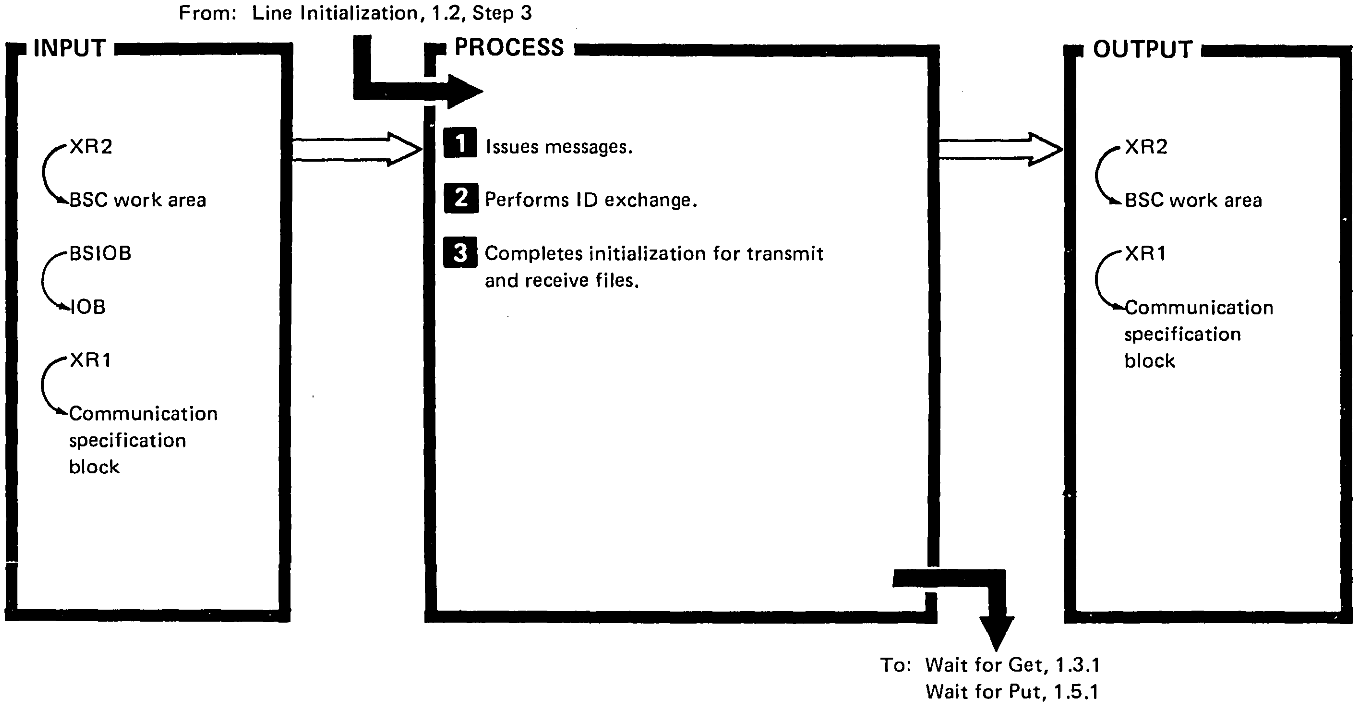
3 Invokes automatic response between files.

OUTPUT

To: Wait for Get, 1.3.1
Wait for Put, 1.5.1

	DESCRIPTION	MODULE/ ROUTINE
1	● For new file, determines whether receive-initial operation should be started or if automatic response is already in effect.	#BSRI
	● Sets up a receive-initial operation to receive an addressing sequence from the control station.	
	● Checks the validity of the response.	#BSCKRI
	● Sends ACK0 when a valid addressing sequence followed by ENQ is received.	
	● Sends EOT if poll is received.	
	● If invalid response, retries the operation until the error retry count is exceeded.	
2	● Sets up a receive-initial operation to receive a polling sequence from the control station.	#BSRI
	● Performs the receive-initial operation.	
	● Checks the validity of the response.	#BSCKRI
	● Starts a 2-second time-out when a valid polling sequence followed by ENQ is received.	
	● Sends RVI if ADDR sequence received.	
	● If invalid response, retries the operation until the error retry count is exceeded.	
3	● Starts receive-initial operation after EOT received.	#BSAUTO
	● Starts receive-initial operation after EOT transmitted for put file.	#BSCL
	● Starts receive-initial operation after EOT transmitted because of permanent error.	#BSMPEOT
	● Checks the validity of the response to receive-initial.	#BSCKRI
	● Transmits EOT to poll, and RVI or NAK to address with a transmit/receive-initial operation.	

Diagram 1.2.1. Multipoint Line Initialization



From: Line Initialization, 1.2, Step 3

INPUT

XR2 → BSC work area

BSIOB → IOB

XR1 → Communication specification block

PROCESS

1 Issues messages.

2 Performs ID exchange.

3 Completes initialization for transmit and receive files.

OUTPUT

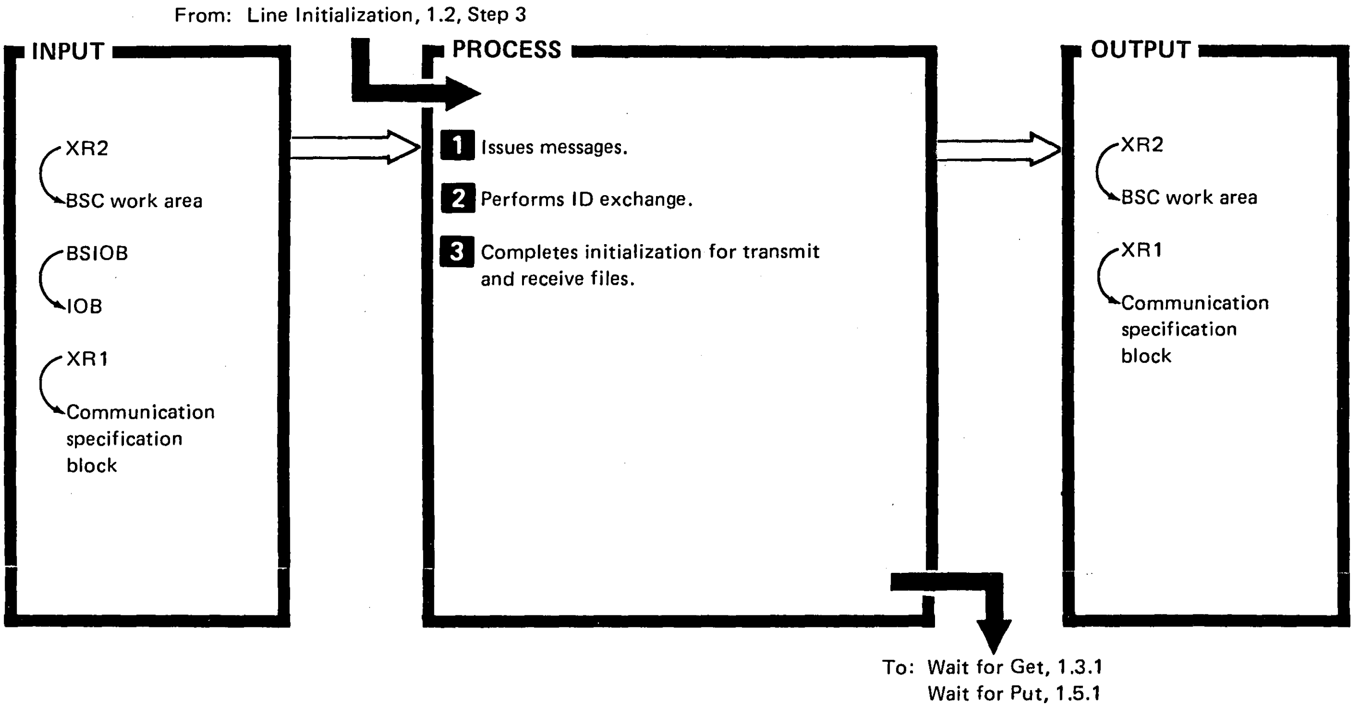
XR2 → BSC work area

XR1 → Communication specification block

To: Wait for Get, 1.3.1
Wait for Put, 1.5.1

DESCRIPTION	MODULE/ ROUTINE
1 Issues a message telling the operator to dial the required number and establishes a line connection.	#BSL0
2 ● Sets up ID field (if $BSTID indicates an ID is to be sent) and ENQ in the send buffer for a transmit/receive operation.	#BSL2
● Performs transmit/receive operation to exchange ID-ENQ sequence.	#BSIO
● If an ID is to be received, compares ID field received with $BSRID for validity.	#BSL2
3 ● If ACK0 is received following an ID exchange for a transmit file, starts a 2-second time-out and returns to the calling routine.	
● If ACK0 is received following an ID exchange for a receive file, sends EOT to release control of the line.	#BSIO
● Starts a receive-only operation after #BSIO sends EOT for a receive file.	#BSL4
● If the ACK0 response is not received for the receive operation, starts a 2-second time-out and retries the operation until the error retry count is exceeded.	#BSL2
● If the error retry count is exceeded, initiates an error message or sets the RPG II permanent-error indicator for the RPG II user.	#BSL0

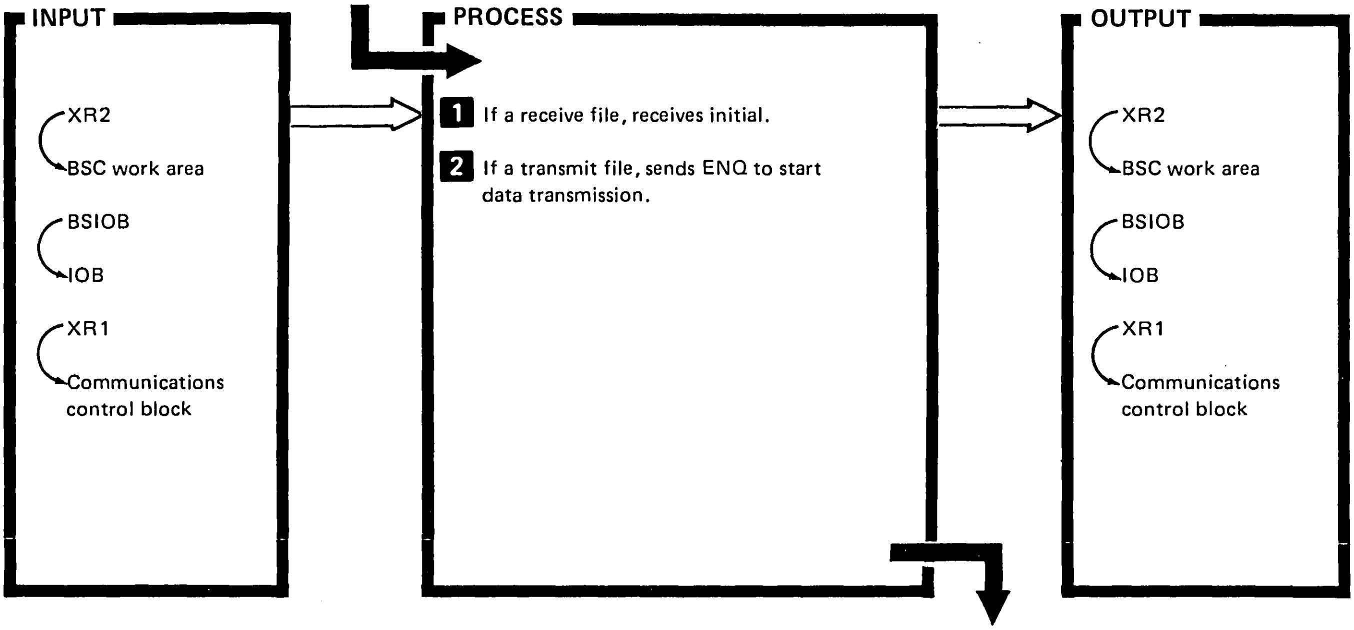
Diagram 1.2.2. Switched Call Line Initialization


From: Line Initialization, 1.2, Step 3
INPUT
XR2
BSC work area
BSIOB
IOB
XR1
Communication specification block
PROCESS
1 Issues messages.
2 Performs ID exchange.
3 Completes initialization for transmit and receive files.
OUTPUT
XR2
BSC work area
XR1
Communication specification block
To: Wait for Get, 1.3.1
Wait for Put, 1.5.1

DESCRIPTION	MODULE/ ROUTINE
1 Issues a message telling the operator to answer to establish the line connection.	#BSLO
2 ● Sets up a receive-initial operation to receive the remote station's ID (if used) and ENQ.	#BSL3
● Performs a receive-initial operation to receive the remote station's ID (if used) followed by ENQ.	#BSIO
● Determines if the remote station has sent ENQ preceded by an optional ID and checks the validity of the ID.	#BSL3
3 ● Sets up to send ACKO preceded by an optional ID if a valid ID-ENQ was received.	
● Sends ID-ACKO sequence. This completes initialization for a receive file.	#BSIO
● Receives EOT from the remote station for a transmit file.	#BSL3
● Sends ENQ if this is a transmit file and EOT was received from the remote station.	#BSIO
● Starts a 2-second time-out and returns to the calling routine if ACKO was the remote station's response to the ENQ for a transmit file. This completes line initialization for a transmit file.	#BSL3
● If the error retry count is exceeded, initiates an error message and disables the line or, for the RPG II user, sets the permanent-error indicator.	#BSLO

Diagram 1.2.3. Switched Answer Line Initialization

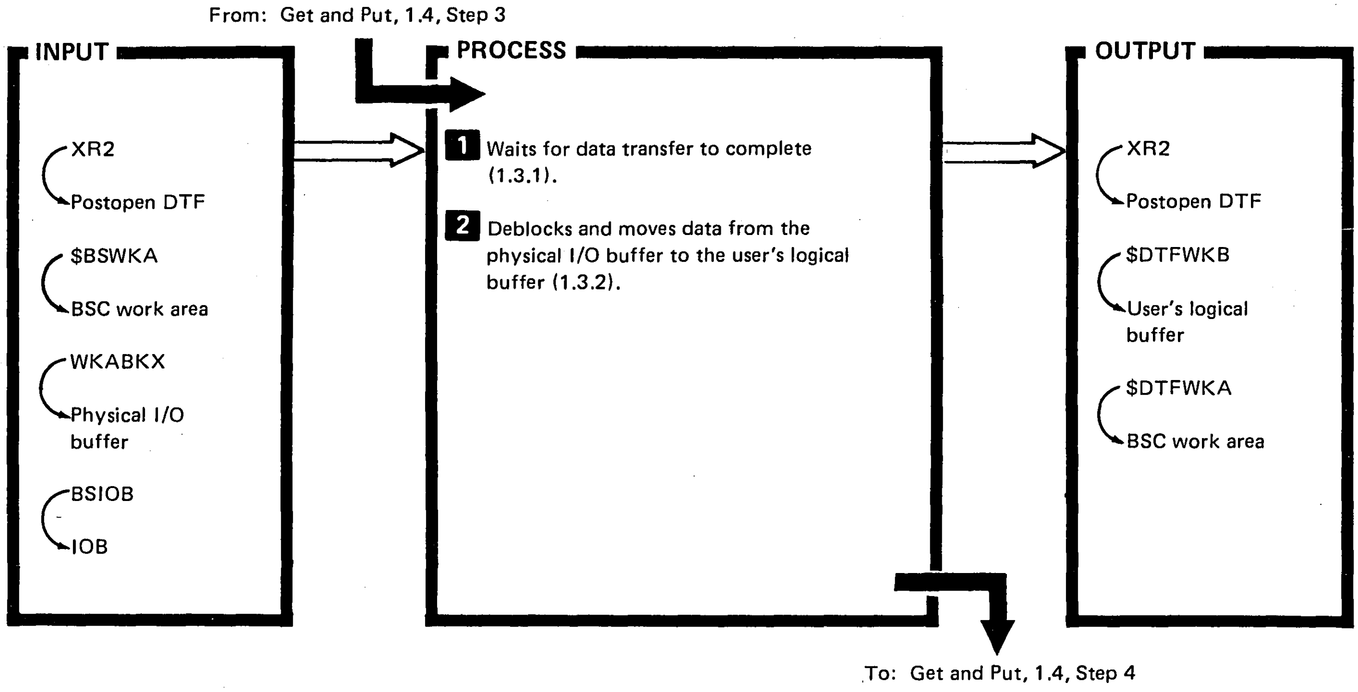
From: Line Initialization, 1.2, Step 3


INPUT
XR2
BSC work area
BSIOB
IOB
XR1
Communications control block
PROCESS
1 If a receive file, receives initial.
2 If a transmit file, sends ENQ to start data transmission.
OUTPUT
XR2
BSC work area
BSIOB
IOB
XR1
Communications control block

To: Line Initialization, 1.2

DESCRIPTION	MODULE/ ROUTINE
1 ● Sets up a receive-initial operation to receive an ENQ.	#BSL4
● Performs the receive-initial operation.	#BSIO
● Determines if an ENQ was received.	#BSL4
● Sends an ACK0 response to ENQ.	#BSIO
● Returns to the calling routine.	#BSL4
● If an ENQ was not received, retries the operation until the error retry count is exceeded.	
● If the retries fail, initiates an error message or sets the RPG II permanent-error indicator.	#BSL0
2 ● Sends ENQ.	#BSIO
● Checks for an ACK0 response to ENQ.	#BSL4
● If ACK0 is received, starts a 2-second time-out and returns to the calling routine.	
● If ACK0 is not received, retries the operation until the error retry count is exceeded.	
● If the retries fail, initiates an error message or, for the RPG II user, sets the permanent-error indicator.	#BSL0

Diagram 1.2.4. Point-to-Point, Nonswitched Line Initialization

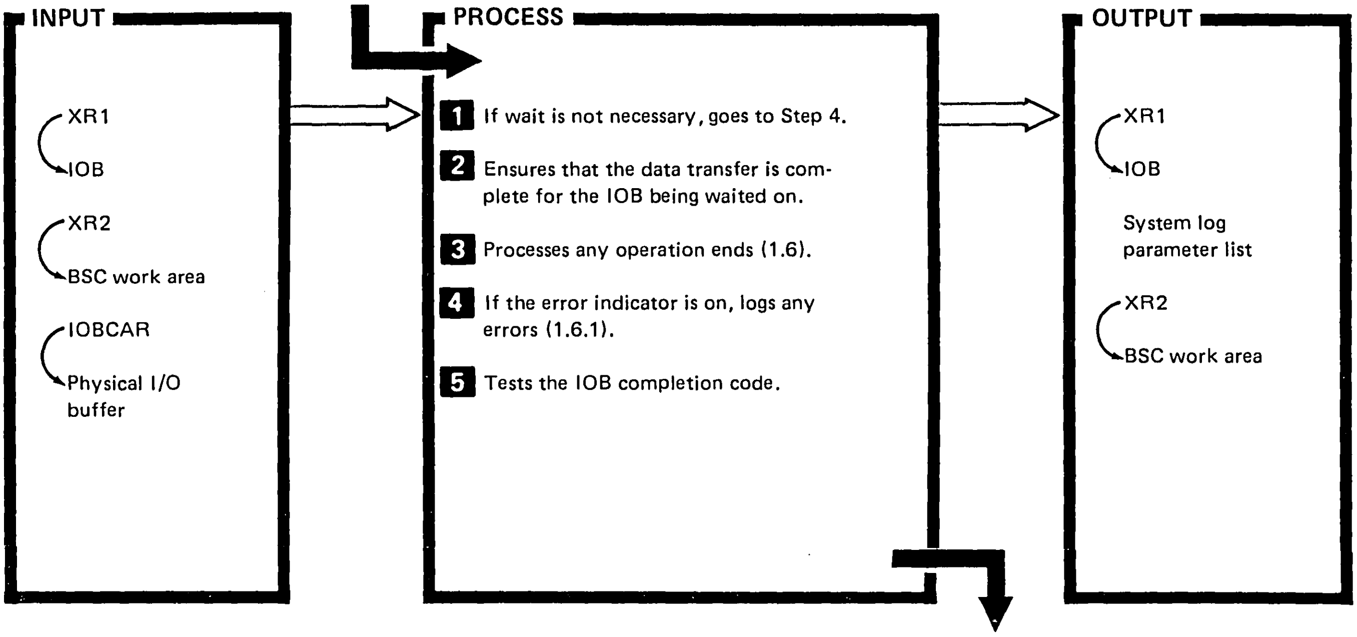


1 Waits for data transfer to complete (1.3.1).

2 Deblocks and moves data from the physical I/O buffer to the user's logical buffer (1.3.2).

Diagram 1.3. Get

From: Get, 1.3, Step 1



INPUT

XR1 → IOB

XR2 → BSC work area

IOBCAR → Physical I/O buffer

PROCESS

1 If wait is not necessary, goes to Step 4.

2 Ensures that the data transfer is complete for the IOB being waited on.

3 Processes any operation ends (1.6).

4 If the error indicator is on, logs any errors (1.6.1).

5 Tests the IOB completion code.

OUTPUT

XR1 → IOB

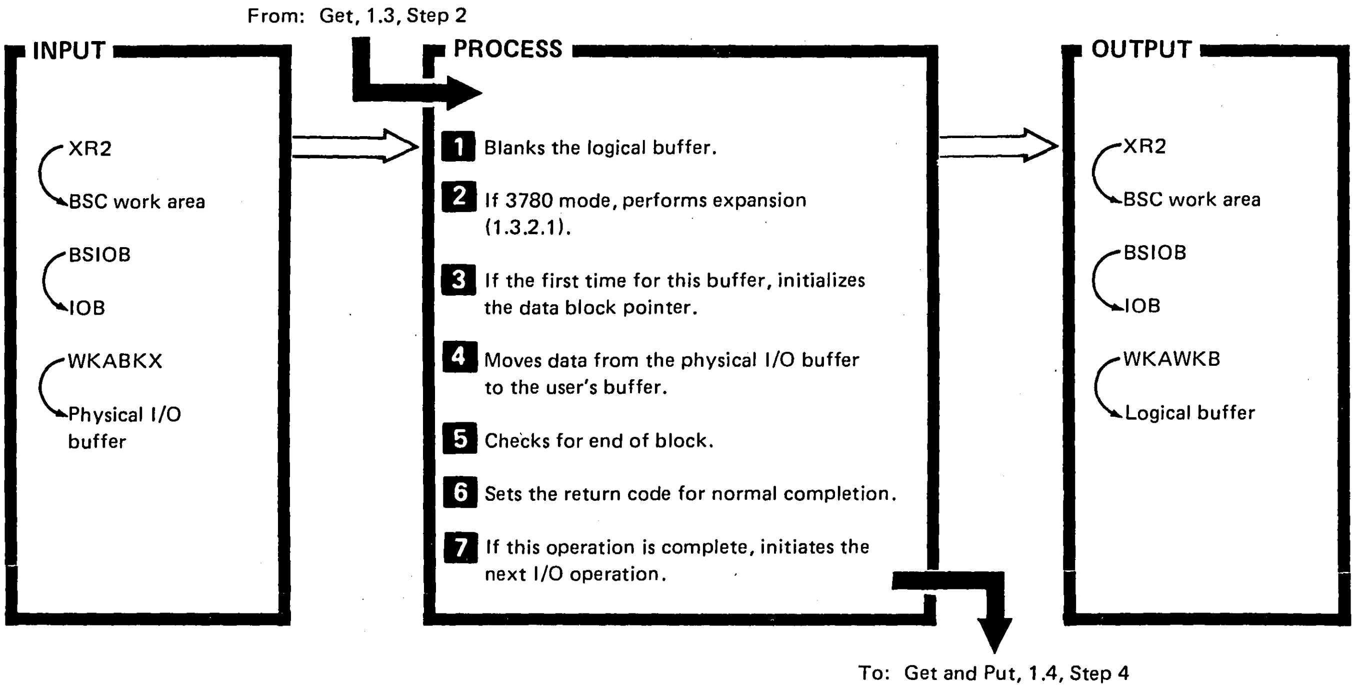
System log parameter list

XR2 → BSC work area

To: Get, 1.3, Step 2

DESCRIPTION	MODULE/ ROUTINE
1 If a permanent error has occurred or a data transfer is not in progress (IOBCMP<84), goes to Step 4.	#BSIW
2 Issues a wait to ensure that data transfer is complete. IOBCAR is the address of the last byte received. IOBRLN is the length of the data received.	
3 –	#BSHB
4 Checks BSFL3D in the work area to see if an error should be logged.	#BSIW
5 If IOBCMP equals hex 40 or hex 42, returns to the caller. Otherwise, displays an operator message and sets the permanent error indicator for the RPG user.	

Diagram 1.3.1. Wait for Get


From: Get, 1.3, Step 2

INPUT

XR2
BSC work area

BSIOB
IOB

WKABKX
Physical I/O buffer

PROCESS

1 Blanks the logical buffer.

2 If 3780 mode, performs expansion (1.3.2.1).

3 If the first time for this buffer, initializes the data block pointer.

4 Moves data from the physical I/O buffer to the user's buffer.

5 Checks for end of block.

6 Sets the return code for normal completion.

7 If this operation is complete, initiates the next I/O operation.

OUTPUT

XR2
BSC work area

BSIOB
IOB

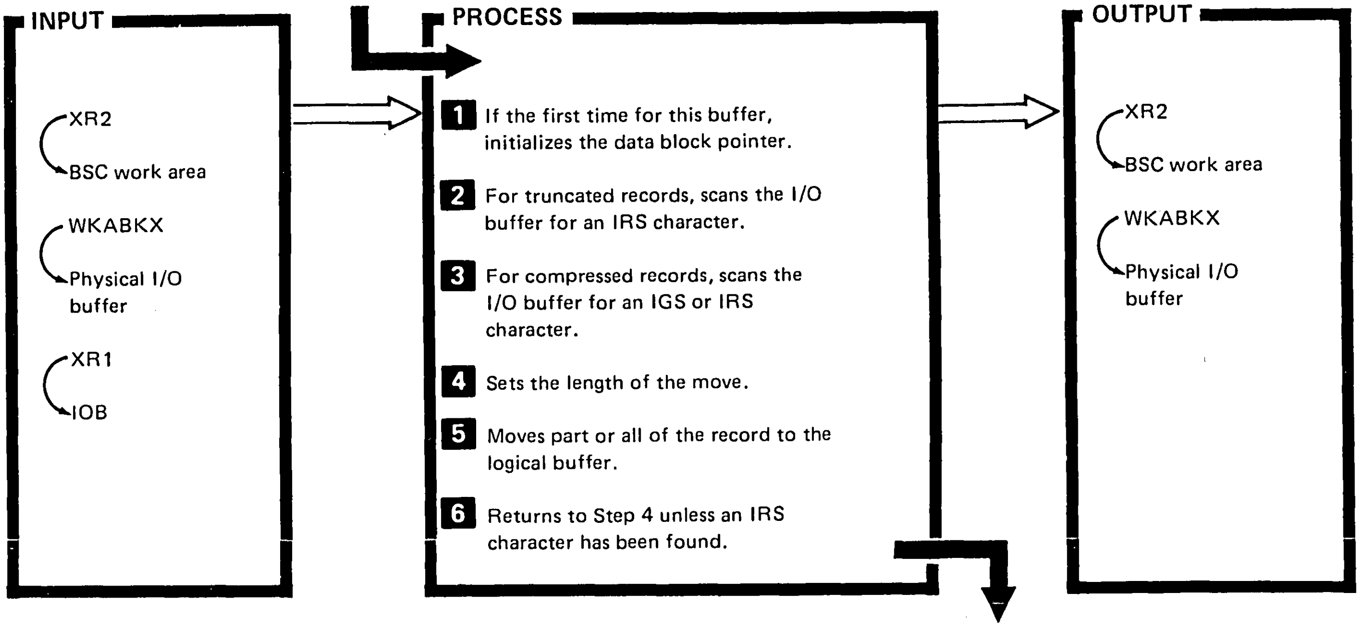
WKAWKB
Logical buffer

To: Get and Put, 1.4, Step 4

DESCRIPTION	MODULE/ ROUTINE
1 –	#BSMG
2 –	
3 –	
4 WKAOPC indicates the direction of the move, and WKARCL indicates the length of the move.	#BSMO
5 Checks for end of block to determine if this operation is complete.	
6 Sets IOBCMP field in the IOB to hex 40.	
7 Cancels 2-second time-out if line is not busy.	

Diagram 1.3.2. Data Deblock

From: Data Deblock, 1.3.2, Step 1



INPUT

XR2 → BSC work area

WKABKX → Physical I/O buffer

XR1 → IOB

PROCESS

1 If the first time for this buffer, initializes the data block pointer.

2 For truncated records, scans the I/O buffer for an IRS character.

3 For compressed records, scans the I/O buffer for an IGS or IRS character.

4 Sets the length of the move.

5 Moves part or all of the record to the logical buffer.

6 Returns to Step 4 unless an IRS character has been found.

OUTPUT

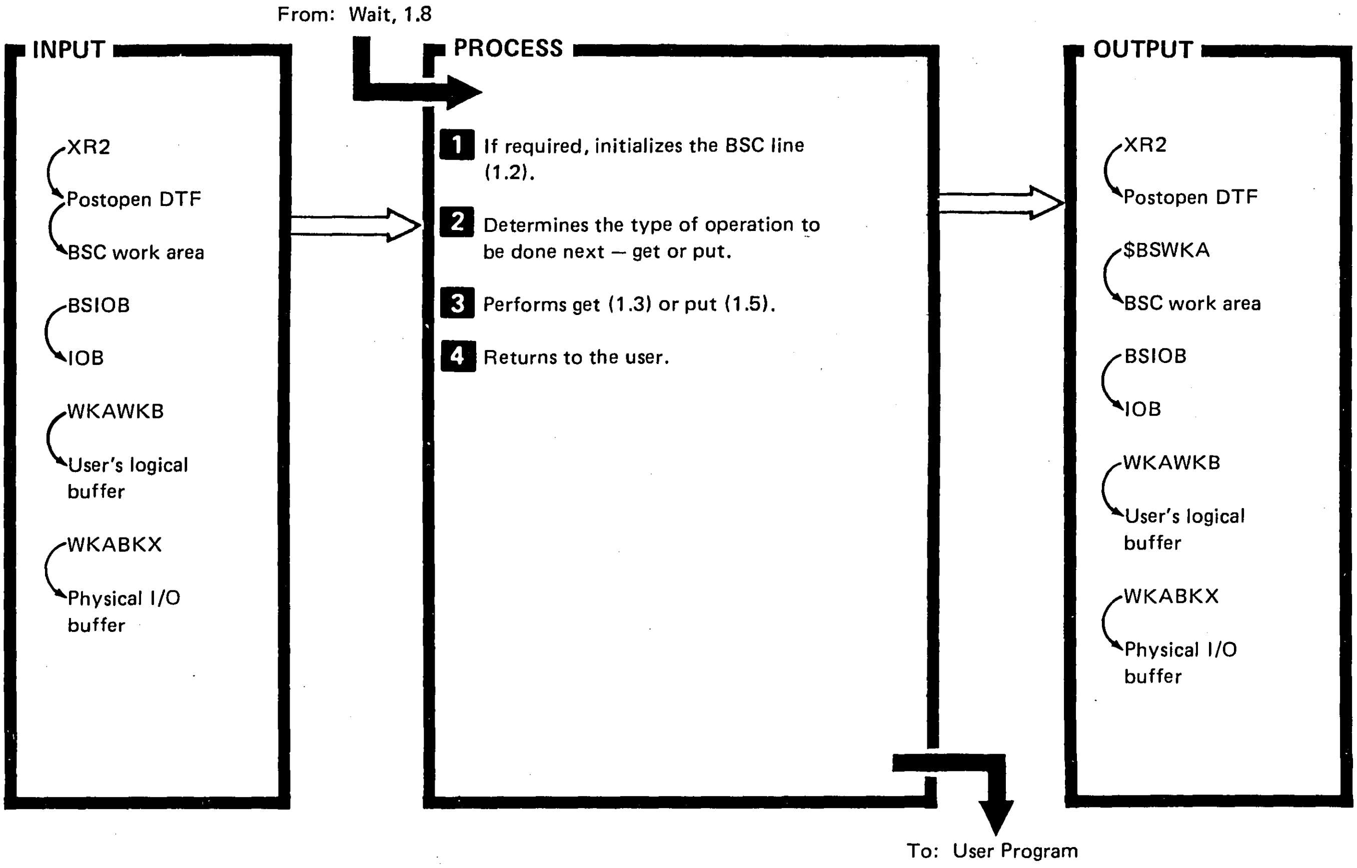
XR2 → BSC work area

WKABKX → Physical I/O buffer

To: Data Deblock, 1.3.2, Step 4

DESCRIPTION	MODULE/ ROUTINE
1 –	#BSXP
2 –	
3 –	
4 Sets WKAERL to the length of the data to be moved from the I/O buffer to the logical buffer.	
5 WKAOPC indicates the direction of the move, and WKAERL indicates the length of the move.	#BSMO
6 –	#BSXP

Diagram 1.3.2.1. Expansion



From: Wait, 1.8

INPUT

XR2 → Postopen DTF → BSC work area

BSIOB → IOB

WKAWKB → User's logical buffer

WKABKX → Physical I/O buffer

PROCESS

1 If required, initializes the BSC line (1.2).

2 Determines the type of operation to be done next — get or put.

3 Performs get (1.3) or put (1.5).

4 Returns to the user.

OUTPUT

XR2 → Postopen DTF

$BSWKA → BSC work area

BSIOB → IOB

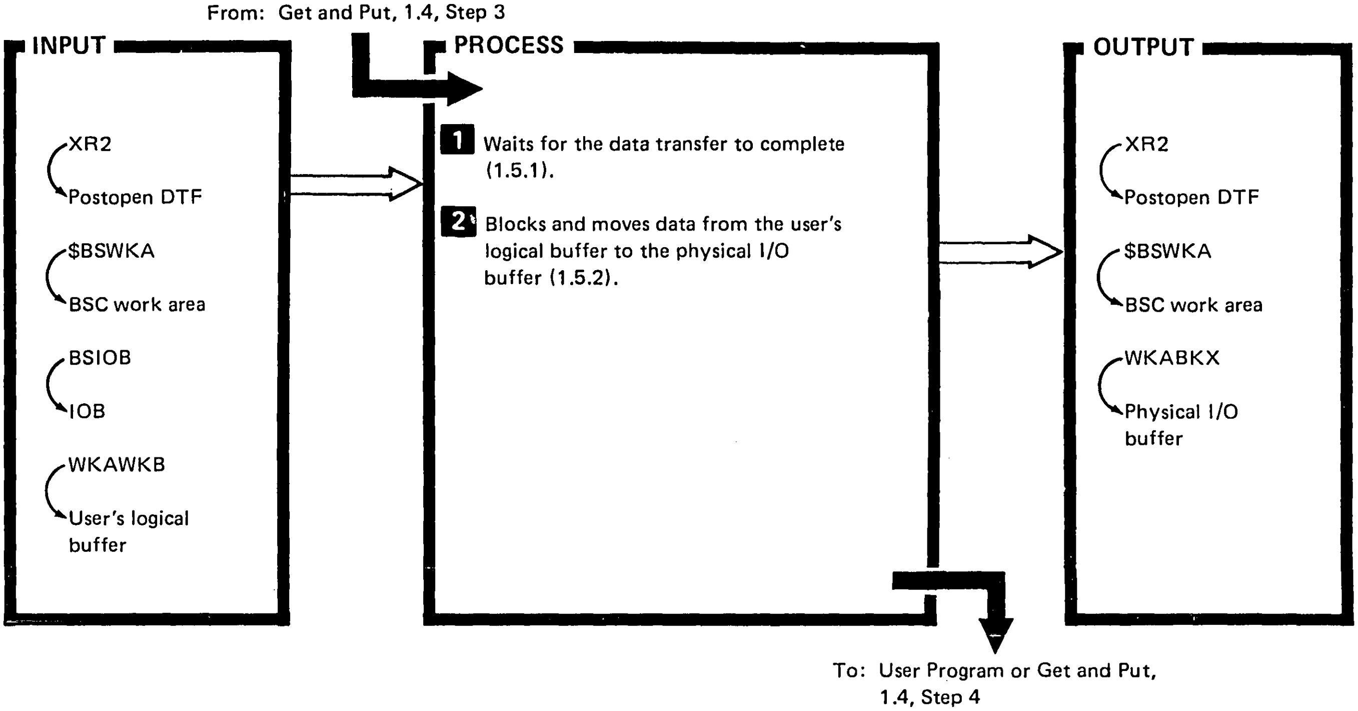
WKAWKB → User's logical buffer

WKABKX → Physical I/O buffer

To: User Program

DESCRIPTION	MODULE/ ROUTINE
1 –	#BSLO
2 Checks value of WKAAT1 to determine whether a put or get operation is to be done next.	#BSDB
3 –	#BSMP
4 Posts system intertask routine (#SVTTC), which exits back to the user program.	#BSPOST

Diagram 1.4. Get and Put

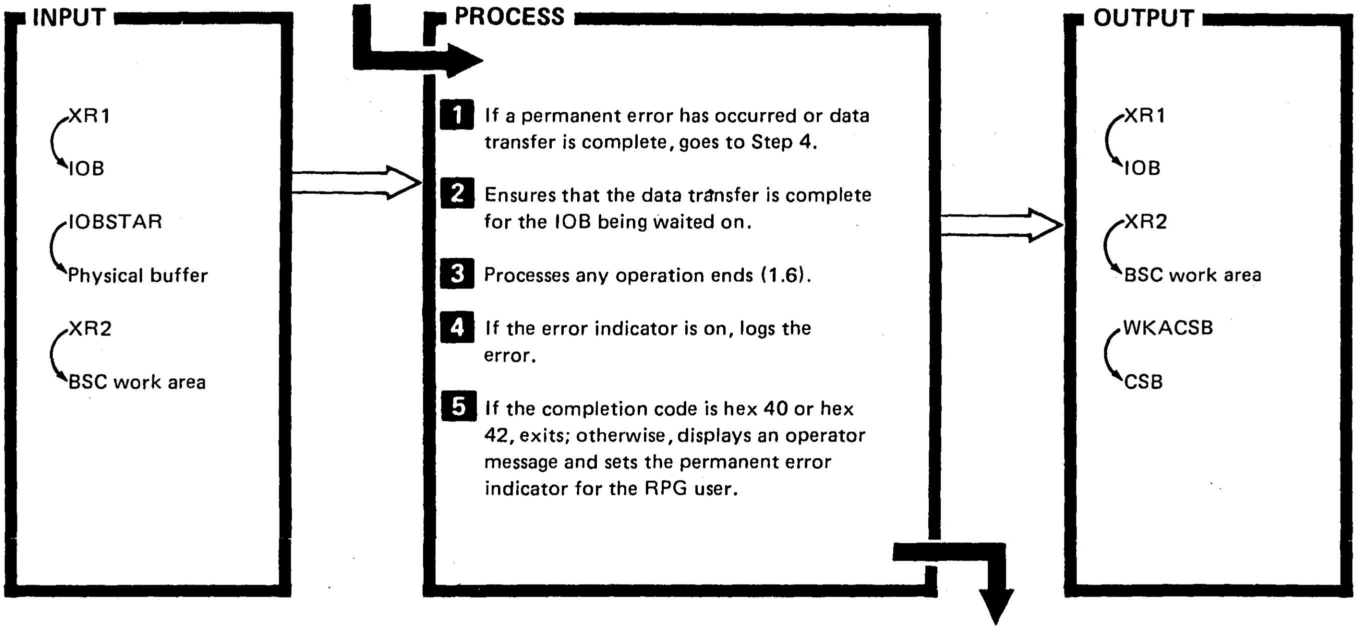


1. Waits for the data transfer to complete (1.5.1).

2. Blocks and moves data from the user's logical buffer to the physical I/O buffer (1.5.2).

Diagram 1.5. Put

From: Put, 1.5, Step 1



1 If a permanent error has occurred or data transfer is complete, goes to Step 4.

2 Ensures that the data transfer is complete for the IOB being waited on.

3 Processes any operation ends (1.6).

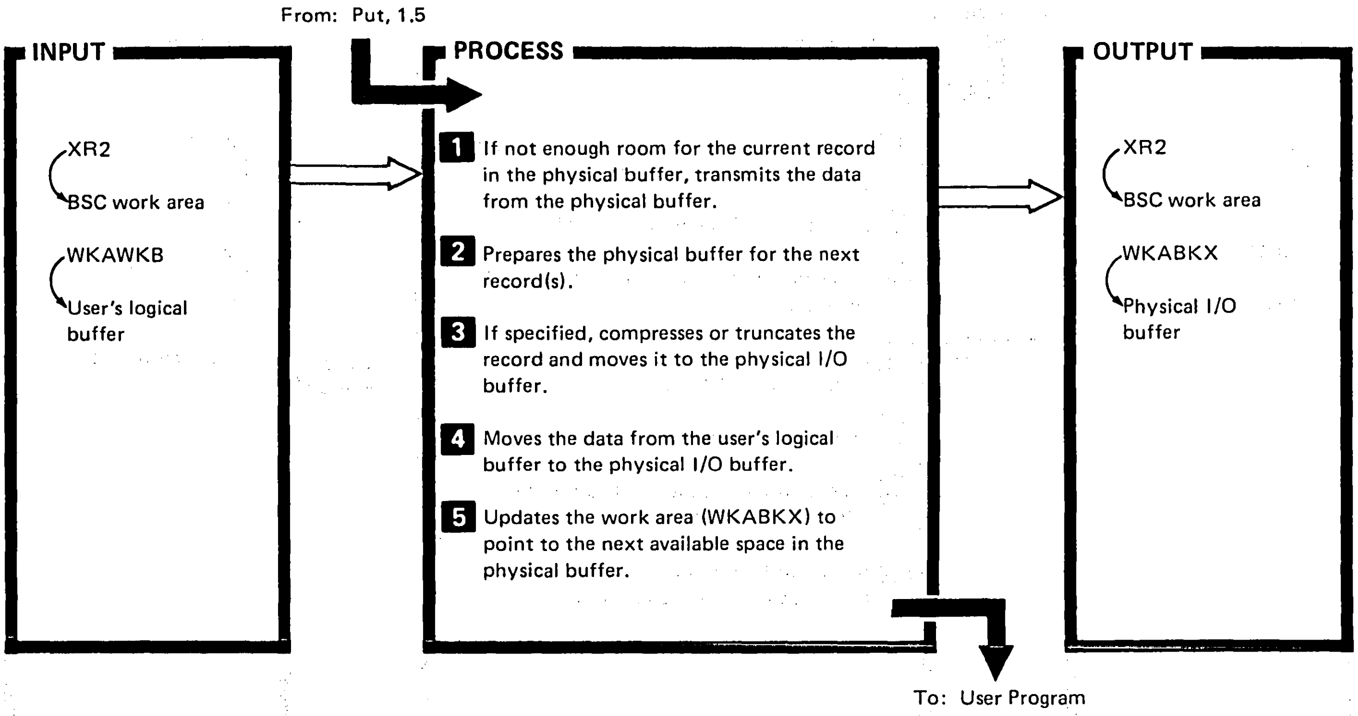
4 If the error indicator is on, logs the error.

5 If the completion code is hex 40 or hex 42, exits; otherwise, displays an operator message and sets the permanent error indicator for the RPG user.

To: Put, 1.5, Step 2

DESCRIPTION	MODULE/ ROUTINE
1 –	#BSIW
2 Issues wait to ensure that the data transfer is complete.	
3 –	
4 If LOGPROC in the BSC work area is on, logs an error in the BSC error history table.	
5 –	

Diagram 1.5.1. Wait for Put



From: Put, 1.5

INPUT

XR2
BSC work area

WKAWKB
User's logical buffer

PROCESS

1 If not enough room for the current record in the physical buffer, transmits the data from the physical buffer.

2 Prepares the physical buffer for the next record(s).

3 If specified, compresses or truncates the record and moves it to the physical I/O buffer.

4 Moves the data from the user's logical buffer to the physical I/O buffer.

5 Updates the work area (WKABKX) to point to the next available space in the physical buffer.

OUTPUT

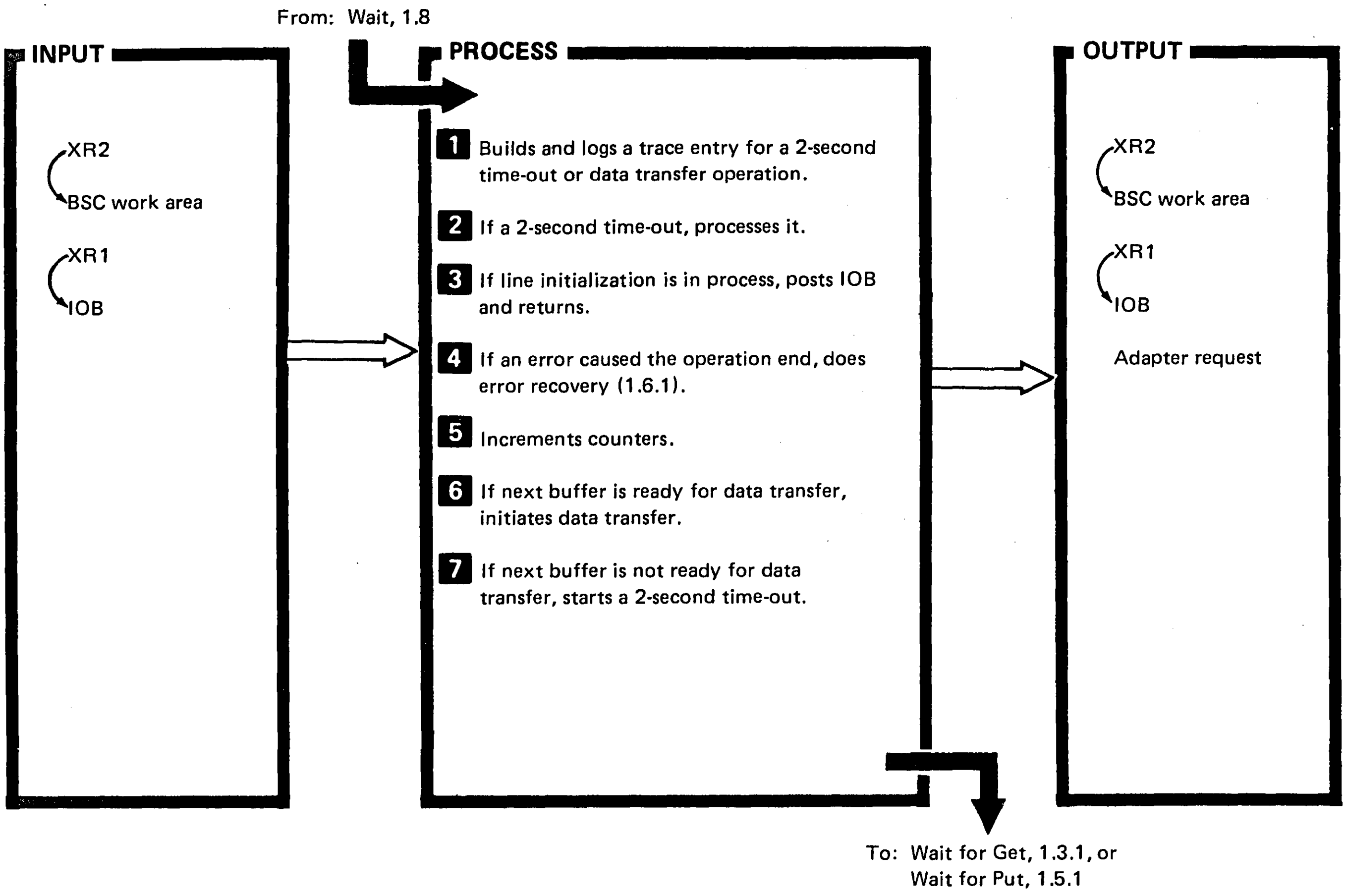
XR2
BSC work area

WKABKX
Physical I/O buffer

To: User Program

DESCRIPTION	MODULE/ ROUTINE
1 Uses information from the work area and IOB to determine if the physical buffer can hold the current record.	#BSMP
2 Sets WKABKX to the start of the physical buffer. Moves STX or DLE-STX to the start of the buffer. (DLE is used for transparent EBCDIC.)	
3 ● For compression, replaces two or more blanks with IGS and blank count. ● For truncation, omits all trailing blanks from the end of the record.	#BSCP
● Inserts IRS at the end of the record. ● Moves data from the logical buffer to the physical buffer.	#BSMP
4 WKAOPC indicates the direction of the move, and WKARCL indicates the length of the move.	
5 –	

Diagram 1.5.2. Data Block



From: Wait, 1.8

INPUT

XR2 → BSC work area

XR1 → IOB

PROCESS

1 Builds and logs a trace entry for a 2-second time-out or data transfer operation.

2 If a 2-second time-out, processes it.

3 If line initialization is in process, posts IOB and returns.

4 If an error caused the operation end, does error recovery (1.6.1).

5 Increments counters.

6 If next buffer is ready for data transfer, initiates data transfer.

7 If next buffer is not ready for data transfer, starts a 2-second time-out.

OUTPUT

XR2 → BSC work area

XR1 → IOB

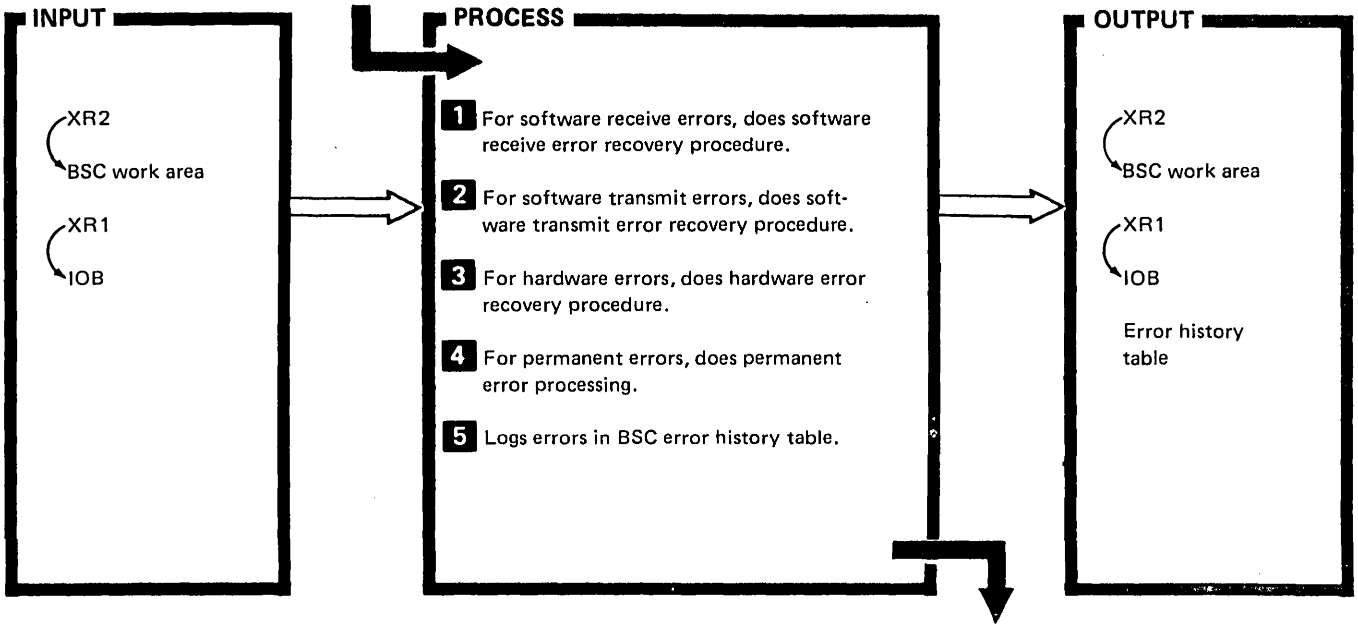
Adapter request

To: Wait for Get, 1.3.1, or
Wait for Put, 1.5.1

DESCRIPTION	MODULE/ ROUTINE
1 ● If a 2-second time-out, builds sense and Q- and R-bytes in work area (TRCSNS, TRCQR). Sets transmit and receive bytes to hex FF. ● For transmit or receive operations, saves first and last 3 bytes transmitted and first and last 2 bytes received. For receive-initial or receive-only operations, saves only receive bytes. ● Uses system trace facility to log operation end.	#BSHB
2 For a receive 2-second time-out, sends WACK; for a transmit 2-second time-out, sends TTD.	
3 If line initialization is in process, posts IOBCMP with normal completion (hex 40) and returns.	
4 –	
5 Increments BSLTBTFJ for transmit files or BSLTBRFJ for receive files. These will be written to the text and error counter table on disk.	
6 –	
7 –	

Diagram 1.6. Processing Operation Ends

From: Processing Operation Ends, 1.6, Step 4



INPUT

XR2 → BSC work area

XR1 → IOB

PROCESS

1. For software receive errors, does software receive error recovery procedure.
2. For software transmit errors, does software transmit error recovery procedure.
3. For hardware errors, does hardware error recovery procedure.
4. For permanent errors, does permanent error processing.
5. Logs errors in BSC error history table.

OUTPUT

XR2 → BSC work area

XR1 → IOB

Error history table

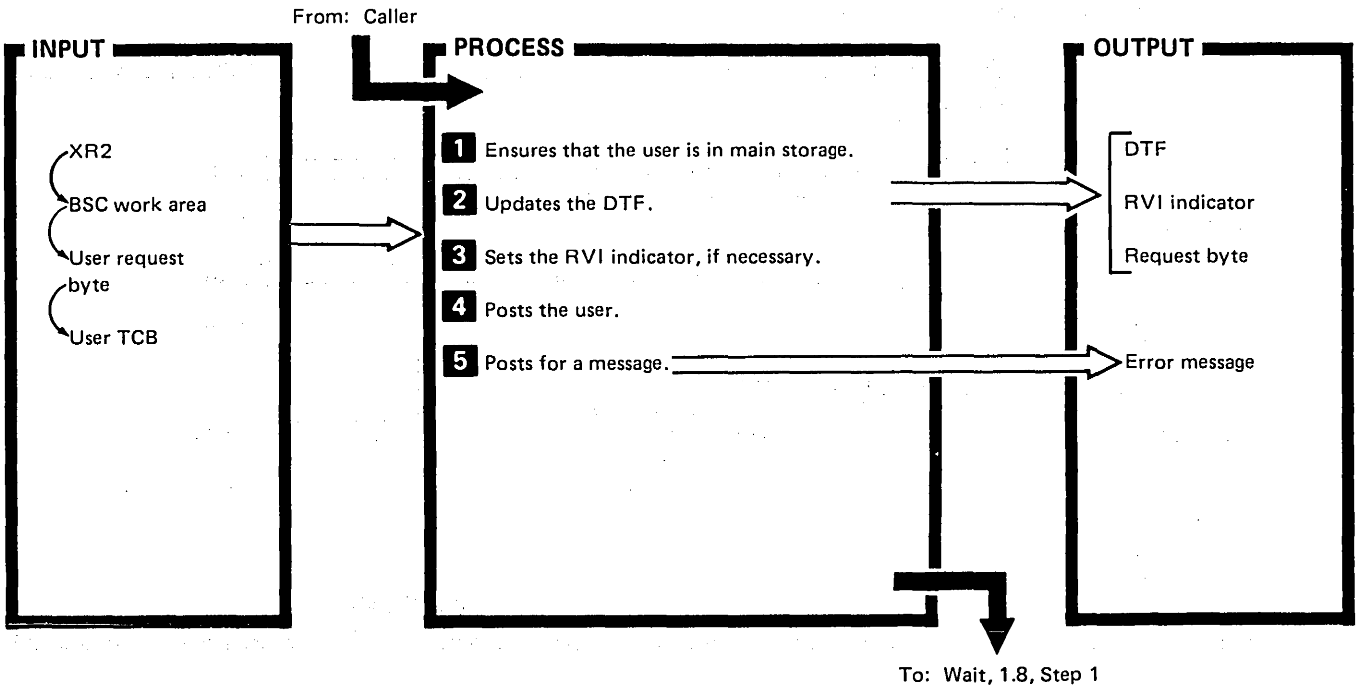
To: Processing Operation Ends, 1.6, Step 5

DESCRIPTION	MODULE/ ROUTINE
1 ● When DISC is received, sets off BSC active indicator in the work area (BSFLGD). When preceded by TTD, posts a permanent error in the BSC work area (BSLBSCCC) and increments the forward-abort-received counter (BSLFARFS). If not preceded by TTD, posts BSLBSCCC with EOF and resets the interrupt. ● When an EOT is received that is preceded by a TTD, posts a permanent error in the BSC work area (BSLBSCCC) and increments the forward-abort-received counter (BSLFARFS). When an EOT is received that is not preceded by a TTD, posts BSLBSCCC with EOF and resets the interrupt. ● For an invalid response (a response other than ENQ as the last character received), retries the operation. ● When ENQ is received in response to WACK, initiates data transfer or starts a 2-second time-out. ● Responds NAK to TTD.	#BSHB
2 ● When an invalid ACK is received, sends ENQ. If the response is another invalid ACK, retransmits text. ● If NAK is received in response to TTD (forward abort is in progress), posts a permanent error in IOBCMP and sends EOT. When NAK is received in response to TTD, sends the next block of text or starts a 2-second time-out. When NAK is the response to text transmitted, retransmits the block of text. ● If WACK is received, sends ENQ. ● When RVI is received, sets RVIRCV flag and continues with successful text transmission processing. ● When EOT is received, posts a permanent error in the BSC work area (BSLBSCCC) and increments the abort-received counter in the work area (BSLABTFS). ● When disconnect is received, sets the adapter-inactive indicator in the work area, posts lost connection in the BSC work area, and increments the abort-received counter in the work area (BSLABTFJ). ● When ENQ is received, sends ENQ to an invalid response of ENQ. ● Any other invalid response is not recognized and ENQ is sent.	

Diagram 1.6.1 (Part 1 of 2). Error Recovery Procedures

DESCRIPTION	MODULE/ ROUTINE
3 ● For an ASCII error, posts a permanent error in the BSC work area. ● When a transmit-adapter check has occurred, increments the transmit-adapter-check counter (BSLACTFJ) in the BSC work area, posts the error in the work area (BSLBSCCC), and sends ENQ. When a receive adapter check has occurred, increments the receive-adapter check counter (BSLACRFT) in the BSC work area, posts the error in the work area (BSLBSCCC), and does a receive-only operation. ● For a lost connection, increments the lost-connection counter (BSLDTOFJ) in the BSC work area and posts the error in the work area (PBLBSCCC). ● For receive time-out errors, increments the receive time-out counter (BSLRTOFJ) in the BSC work area and sets up the IOB to do a receive-only operation. ● When a data check has occurred, increments the data-check counter (BSLDCKFS) in the BSC work area (BSLPSCCC), and sends NAK.	
4 ● Builds the error history table list. ● If BSC is active for receive files, sends EOT or DISC. For transmit files, sends forward abort sequence: TTD followed by EOT or DISC. ● Disables the adapter when going inactive. ● Posts IOB (IOBCMP) with error condition from the BSC work area (BSLBSCCC).	
5 The following bytes are recorded: ● Q-byte ● R-byte ● Sense byte ● Error retry count ● Completion code ● Multipoint tributary station address	

Diagram 1.6.1 (Part 2 of 2). Error Recovery Procedures



1 Ensures that the user is in main storage.

2 Updates the DTF.

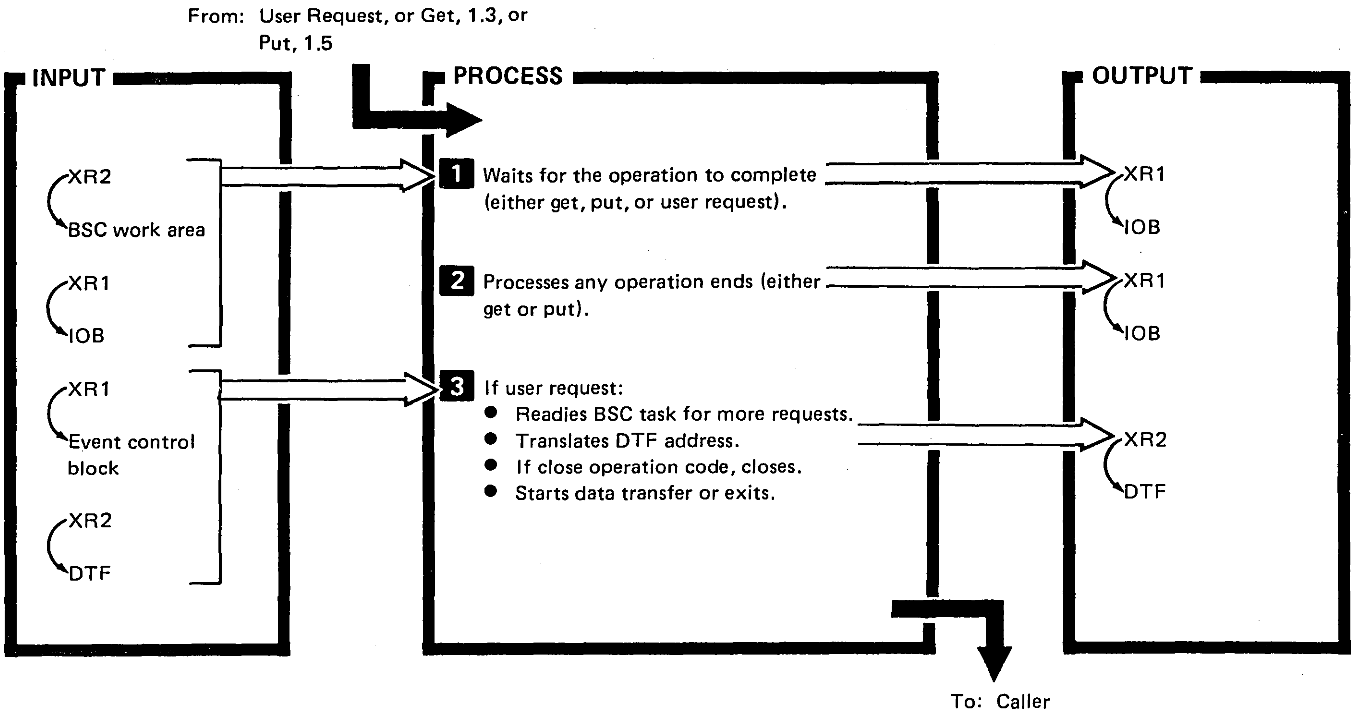
3 Sets the RVI indicator, if necessary.

4 Posts the user.

5 Posts for a message.

DESCRIPTION	MODULE/ ROUTINE
1 Posts the system task-to-task interface routine (#SVTTC) with completion code indicating that control should be returned to the BSC task.	#BSIW
2 Moves DTF bytes from the BSC work area to the BSC DTF.	
3 If the operation end processor (#BSHB) has received RVI and set the RVI indicator in the work area, translates the RVI indicator address and sets the indicator.	
4 Sets completion code indicating return to user, and posts the system task-to-task interface routine (#SVTTC).	
5 If a BSC error has occurred, posts the system task-to-task interface routine (#SVTTC) indicating that the BSC message transient (#BSMD) should be called.	

Diagram 1.7. Post



1 Waits for the operation to complete (either get, put, or user request).

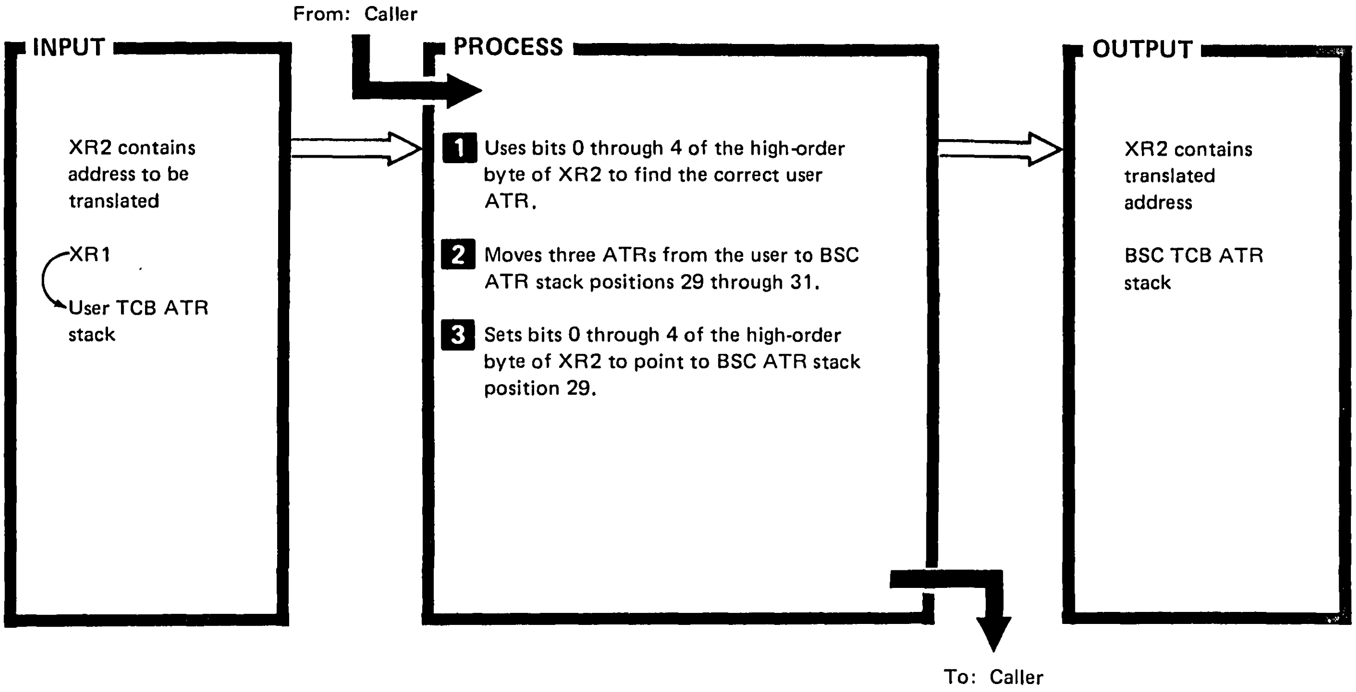
2 Processes any operation ends (either get or put).

3 If user request:

- Readies BSC task for more requests.
- Translates DTF address.
- If close operation code, closes.
- Starts data transfer or exits.

DESCRIPTION	MODULE/ ROUTINE
1 For a user request, returns control after the wait with XR1 pointing to the BSC TCB.	#BSIW
2 –	
3 ● Issues PIQ to allow more user requests. ● Logical address of DTF must be translated to a BSC task address.	
● If the DTF operation code is close or terminate, closes the file or terminates the BSC task. ● If this request is from the user, processes the request; if this request is due to a BSC-initiated action (such as posting an error message), returns to the calling program.	#BSCL

Diagram 1.8. Wait



From: Caller

INPUT

XR2 contains address to be translated

XR1

User TCB ATR stack

PROCESS

1 Uses bits 0 through 4 of the high-order byte of XR2 to find the correct user ATR.

2 Moves three ATRs from the user to BSC ATR stack positions 29 through 31.

3 Sets bits 0 through 4 of the high-order byte of XR2 to point to BSC ATR stack position 29.

OUTPUT

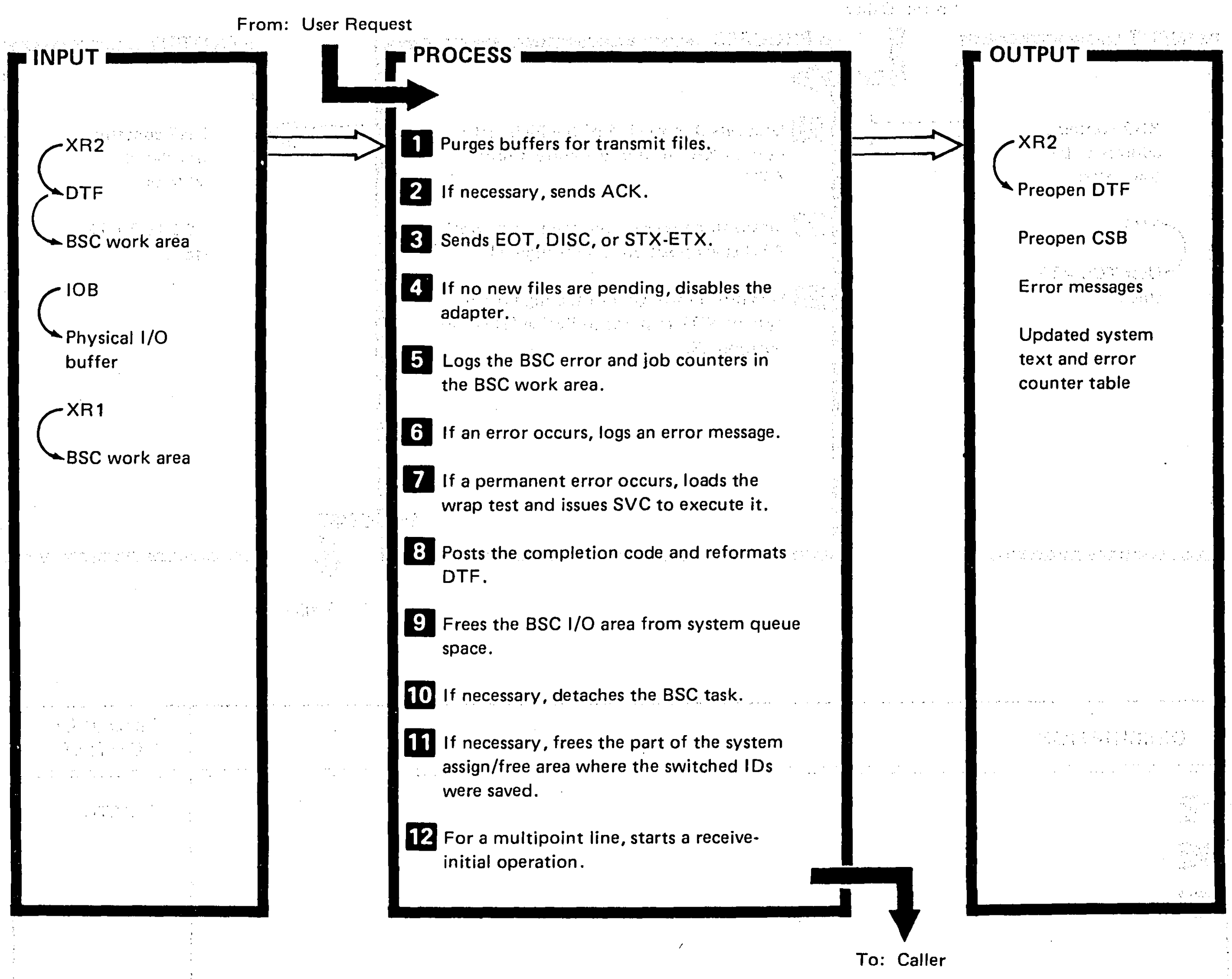
XR2 contains translated address

BSC TCB ATR stack

To: Caller

DESCRIPTION	MODULE/ ROUTINE
1 –	#BSATR
2 –	
3 –	

Diagram 1.9. ATR Translation



From: User Request

INPUT

- XR2 → DTF
- DTF → BSC work area
- IOB → Physical I/O buffer
- XR1 → BSC work area

PROCESS

1 Purges buffers for transmit files.

2 If necessary, sends ACK.

3 Sends EOT, DISC, or STX-ETX.

4 If no new files are pending, disables the adapter.

5 Logs the BSC error and job counters in the BSC work area.

6 If an error occurs, logs an error message.

7 If a permanent error occurs, loads the wrap test and issues SVC to execute it.

8 Posts the completion code and reformats DTF.

9 Frees the BSC I/O area from system queue space.

10 If necessary, detaches the BSC task.

11 If necessary, frees the part of the system assign/free area where the switched IDs were saved.

12 For a multipoint line, starts a receive-initial operation.

OUTPUT

- XR2 → Preopen DTF
- Preopen CSB
- Error messages
- Updated system text and error counter table

To: Caller

DESCRIPTION	MODULE/ ROUTINE
Note: If BSC close is being called because of user or task termination, executes only Steps 4, 5, 9, and 10.	
1 If file being closed is a get file, file pending is a get file, and 3740 multiple file support is being used, then goes to Step 4.	#BSCL
2 If the file being closed is a get file, 3740 multiple file support is being used, and either no new files are pending or the pending file is a put file, then transmits ACK.	
3 ● If no new files are pending and the file being closed is on a nonswitched line, or if a new file is pending, then transmits EOT. ● If no new file on a switched line is being closed, then transmits DISC. ● If the file pending is a put file, file being closed is a put file, and 3740 multiple file support is being used, then transmits STX-ETX (null record).	
4 –	

Diagram 1.10 (Part 1 of 2). BSC Close/Terminate

DESCRIPTION	MODULE/ ROUTINE
5 Stores job counters in BSC text and error counter table on disk using control storage transient. Control storage transient also adds these counters to cumulative counters.	
6 –	
7 –	
8 –	
9 –	
10 If $BSOPC in DTF is 0, terminates the BSC task.	
11 –	
12 If an EOT is to be transmitted for a transmit file and if a new file is pending or the user issued a put EOF, transmits EOT with a transmit/receive-initial operation before returning to the caller.	

Diagram 1.10 (Part 2 of 2). BSC Close/Terminate

Program Organization

This section includes:

- A storage map indicating the layout of the program
- A description of each module within the program
- A description of the control flow between the modules

BSC STORAGE MAP

The BSC mainline code, which requires 6K bytes, is loaded into the user portion of main storage at the time the user allocates the BSC files. Because the nucleus can vary and the BSC mainline can be broken into 2K segments, the storage map shown in Figure 1-2 is only a logical representation of how BSC appears in main storage.

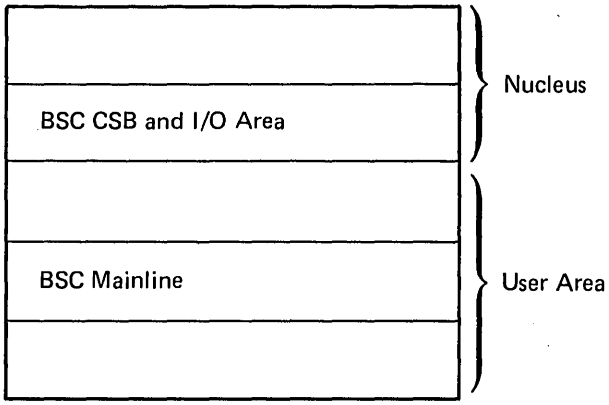
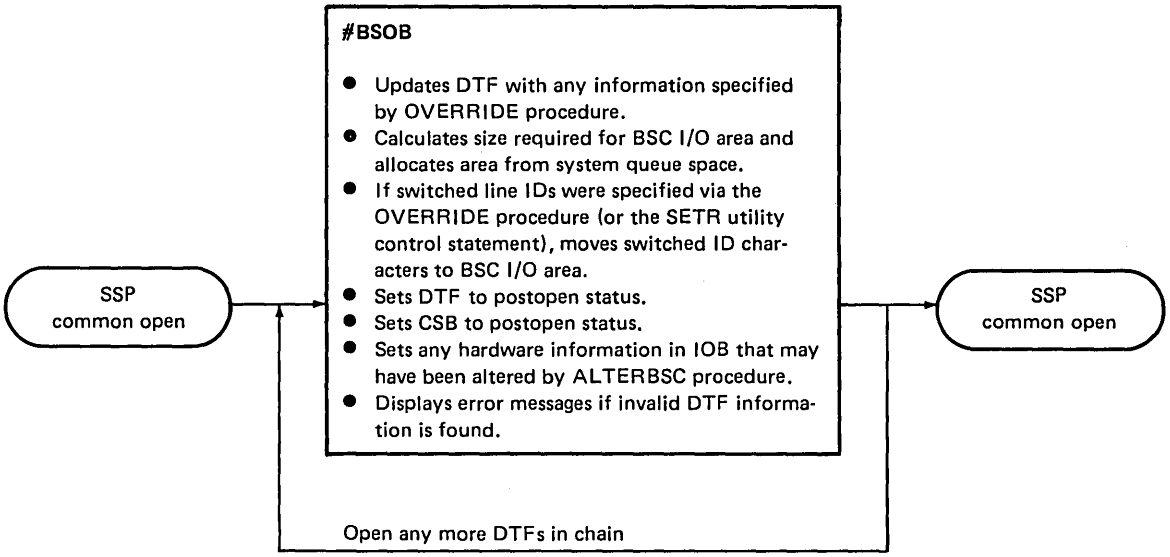


Figure 1-2. BSC Storage Map

CONTROL FLOW DIAGRAMS

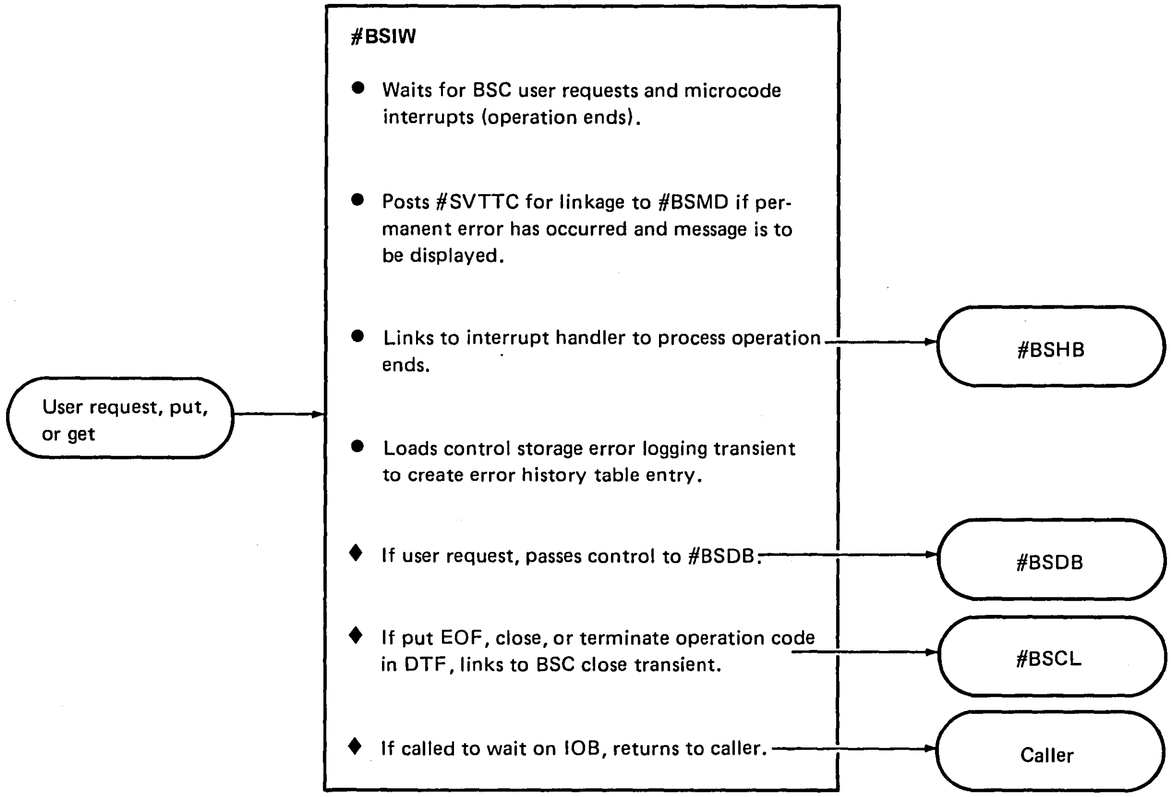
Figures 1-3 through 1-16 are control flow diagrams that describe each module within the program and the control flow between the modules. A bullet (●) indicates a process step and a diamond (♦) indicates a decision.



#BSOB

- Updates DTF with any information specified by OVERRIDE procedure.
- Calculates size required for BSC I/O area and allocates area from system queue space.
- If switched line IDs were specified via the OVERRIDE procedure (or the SETR utility control statement), moves switched ID characters to BSC I/O area.
- Sets DTF to postopen status.
- Sets CSB to postopen status.
- Sets any hardware information in IOB that may have been altered by ALTERBSC procedure.
- Displays error messages if invalid DTF information is found.

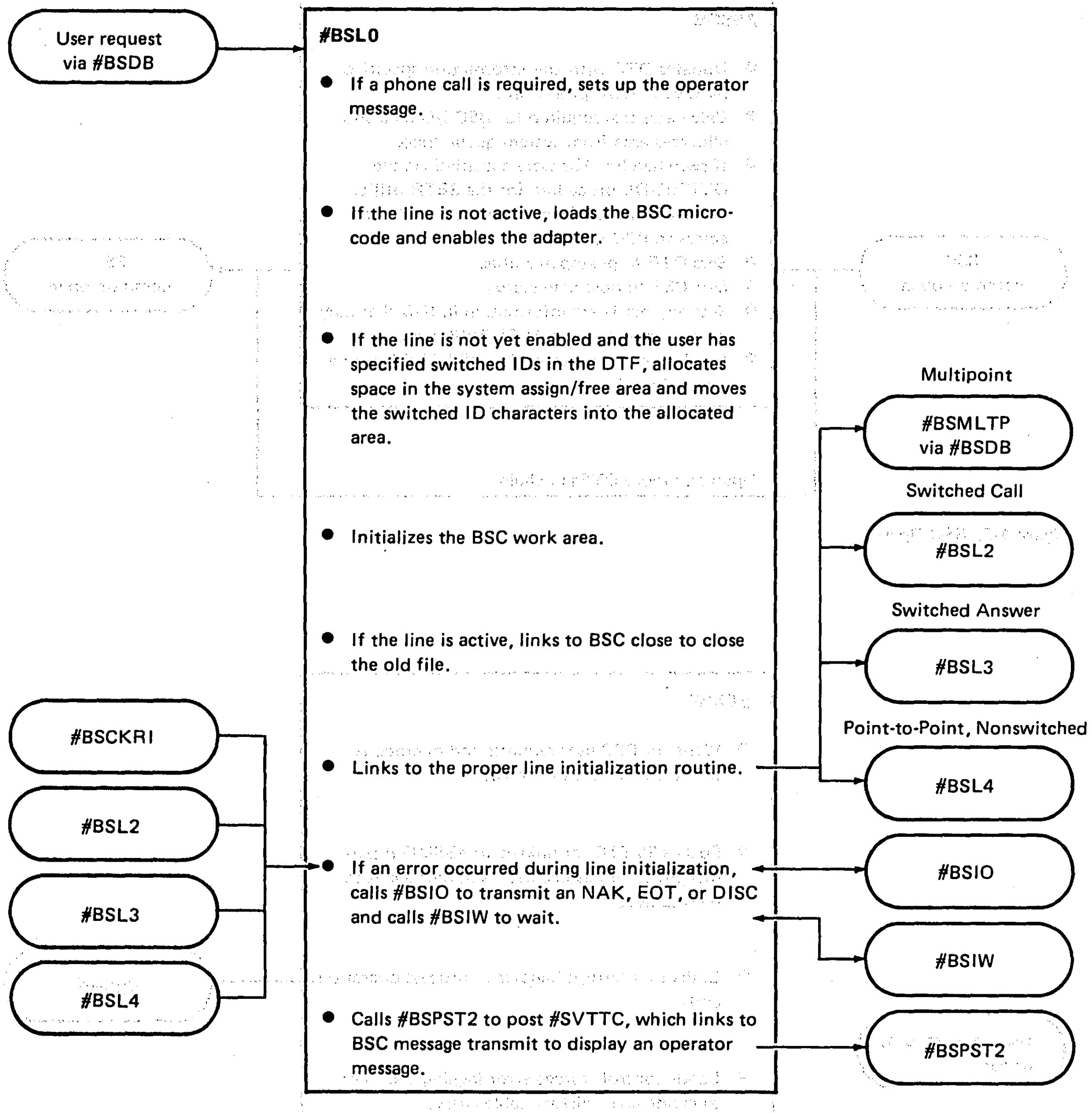
Figure 1-3. BSC Open



#BSIW

- ● Waits for BSC user requests and microcode interrupts (operation ends).
- ● Posts #SVTTC for linkage to #BSMD if permanent error has occurred and message is to be displayed.
- ● Links to interrupt handler to process operation ends.
- ● Loads control storage error logging transient to create error history table entry.
- ♦ If user request, passes control to #BSDB.
- ♦ If put EOF, close, or terminate operation code in DTF, links to BSC close transient.
- ♦ If called to wait on IOB, returns to caller.

Figure 1-4. BSC Wait



#BSL0

- If a phone call is required, sets up the operator message.
- If the line is not active, loads the BSC microcode and enables the adapter.
- If the line is not yet enabled and the user has specified switched IDs in the DTF, allocates space in the system assign/free area and moves the switched ID characters into the allocated area.
- Initializes the BSC work area.
- If the line is active, links to BSC close to close the old file.
- Links to the proper line initialization routine.
- If an error occurred during line initialization, calls #BSIO to transmit an NAK, EOT, or DISC and calls #BSIW to wait.
- Calls #BSPST2 to post #SVTTC, which links to BSC message transmit to display an operator message.

Figure 1-5. Line Initialization Mainline

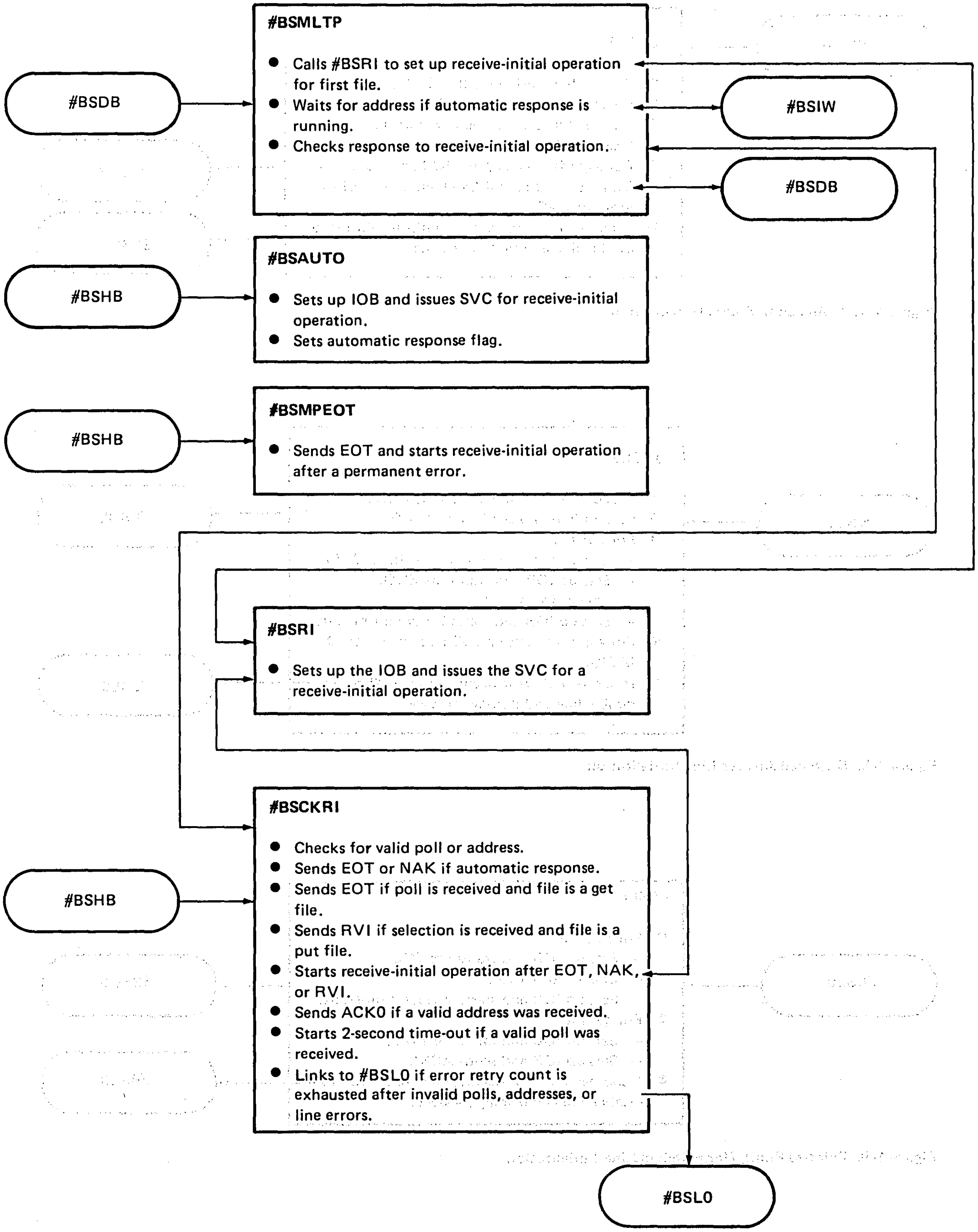
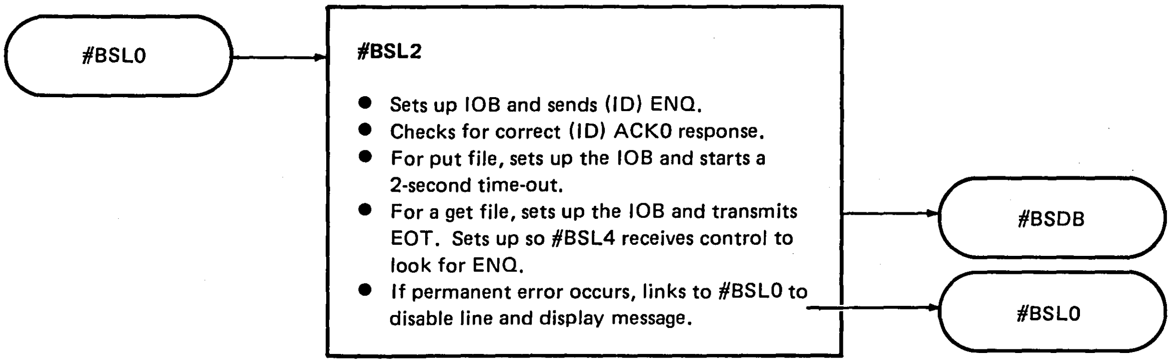

#BSMLTP
● Calls #BSRI to set up receive-initial operation for first file.
● Waits for address if automatic response is running.
● Checks response to receive-initial operation.
#BSDB
#BSIW
#BSDB
#BSAUTO
● Sets up IOB and issues SVC for receive-initial operation.
● Sets automatic response flag.
#BSHB
#BSMPEOT
● Sends EOT and starts receive-initial operation after a permanent error.
#BSHB
#BSRI
● Sets up the IOB and issues the SVC for a receive-initial operation.
#BSCKRI
● Checks for valid poll or address.
● Sends EOT or NAK if automatic response.
● Sends EOT if poll is received and file is a get file.
● Sends RVI if selection is received and file is a put file.
● Starts receive-initial operation after EOT, NAK, or RVI.
● Sends ACK0 if a valid address was received.
● Starts 2-second time-out if a valid poll was received.
● Links to #BSL0 if error retry count is exhausted after invalid polls, addresses, or line errors.
#BSHB
#BSL0

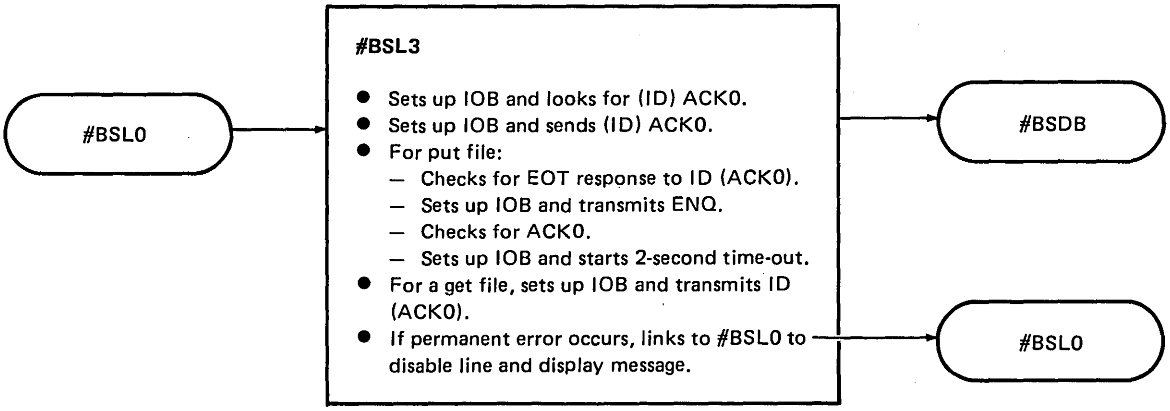
Figure 1-6. Multipoint Line Initialization



#BSL2

- Sets up IOB and sends (ID) ENQ.
- Checks for correct (ID) ACK0 response.
- For put file, sets up the IOB and starts a 2-second time-out.
- For a get file, sets up the IOB and transmits EOT. Sets up so #BSL4 receives control to look for ENQ.
- If permanent error occurs, links to #BSL0 to disable line and display message.

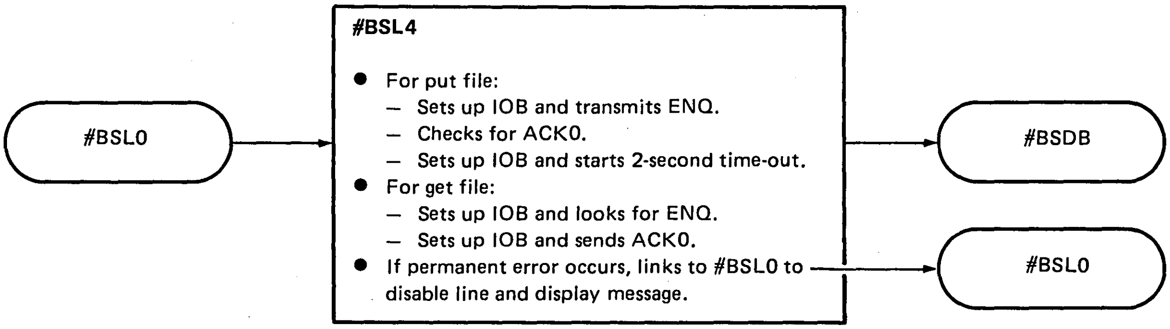
Figure 1-7. Switched Call Line Initialization



#BSL3

- Sets up IOB and looks for (ID) ACK0.
- Sets up IOB and sends (ID) ACK0.
- For put file:
  - Checks for EOT response to ID (ACK0).
  - Sets up IOB and transmits ENQ.
  - Checks for ACK0.
  - Sets up IOB and starts 2-second time-out.
- For a get file, sets up IOB and transmits ID (ACK0).
- If permanent error occurs, links to #BSL0 to disable line and display message.

Figure 1-8. Switched Answer Line Initialization



#BSL4

- For put file:
  - Sets up IOB and transmits ENQ.
  - Checks for ACK0.
  - Sets up IOB and starts 2-second time-out.
- For get file:
  - Sets up IOB and looks for ENQ.
  - Sets up IOB and sends ACK0.
- If permanent error occurs, links to #BSL0 to disable line and display message.

Figure 1-9. Point-to-Point, Nonswitched Line Initialization

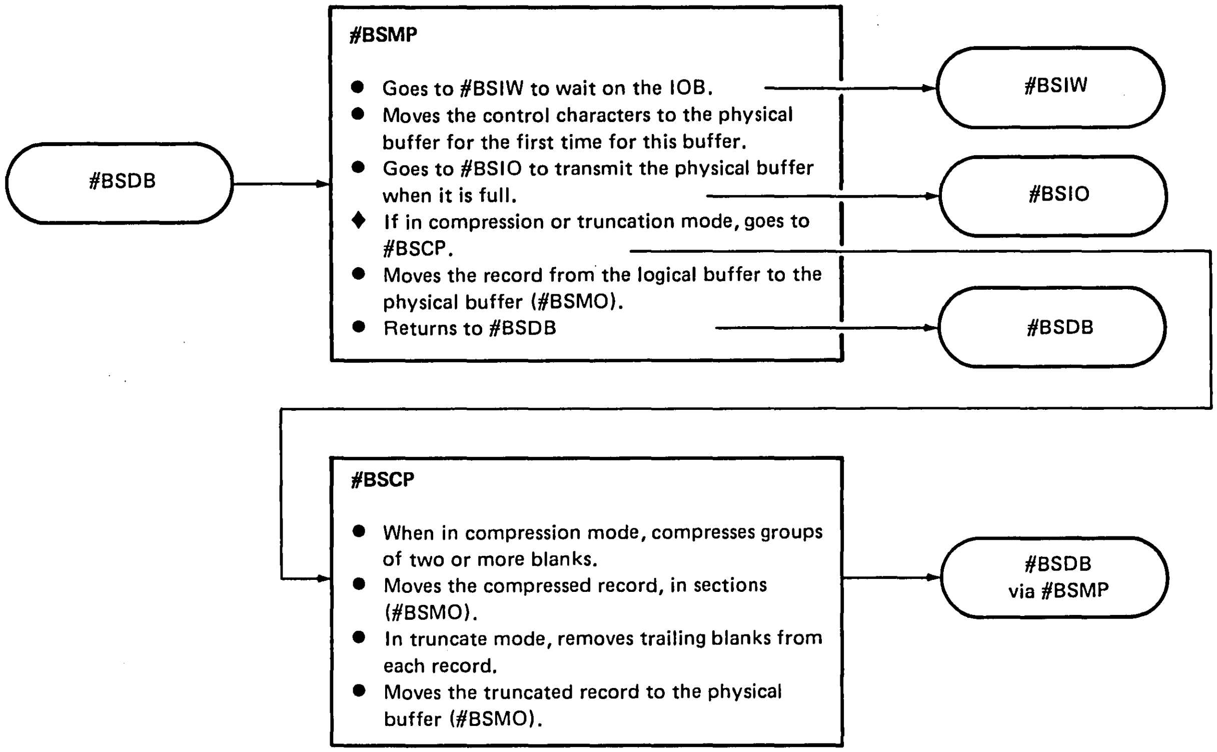
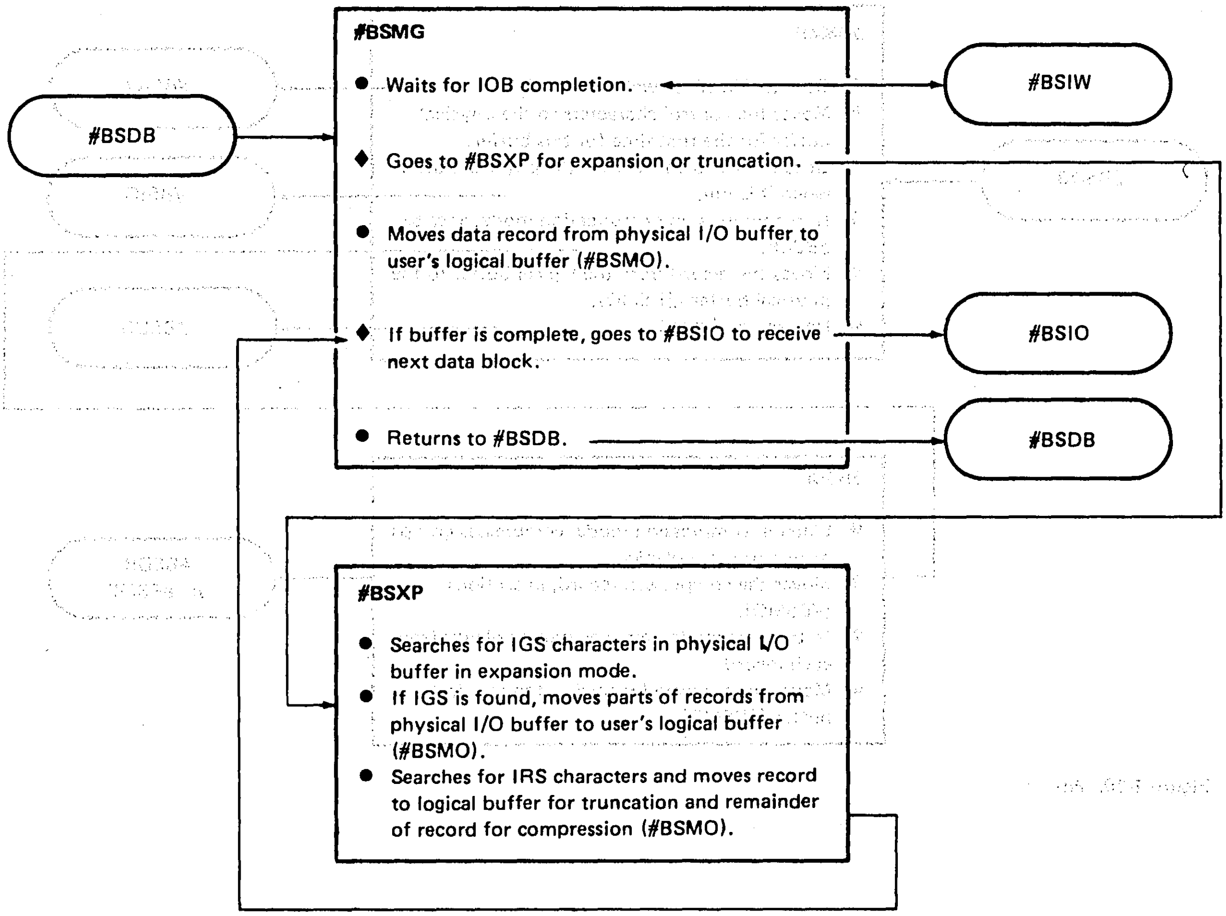

#BSDB
#BSMP
● Goes to #BSIW to wait on the IOB.
● Moves the control characters to the physical buffer for the first time for this buffer.
● Goes to #BSIO to transmit the physical buffer when it is full.
♦ If in compression or truncation mode, goes to #BSCP.
● Moves the record from the logical buffer to the physical buffer (#BSMO).
● Returns to #BSDB
#BSIW
#BSIO
#BSDB
#BSCP
● When in compression mode, compresses groups of two or more blanks.
● Moves the compressed record, in sections (#BSMO).
● In truncate mode, removes trailing blanks from each record.
● Moves the truncated record to the physical buffer (#BSMO).
#BSDB
via #BSMP

Figure 1-10. Put



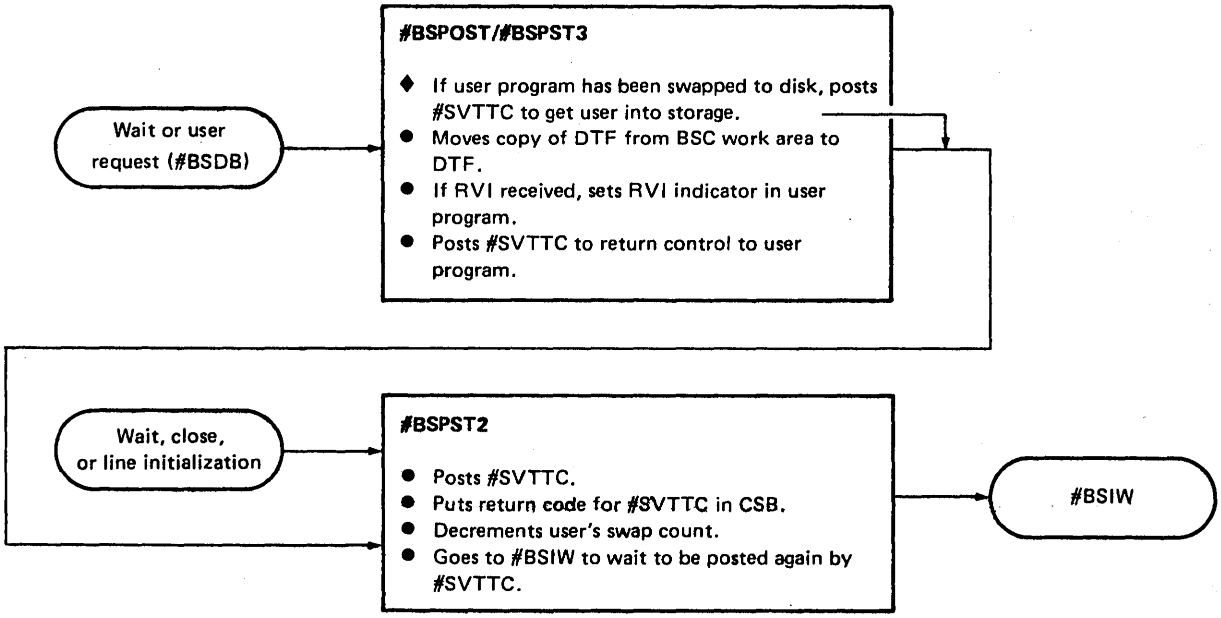
#BSMG

- ● Waits for IOB completion.
- ♦ Goes to #BSXP for expansion or truncation.
- ● Moves data record from physical I/O buffer to user's logical buffer (#BSMO).
- ♦ If buffer is complete, goes to #BSIO to receive next data block.
- ● Returns to #BSDB.

#BSXP

- ● Searches for IGS characters in physical I/O buffer in expansion mode.
- ● If IGS is found, moves parts of records from physical I/O buffer to user's logical buffer (#BSMO).
- ● Searches for IRS characters and moves record to logical buffer for truncation and remainder of record for compression (#BSMO).

Figure 1-11. Get



Wait or user request (#BSDB)
Wait, close, or line initialization
#BSIW

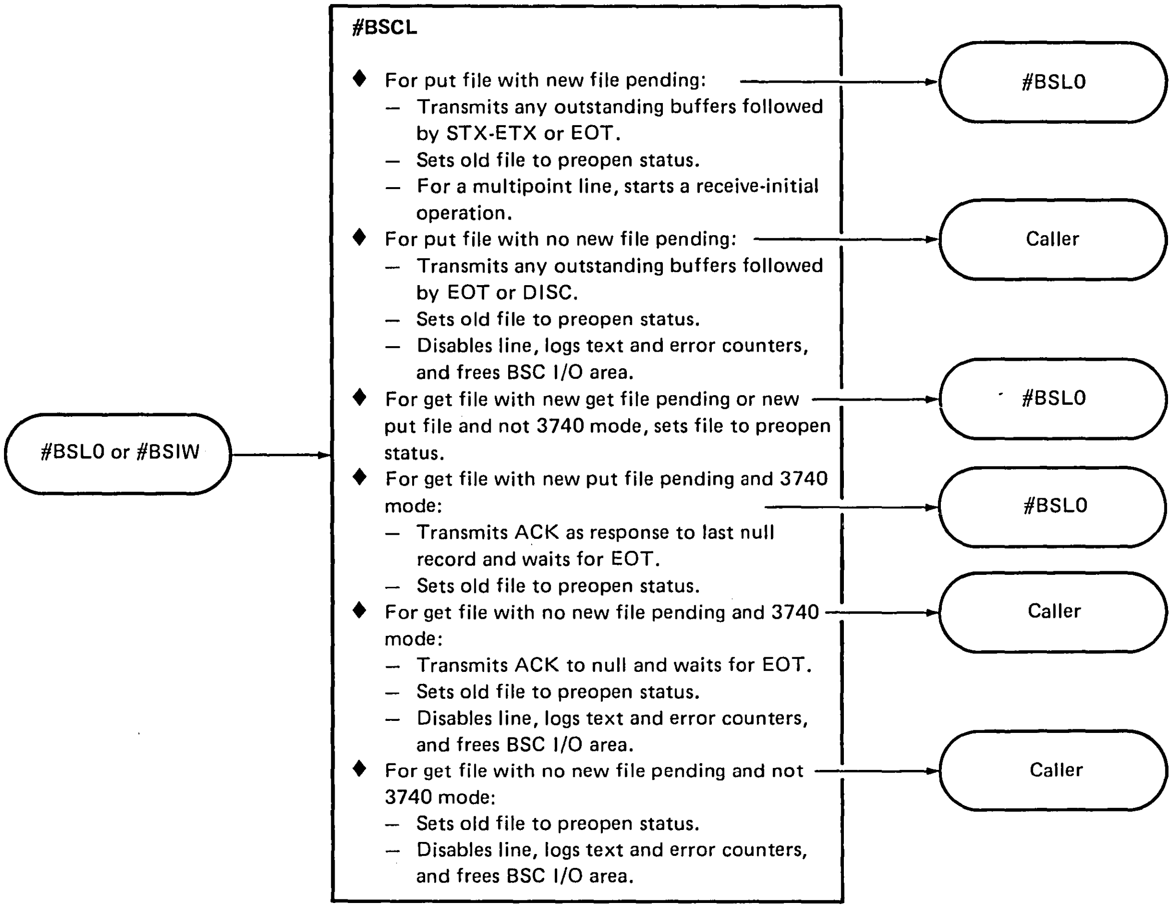
#BSPOST/#BSPST3

- ♦ If user program has been swapped to disk, posts #SVTTC to get user into storage.
- ● Moves copy of DTF from BSC work area to DTF.
- ● If RVI received, sets RVI indicator in user program.
- ● Posts #SVTTC to return control to user program.

#BSPST2

- ● Posts #SVTTC.
- ● Puts return code for #SVTTC in CSB.
- ● Decrements user's swap count.
- ● Goes to #BSIW to wait to be posted again by #SVTTC.

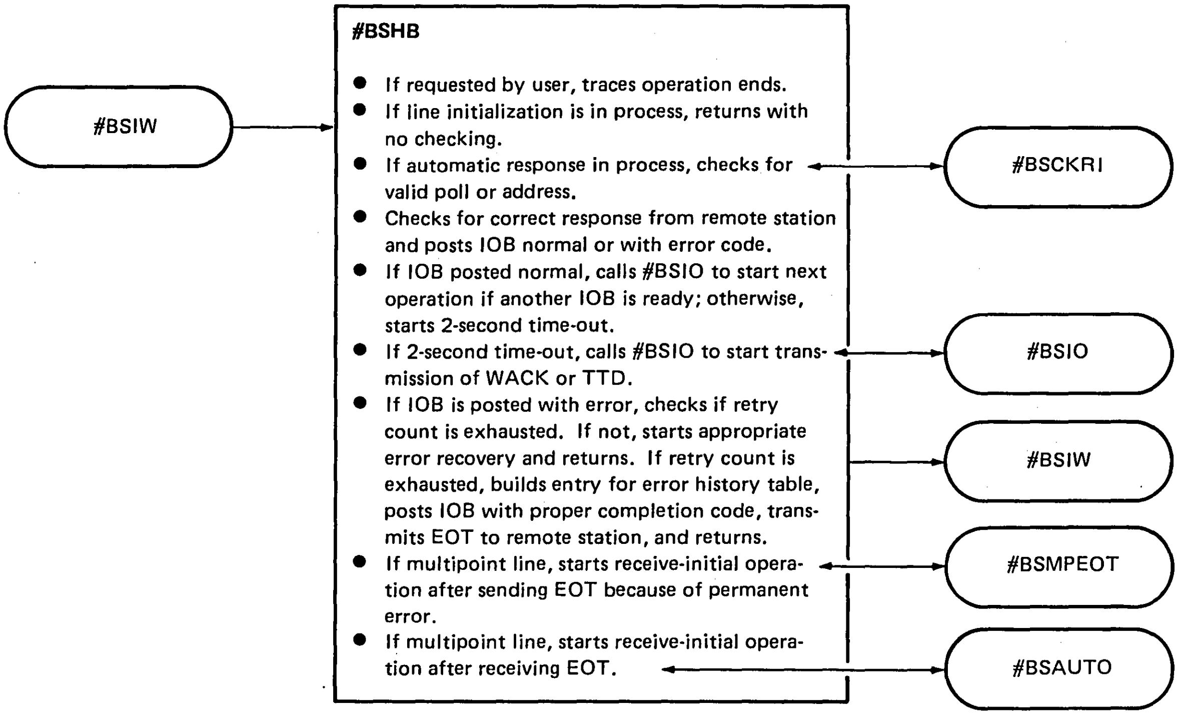
Figure 1-12. Post



#BSCL

- ♦ For put file with new file pending:
  - Transmits any outstanding buffers followed by STX-ETX or EOT.
  - Sets old file to preopen status.
  - For a multipoint line, starts a receive-initial operation.
- ♦ For put file with no new file pending:
  - Transmits any outstanding buffers followed by EOT or DISC.
  - Sets old file to preopen status.
  - Disables line, logs text and error counters, and frees BSC I/O area.
- ♦ For get file with new get file pending or new put file and not 3740 mode, sets file to preopen status.
- ♦ For get file with new put file pending and 3740 mode:
  - Transmits ACK as response to last null record and waits for EOT.
  - Sets old file to preopen status.
- ♦ For get file with no new file pending and 3740 mode:
  - Transmits ACK to null and waits for EOT.
  - Sets old file to preopen status.
  - Disables line, logs text and error counters, and frees BSC I/O area.
- ♦ For get file with no new file pending and not 3740 mode:
  - Sets old file to preopen status.
  - Disables line, logs text and error counters, and frees BSC I/O area.

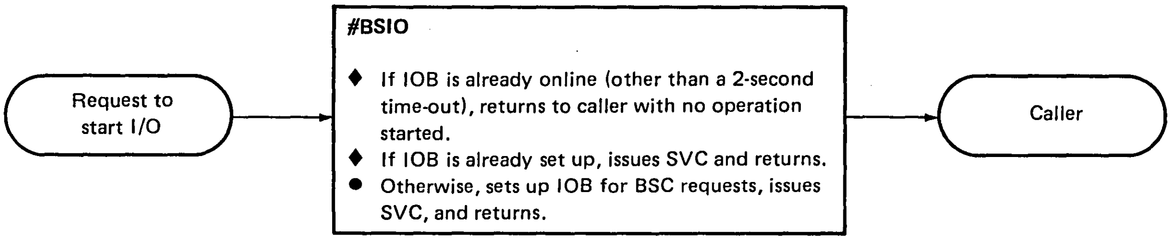
Figure 1-13. Close



#BSHB

- If requested by user, traces operation ends.
- If line initialization is in process, returns with no checking.
- If automatic response in process, checks for valid poll or address.
- Checks for correct response from remote station and posts IOB normal or with error code.
- If IOB posted normal, calls #BSIO to start next operation if another IOB is ready; otherwise, starts 2-second time-out.
- If 2-second time-out, calls #BSIO to start transmission of WACK or TTD.
- If IOB is posted with error, checks if retry count is exhausted. If not, starts appropriate error recovery and returns. If retry count is exhausted, builds entry for error history table, posts IOB with proper completion code, transmits EOT to remote station, and returns.
- If multipoint line, starts receive-initial operation after sending EOT because of permanent error.
- If multipoint line, starts receive-initial operation after receiving EOT.

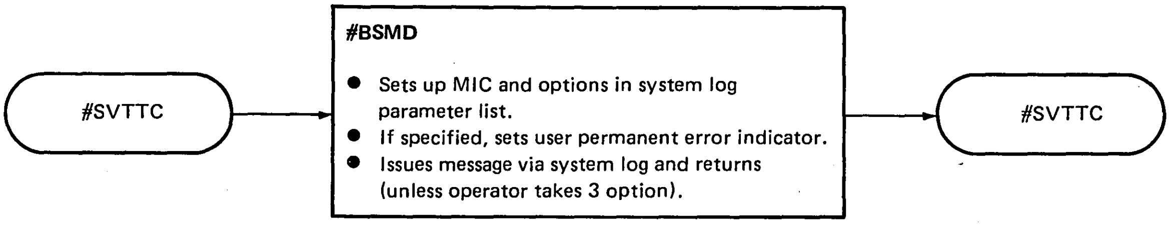
Figure 1-14. Operation End Processing



#BSIO

- ◆ If IOB is already online (other than a 2-second time-out), returns to caller with no operation started.
- ◆ If IOB is already set up, issues SVC and returns.
- Otherwise, sets up IOB for BSC requests, issues SVC, and returns.

Figure 1-15. BSC IOS



#BSMD

- Sets up MIC and options in system log parameter list.
- If specified, sets user permanent error indicator.
- Issues message via system log and returns (unless operator takes 3 option).

Figure 1-16. Message Logging

Directory

This section consists of a quick-reference table that can be used to find BSC modules on microfiche program listings. The table contains:

- Module/routine name
- Diagram number
- Descriptive name
- Function

Module/ Routine Name	Diagram Number	Descriptive Name	Function
#BSCM	––	Resident data management (BSC mainline)	Processes all BSC user requests. The following routines (#BSATR through #BSXP) are all part of this module.
#BSATR	1.9	Address translation	Converts user's logical address to BSC logical address and updates BSC ATR stack.
#BSAUTO	1.2.1	Receive-initial initiator	Sets up IOB and issues SVC for receive-initial command. Starts automatic response.
#BSCKRI	1.2.1	Check receive-initial command	Checks for poll or address response with NAK, EOT, ACK0; or starts 2-second time-out. Starts another receive-initial command after NAK or EOT.
#BSCP	1.5.2	Blank compression and expansion	Removes blanks and inserts 3780 control characters.
#BSDB	1.4	Data management for get and put	Invokes module necessary to receive and/or transmit data.
#BSHB	1.3.1 1.6	Operation end processor	Analyzes operation end interrupts and (1) gives control to error recovery routines, or (2) sets good completion code in IOB.
#BSIO	1.2.1 1.2.2 1.2.3 1.2.4	BSC IOS	Sets up IOB and issues SVC to start BSC I/O operations.
#BSIW	1.3.1 1.5.1 1.7 1.8	Wait	Waits for completion of BSC I/O operations of for user requests.
#BSMG	1.3.2	Data receive	Deblocks the receive buffer.

Module/ Routine Name	Diagram Number	Descriptive Name	Function
#BSCM (continued)			
#BSMLTP	1.2.1	Multipoint router on new file	Sets up receive-initial command if automatic response not on or waits for address if automatic response started.
#BSMO	1.3.2.1	Move	Moves data from physical buffer to user's logical buffer for receive, and from user's logical buffer to physical buffer for transmit.
#BSMP	1.4 1.5.2	Data blocking	Blocks the data records.
#BSMPEOT	1.2.1	Send EOT	Sends EOT and starts receive-initial processing.
#BSPOST	1.4	Post	Restores user's DTF, sets RVI indicator if required, and posts system task-to-task interface routine (#SVTTC) to return to user.
#BSRI	1.2.1	Receive-initial initiator	Sets up IOB and issues SVC for receive-initial command.
#BSXP	1.3.2.1	Blank expansion	Removes 3780 control characters and inserts blanks.
#BSCL	1.2 1.8 1.10	Close	Closes BSC files and terminates BSC task.
#BSLO	1.2 1.4	Line initialization	Determines line type and calls appropriate transient.
#BSL2	1.2.2	Line initialization	Initializes switched-call line.
#BSL3	1.2.3	Line initialization	Initializes switched-answer line.
#BSL4	1.2.2 1.2.4	Line initialization	Initializes point-to-point, nonswitched line.
#BSMD	None	BSC message transient	Sets up parameter list and calls the system logging routine to display BSC error messages. Sets permanent-error indicator if required (RPG II only).
#BSOB	1.1	Open	Opens BSC files.

Data Areas

BSC uses the following data areas, which are described in the *Data Areas Handbook* :

- BSC DTF
- BSC IOB
- BSC CSB
- BSC work area
- BSC I/O area
- BSC buffers

Part 2. MULTI-LEAVING Remote Job Entry (MRJE) Utility

Introduction

The IBM System/34 MULTI-LEAVING Remote Job Entry (MRJE) SSP utility provides the ability to submit, execute, and obtain results of jobs from a host system via the communications adapter. Host systems are:

- ASP under OS/VS2
- HASP II under OS/VS2
- RES under OS/VS1
- JES2 under OS/VS2
- JES3 under OS/VS2
- VM/370 RSCS

These host systems support a line discipline (MULTI-LEAVING) that permits maximum overlap of I/O operations at the remote station and the host system.

Communication is in EBCDIC and text transparency can be used.

The MRJE utility can run either in attended mode or automatic mode. For attended mode operation, the operator must be present while the MRJE utility is running. For automatic mode operation, the operator need not be present to handle MRJE halts.

Input to the host system can be any job that can be entered at the host system from similarly functioning I/O devices. MRJE input can be entered from the keyboard, from a disk file, or from a system library member.

Output from the host system can be returned to the MRJE utility, directed to another RJE work station, or directed to the host system's I/O devices. Output files created by MRJE are sequential files that can be processed by customer-written RPG II programs (except TDISK files), by customer-written assembler programs (except TDISK files), or by the data communications print utility ($DCSUP). MRJE operator messages received from the host system are directed to the system console and entered in the history file.

FUNCTIONAL ORGANIZATION

Execution of the MRJE utility can be divided into three general functional areas: initialization, processing, and termination.

Initialization

The MRJE utility is loaded by entering the MRJE procedure command from the system console.

During initialization, MRJE performs the following operations:

1. Displays the initialization display at the console. The format used can be either the IBM-supplied format (#MR01) or a user-written format. The operator can modify any of the parameters and then press the Enter key or simply press the Enter key to have MRJE use all default values.

2. Builds and initializes the following data areas:
   a. The program function control blocks (PFCBs) for the tasks performed during MRJE processing.
   b. The communication and control table (CCT) containing information on the status of the I/O devices, the addresses of the MRJE modules, information relating to the printer, task-to-task communication information, and other information required by more than one module.

3. Builds the total MRJE utility and the individual task buffers based on the information given on the initialization display.

4. Determines the space remaining in the region allocated to the MRJE mainline program. If the space remaining is large enough for the MRJE BSC module and buffers (as specified on the initialization display), MRJE loads the BSC open module via system allocate and builds the BSC buffers. If the space is not large enough, MRJE issues a message to the operator indicating that not enough storage is available.

5. Establishes task-to-task communications between the MRJE mainline and the MRJE BSC module.

6. Creates termination exit blocks to handle MRJE and BSC abnormal termination.

Processing

During processing, MRJE performs the following functions:

- BSC: Transmits and receives data, and maintains communications with the host system.

- Printer: Processes all printer data from the host system. The printer data is either printed, written to the TDISKPR1 file for standard print forms, or written to individual files for special printer forms on disk.

- Console output: Processes MRJE messages and all console output from the host system.

- Console input: Processes MRJE statements and host system commands.

- Reader: Processes MRJE statements and jobs entered at the display station for the host system.

- Punch: Processes all punch data from the host system. The punch data is written to individual files on disk.

Associated with each function is a PFCB, which contains control and status information for the task, and an IOB, DTF, or TUB address. The PFCBs are chained as shown in Figure 2-1.

Dispatching Logic

The dispatching supervisor ($MRJE) coordinates all MRJE processing by continually monitoring the PFCBs. When a PFCB indicates that there is work for the task to perform, the dispatching supervisor passes control to the dispatching address stored in the PFCB. The dispatching address is the address of the routine that receives control when the task is dispatched.

Upon completion of an operation by a task, control is returned to the dispatching supervisor. For example, after the print task issues a supervisor call (SVC) to start an I/O operation, the task sets a bit in the printer PFCB indicating that it is waiting for completion of an I/O operation. Then, the task returns control to the dispatching supervisor. The dispatching supervisor checks the remaining PFCBs and TUB for work to be done and dispatches the tasks in turn if work is found. When all PFCBs have been checked, and no more work can be done, the dispatching supervisor issues a wait-on-any SVC to return control to the SSP. When an external event occurs, such as the completion of an MRJE I/O operation, the MRJE dispatcher is posted out of its wait. The MRJE dispatcher then identifies the event that occurred, posts the appropriate function, and returns to its dispatching loop.

Termination

During termination, MRJE (1) disables the communications adapter, (2) closes the TDISKPR1 file if required, (3) displays the BSC statistics for the session, (4) updates the BSC I/O counters, (5) terminates the BSC task, and (6) terminates the MRJE task.

INPUT/OUTPUT

Input required by System/34 MRJE consists of the following:

- Program load data from the MRJE procedure command entered at the system console

- Configuration, carriage, and sign-on information from the initialization display

- MRJE utility control statements

- Print and punch output data and messages from the host system

- JCL, data, and host system commands entered from the System/34

Output depends on the type of program being run.

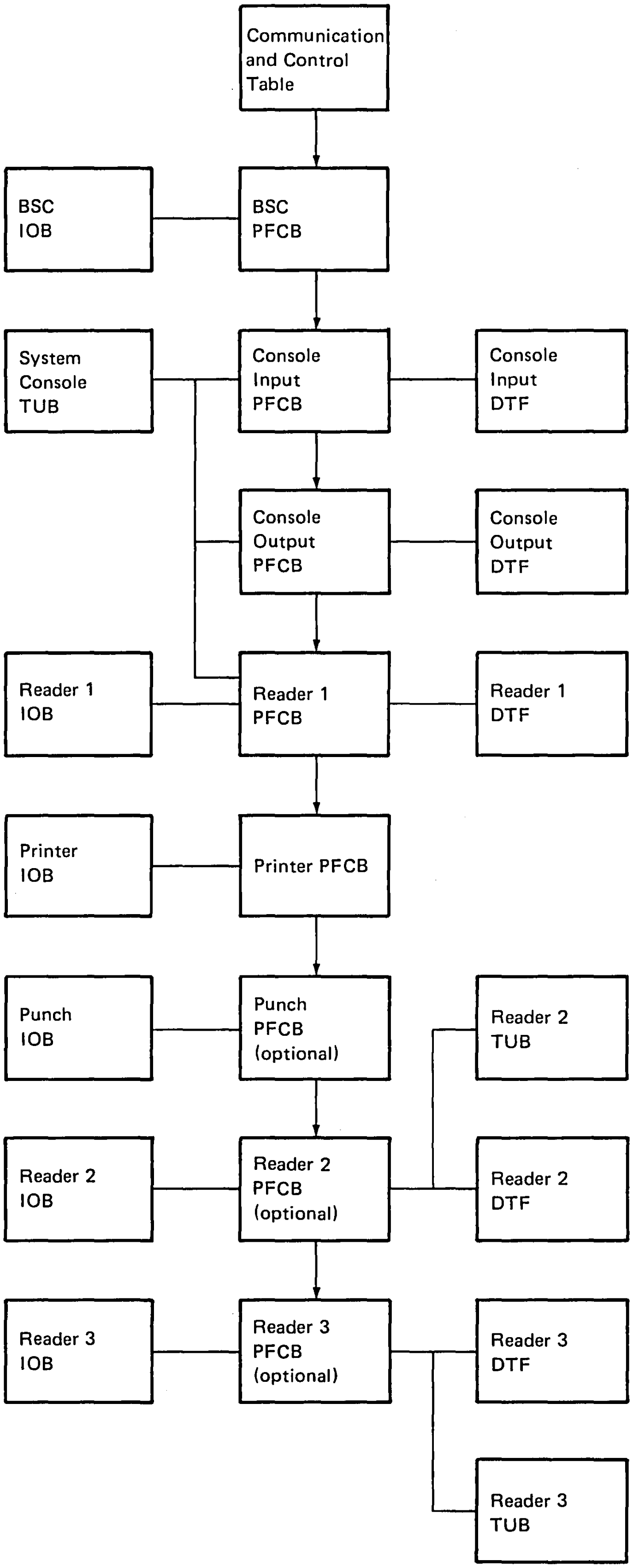


Figure 2-1. Chain of PFCBs for MRJE

SYSTEM INTERFACES

The System/34 MRJE utility consists of two main storage tasks:

- The BSC task, which controls the BSC line. The BSC task interfaces with the main MRJE task, the BSC control storage module, and the BSC hardware. The BSC task also controls the flow of data over the BSC line.

- The main MRJE task, which controls the remainder of the MRJE program. The main MRJE task supervises the loading and dispatching of the MRJE main storage modules that execute in the MRJE-allocated overlay areas. The main MRJE task also controls the overall flow and scheduling of work within MRJE.

All of the MRJE main storage modules reside in the system library.

SSP Interfaces

MRJE operates under control of the System/34 SSP. The SSP controls display station data management, allocation and deallocation of printer and disk files, task-to-task communication, get-page and free-page operations, BSC trace, finding and loading of the MRJE main storage modules, system log operations, history file put operations, and program termination.

MRJE operates as a stand-alone MRT (multiple requestor terminal) program and is also defined as an NEP (never-ending program). The main MRJE task is loaded by the SSP and runs in its own storage area. The main MRJE task uses the system allocate function to start the BSC task. The two MRJE tasks then share a common storage area. The BSC task is nonswappable and the main MRJE task is swappable. The various MRJE modules are loaded and run, as needed, in overlay areas that are part of the MRJE storage area.

Control Storage Interface

The BSC open module (#MRBO) loads the BSC control storage microcode. #MRBO loads the microcode using the System/34 load control storage transient SVC instruction.

Hardware Interface

The IOB (input/output block) and System/34 supervisor calls (SVCs) provide the interface to the BSC hardware. The MRJE BSC task uses transmit/receive operations with buffer overlay (TRRCVOL) and, for some special operations, uses transmit/receive operations without buffer overlay.

ERROR RECOVERY

Types of Errors

MRJE errors can be caused by:

- System console

- Display station

- Disk

- Printer

- BSC

- Operator or programmer

System Console Errors

If an error occurs on the system console, MRJE abnormally terminates. The communications adapter is disabled and the TDISKPR1 file, if one exists, is closed as if the operator had selected a 3 option for an error message or a CANCEL control command.

Display Station Errors

If an error occurs on a display station (other than the system console) that is a reader, that reader task is marked as having a display station error. Any subsequent messages to that display station are logged only to the history file. MRJE assumes default responses to halts. When the next input statement is to be read from the display station, the display station is released and that reader task is terminated.

Disk Errors

A disk write error causes a hard system error (a processor check). A disk read error that occurs while the TDISKPR1 file is being processed causes a message to be displayed and the print task to be deactivated.

A disk read error that occurs while the reader task is reading from the disk causes a message to be displayed and the reader task to be deactivated. If a disk error occurs while putting the BSC I/O counters to disk, a message is displayed and the counters are not updated.

Printer Errors

When a printer error occurs, and MRJE is in automatic mode, the print queue is cleared and quiesced, a message is displayed, and the print task is deactivated. If MRJE is not in automatic mode, normal SSP error recovery occurs.

BSC Errors

When a BSC error occurs, the appropriate message is displayed and the error is recorded in the PFCB for the BSC task. When MRJE terminates, the error counters are updated with the information in the BSC PFCB.

Operator or Programmer Errors

If the operator keys invalid program load data when starting MRJE, an appropriate message is displayed and MRJE terminates. If the operator keys invalid data into the initialization display, the fields in error are redisplayed in reverse image, the audible alarm is sounded, and an error message is displayed. The operator can then enter the correct field values or change any of the original values.

If an error is found in an MRJE utility control statement, one of the following occurs:

- If the statement in error was entered from the keyboard, an appropriate error message is displayed and execution continues. The operator can then enter the correct statement or any other statement.

- If the statement in error was read from a disk file, an appropriate error message that allows a 0 or 1 option is displayed. If the operator selects a 0 option, the statement in error is ignored and execution continues with input from the disk file. If the operator selects a 1 option, the keyboard is unlocked and the operator can then key in the corrected statement. After the new statement is processed, execution continues with input from the disk file.

SYSTEM REQUIREMENTS

MRJE operates in the user program area under control of the System Support Program Product (SSP) and requires supervisor and input/output supervisor (IOS) services provided by the SSP.

The minimum system configuration required to support MRJE is:

- 16K region size
- Communications adapter
- EBCDIC capability specified

A region size greater than the minimum required allows more tasks (punch and additional readers), allows full compression, and can improve performance by increasing the number of BSC buffers.

Method of Operation

This section contains the logical functions of MRJE using an overview (see Diagram 2) and an input-process-output diagram for each function.

For detailed information on input-process-output diagrams, see *Diagram Techniques* in *How to Use This Manual* at the front of this publication.

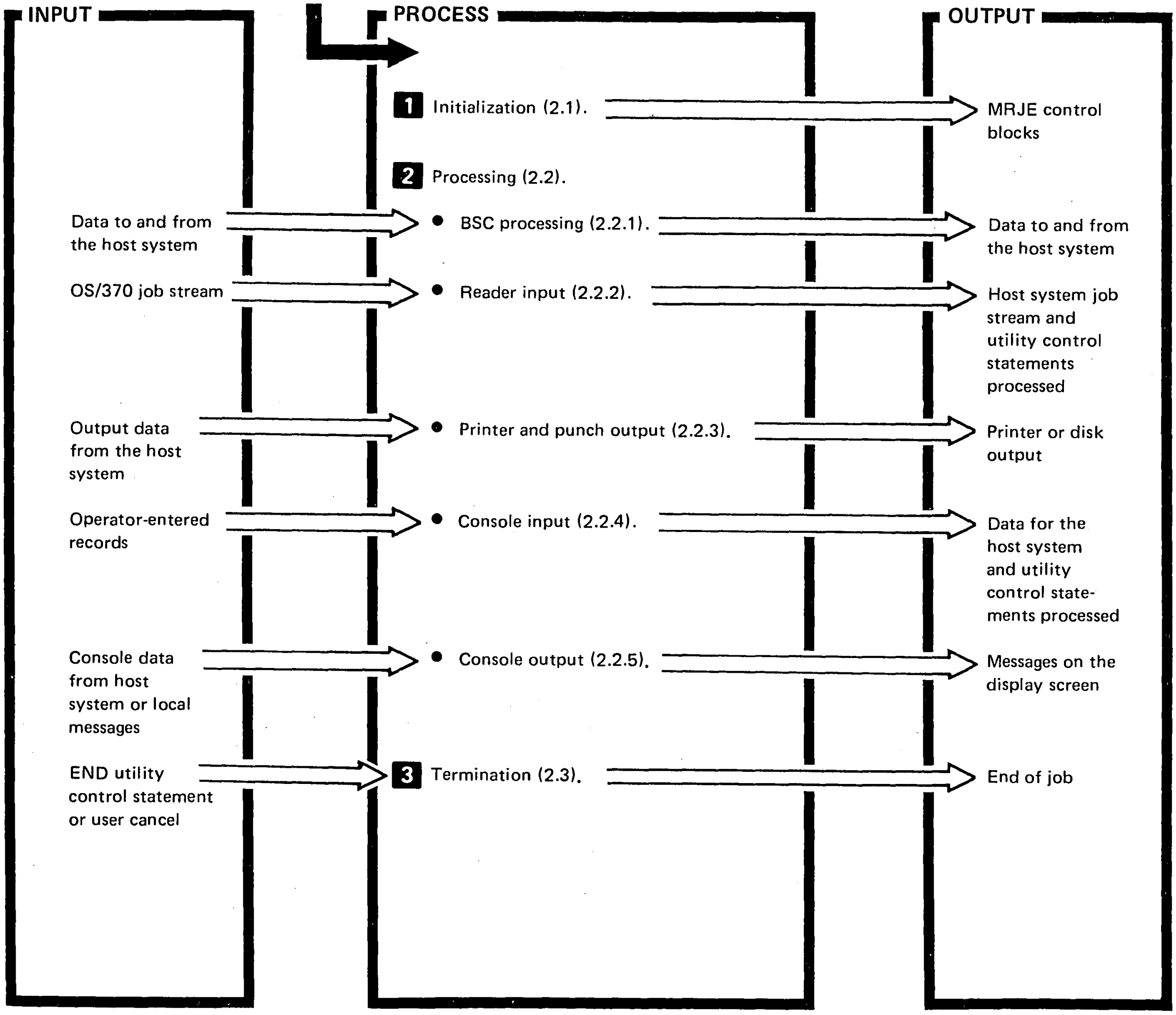
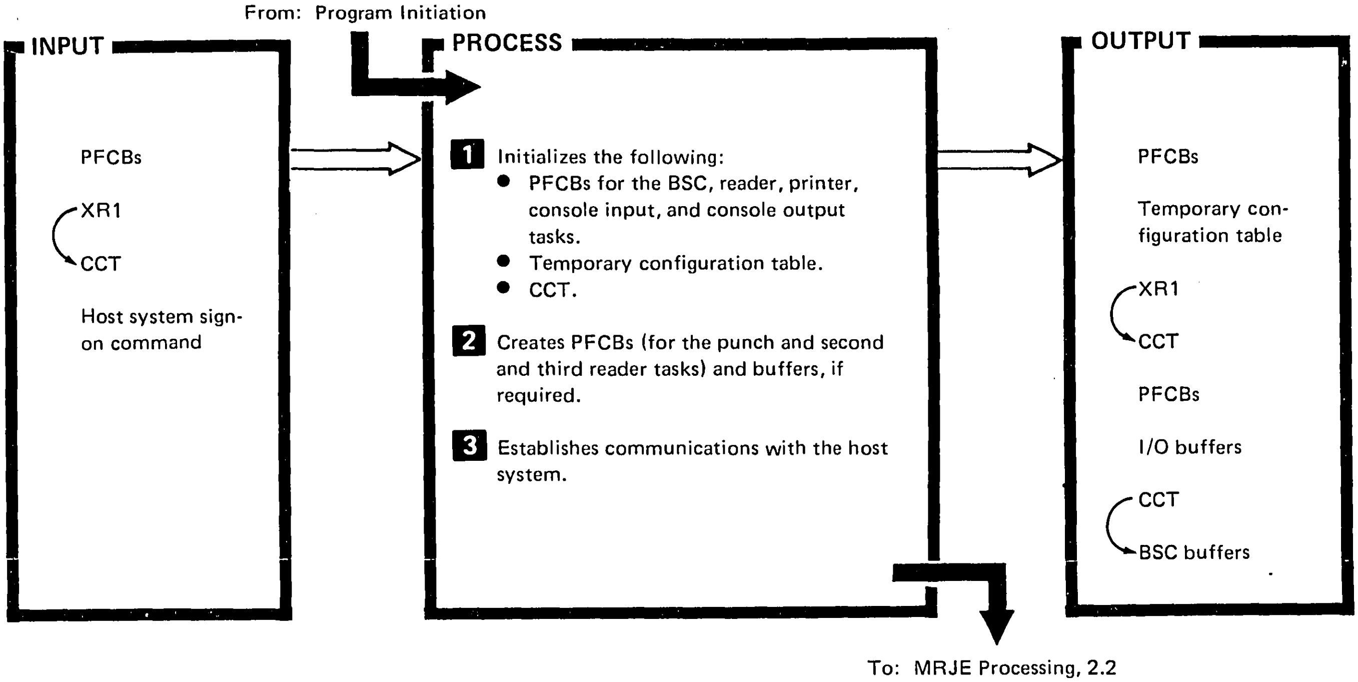

INPUT
PROCESS
OUTPUT
1 Initialization (2.1).
MRJE control blocks
2 Processing (2.2).
Data to and from the host system
• BSC processing (2.2.1).
Data to and from the host system
OS/370 job stream
• Reader input (2.2.2).
Host system job stream and utility control statements processed
Output data from the host system
• Printer and punch output (2.2.3).
Printer or disk output
Operator-entered records
• Console input (2.2.4).
Data for the host system and utility control statements processed
Console data from host system or local messages
• Console output (2.2.5).
Messages on the display screen
END utility control statement or user cancel
3 Termination (2.3).
End of job

Diagram 2. MRJE Overview



From: Program Initiation

INPUT

PFCBs

XR1

CCT

Host system sign-on command

PROCESS

1 Initializes the following:
- PFCBs for the BSC, reader, printer, console input, and console output tasks.
- Temporary configuration table.
- CCT.

2 Creates PFCBs (for the punch and second and third reader tasks) and buffers, if required.

3 Establishes communications with the host system.

OUTPUT

PFCBs

Temporary configuration table

XR1

CCT

PFCBs

I/O buffers

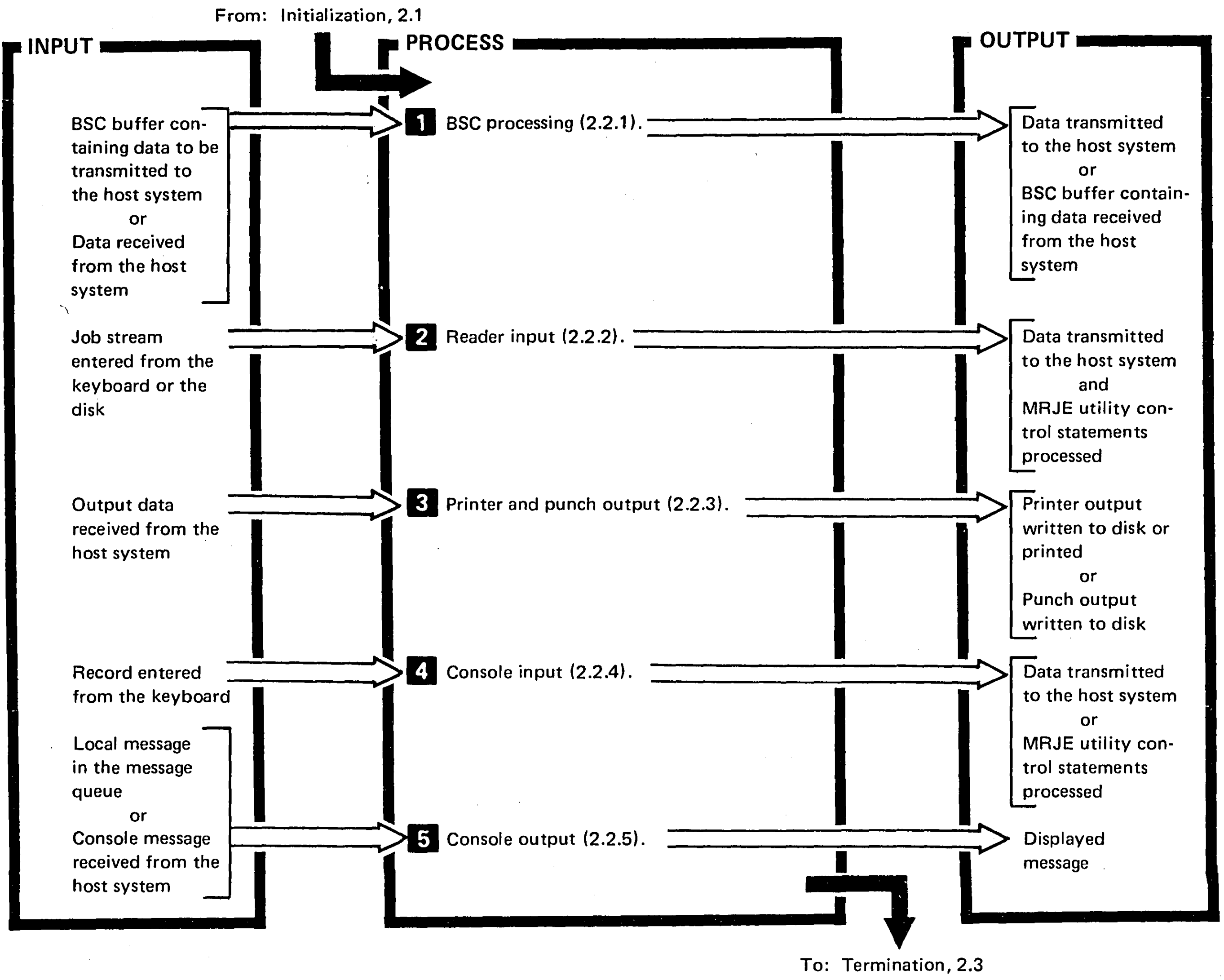
CCT

BSC buffers

To: MRJE Processing, 2.2

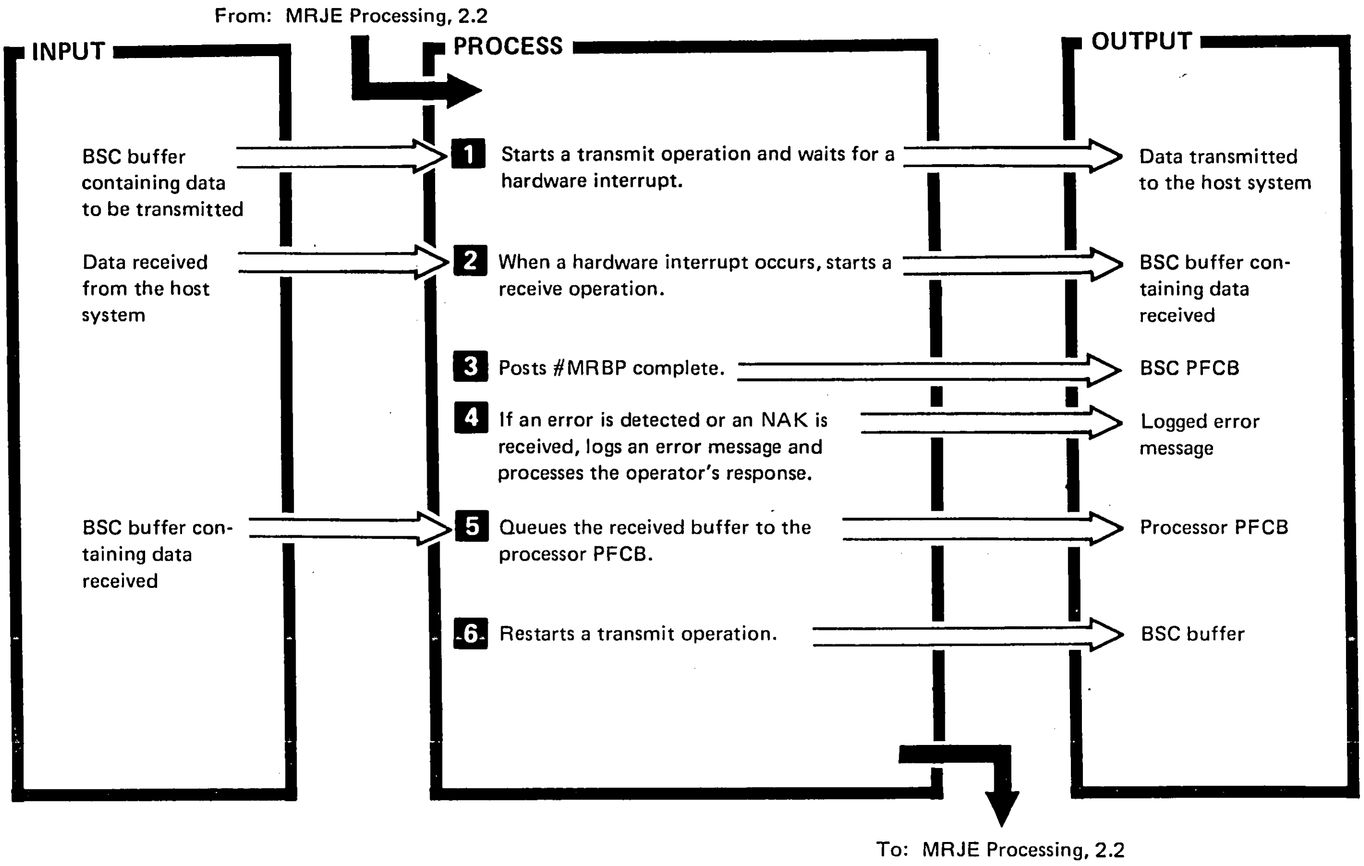
DESCRIPTION	MODULE/ ROUTINE
1 Reads the initialization display from the system console and uses it to complete the fields in the control blocks	#MRI1
2 ● Loads and initializes the buffer manager (#MRBM) and, if required, the full compression routine (#MRFC). ● Divides the remaining main storage region into transient areas; I/O buffers for the processors; BSC buffers; and PFCBs, DTFs, and IOBs for the additional reader and punch functions. ● Chains the BSC buffers and places the address of the first BSC buffer in the CCT.	#MRI6
3 To establish communications, the operator dials the host system if dialing is required. MRJE then sends a sign-on command (entered from the system console initialization display) to the host system. BSC open (#MRBO) sets up to send the sign-on command, while the BSC I/O processor (#MRBP) actually sends the command.	#MRBO #MRBP

Diagram 2.1 Initialization


From: Initialization, 2.1
INPUT
PROCESS
OUTPUT
BSC buffer containing data to be transmitted to the host system or Data received from the host system
1 BSC processing (2.2.1).
Data transmitted to the host system or BSC buffer containing data received from the host system
Job stream entered from the keyboard or the disk
2 Reader input (2.2.2).
Data transmitted to the host system and MRJE utility control statements processed
Output data received from the host system
3 Printer and punch output (2.2.3).
Printer output written to disk or printed or Punch output written to disk
Record entered from the keyboard
4 Console input (2.2.4).
Data transmitted to the host system or MRJE utility control statements processed
Local message in the message queue or Console message received from the host system
5 Console output (2.2.5).
Displayed message
To: Termination, 2.3

DESCRIPTION	MODULE/ ROUTINE
The following functions are performed, as required, during MRJE processing:	
1 BSC processing.	#MRBP
2 Reader input.	#MRRP, #MRKP, or #MRFR
3 Printer and punch output.	#MRDP or #MRUP
4 Console input.	#MRCP
5 Console output.	#MRCD or #MRHD

Diagram 2.2. MRJE Processing



From: MRJE Processing, 2.2

INPUT

BSC buffer containing data to be transmitted

Data received from the host system

BSC buffer containing data received

PROCESS

1 Starts a transmit operation and waits for a hardware interrupt.

2 When a hardware interrupt occurs, starts a receive operation.

3 Posts #MRBP complete.

4 If an error is detected or an NAK is received, logs an error message and processes the operator's response.

5 Queues the received buffer to the processor PFCB.

6 Restarts a transmit operation.

OUTPUT

Data transmitted to the host system

BSC buffer containing data received

BSC PFCB

Logged error message

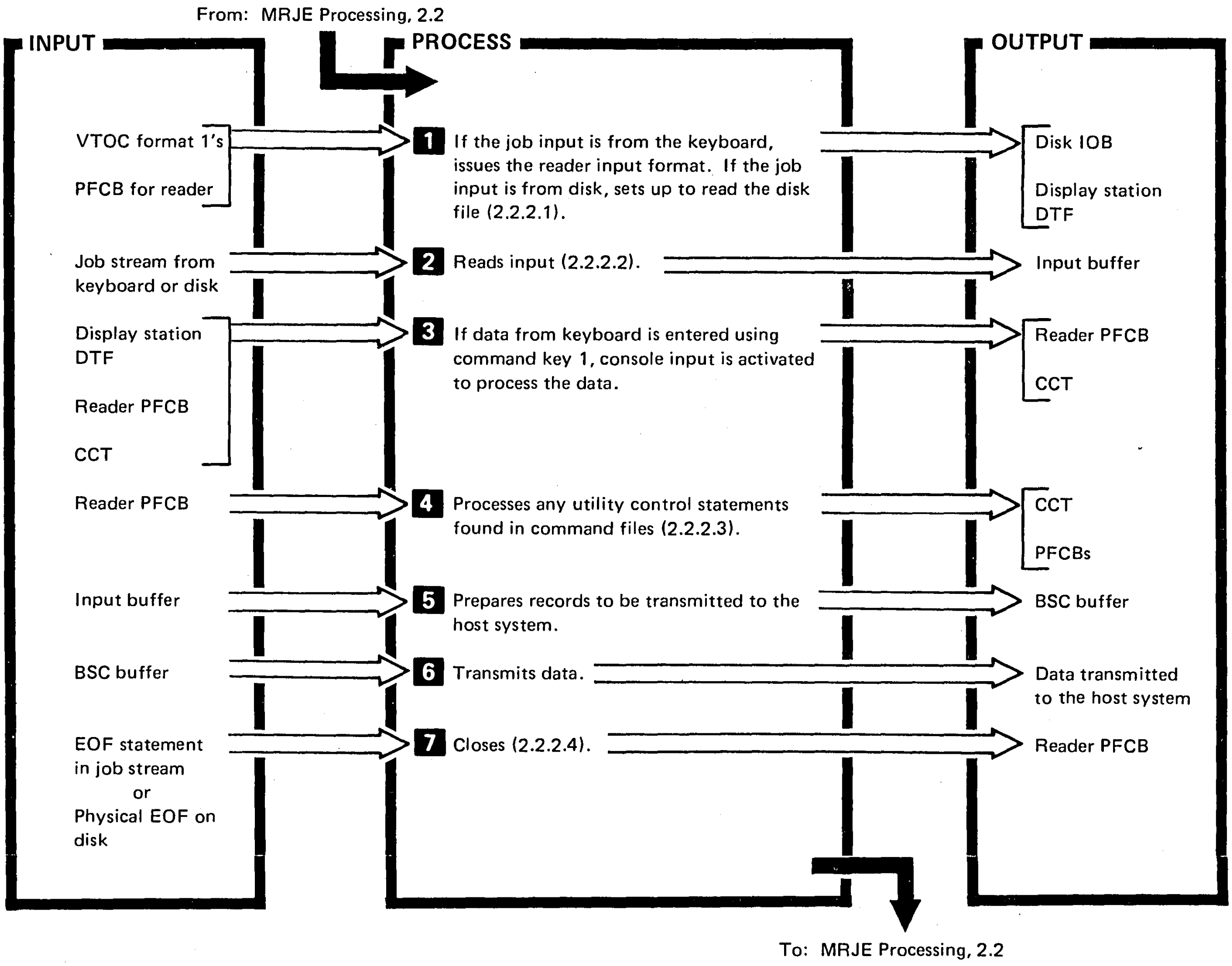
Processor PFCB

BSC buffer

To: MRJE Processing, 2.2

DESCRIPTION	MODULE/ ROUTINE
1 After #MRBP receives control via a BSC I/O post, it determines if there is data to send to the host system or if the host system is transmitting text. In either case, a BSC transmit operation is started. If there is no data to send to the host system and the host system is not transmitting text, #MRBP starts a 2-second time-out operation; on completion of the time-out operation, a transmit operation is started.	#MRBP
2 On completion of the BSC transmit operation, control is given to the interrupt portion of #MRBP (BSCAIL). The interrupt portion starts a BSC receive operation.	
3 On completion of the BSC receive operation, the interrupt portion posts the mainline portion of #MRBP complete.	
4 If there are errors, or if an NAK is received, control is passed to the BSC error recording routine (#MRBE) to log an error message and to process the operator's response. If the operator responds by selecting a 2 option, control is passed to the terminator (#MRTM) through the MRJE dispatcher.	#MRBE
5 The received BSC buffer is inspected for text received. If text was received, #MRBP queues the BSC buffer to the PFCB determined by the record control byte.	#MRBP
6 A BSC transmit operation is restarted, repeating Steps 1 through 5.	

Diagram 2.2.1. BSC Processing



1 If the job input is from the keyboard, issues the reader input format. If the job input is from disk, sets up to read the disk file (2.2.2.1).

2 Reads input (2.2.2.2).

3 If data from keyboard is entered using command key 1, console input is activated to process the data.

4 Processes any utility control statements found in command files (2.2.2.3).

5 Prepares records to be transmitted to the host system.

6 Transmits data.

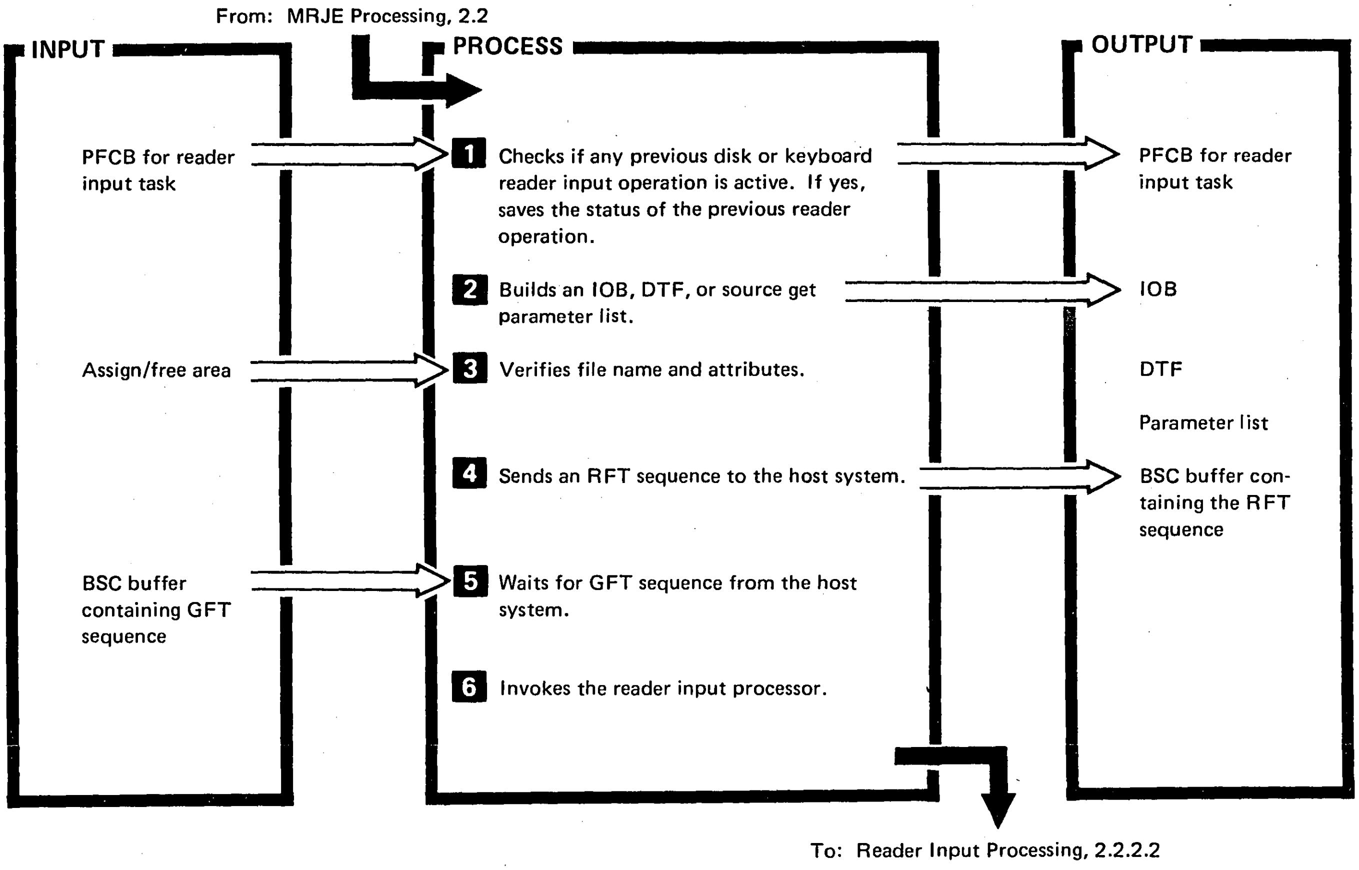
7 Closes (2.2.2.4).

DESCRIPTION	MODULE/ ROUTINE
This operation is performed under the logical reader input task. Reader input from the keyboard is invoked by the MODIFY utility control statement or during initialization. Reader input from disk is invoked by the READFILE utility control statement or during initialization.	
1 ● If input is from the keyboard, displays the reader input format.	#MRKP
● If input is from disk, the open function ensures the readiness of the file or library member to be read. Checks disk file labels to ensure that the proper files are available and that the attributes of the data (for example, record length) are valid for reader input.	#MRRO
2 Places the input entered from the keyboard or disk into the reader input buffer.	#MRKP, #MRRP, or #MRFR
3 If keyboard input and the DTF indicates command key 1, activates the console input task through the reader task. The reader task is temporarily halted.	#MRKP

Diagram 2.2.2 (Part 1 of 2). Reader Input

DESCRIPTION	MODULE/ ROUTINE
4 If input is from the command file, scans the input for utility control statements. If a utility control statement is found, invokes the command processor (#MRCO) to process it.	#MRRP, #MRFR, #MRKP, and #MRCO
5 Prepares records to be sent to the host system for transmission by compressing duplicate characters (if compression was requested) and by truncating trailing blanks and adding appropriate control information to the records. Then, places the records into the BSC buffer.	#MRRP, #MRFR, #MRBM, and #MRFC
6 When the BSC buffer is full, transmits the buffer to the host system.	#MRBP and #MRBM
7 Upon detection of an EOF utility control statement or a physical end of file on disk, closes the job input operation. Sends a logical end of file to the host system and updates the PFCB to indicate the closing of the reader task.	#MRRP, #MRFR, and #MRRC

Diagram 2.2.2 (Part 2 of 2). Reader Input



From: MRJE Processing, 2.2

INPUT

PFCB for reader input task

Assign/free area

BSC buffer containing GFT sequence

PROCESS

1 Checks if any previous disk or keyboard reader input operation is active. If yes, saves the status of the previous reader operation.

2 Builds an IOB, DTF, or source get parameter list.

3 Verifies file name and attributes.

4 Sends an RFT sequence to the host system.

5 Waits for GFT sequence from the host system.

6 Invokes the reader input processor.

OUTPUT

PFCB for reader input task

IOB

DTF

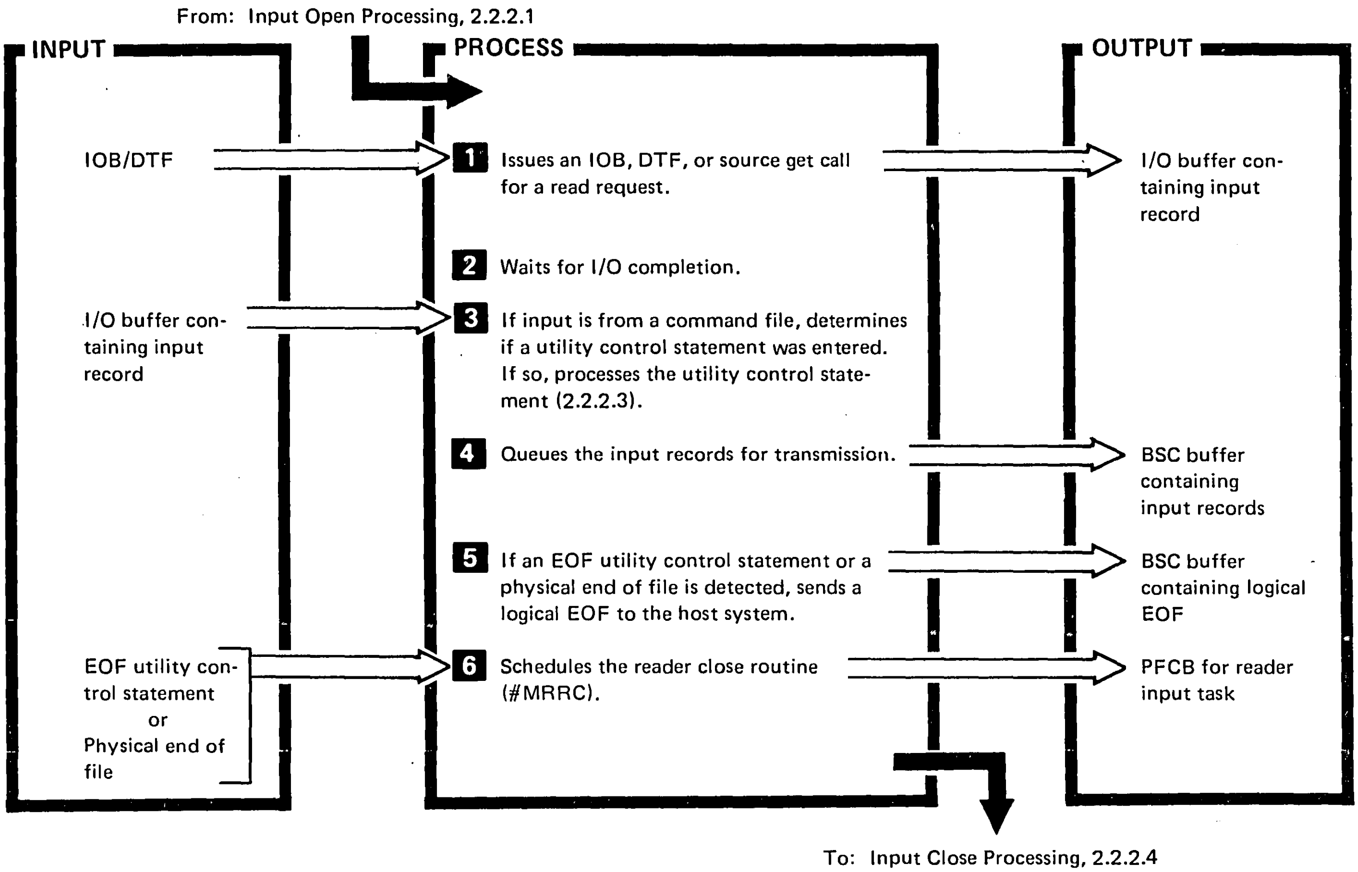
Parameter list

BSC buffer containing the RFT sequence

To: Reader Input Processing, 2.2.2.2

DESCRIPTION	MODULE/ ROUTINE
1 Checks the reader input PFCB to determine if a previous command file is being read. If so, saves the status of the file.	#MRRC
2 Creates an IOB, DTF, or source get parameter list to be used by the reader input processor (#MRRP, #MRFR, or #MRKP).	#MRRO
3 –	
4 Issues a message to indicate the reader is activated.	#MRRO
5 –	
6 –	

Diagram 2.2.2.1. Input Open Processing



From: Input Open Processing, 2.2.2.1

INPUT

IOB/DTF

I/O buffer containing input record

EOF utility control statement or Physical end of file

PROCESS

1 Issues an IOB, DTF, or source get call for a read request.

2 Waits for I/O completion.

3 If input is from a command file, determines if a utility control statement was entered. If so, processes the utility control statement (2.2.2.3).

4 Queues the input records for transmission.

5 If an EOF utility control statement or a physical end of file is detected, sends a logical EOF to the host system.

6 Schedules the reader close routine (#MRRC).

OUTPUT

I/O buffer containing input record

BSC buffer containing input records

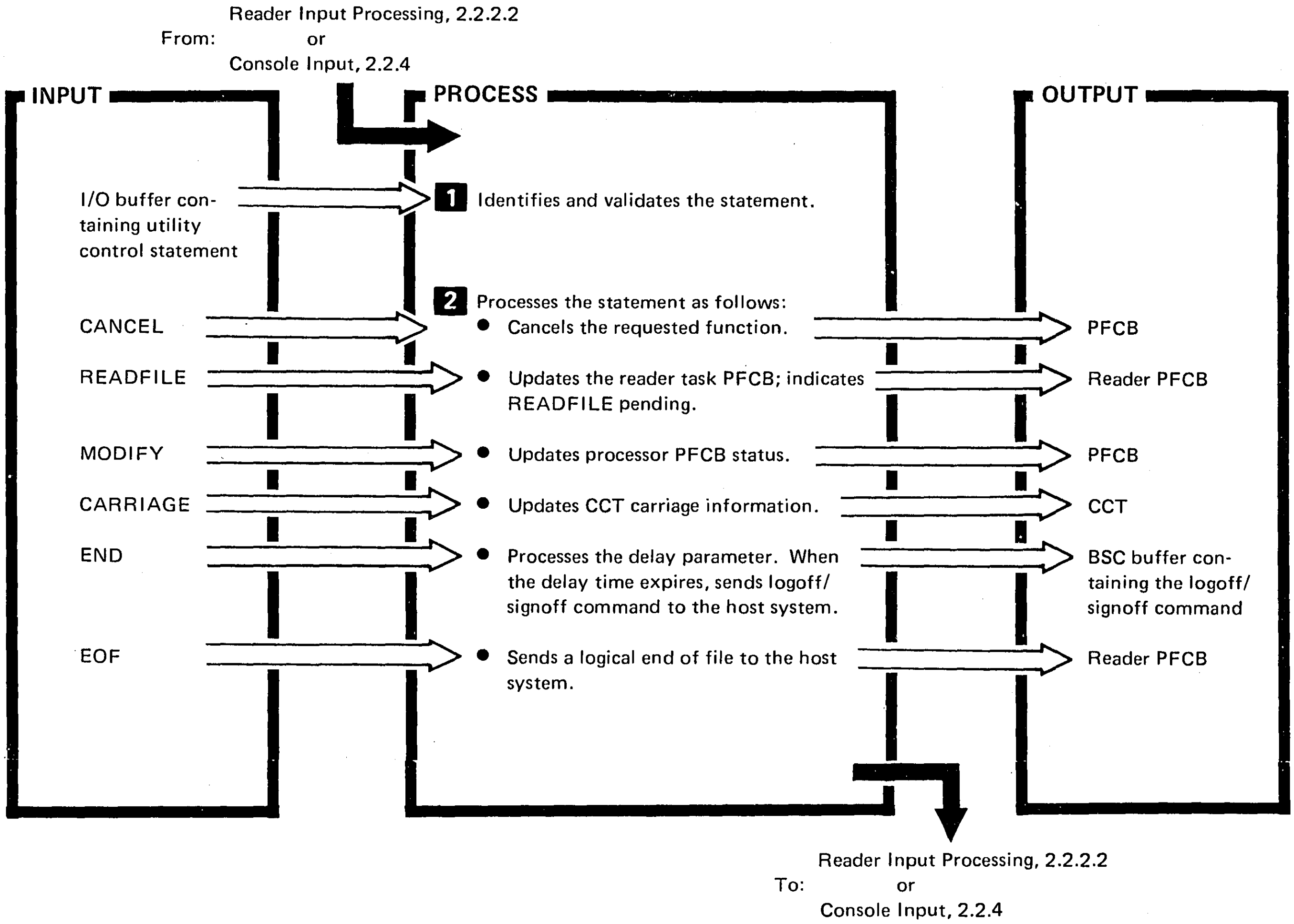
BSC buffer containing logical EOF

PFCB for reader input task

To: Input Close Processing, 2.2.2.4

DESCRIPTION	MODULE/ ROUTINE
1 Issues an I/O request for the next block of data.	#MRRP, #MRKP, or #MRFR
2 For disk input, blocks/deblocks the records into 80-byte physical records upon completion of the read operation.	#MRRP or #MRFR
3 If command file is specified, scans the whole file for utility control statements; if utility control statements are found, passes control to the command processor (#MRCO).	#MRCO
4 Compresses and/or truncates the record and inserts it into the BSC buffer.	#MRBM #MRFC
5 Repeats Steps 1 through 4 until an EOF utility control statement is entered or a physical end of file is encountered. When either of these conditions exists, sends a logical end of file to the host system. If the reader was the last active data stream and an END statement had been entered previously, the interval timer is started with the delay value specified in the END statement. **6** –	#MRRC

Diagram 2.2.2.2. Reader Input Processing


From: Reader Input Processing, 2.2.2.2 or Console Input, 2.2.4
INPUT
PROCESS
OUTPUT
I/O buffer containing utility control statement
CANCEL
READFILE
MODIFY
CARRIAGE
END
EOF
PFCB
Reader PFCB
PFCB
CCT
BSC buffer containing the logoff/signoff command
Reader PFCB
To: Reader Input Processing, 2.2.2.2 or Console Input, 2.2.4

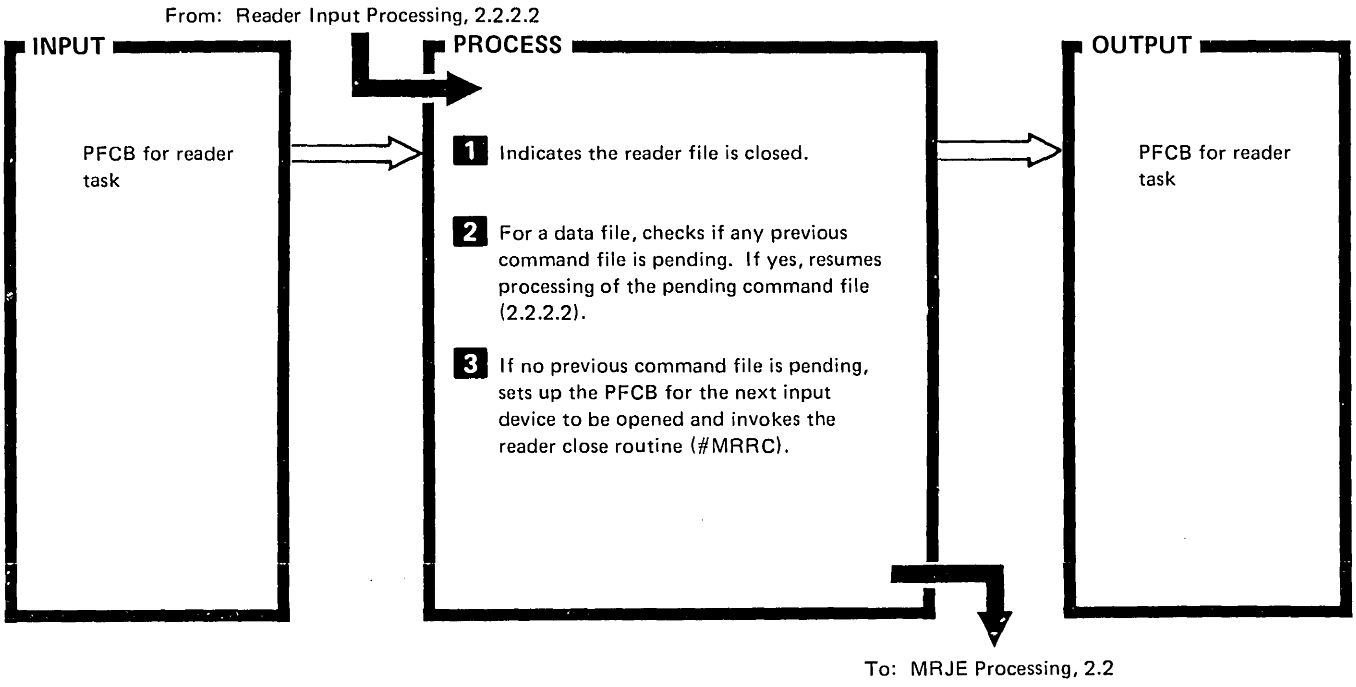
1 Identifies and validates the statement.

2 Processes the statement as follows:

- Cancels the requested function.
- Updates the reader task PFCB; indicates READFILE pending.
- Updates processor PFCB status.
- Updates CCT carriage information.
- Processes the delay parameter. When the delay time expires, sends logoff/signoff command to the host system.
- Sends a logical end of file to the host system.

DESCRIPTION	MODULE/ ROUTINE
1 –	#MRCO
2 Scans the utility control statement for the keyword and processes it further (by other routines) as follows:	#MRCO and #MRCS
● Processes the CANCEL statement by terminating the specified function.	#MRCO
● Processes the READFILE statement by updating the reader input PFCB.	#MRRF
● Processes the MODIFY statement by updating the correct PFCB.	#MRMO
● Processes the CARRIAGE statement by updating the CCT.	#MRCR
● Processes the END command by saving the delay value. When the printer, punch, and reader data streams close, sets the interval timer. When the delay time expires, sends the logoff/signoff command to the host system. Any time a print or punch data stream begins, the timer is canceled and reset when all data streams are closed.	#MRCO #MRKP #MROP
● –	

Diagram 2.2.2.3. Utility Control Statement Processing



From: Reader Input Processing, 2.2.2.2

INPUT

PFCB for reader task

PROCESS

1 Indicates the reader file is closed.

2 For a data file, checks if any previous command file is pending. If yes, resumes processing of the pending command file (2.2.2.2).

3 If no previous command file is pending, sets up the PFCB for the next input device to be opened and invokes the reader close routine (#MRRC).

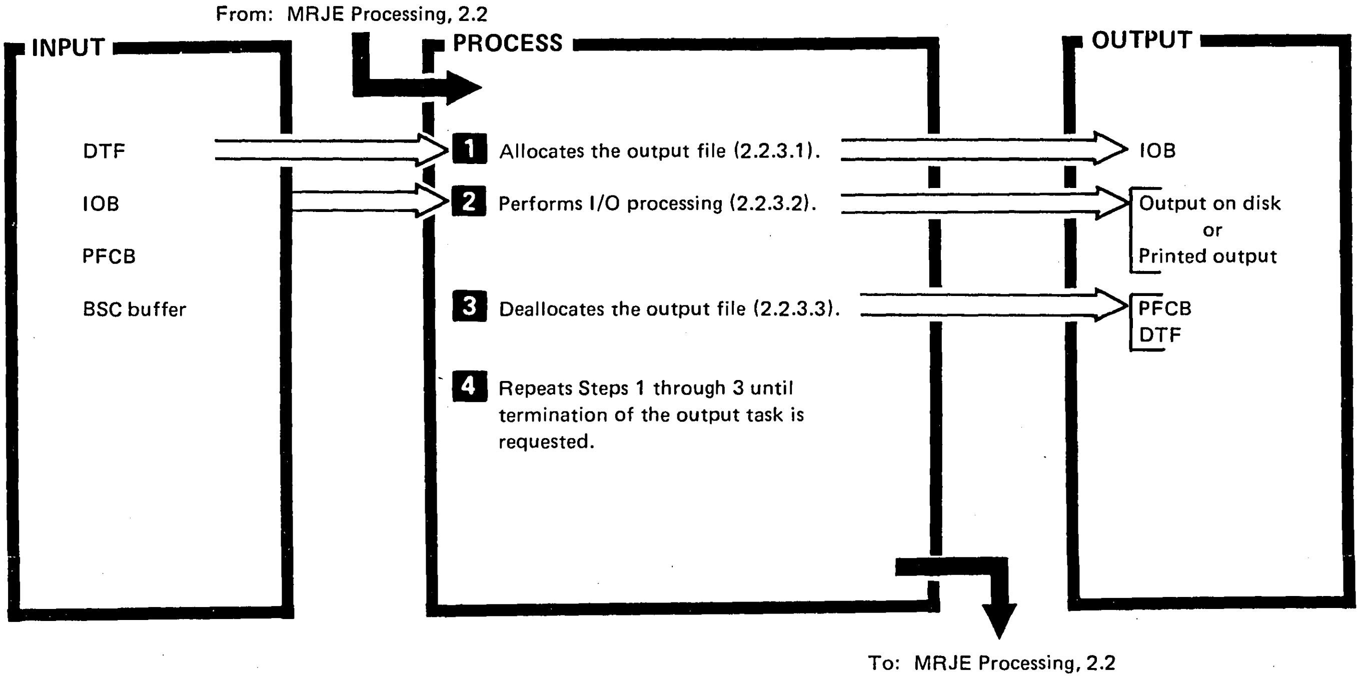
OUTPUT

PFCB for reader task

To: MRJE Processing, 2.2

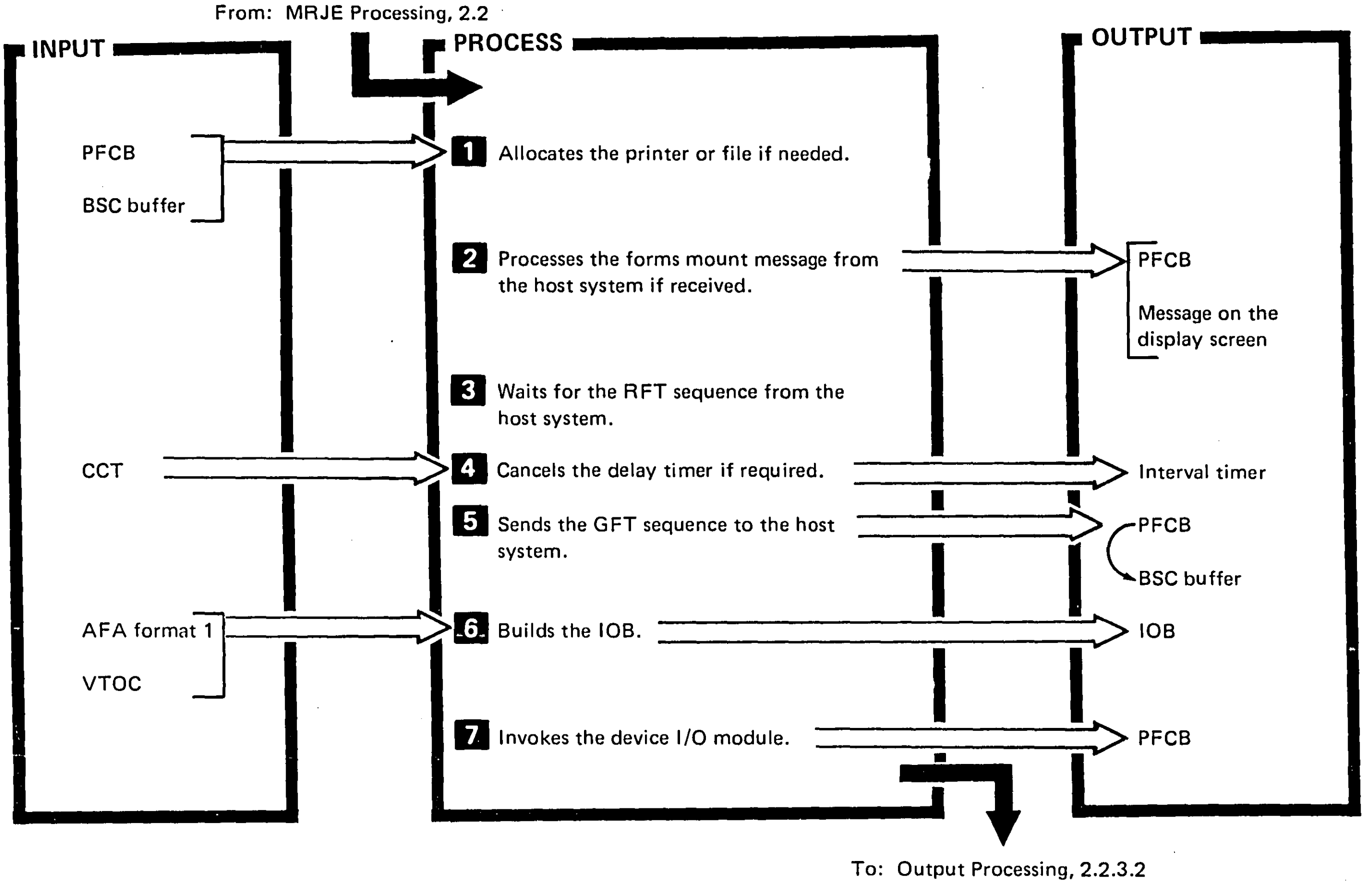
DESCRIPTION	MODULE/ ROUTINE
1 Updates the PFCB to indicate that the reader file is closed.	#MRRC
2 If a previous command file is pending, resumes processing the previous command file. If the previous command file is from disk, then rebuilds the disk IOB and goes to the reader disk input processor. If the previous command file is from keyboard, then goes to the keyboard reader processor (#MRKP).	
3 If no previous command file is pending, sends a logical end of file to the host system and sets the dispatching address in the PFCB to the address of the reader close routine. If the reader was the last active data stream and an END statement had been entered previously, the interval timer is set with the delay value specified on the END statement.	

Diagram 2.2.2.4. Input Close Processing


From: MRJE Processing, 2.2
INPUT
DTF
IOB
PFCB
BSC buffer
PROCESS
1 Allocates the output file (2.2.3.1).
2 Performs I/O processing (2.2.3.2).
3 Deallocates the output file (2.2.3.3).
4 Repeats Steps 1 through 3 until termination of the output task is requested.
OUTPUT
IOB
Output on disk
or
Printed output
PFCB
DTF
To: MRJE Processing, 2.2

DESCRIPTION	MODULE/ ROUTINE
Job output consists of processing printer and punch data streams sent by the host system. Processing is performed under the printer and punch PFCBs.	
1 Job output begins with the open function which: ● Allocates the printer or a disk file. ● Builds an IOB. ● Waits for an RFT sequence from the host system. ● Sends a GFT sequence to the host system on receipt of the RFT sequence. ● Exits to the proper device I/O module.	#MROP, #MROD, and #MRBM
2 As data is received from the host system, decompresses data unless it is going to the TDISKPR1 file (removes embedded control characters and puts the data in a readable form). Writes the data to the assigned device. The decompression/write cycle continues until a logical end-of-file sequence is received from the host system.	#MRBM and #MRDP or #MRUP
3 When an EOF condition occurs, the close routine (#MRCL) updates the PFCB to indicate that the printer/punch function is closed and that the assigned device/file is closed. If output was to disk, deallocates the file unless the output was to TDISKPR1.	#MRCL
4 –	

Diagram 2.2.3. Printer and Punch Output



1 Allocates the printer or file if needed.

2 Processes the forms mount message from the host system if received.

3 Waits for the RFT sequence from the host system.

4 Cancels the delay timer if required.

5 Sends the GFT sequence to the host system.

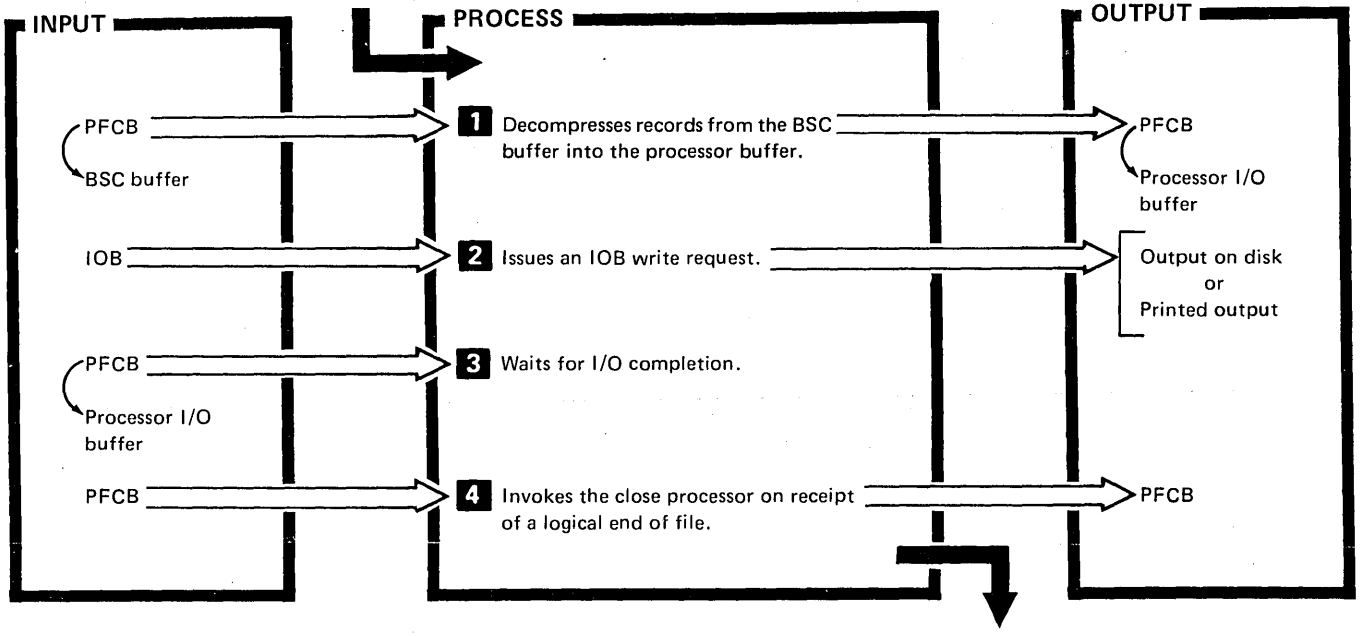
6 Builds the IOB.

7 Invokes the device I/O module.

DESCRIPTION	MODULE/ ROUTINE
1 –	#MROP and #MROD
2 Processes the forms mount messages from the host system to ensure that the correct forms are mounted or that the data is going to be written to disk.	
3 –	
4 If an END statement had been entered previously, cancels the interval timer.	
5 –	
6 Builds the IOB and any other required control blocks (in the work area).	
7 Dispatches the correct output processor (#MRDP for disk output or #MRUP for printer output) by setting the dispatching address in the PFCB to the address of the processor.	#MRDP and #MRUP

Diagram 2.2.3.1. Output Open Processing

From: Output Open Processing, 2.2.3.1



1 Decompresses records from the BSC buffer into the processor buffer.

2 Issues an IOB write request.

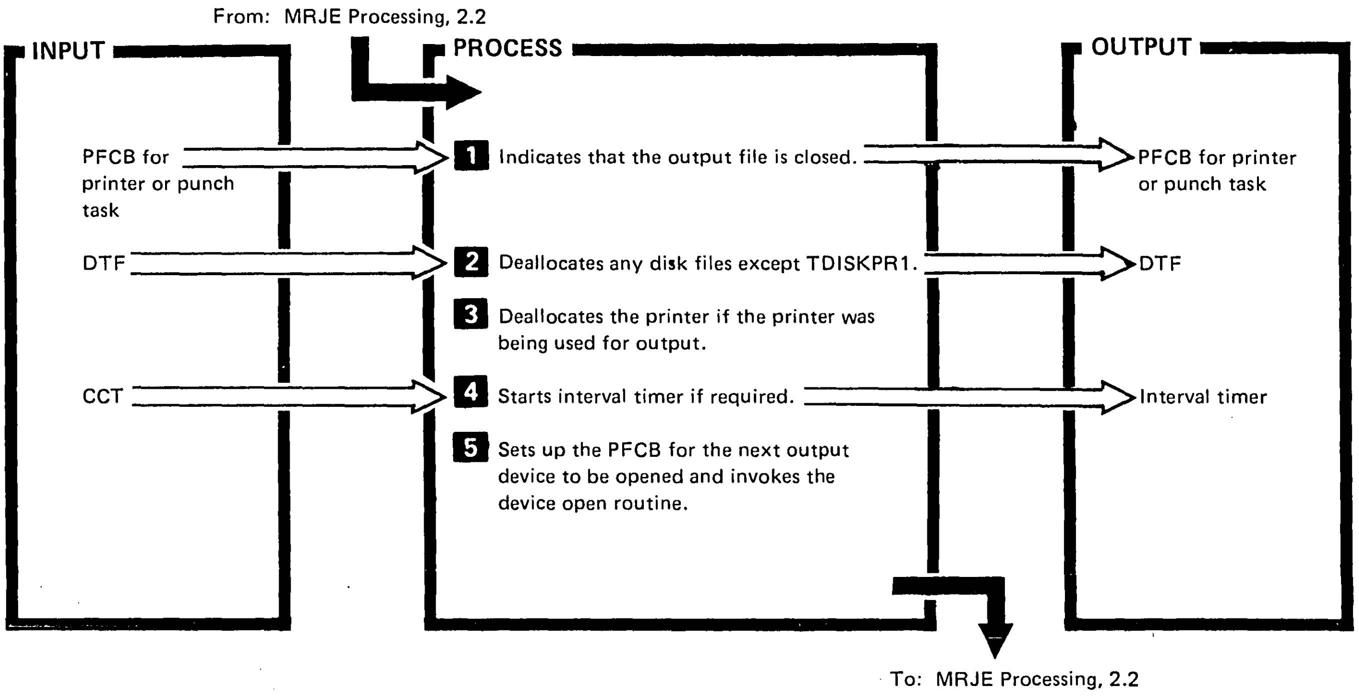
3 Waits for I/O completion.

4 Invokes the close processor on receipt of a logical end of file.

To: Output Close Processing, 2.2.3.3

DESCRIPTION	MODULE/ ROUTINE
1 Receives and decompresses the next record.	#MRBM
2 Writes the record to the current device.	#MRDP or #MRUP
3 –	
4 Repeats Steps 1 through 3 until a logical end of file is received from the host system. When a logical end of file is received, the I/O processor sets the dispatching address in the PFCB to the address of the close routine (#MRCL).	

Diagram 2.2.3.2 Output Processing


From: MRJE Processing, 2.2
INPUT
PFCB for printer or punch task
DTF
CCT
PROCESS
OUTPUT
PFCB for printer or punch task
DTF
Interval timer
To: MRJE Processing, 2.2

1 Indicates that the output file is closed.

2 Deallocates any disk files except TDISKPR1.

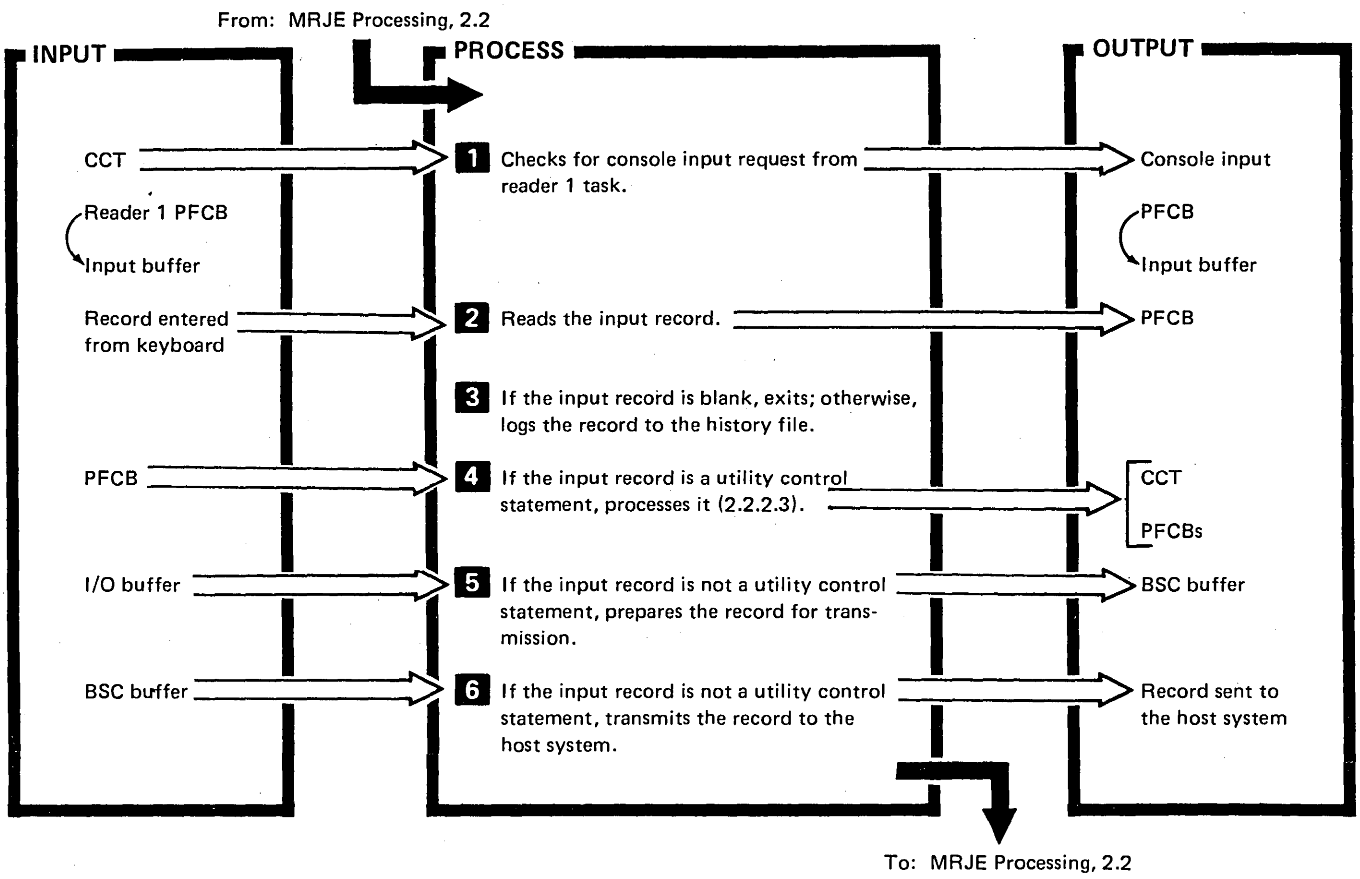
3 Deallocates the printer if the printer was being used for output.

4 Starts interval timer if required.

5 Sets up the PFCB for the next output device to be opened and invokes the device open routine.

DESCRIPTION	MODULE/ ROUTINE
1 Updates the PFCB to indicate that the output file is closed and performs all cleanup processing.	#MRCL
2 –	
3 –	
4 If the output file was the last active data stream and an END statement had been entered previously, the system interval timer is started using the delay value specified on the END statement.	
5 Updates the PFCB for the next device to be used (the device specified on the MODIFY utility control statement) and sets the dispatching address in the PFCB to the address of the open routine (#MROP).	

Diagram 2.2.3.3. Output Close Processing



1 Checks for console input request from reader 1 task.

2 Reads the input record.

3 If the input record is blank, exits; otherwise, logs the record to the history file.

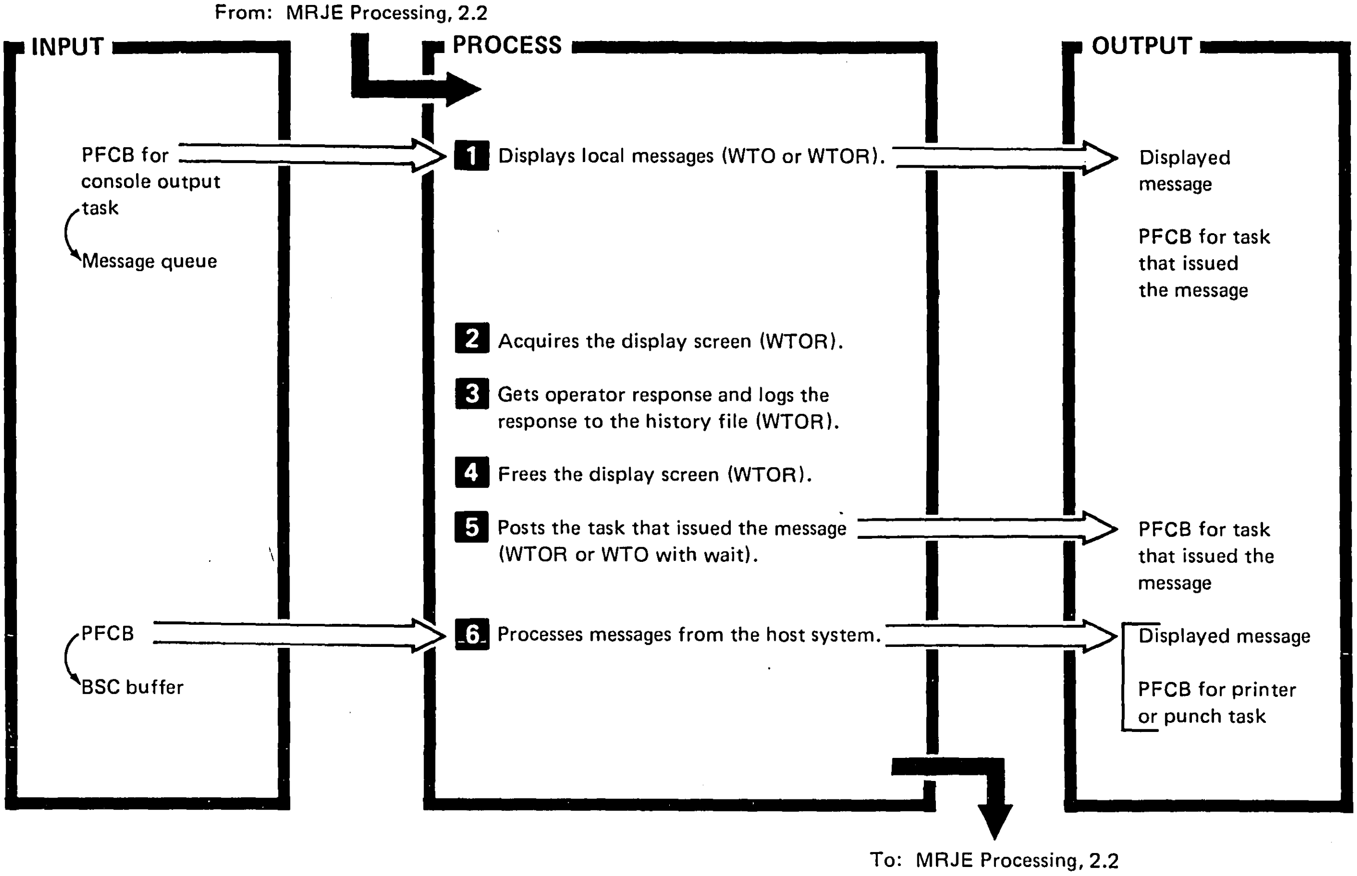
4 If the input record is a utility control statement, processes it (2.2.2.3).

5 If the input record is not a utility control statement, prepares the record for transmission.

6 If the input record is not a utility control statement, transmits the record to the host system.

DESCRIPTION	MODULE/ ROUTINE
The dispatcher invokes the console input processor when the system console is not being used by the keyboard reader input function and is not being used for a message response.	
1 If a reader console input request is pending and not processed, checks the reader 1 input buffer. If blank, normal console input proceeds; otherwise, copies the reader buffer to the console buffer and processes it. If a reader console input request is pending and processed, posts the reader 1 task, suspends console input processing, and exits.	#MRCP
2 Reads the input record and checks to determine if it is a utility control statement.	
3 Blank records are ignored; all others are logged to the history file.	
4 Decodes the utility control statement and takes the correct action.	#MRCO, #MRCS, and either #MRCR, #MRRF, or #MRMO
5 If the input record is not a utility control statement, the console input processor compresses the record and inserts it into the BSC buffer.	#MRBM and #MRFC
6 Transmits this buffer to the host system.	

Diagram 2.2.4. Console Input


From: MRJE Processing, 2.2
INPUT
PROCESS
OUTPUT
PFCB for console output task
Message queue
PFCB
BSC buffer
Displayed message
PFCB for task that issued the message
PFCB for task that issued the message
Displayed message
PFCB for printer or punch task
To: MRJE Processing, 2.2

1 Displays local messages (WTO or WTOR).

2 Acquires the display screen (WTOR).

3 Gets operator response and logs the response to the history file (WTOR).

4 Frees the display screen (WTOR).

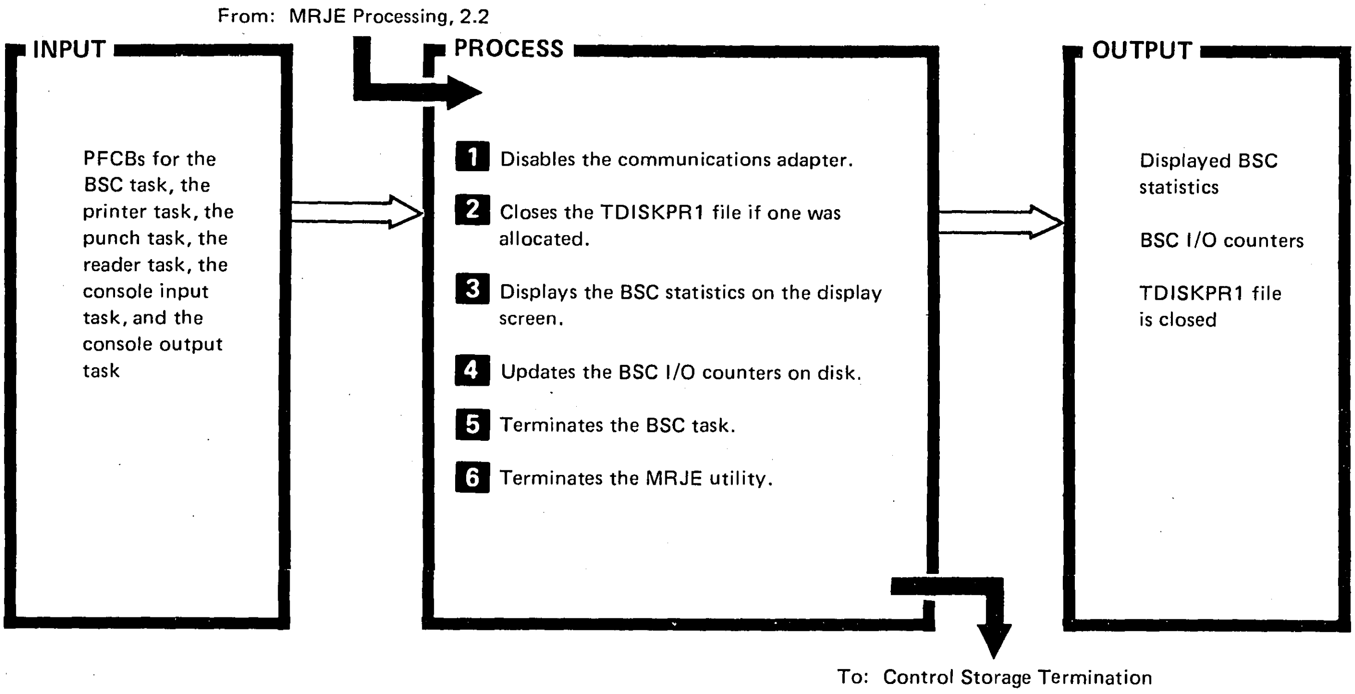
5 Posts the task that issued the message (WTOR or WTO with wait).

6 Processes messages from the host system.

DESCRIPTION	MODULE/ ROUTINE
1 ● Retrieves the message request element from the message queue. ● Fetches the message text from the message member. ● Displays the message and logs the message to the history file. 2 Because the display screen might already be in use by the console input or reader task, the display screen must be acquired to request the operator response. 3 Displays a message that requests a reply to the local message. The operator response is checked to ensure that a valid option was selected. If the response is not valid, redisplays the message requesting a reply along with an error message until a valid response is entered. Logs the valid response to the history file. 4 Frees the display screen for use by any other functions requiring operator input. 5 Posts the task (PFCB) under which the message was generated, if required. *Note:* A WTO with wait is a message that does not require an operator response, but does suspend the function until the message is displayed to the operator.	#MRCD
6 ● Decompresses the message and scans for a mount message. ● Processes the mount message, if found. ● Displays the message.	#MRBM and #MRHD

Diagram 2.2.5. Console Output

From: MRJE Processing, 2.2



INPUT

PFCBs for the BSC task, the printer task, the punch task, the reader task, the console input task, and the console output task

PROCESS

1 Disables the communications adapter.

2 Closes the TDISKPR1 file if one was allocated.

3 Displays the BSC statistics on the display screen.

4 Updates the BSC I/O counters on disk.

5 Terminates the BSC task.

6 Terminates the MRJE utility.

OUTPUT

Displayed BSC statistics

BSC I/O counters

TDISKPR1 file is closed

To: Control Storage Termination

DESCRIPTION	MODULE/ ROUTINE
Termination of the MRJE session is caused by one of the following: ● The successful transmission of a logoff/signoff command to the host system and the completion of all functions. ● A 2 option chosen in response to a BSC disconnect message. ● A system CANCEL. ● A processor check in the MRJE or BSC task, or a permanent I/O error on the system console. ● A 2 or 3 option taken during an inquiry request from the system console.	
1 Determines if the communications adapter is disabled. If not, issues an adapter disable instruction.	$MRJE, #MRTM, and #MRBP
2 –	
3 Converts the BSC line statistics in the BSC PFCB to EDCDIC, displays them on the display screen, and logs them to the history file.	
4 –	
5 The BSC task is posted and goes to end of job.	#MRJE and #MRBP
6 The MRJE dispatcher goes to end of job.	$MRJE

Diagram 2.3. Termination

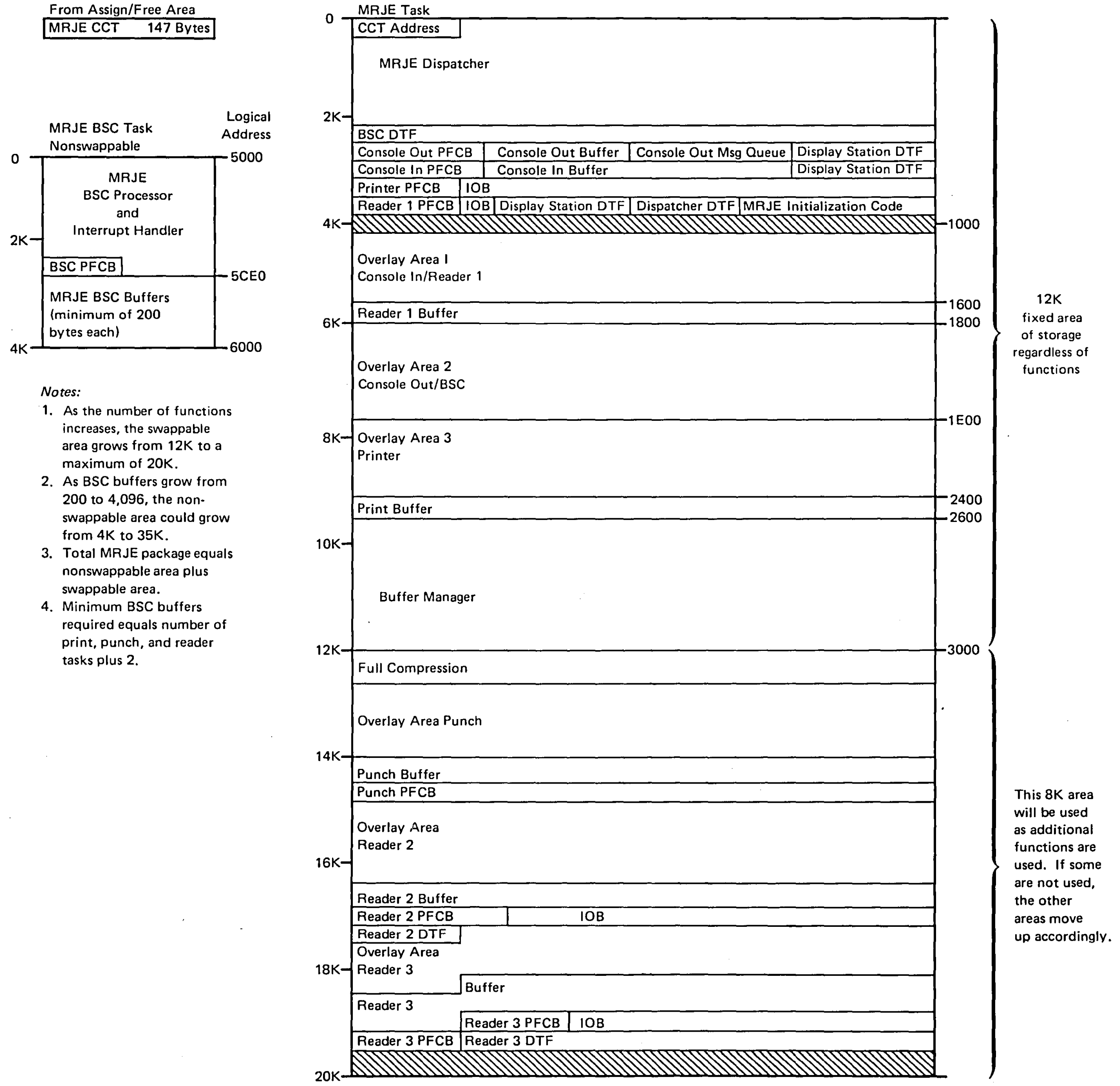
Program Organization

This section includes:

- A storage map indicating the layout of the program
- A description of each module within the program
- A description of the control flow between the modules

MRJE STORAGE MAP

Figure 2-2 shows the main storage layout of the MRJE utility.



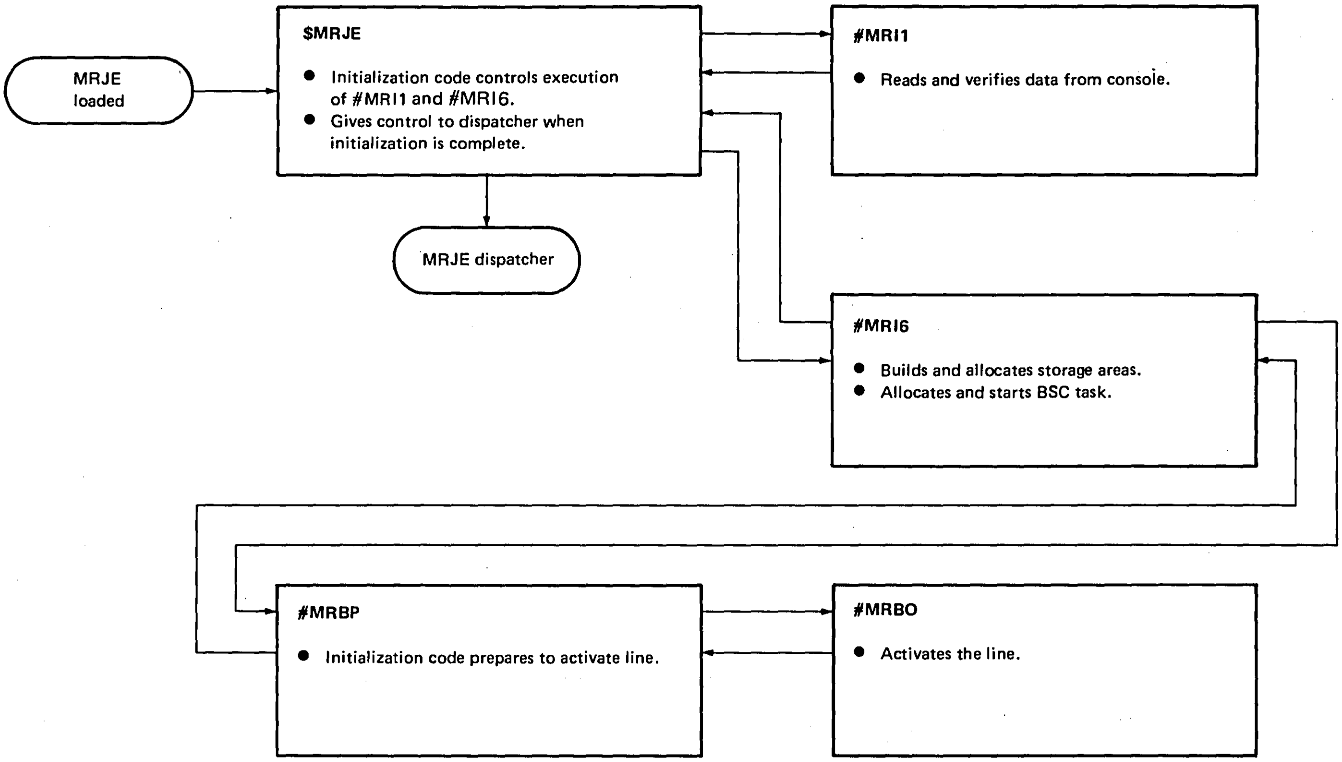
Notes:

1. As the number of functions increases, the swappable area grows from 12K to a maximum of 20K.
2. As BSC buffers grow from 200 to 4,096, the nonswappable area could grow from 4K to 35K.
3. Total MRJE package equals nonswappable area plus swappable area.
4. Minimum BSC buffers required equals number of print, punch, and reader tasks plus 2.

Figure 2-2. MRJE Storage Map

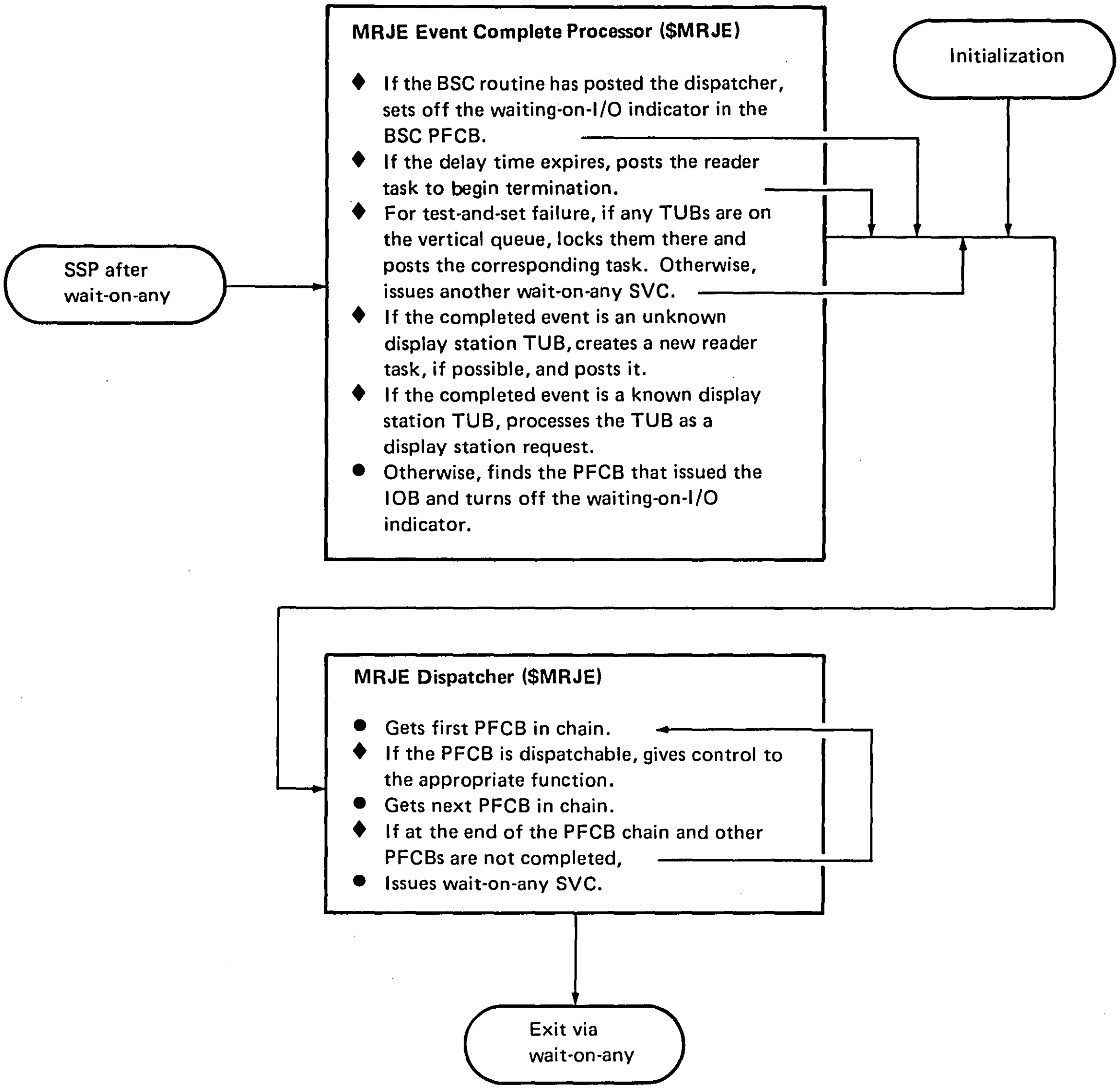
CONTROL FLOW DIAGRAMS

Figures 2-3 through 2-9 are control flow diagrams that describe each module within the program and the control flow between the modules. A bullet (●) indicates a process step and a diamond (♦) indicates a decision.



MRJE loaded
$MRJE
● Initialization code controls execution of #MRI1 and #MRI6.
● Gives control to dispatcher when initialization is complete.
MRJE dispatcher
#MRI1
● Reads and verifies data from console.
#MRI6
● Builds and allocates storage areas.
● Allocates and starts BSC task.
#MRBP
● Initialization code prepares to activate line.
#MRBO
● Activates the line.

Figure 2-3. MRJE Initialization



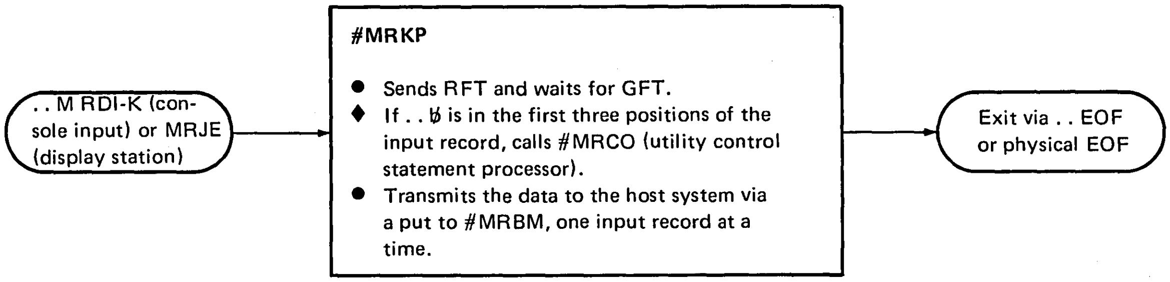
MRJE Event Complete Processor ($MRJE)

- ◆ If the BSC routine has posted the dispatcher, sets off the waiting-on-I/O indicator in the BSC PFCB.
- ◆ If the delay time expires, posts the reader task to begin termination.
- ◆ For test-and-set failure, if any TUBs are on the vertical queue, locks them there and posts the corresponding task. Otherwise, issues another wait-on-any SVC.
- ◆ If the completed event is an unknown display station TUB, creates a new reader task, if possible, and posts it.
- ◆ If the completed event is a known display station TUB, processes the TUB as a display station request.
- ● Otherwise, finds the PFCB that issued the IOB and turns off the waiting-on-I/O indicator.

MRJE Dispatcher ($MRJE)

- ● Gets first PFCB in chain.
- ◆ If the PFCB is dispatchable, gives control to the appropriate function.
- ● Gets next PFCB in chain.
- ◆ If at the end of the PFCB chain and other PFCBs are not completed,
- ● Issues wait-on-any SVC.

Figure 2-4. MRJE Mainline


. . M RDI-K (console input) or MRJE (display station)
Exit via . . EOF or physical EOF

#MRKP

- ● Sends RFT and waits for GFT.
- ◆ If . . ƀ is in the first three positions of the input record, calls #MRCO (utility control statement processor).
- ● Transmits the data to the host system via a put to #MRBM, one input record at a time.

Figure 2-5. Reader (Display Station)

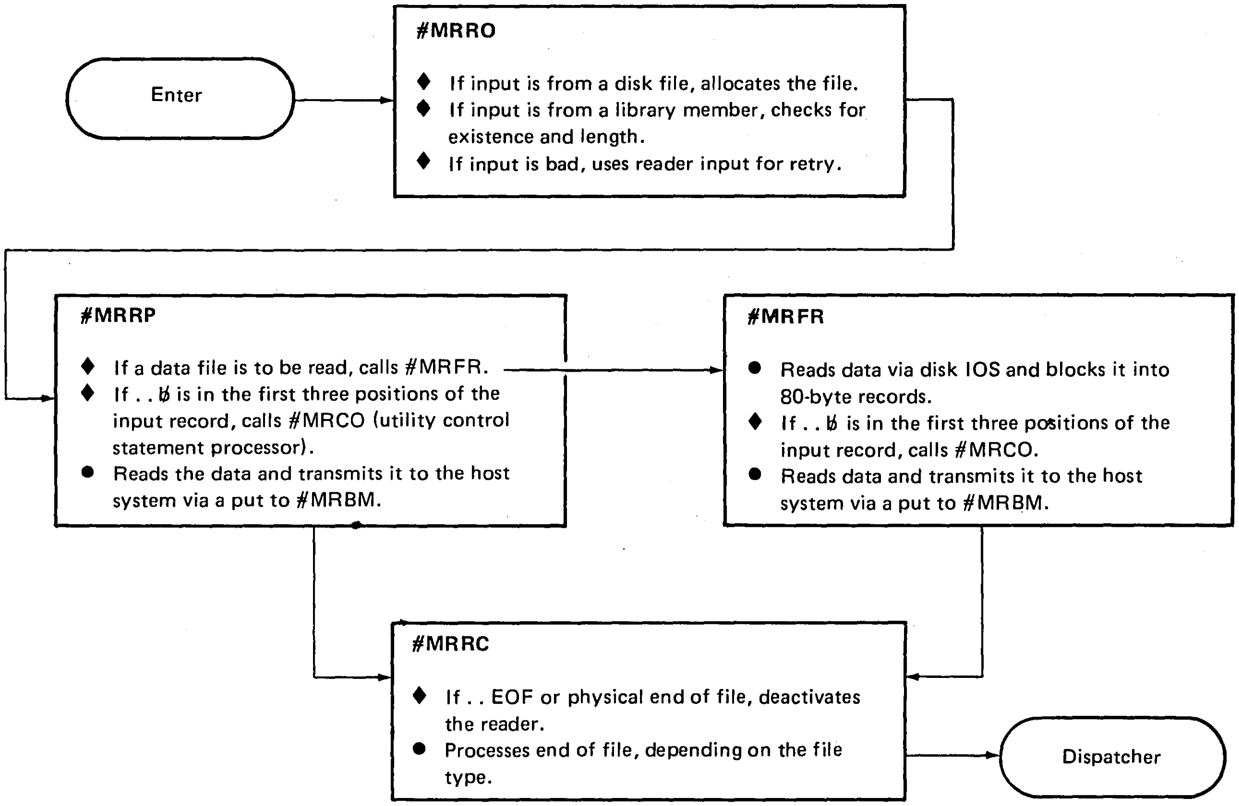

Enter
#MRRO
♦ If input is from a disk file, allocates the file.
♦ If input is from a library member, checks for existence and length.
♦ If input is bad, uses reader input for retry.
#MRRP
♦ If a data file is to be read, calls #MRFR.
♦ If . . ␢ is in the first three positions of the input record, calls #MRCO (utility control statement processor).
● Reads the data and transmits it to the host system via a put to #MRBM.
#MRFR
● Reads data via disk IOS and blocks it into 80-byte records.
♦ If . . ␢ is in the first three positions of the input record, calls #MRCO.
● Reads data and transmits it to the host system via a put to #MRBM.
#MRRC
♦ If . . EOF or physical end of file, deactivates the reader.
● Processes end of file, depending on the file type.
Dispatcher

Figure 2-6. Reader (Disk)

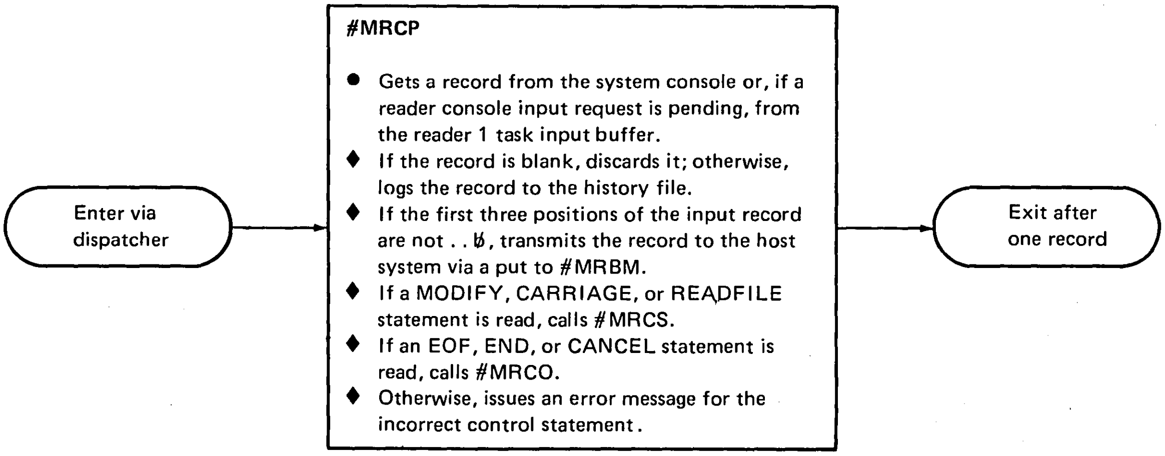

Enter via dispatcher
#MRCP
● Gets a record from the system console or, if a reader console input request is pending, from the reader 1 task input buffer.
♦ If the record is blank, discards it; otherwise, logs the record to the history file.
♦ If the first three positions of the input record are not . . ␢, transmits the record to the host system via a put to #MRBM.
♦ If a MODIFY, CARRIAGE, or READFILE statement is read, calls #MRCS.
♦ If an EOF, END, or CANCEL statement is read, calls #MRCO.
♦ Otherwise, issues an error message for the incorrect control statement.
Exit after one record

Figure 2-7. Console Input

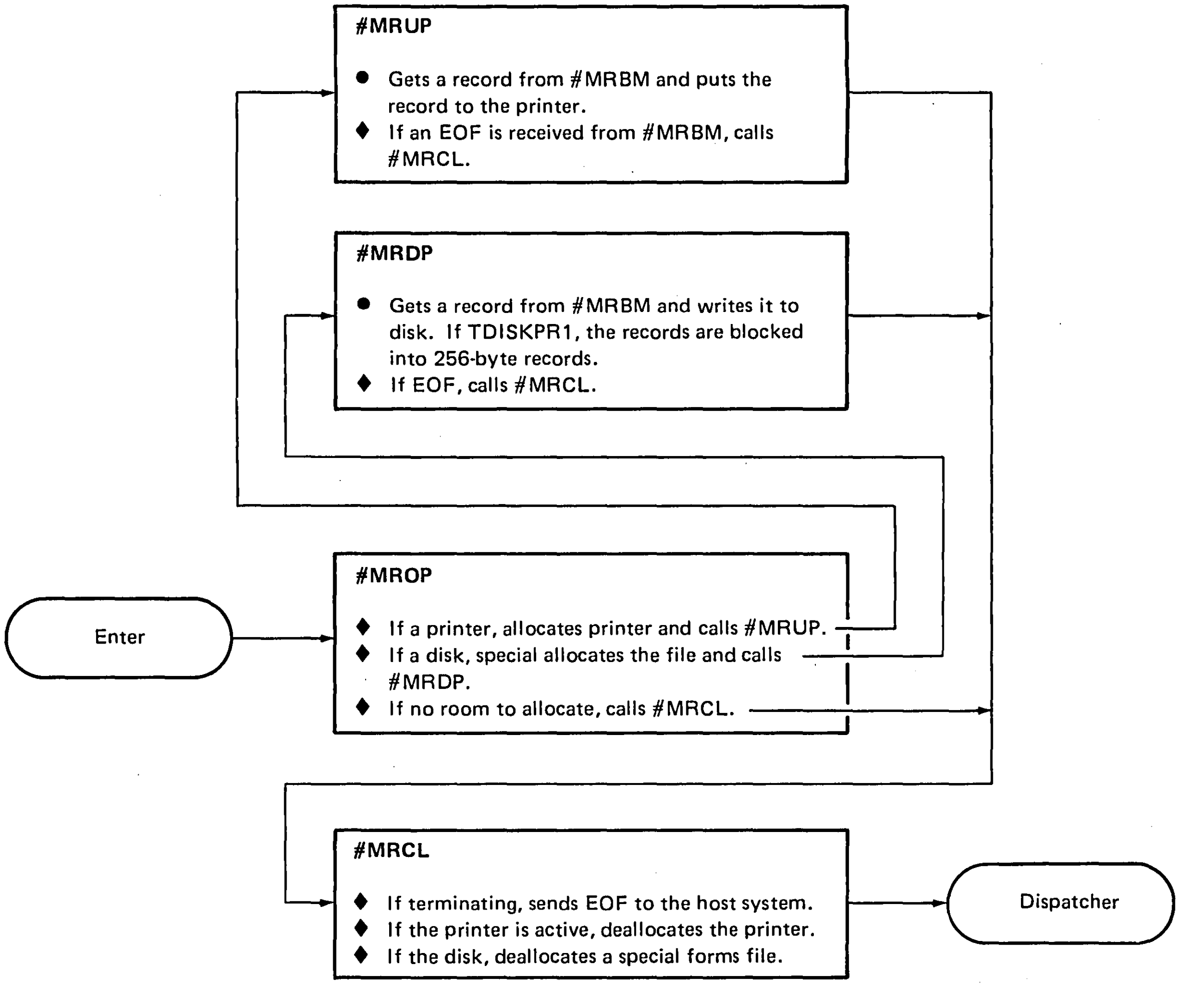

#MRUP
● Gets a record from #MRBM and puts the record to the printer.
♦ If an EOF is received from #MRBM, calls #MRCL.
#MRDP
● Gets a record from #MRBM and writes it to disk. If TDISKPR1, the records are blocked into 256-byte records.
♦ If EOF, calls #MRCL.
Enter
#MROP
♦ If a printer, allocates printer and calls #MRUP.
♦ If a disk, special allocates the file and calls #MRDP.
♦ If no room to allocate, calls #MRCL.
#MRCL
♦ If terminating, sends EOF to the host system.
♦ If the printer is active, deallocates the printer.
♦ If the disk, deallocates a special forms file.
Dispatcher

Figure 2-8. Disk or Printer Output

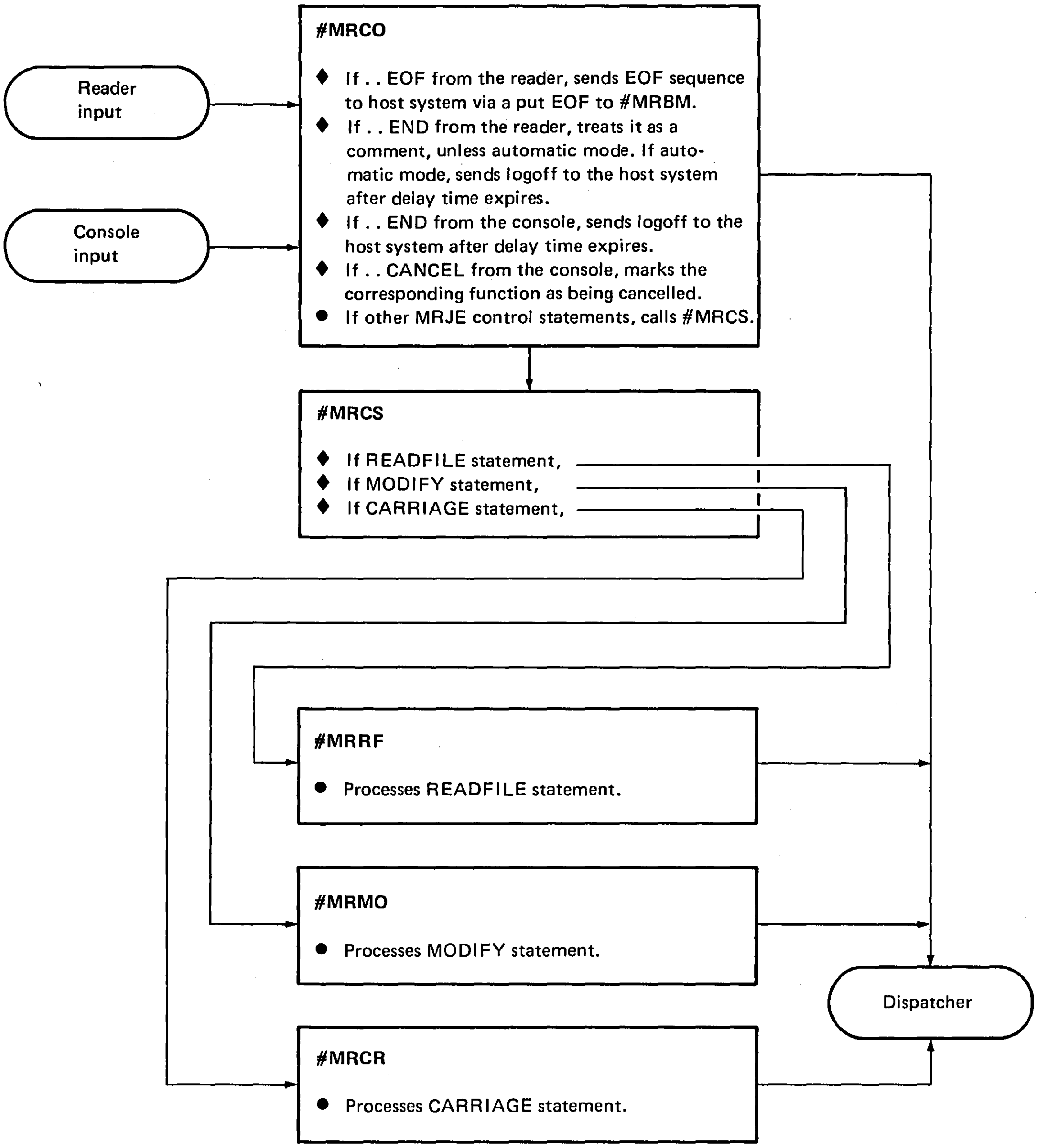

Reader input
Console input
#MRCO
♦ If . . EOF from the reader, sends EOF sequence to host system via a put EOF to #MRBM.
♦ If . . END from the reader, treats it as a comment, unless automatic mode. If automatic mode, sends logoff to the host system after delay time expires.
♦ If . . END from the console, sends logoff to the host system after delay time expires.
♦ If . . CANCEL from the console, marks the corresponding function as being cancelled.
● If other MRJE control statements, calls #MRCS.
#MRCS
♦ If READFILE statement,
♦ If MODIFY statement,
♦ If CARRIAGE statement,
#MRRF
● Processes READFILE statement.
#MRMO
● Processes MODIFY statement.
#MRCR
● Processes CARRIAGE statement.
Dispatcher

Figure 2-9. Utility Control Statement Processor

Directory

This section consists of a quick-reference table that can be used to find MRJE modules on microfiche program listings. The table contains:

- Module name
- Diagram number
- Descriptive name
- Function

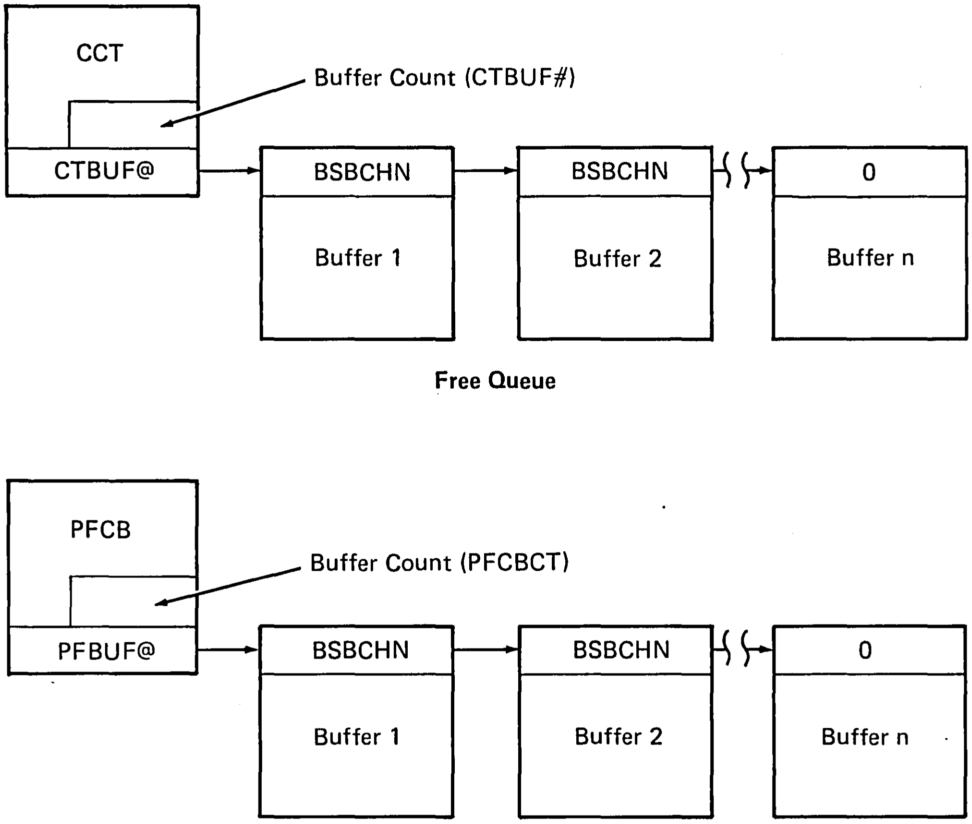
Module Name	Diagram Number	Descriptive Name	Function
$MRJE	–	Dispatching supervisor	Dispatches control to tasks whose PFCBs indicate they have work to do. Also processes waits and posts.
#MRBE	2.2.1	BSC error recording routine	Uses system logging to handle BSC errors.
#MRBM	2.2.2 2.2.3 2.2.4 2.2.5	Buffer manager	Manages the BSC buffers by performing moves, puts, gets, and purges as required.
#MRBO	2.1	BSC open	Establishes communications with the host system and sets up to send sign-on command.
#MRBP	2.2 2.2.1	BSC I/O processor	BSC mainline I/O processor and interrupt handler.
#MRCD	2.2 2.2.5	Console output processor	Controls all informational and operator reply messages.
#MRCL	2.2.3	Printer/punch close routine	Closes all disk files except for TDISKPR1, or deallocates the printer. Sets up the PFCB for the next device to be opened.
#MRCO	2.2.2 2.2.4	Utility control statement processor	Processes MRJE utility control statements.
#MRCP	2.2	Console input processor	Processes all information entered for the console input task.
#MRCR	2.2.2.3 2.2.4	CARRIAGE utility control statement processor	Processes the CARRIAGE utility control statement.
#MRCS	2.2.2.3 2.2.4 2.2.5	Utility control statement scanner	Scans a utility control statement and invokes the appropriate processor.

Module Name	Diagram Number	Descriptive Name	Function
#MRDP	2.2 2.2.3	Disk output processor	Processes all MRJE disk output.
#MRFC	2.2.2 2.2.4	Full compression routine	Compresses console input or reader input data that is transmitted to the host system.
#MRFR	2.2 2.2.2	Reader disk file	Processes all information read from disk files by the reader task.
#MRHD	2.2 2.2.5	Host message processor	Displays received host messages on the system console and processes mount messages from the host system.
#MRI1	2.1	Program load data and initialization processor	Processes the program load data and the initialization display information.
#MRI6	2.1	Storage allocation routine	Determines the limits of MRJE storage and creates and initializes the MRJE data areas.
#MRKP	2.2 2.2.2	Reader keyboard input processor	Processes all information entered from a keyboard under the reader task.
#MRMO	2.2.2.3 2.2.4	MODIFY utility control statement processor	Processes the MODIFY utility control statement.
#MROD	2.2.3.1	Open disk processing routine	Performs the open operations for MRJE disk output.
#MROP	2.2.3.1	Open printer processing routine	Performs the open operations for all MRJE printer output.
#MRRC	2.2	Reader close routine	Performs the close operations for information read by the reader task.
#MRRF	2.2.2.3 2.2.4	READFILE utility control statement processor	Processes the READFILE utility control statement.
#MRRO	2.2.2	Reader open routine	Performs the open operations for disk files or library members read by the reader task.
#MRRP	2.2 2.2.2	Reader source get input processor	Processes all information read from members.
#MRTM	2.3	Terminator	Disables the adapter and displays a summary of BSC errors, deallocates TDISKPR1, and updates BSC I/O counters.
#MRUP	2.2 2.2.3	Printer output processor	Controls all printer processing.

Data Areas

BSC BUFFER

BSC buffers are maintained in a free queue until needed for data transmission or until needed as buffers for receiving data from the host system. Following their use, buffers are returned to the free queue for later use. While in use, BSC buffers are maintained in a processor queue. The free queue is addressed by the field CTBUF@ in the CCT while the processor queue is addressed by the field PFBUF@ in each task's PFCB. Figure 2-10 illustrates BSC buffer queueing.



Free Queue

Processor Queue

Figure 2-10. Illustration of Buffer Queueing

Figure 2-11 shows the general format of a BSC buffer that is used for communication between the central and remote computers.

DLE	BSC leader (DLE if transparency, SOH if no transparency)	These 2 bytes are overlaid by BSBCHN (chain pointer to next buffer).
STX	BSC start of text	
BCB	Block control byte	
FCS	Function control sequence	FCS bytes become the pointer to the next RCB when the BSC buffer becomes a processor buffer.
FCS	Function control sequence	
RCB	Record control byte for record 1	
SRCB	Subrecord control byte for record 1	
SCB	String control byte for record 1	
DATA	Character string (1-63 characters)	
SCB	String control byte for record 1	
DATA	Character string (1-63 characters)	
SCB	Terminating string control byte for record 1	
RCB	Record control byte for record 2	
SRCB	Subrecord control byte for record 2	
SCB	String control byte for record 2	
DATA	Character string (1-63 characters)	
SCB	Terminating string control byte for record 2	
RCB	Record control byte transmission block terminator	
DLE	BSC leader (DLE if transparency, SYN if no transparency) (not found in BSC buffer)	
ETB	BSC ending sequence	

Figure 2-11. General Format of a BSC Buffer

A typical BSC buffer for the MRJE utility is formatted as shown in Figure 2-12.

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
00	BSBCHN	2	Chain pointer (address +1) to next buffer. Zero if last buffer on the chain. Overlaid during transmit and receive operations.
00	BSBSOH	1	BSC start of header (SOH), or data link escape (DLE) if transparency feature used.
01	BSBSTX	1	BSC start of text (STX).
02	BSBBCB	1	Block control byte (BCB). 0 X X X C C C C 0 7 Usage: Transmission block status and sequence count. Bit meanings: 0 = 1 (must always be on). XXX = Control information as follows: = 000 Normal block = 001 Bypass sequence count validation = 010 Reset expected block sequence count to CCCC = 011 Reserved = 100 Reserved = 101 Not supported = 110 Not supported = 111 Reserved for future expansion CCCC = Modulus 16 block sequence count.
03	BSBFCS	2	Function control sequence (FCS). O S R R A B C D | O T R R W X Y Z 0 7 8 15 Usage: To control the flow of individual function streams. Bit meanings: O = 1 (must always be on).

Figure 2-12 (Part 1 of 6). Sample BSC Buffer for MRJE

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
			S = 1 Suspend *all* stream transmission (wait-a-bit). 0 Normal state T = Remote console stream identifier 1 Console on 0 Console off RR = Reserved for future expansion A = 1 Printer 1 on 0 Printer 1 off Z = 1 Punch 1 on 0 Punch 1 off *Note:* Bits BCD and WXY are not supported on System/34 and are always 1's. If a bit is on = continue function transmission; if a bit is off = suspend function transmission. individual functions.)
05	BSBRCB	1	Record control byte (RCB) | O I I I T T T T | 0 7 Usage: To identify each record type within a transmission block. Bit meanings: O = 0 End of transmission block (IIITTTT = 0) 1 All other RCBs III = Stream identifier is used to identify streams of multiple identifical functions (multiple print streams to a multiple printer terminal, and so on). = Control information if TTTT = 0 (control record) = 000 Reserved for future expansion = 001 Request to initiate a function transmission (prototype RCB for function in SRCB) = 010 Permission to initiate a function transmission (RCB for function contained in SRCB) = 011 Reserved = 100 Reserved = 101 Not supported = 110 Not supported = 111 General control record (type indicated in SRCB)

Figure 2-12 (Part 2 of 6). Sample BSC Buffer for MRJE

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
			TTTT = Record-type identifier = 0000 Control record = 0001 Operator message display request = 0010 Operator command = 0011 Normal input record = 0100 Normal print record = 0101 Normal punch record = 0110-1111 Not supported Examples: hex 91 = Console output hex 92 = Console input hex 93 = Reader 1 hex 94 = Printer 1 hex 95 = Punch 1 hex A3 = Reader 2 hex B3 = Reader 3
06	BSBSRC	1	Subrecord control byte (SRCB) O S S S S S S S 0 7 Usage: To provide supplemental information about a record. Bit meanings: O = 1 (must always be on) SSSSSSS = Additional record information; actual content is dependent on following record type listed: *SRCB for General Control Record* (character) 0 7 Usage: To identify the type of generalized control records. Bit meanings: character = A Initial terminal signon/logon = B Final terminal signoff/logoff = C Print initialization record = D Punch initialization record = E Input initialization record = F Data set transmission initialization

Figure 2-12 (Part 3 of 6). Sample BSC Buffer for MRJE

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description

= G System configuration status
= H Diagnostic control record
= I to R Reserved
= S to Z Not supported

SRCB for Print Records (Carriage Control Byte)

```
O M C C C C C C
0             7
```

Usage: To provide carriage control information for print records.

Bit meanings:

O = 1 (must always be on)

M = 0 Normal carriage control
1 Reserved for future use

CCCCCC = Carriage control information
= 1000NN Space immediately NN spaces
= 11NNNN Skip immediately to channel NNNN
= 0000NN Space NN lines after print
= 01NNNN Skip to channel NNNN after print
= 000000 Suppress space

SRCB for Punch Records

```
O M M B R R S S
0             7
```

Usage: To provide additional information for punch records.

Bit meanings:

O = 1 (must always be on)

MM = 00 SCB count units = 1
= 01 SCB count units = 2
= 10 SCB count units = 4
= 11 Reserved

B = 0 Normal EBCDIC card image
1 Not supported

RR = Reserved for future expansion

SS = Punch stacker select information

Figure 2-12 (Part 4 of 6). Sample BSC Buffer for MRJE

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description

SRCB for Input Records

```
| O M M B R R R R |
  0             7
```

Usage: To provide additional information for input records.

Bit meanings:

O = 1 (must always be on)

MM = 00 SCB count units = 1
 = 01 SCB count units = 2
 = 10 SCB count units = 4
 = 11 Reserved

B = 0 Normal EBCDIC card image
 1 Not supported

RRRR = Reserved

SRCB for Terminal Message Routing Record

```
| O T T T T T T T |
  0             7
```

Usage: To indicate the destination of a terminal message.

Bit meanings:

O = 1 (must always be on)

TTTTTTT = Remote system number
 = 0 Broadcast to all remote systems

SRCB for RFT or GFT Records

```
| O I I I T T T T |
  0             7
```

Usage: To identify the function for RFT or GFT control records.

Bit meanings:

O = 1 (must always be on)

Figure 2-12 (Part 5 of 6). Sample BSC Buffer for MRJE

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
			III = Stream identifier 001 Stream 1 010-111 Not supported TTTT = Record type identifier 0011 Normal reader record 0100 Normal printer record 0101 Normal punch record 0110-1111 Not supported Examples: (These are the only identifiers supported on System/34). hex 94 = Printer 1 hex 95 = Punch 1 hex 93 = Reader 1
07	BSBSCB	1	String control byte (SCB) O K L J J J J J 0 7 Usage: Control field for data character strings, used for data compression. Bit meanings: O = 0 End of record (KLJJJJJ = 0) = 1 All other SCBs K = 0 Duplicate character string L = 0 Duplicate character is blank L = 1 Duplicate character is nonblank (and follows SCB) JJJJJ = Duplication count (see Note 2) K = 1 Nonduplicate character string LJJJJJ = Character string length (see Note 2) *Notes:* 1. If KLJJJJJ = 0 and O = 1, SCB indicates record is continued in next transmission block. 2. Count units are normally 1 but may be in any other units. The units utilized may be indicated as function control or dynamically in the SRCB.
08	BSBTXT	Variable Length	Text terminated by next SCB. Last text string is terminated by SCB of zero. Last record in a block terminated by RCB of zero. A logical end of file (EOF) is an initial SCB of zero.

Figure 2-12 (Part 6 of 6). Sample BSC Buffer for MRJE

COMMUNICATION AND CONTROL TABLE

The communication and control table (see Figure 2-13) contains addresses of the MRJE modules, information on the status of I/O devices, printer carriage control information, and other information of interest to more than one module.

The address of the CCT is contained in low storage of the MRJE utility immediately following the characters CCT@. Also, each PFCB has the address of the CCT at PFCCT@.

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0	CTPFC@	2	Address of first PFCB to be dispatched.
2	CTBUF1	2	Length of BSC buffers.
4	CTBBUF	2	Length of BSC buffers rounded up to nearest multiple of 8.
6	CTBUF@	2	Address of free buffer pool.
8	CTBUF#	1	Number of free buffers.
9	CTFLG1	1	Flag byte 1: hex 80 = MRJE has been initialized hex 40 = Processor has work hex 10 = Compression requested by user hex 08 = Signoff/logoff command has been read hex 04 = BSC connection has ended hex 02 = BSC error, 2 option taken hex 01 = Automatic mode
A	CTFLG2	1	Flag byte 2: hex 40 = TDISKPR1 defined hex 20 = PDISKPR1 defined hex 10 = Current IOB is in a translated area hex 02 = System console is a reader hex 01 = Temporary work bit
B	CTFLG3	1	Flag byte 3: hex 80 = MRJE BSC trace active hex 40 = BSC task has a processor check hex 20 = Console input request from reader 1 task outstanding hex 10 = Termination of MRJE hex 08 = Console released hex 04 = Logoff sent to host hex 02 = Controlled cancel issued hex 01 = Termination, termination event block is valid

Figure 2-13 (Part 1 of 5). Communication and Control Table

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
C	CTFHOS	1	Host program type: hex FA = VM hex F9 = ASP hex F7 = HASP II hex F3 = JES III hex F2 = JES II hex F1 = RES
D	CTBSCDTF	2	BSC DTF address for #MRBP.
F	CTCNTUB@	2	Address of console TUB.
11	CTCNTMID	2	Console terminal ID.
13	CTCSB@	2	Address of BSC CSB.
15	CTMSG1	2	NPPMSG1 address save area.
17	CTMSG2	2	NPPMSG2 address save area.
19	CTPGSRT	2	Address of start area to build MRJE.
1B	CTPGEND	2	Address of last byte (plus 1) of available area.
1D	CTFORMAT	2	Address of configuration information save area.
1F	CTBSLN	1	Reserved.
20	CTSTDPR1	2	Printer disk block size (standard).
22	CTSPCPR1	2	Printer disk block size (special).
24	CTSPCPU1	2	Punch disk block size.
26	CTTCBTWA	2	Disk address of task work area.
28	CTCARG	1	Number of lines per page.
29	CTCHN1	1	Channel 1 equivalence.
2A	CTCHN2	1	Channel 2 equivalence.
2B	CTCHN3	1	Channel 3 equivalence.
2C	CTCHN4	1	Channel 4 equivalence.
2D	CTCHN5	1	Channel 5 equivalence.
2E	CTCHN6	1	Channel 6 equivalence.

Figure 2-13 (Part 2 of 5). Communication and Control Table

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
2F	CTCHN7	1	Channel 7 equivalence.
30	CTCHN8	1	Channel 8 equivalence.
31	CTCHN9	1	Channel 9 equivalence.
32	CTCH10	1	Channel 10 equivalence.
33	CTCH11	1	Channel 11 equivalence.
34	CTCH12	1	Channel 12 equivalence.
35	CTJOB#	4	File sequence number.
39	CTDKID	1	First character of disk file form number.
3A	CTRES2	1	Reserved.
3B	CTRES3	1	Reserved.

The following 28 entries contain the addresses of the routines listed:

3C	CTDSGTUB	2	DSGTUB—get exclusive use of TUB.
3E	CTDSNEXT	2	DSNEXT—return and skip turn.
40	CTCONBUF	2	CONBUFO—address of console buffer.
42	CTBMGET	2	BMGET—get record from BSC buffer.
44	CTBMPUT	2	BMPUT—put record into BSC buffer.
46	CTBMPURG	2	BMPURG—purge buffers.
48	CTBMGFT	2	BMGFT—put GFT sequence in buffer.
4A	CTBMRFT	2	BMRFT—put RFT sequence in buffer.
4C	CTBMEOF	2	BMEOF—put logical EOF in buffer.
4E	CTBMQBUF	2	BMQBUF—add current BSC buffer to queue.
50	CTBMFCON	2	BMFCON—turn on FCS bit in BSC PFCB.
52	CTBMFCOF	2	BMFCOF—turn off FCS bit in BSC PFCB.
54	CTDSIOWT	2	DSIOWT—wait for I/O completion.
56	CTDSRFT	2	DSRFT—wait for RFT.

Figure 2-13 (Part 3 of 5). Communication and Control Table

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
58	CTDSGFT	2	DSGFT—wait for GFT.
5A	CTDSSTRT	2	DSSTRT—wait for MODIFY statement.
5C	CTDSWTO	2	DSWTO—wait for WTO.
5E	CTDSWTOR	2	DSWTOR—wait for WTOR.
60	CTBMFBUF	2	BMFBUF—free a BSC buffer.
62	CTMRFC	2	#MRFC—full compression module.
64	CTBMSTRT	2	BMSTRT—send start command to host system.
66	CTDSWBUF	2	DSWBUF—wait for buffers.
68	CTDSWORK	2	DSWORK—wait for work.
6A	CTBMCAN	2	BMCAN—send cancel command to host system.
6C	CTBMDCOM	2	BMDCOM—decompress a record.
6E	CTDSSKIP	2	DSSKIP—skip a dispatching turn.
70	CTDSGCRT	2	DSGCRT—acquire display screen.
72	CTDSFCRT	2	DSFCRT—free display screen.
74	CTOVRLY1	2	Address of overlay area 1.
76	CTOVRLY2	2	Address of overlay area 2.
78	CTOVRLY3	2	Address of overlay area 3.
7A	CTPNTBUF	2	Address of print buffer.
7C	CTRD1BUF	2	Address of reader 1 buffer.
7E	CTBUFMAN	2	Address of buffer manager.
80	CTTCB@	2	Address of MRJE TCB.
82	CTJCB@	2	Address of MRJE JCB.
84	CTWSINXA	2	Address of format index area.
86	CTRDTABL	2	Address of active reader table.
88	CTSIGN@	2	Address of rightmost byte of sign-on area.

Figure 2-13 (Part 4 of 5). Communication and Control Table

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
8A	CTTIMER	6	Delay value from END statement.
90	CTDLAYCT	1	Count of active data streams.
92	CTRDCNT	2	Counter for FSB names for data files.

Figure 2-13 (Part 5 of 5). Communication and Control Table

PROGRAM FUNCTION CONTROL BLOCKS

The MRJE utility uses the PFCBs (program function control blocks) to control the dispatching of the logical processors and to pass information between the various processors and $MRJE. There is a PFCB for each of the following:

- BSC
- Console input
- Console output
- Reader (up to 3)
- Printer
- Punch (optional)

The beginning of each PFCB has the same description (see Figure 2-14). The remainder of the bytes are described separately in Figures 2-15 through 2-19. (Printer and punch PFCBs are described together.)

The address of the first PFCB is contained in the CCT at CTPFC@. The address of the next PFCB in the chain is contained at PFNEXT in each PFCB. The last PFCB in the chain has an address of hex 0000 at PFNEXT.

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0	PFNEXT	2	Address of next PFCB in chain.
2	PFDTF@	2	Address of current processor DTF.
4	PFIOB@	2	Address of current processor IOB.
6	PFLAG1	1	PFCB flag byte 1: hex 80 = Processor waiting for I/O hex 40 = Processor waiting for buffers hex 20 = Processor waiting for GFT/RFT hex 10 = BSC SVC request pending hex 08 = General wait, between tasks hex 04 = Processor waiting for console message hex 02 = Processor waiting for MODIFY statement hex 01 = Processor waiting for work hex 00 = Processor is dispatchable
7	PFLAG2	1	PFCB flag byte 2: hex 80 = Option 1 halt error recovery (console I/O and reader) User requested task termination (print/punch) BSC post switch (BSC) hex 40 = Receive switch (BSC) End of file hex 20 = Two-second timer switch (BSC) I/O error hex 10 = End of extent (print/punch) Timer expired (console I/O and reader) Wait-a-bit (BSC) hex 08 = Local message switch (BSC) RFT/GFT received hex 04 = FCS on (BSC) Forms mount or READFILE pending hex 02 = FCS off (BSC) Terminate the task (print/punch) hex 01 = Signoff/logoff indicator (BSC) Controlled cancel pending
8	PFLAG3	1	PFCB flag byte 3: hex 80 = RFT needed (console I/O and reader) hex 40 = Signoff/logoff command indicator (console I/O) Data written to disk file (print/punch) hex 20 = Start command required (console I/O) Decompression required for retry (print/punch) hex 10 = Console interrupt honored (console I/O) PDISKPR1 file being used (print/punch) hex 08 = First-time switch (reader)

Figure 2-14 (Part 1 of 3). Common PFCB Bytes

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
			hex 04 = Log device message required (console and reader) #MRCL is processing an error (print/punch) hex 02 = MODIFY statement pending hex 01 = #MRCL will switch output device (print/punch) Disk data file read pending or library member remaining bytes pending (reader)
9	PFLAG4	1	SRCB for print task.
A	PFBUF@	2	Address of BSC buffer chain.
C	PFMBCT	1	Maximum BSC buffer count.
D	PFCBCT	1	Current BSC buffer count.
E	PFPRI@	2	Primary dispatching address.
10	PFRES@	2	Processor resident code address.
12	PFMSG#	2	MIC for output.
14	PFCCT@	2	CCT address.
16	PFRCB1	1	Processor record control byte: hex B3 = Reader 3 PFCB hex A3 = Reader 2 PFCB hex 95 = Punch PFCB hex 94 = Printer PFCB hex 93 = Reader 1 PFCB hex 92 = Console input PFCB hex 91 = Console output PFCB hex 00 = BSC PFCB
17	PFPRIQ	1	Primary device Q-byte: hex E0 = Printer hex A0 = Disk hex 10 = keyboard, display screen
18	PFTMID	2	Work station terminal ID.
1A	PFTUB@	2	Work station TUB address.
1C	PFMSGR	1	Operator reply to message.

Figure 2-14 (Part 2 of 3). Common PFCB Bytes

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
1D	PFLFLG	1	Transient load list flags: hex 80 = Leave module in transient area hex 10 = Transient area shared hex 02 = Return to dispatcher via PFSAR@ hex 01 = Dispatch resident code
1E	PFNXNM	6	Name of next module to execute.
24	PFTRAR	2	Address of transient control.
26	PFLAG5	1	PFCB flag byte 5: hex 80 = Printer already allocated (printer) Current EOF flag (reader) hex 40 = TDISKPR1 already allocated (printer) EOF save flag (reader) Processing user request (BSC) hex 20 = IOB address not real hex 10 = #MRBO terminating abnormally (BSC) Reader display station has been released (reader) hex 08 = Display station is invited hex 04 = Display station has a permanent I/O error
27	PFMSGV	10	Variable data for message output.
31	PFSAR@	2	Save area for dispatcher service.
33	PFPBF@	2	Address of processor buffers.
35	PFSXR1	2	Save area for XR1.
37	PFSARR	2	Save area for ARR.
39	PFHXR1	2	Hold area for XR1.
3B	PFHARR	2	Hold area for ARR.
3D	PFRCNT	3	Input/output record count.
40	—	1	Reserved.

Figure 2-14 (Part 3 of 3). Common PFCB Bytes

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0 through 40	See Figure 2-14.		
41	PFBBSN	2	BSC sense information.
43	PFBBLN	2	Length of transmit/receive operation.
45	PFBRW1	2	Start of receive address for wait-a-bit.
47	PFBRW2	2	End of receive address for wait-a-bit.
49	PFBXW1	2	Start of transmit address for wait-a-bit.
4B	PFBWX2	2	End of transmit address for wait-a-bit.
4D	PFBR1	2	Start of receive address.
4F	PFBR2	2	End of receive address.
51	PFBX1	2	Start of transmit address.
53	PFBX2	2	End of transmit address.
55	PFBX1S	2	Save start of transmit address.
57	PFBX2S	2	Save end of transmit address.
59	PFBRBC	1	Expected BCB from host system.
5A	PFBSBC	1	Next transmitted BCB to host system.
5B	PFBNFS	2	Next transmitted FCS to host system.
5D	PFBRFS	2	Last received FCS from host system.
5F	PFBXFS	2	Last transmitted FCS to host system.
61	PFBWRK	1	BSC work byte.
62	PFBSSN	1	Temporary BSC sense save.
63	PFBCTR	1	Bad receipt counter.
64	PFBMER	1	Maximum BSC error count.
65	PFBSEQ	5	DLE-STX-BCB-FCS-FCS.
6A	PFBERR	9	DLE-STX-LBCBKSP-FCS-FCS-ERR-ERR-ERR-ETB.
73	PFBXWB	9	Wait-a-bit sequence.

Figure 2-15 (Part 1 of 2). BSC PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
7C	PFBRWB	9	Receive area for wait-a-bit reply.
85	PFBSTX	2	DLE-STX.
87	PFBETB	2	Logical EOB-physical ETB.
89	PFBACK	2	DLE-ACK0.
8B	PFBTBS	2	Number of text blocks sent.
8D	PFBTBR	2	Number of text blocks received.
8F	PFBNKR	2	Number of NAKs received.
91	PFBLDE	2	Number of lost data errors.
93	PFBINR	2	Number of invalid responses.
95	PFBRTO	2	Number of receive time-outs.
97	PFBDCK	2	Number of data checks.
99	PFBACT	2	Number of transmit adapter checks.
9B	PFBACR	2	Number of receive adapter checks.
9D	PFBASC	2	Not used.
9F	PFBABR	2	Number of aborts received.
A1	PFBDTO	2	Number of disconnect time-outs.
A3	PFBBRV	2	Reserved.
A5	PFBBMG	4	BSC PFCB message area.
A9	–	3	Reserved.
AC	PFBIA@	2	Interrupt routine entry point.
AE	PFBIS@	2	Interrupt routine switch address.
B0	PFBGB@	2	Get buffer subroutine entry point.
B2	PFBP1@	2	Interrupt routine post address.
B4	PFBP2@	2	Interrupt routine post address.
B6	PFBQR	2	Q- and R-bytes of last SIO.
B8	PFBCAR	2	Current address register at last SIO.

Figure 2-15 (Part 2 of 2). BSC PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0 through 40	See Figure 2-14.		
41	PFREC@	2	Input record address.
43	PFRLEN	2	Input record length.
45	–	18	Reserved.
57	WACIB	16	Console input IOB.
67	WACSIZ	133	Console input buffer.

Figure 2-16. Console Input PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0 through 40	See Figure 2-14.		
41	PFREC@	2	Output record address.
43	PFRLEN	2	Output record length.
45	PFMSGQ	2	Message queue address.
47	PFMQCT	1	Message queue available counter (8=empty, 0=full).
48	PFQNXT	2	Next message to be processed by console output.
4A	WAMSGQ	152	Local message queue.

Figure 2-17. Console Output PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0 through 40	See Figure 2-14.		
41	PFREC@	2	Input record address.
43	PFRLEN	2	Input record length.
45	PFREAD	8	Current READFILE name.
4D	PFRFTY	1	Current READFILE type.

Figure 2-18 (Part 1 of 2). Reader PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
4E	PFRCMD	1	Current READFILE command indicator.
4F	PFRCRA	3	Next available sector address in READFILE.
52	PFREND	3	Ending sector address in READFILE.
55	PFRDSV	8	File name from READFILE save area.
5D	PFRTYS	1	File member ‡ype save area.
5E	PFRCMS	1	File indicator save area.
5F	PFRDAT	6	File creation date.
65	PFQSAV	1	Device Q-byte.
66	PFRLNS	2	Record length in READFILE.
68	PFRSSV	3	Disk file current address or library member. I/O buffer offset and displacement save area.
6B	PFRDSS	3	Disk file current address or library member. I/O buffer offset and displacement save area.
6E	PFNAME	2	Count value for FSB file names.
70	PFNAMS	2	Save count value for FSB file names.
72	WARRCL	2	Logical record length.
74	WARLSR	4	Next available record address.
78	WAREND	3	File end sector address.
7B	WARF1	64	Format 1 work area.
or	WARBUF	93	Input buffer
D9	WRCUOF	2	Cürrent disposition in processor buffer.
DB	WRCURC	2	Byte count in processor buffer.
DD	WCUOFS	2	Processor buffer disposition.
DF	WCURCS	2	Processor buffer count.
E1	WARIOB	28	Disk IOB for reader.

Figure 2-18 (Part 2 of 2). Reader PFCB

Displacement of Leftmost Byte (Hexadecimal)	Label	Length in Bytes (Decimal)	Description
0 through 40	See Figure 2-14.		
41	PFREC@	2	Output record address.
43	PFRLEN	2	Output record length.
45	PFDEOD	2	Sector address of end of data.
47	PFCURC	2	Byte count in processor buffer.
49	PFNXFM	4	Next forms mount number.
4D	PFRS#	5	RES response number (first 2 bytes only).
or	PFRM#	5	HASP/JES2 remote number (first 4 bytes only).
or	PFRMID	5	ASP/JES3 remote ID.
52	PFQMDS	1	MODIFY statement Q-byte.
53	—	4	Reserved.
57	PFTCNT	3	TDISKPR1 record count.
5A	PFTCNS	3	TDISKPR1 record count save.
5D	PFHOLD1	2	Work area.
5F	PFHOLD2	2	Work area.
61	PFHCNT1	3	Work area.
64	PFHDNT2	3	Work area.
67	PFRRCN	3	File record count.
6A	PFSCUR	1	Save PFCURC for TDISK.
6B	PFAFAA	2	Address of current format 1 in assign/free area.
6D	PFCNLT	13	Temporary carriage table.
7A	WADSWA	64	Disk I/O area.
BA	WADBMW	20	Buffer manager work area.
CE	WAIOB@	28	Disk IOB for printer/punch.

Figure 2-19. Printer/Punch PFCB

Introduction

The data communications print utility is an SSP utility that prints data that was written to disk during an MRJE utility session. The print utility can process any number of disk files and any number of contiguous records within each file. For a description of the utility control statements required to run the utility, see the *Data Communications Reference Manual.* Figure A-1 describes the function of this utility.

SYSTEM REQUIREMENTS

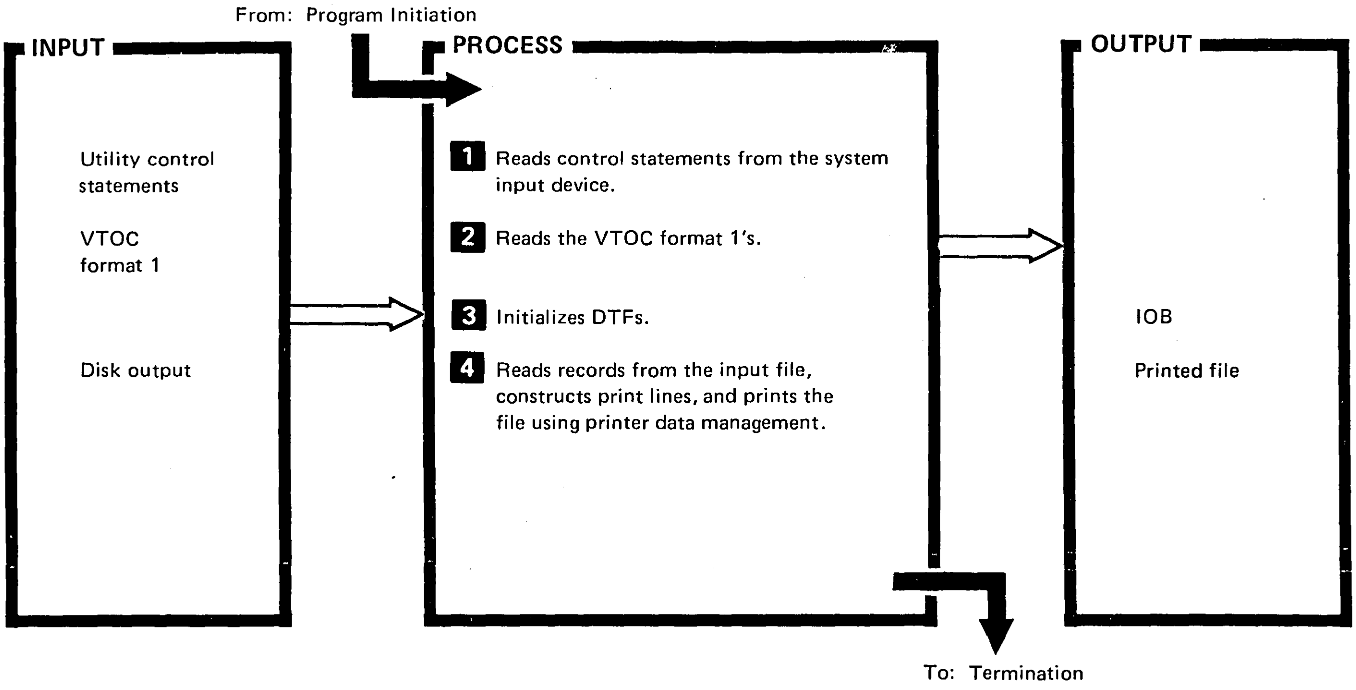
The data communications print utility is part of the System/34 SSP and executes in 8K bytes of main storage.

ERROR HANDLING

When an error condition is detected by this utility, it is processed through the system logging function. Operator intervention is required to select a retry of the operation or a controlled cancel. On a controlled cancel, control is returned to the utility, which calls end of job.

Method of Operation

Figure A-1 describes the logical function of $DCSUP using an input-process-output diagram. For detailed information on the input-process-output diagram, see *Diagram Techniques* in *How to Use This Manual* at the front of this publication.



1 Reads control statements from the system input device.

2 Reads the VTOC format 1's.

3 Initializes DTFs.

4 Reads records from the input file, constructs print lines, and prints the file using printer data management.

DESCRIPTION	MODULE/ ROUTINE
1 Reads the COPYFILE, SELECT, and CARRIAGE utility control statements from the system input device. 2 Reads the VTOC format 1's describing the MRJE/WS disk output files to be used as input to the utility. 3 The information contained in the VTOC format 1's is used to format the disk IOB and do an initial 2-sector read of the input file. 4 Reads records from the data communication utility disk files and construbts print lines, converting the host system carriage control information into the proper carriage command. The print lines are printed using printer IOS. When end of file is reached for the input disk file, the print utility reads another utility control statement from the system input device. The print utility goes to end of job after reading END or after an operator responds to a message with a 2 option.	$DCSUP

Figure A-1. Printing Disk Output

ACK0: The even-numbered positive acknowledge sequence.

ACK1: The odd-numbered positive acknowledge sequence.

address translation register: ATR.

addressing: The means by which a sending or control station selects the unit to which it will send a message.

American National Standard Code for Information Interchange: ASCII.

ASCII: American National Standard Code for Information Interchange.

ASP: Asymmetric multiprocessing system.

asymmetric multiprocessing system: ASP.

ATR: Address translation register.

auto-answer: A machine feature that permits a station to respond to a call it receives over a switched line without operator action.

auto-call: A machine feature that permits a station to initiate a connection with another station over a switched line without operator action.

basic telecommunications access method: BTAM.

BCB: Block control byte.

BCC: Block check character.

binary synchronous communications (BSC): A flexible form of line control that provides a set of rules for transferring data over a communications line connecting two or more devices that use a communications adapter.

block control byte: BCB.

block check character: BCC.

bps: Bits per second.

BSC: Binary synchronous communications.

BTAM: Basic telecommunications access method.

call: The action performed by the requesting party, the operations necessary to make a request, or the effective use made of a connection between two stations.

CCP: Communications control program.

CCT: Communication and control table.

central station: See *control station.*

clocking: A method of controlling the number of data bits sent on a data communications line in a given time.

command file: A disk file, procedure member, or source member that is designated as a command file by a data communications utility program. A command file can contain utility control statements and/or records to be transmitted to the host system.

communication and control table: CCT.

communication specification block: CSB.

communications adapter: A hardware feature that enables System/34 to become a part of a data communications network.

communications control program: CCP.

compression: A technique for removing strings of duplicate characters and truncating trailing blanks before transmitting data.

configuration record: See *system configuration record; display station configuration record.*

control station: The primary or controlling computer in a multipoint data communications configuration. The control station controls the sending and receiving of data.

conversational file: A BSC file that allows receiving or sending data characters as an acknowledgment instead of the ACK0 or ACK1 sequence.

CSB: Communication specification block.

data communications: The transmission of data between systems and/or remote devices over a communications line.

data file: A disk file procedure member, or source member that is designated as a data file by a data communications utility program. A data file can contain only records to be transmitted to the host system.

data link: The equipment and rules (protocols) used for sending data over a communications line.

data link escape (DLE): A control character used exclusively to provide supplementary line-control signals (control character sequences or DLE sequences).

data mode: A time at which BSC is transmitting or receiving characters on the line.

data stream: All data transmitted over a data link in a single read or write operation.

DISC: The transmission control sequence for disconnect on a switched line.

disconnect time-out: An indication that the BSC station you were communicating with has gone on hook or hung up.

disk operating system: DOS.

display station: An input/output device containing a display screen on which data is displayed and an attached keyboard from which data is entered.

display station configuration record: An area on disk that describes a command display station's environment. The display station's configuration record contains information such as the session date, the work station ID of the printer to be used for the display station's printed output, and the region size for jobs submitted from the display station.

DLE: Data link escape.

DOS: Disk operating system.

duplex: Concurrent transmission and reception of data.

EBCDIC: Extended binary-coded decimal interchange code.

EBCDIC transparency: See *transparent text mode.*

end of transmission: EOT.

ENQ: Enquiry character.

EOT: End of transmission.

error history file: A push-down stack of the last BSC errors that have occurred.

ETB: End-of-transmission-block character.

ETX: End-of-text character.

expanded communications buffer: A special feature of the 3741 that allows multiple records to be transmitted or received in one block of data.

extended binary-coded decimal interchange code: EBCDIC.

extent: A continuous space on disk or diskette that is occupied by, or reserved for, a particular file.

GFT: Grant function transmission.

grant function transmission: GFT.

half duplex: Data communications in opposite directions, but not at the same time.

host system: The primary or controlling computer in the communications network. See also *control station.*

intermediate block check: A function that permits checking of each record, instead of checking the contents of the total buffer when large buffers of data are received.

interrecord separator: The last character of a record that signals the end of that record and the beginning of another record on a data communications network.

ITB: Intermediate-text-block character.

JES2: Job entry subsystem 2.

JES3: Job entry subsystem 3.

line control characters: See *transmission control characters.*

manual answer: Operator actions to make a station ready when a station receives a call over a switched line.

manual call: Operator actions to make a connection with a station over a switched line.

MLMP: Multiline multipoint.

modem (modulator/demodulator): A device that connects a communications adapter to a communications line.

monitor mode: A time during which the communications adapter is looking for BSC synchronization characters.

MRJE: MULTI-LEAVING Remote Job Entry.

MSP/7: Modular system programs.

MULTI-LEAVING Remote Job Entry (MRJE): An SSP function that allows the user to communicate with a System/370 over a communications line using BSC.

multidropped terminal: See *tributary station.*

multiline multipoint: MLMP.

multipoint data link: A network configuration in which connected stations communicate with each other over a common communications line on a time-shared basis. The primary station controls and maintains the data link.

NAK: Negative acknowledgment character.

NCP: Network control program.

network: A configuration by which two or more stations can communicate.

network control program: NCP.

nonswitched line: A connection between systems or devices that does not have to be made by dialing.

operating system: OS.

OS: Operating system.

PFCB: Program function control block.

point-to-point line. A data communications facility that connects a single remote station to a data processing system. A point-to-point line can be either switched or nonswitched.

polling: In a multipoint environment, a request to send, transmitted from the primary station to a specific secondary station.

POWER: Priority output writers, execution processors, and input readers.

program function control block: PFCB.

receive mode: A time during which the BSC adapter looks for synchronization characters and then stores the data characters in main storage.

receive time-out: An indication that no data has been received by this communications adapter in a given period of time.

remote entry services: RES.

remote job entry: RJE.

remote spooling communications system: RSCS.

request function transmission: RFT.

RES: Remote entry services.

reverse interrupt (RVI): A request by the receiving station to the sending station to stop transmitting and prepare to receive a message.

RFT: Request function transmission.

RJE: Remote job entry.

RSCS: Remote spooling communications system.

RVI: Reverse interrupt character.

session: The period of time during which programs or devices can communicate with each other.

SOH: Start-of-heading character.

standby line: A modem feature that allows a point-to-point, nonswitched line modem to also function on a point-to-point, switched line.

station: A system or device that can send or receive data over a communications line.

STX: Start-of-text character.

switched line: A connection between two stations that is established by dialing.

SYN: Line synchronization character.

system configuration record: Information stored on disk that describes system characteristics and programming support; for example, system data format, disk capacity, and main storage capacity.

system console: A display station designated to activate specific system functions, and to control and monitor system operation, in addition to performing as a command display station.

task control block: TCB.

TCAM: Telecommunications access method.

TCB: Task control block.

telecommunications access method: TCAM.

terminal unit block: TUB.

text transparency: A provision that allows BSC to send and receive messages containing any or all of the 256 character combinations in EBCDIC, including transmission control characters. EBCDIC and control characters are all sent as text.

transmission control characters: Special characters that are included in a message to control communication over a data link.

transparency: See *transparent text mode.*

transparent text mode: A method of binary synchronous transmission in which only transmission control characters preceded by the DLE control character are processed as transmission control characters.

tributary station: A secondary or noncontrolling device in a multipoint data communications configuration.

TTD: Temporary-text-delay sequence.

TUB: Terminal unit block.

virtual machine: VM.

virtual storage: VS.

virtual telecommunications access method: VTAM.

VM: Virtual machine.

VS: Virtual storage.

VTAM: Virtual telecommunications access method.

WACK (wait before transmit positive acknowledgment): The DLE sequence sent by a receiving station to indicate that it cannot receive data at present.

work station: A device that lets a person transmit information to or receive information from a computer, or both, as required to perform the job.

wrap test: A test that checks attachment or controller circuitry without testing the device itself.

WTO: Write to operator.

WTOR: Write to operator with reply.

Index

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

Note: All comments and suggestions become the property of IBM.

● No postage necessary if mailed in the U.S.A.

Name ______________________________

Address ______________________________

____

IBM System/34
System Support Program
Logic Manual: Data Communications

LY21-0051-0

Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold

Fold

IBM®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

System/34 System Support PLM: Data Communications (File No. S34-36) Printed in U.S.A. LY21-0051-0

LY21-0051-0