Transaction Processing Facility

Non-SNA Data Communications Reference

Version 4 Release 1

SH31-0161-02

Transaction Processing Facility

Non-SNA Data Communications Reference

Version 4 Release 1

SH31-0161-02

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

Third Edition (June 2002)

This is a major revision of, and obsoletes, SH31-0161-01 and all associated technical newsletters.

This edition applies to Version 4 Release 1 Modification Level 0 of IBM Transaction Processing Facility, program number 5748-T14, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

IBM welcomes your comments. Address your comments to:

IBM Corporation
TPF Systems Information Development
Mail Station P923
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Contents

Notices

References in this book to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service in this book is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 830A
Mail Drop P131
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this book to non-IBM Web sites are provided for convenience only and do not in any way serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this book or accessed through an IBM Web site that is mentioned in this book.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM

Other company, product, and service names may be trademarks or service marks of others.

About This Book

This book describes the functions provided for the non-Systems Network Architecture (non-SNA) data communications area of the TPF system.

In this book, abbreviations are often used instead of spelled-out terms. Every term is spelled out at first mention followed by the all-caps abbreviation enclosed in parentheses; for example, Systems Network Architecture (SNA). Abbreviations are defined again at various intervals throughout the book. In addition, the majority of abbreviations and their definitions are listed in the master glossary in the *TPF Library Guide*.

Who Should Read This Book

This book is intended for system programmers who are responsible for non-SNA data communication support.

Conventions Used in the TPF Library

The TPF library uses the following conventions:

Conventions	Examples of Usage
italic	Used for important words and phrases. For example: A *database* is a collection of data. Used to represent variable information. For example: Enter **ZFRST STATUS MODULE** *mod*, where *mod* is the module for which you want status.
bold	Used to represent text that you type. For example: Enter **ZNALS HELP** to obtain help information for the ZNALS command. Used to represent variable information in C language. For example: **level**

`monospaced`	Used for messages and information that displays on a screen. For example:
	`PROCESSING COMPLETED`
	Used for C language functions. For example:
	`maskc`
	Used for examples. For example:
	`maskc(MASKC_ENABLE, MASKC_IO);`

bold italic	Used for emphasis. For example: You ***must*** type this command exactly as shown.
Bold underscore	Used to indicate the default in a list of options. For example: **Keyword=OPTION1** | **DEFAULT**
Vertical bar |	Used to separate options in a list. (Also referred to as the OR symbol.) For example: **Keyword=Option1** | **Option2** **Note:** Sometimes the vertical bar is used as a *pipe* (which allows you to pass the output of one process as input to another process). The library information will clearly explain whenever the vertical bar is used for this reason.

Conventions	**Examples of Usage**
CAPital LETters	Used to indicate valid abbreviations for keywords. For example: KEYWord=*option*

Scale	Used to indicate the column location of input. The scale begins at column position 1. The plus sign (+) represents increments of 5 and the numerals represent increments of 10 on the scale. The first plus sign (+) represents column position 5; numeral 1 shows column position 10; numeral 2 shows column position 20 and so on. The following example shows the required text and column position for the image clear card.
	`|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...`
	`LOADER   IMAGE CLEAR`
	Notes:
	1. The word LOADER must begin in column 1.
	2. The word IMAGE must begin in column 10.
	3. The word CLEAR must begin in column 16.

Related Information

A list of related information follows. For information on how to order or access any of this information, call your IBM representative.

IBM Transaction Processing Facility (TPF) 4.1 Books

- *TPF ACF/SNA Data Communications Reference*, SH31-0168
- *TPF General Macros*, SH31-0152
- *TPF Operations*, SH31-0162
- *TPF System Macros*, SH31-0151
- *TPF System Performance and Measurement Reference*, SH31-0170.

Miscellaneous IBM Books

- *IBM Extended Operations Console Facility/2 System Administrator's Guide*, SH31-0105.

Online Information

- *Messages (Online)*
- *Messages (System Error and Offline)*.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other TPF information, use one of the methods that follow. Make sure you include the title and number of the book, the version of your product and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send your comments electronically, do either of the following:
  - Go to http://www.ibm.com/tpf/pubs/tpfpubs.htm.

  There you will find a link to a feedback page where you can enter and submit comments.
  - Send your comments by e-mail to tpfid@us.ibm.com
- If you prefer to send your comments by mail, address your comments to:

  IBM Corporation
  TPF Systems Information Development
  Mail Station P923
  2455 South Road
  Poughkeepsie, NY 12601-5400
  USA
- If you prefer to send your comments by FAX, use this number:
  - United States and Canada: 1 + 845 + 432 + 9788
  - Other countries: (international code) + 845 + 432 +9788

Synchronous Link Control Functional Specification

The Synchronous Link Control Procedure is designed to control the interchange of messages on a point-to-point link between two TPF systems or between a TPF system and any other automatic system. The hardware characteristics of the link are as follows:

- The link will be point to point and may consist of from one to seven full-duplex voice grade lines.
- The transmission code will be 8 bits (7 information bits and 1 parity). The relationship between the seven-bit characters and the IBM eight-bit byte is shown below:
  seven-bit character – b7 b6 b5 b4 b3 b2 b1
  IBM eight-bit – 0 1 2 3 4 5 6 7

  Thus, IBM eight-bit byte X'1F' will be represented as 1/15 in the seven-bit character code. The seven-bit code will be used throughout this document.
- The mode of transmission will be synchronous, serial by bit and by character, with the low order bit b1 transmitted first through to the most significant bit b7 followed by the parity bit b8 which will maintain odd parity.

Processing Description

Messages will be transmitted in variable length blocks, with maximum block length subject to agreement by link. The total message length is restricted by the maximum number of blocks used for one message (sixteen).

A single block will contain characters of only one message; parts of different messages will not be packed into the same block. This applies also to messages from and to indirect terminals attached to the TPF system via the SITA High Level Network.

Error detection based on character and block parity checking is applied to all blocks. Multi-block message protection is by message label and block chaining. The message block identifier, MBI and the link characteristics indicator, LCI are used to make possible correct reassembly of multi-block messages on receipt.

The link will be controlled by means of fixed length link control messages. These Link Control Blocks (LCBs) will indicate correct or incorrect reception of information blocks and provide information about the status of the link. Two types of data messages will be recognized:

Type A Messages having requirements for short transit times of the order of seconds. These will normally be short, single-block messages, but multi-block Type A messages can be handled.

A subset called service messages will be exchanged to enable the TPF processor to stop and start the polling of indirect terminal interchanges and to inform the TPF processor of a failure on the High Level Network.

Type B Conventional Telegraph Messages permitting long transit times in the order of minutes. They may be either single- or multi-block, depending on their length.

The following order of precedence will be observed when transmitting message blocks between the processors:

- Link Control Blocks have first priority
- Type A message blocks have second priority
- Type B message blocks have third priority

The security of Type B messages sent to an SLC link may be achieved by one of two ways.

- The application controls and maintains the security of its messages.
- The application loses control of the message after passing it to SLC, which maintains security thereafter.

The application informs SLC which type of security is being used by setting an indicator in the message status bytes.

Data Areas

CTKD – Keypoint Record D

Content and Usage

Keypoint Record D (CTKD) is used exclusively by the TPF Synchronous Link Control Package (18-SLC) and contains two tables, the Link Routing Table and the Link Pool Availability Directory.

Type Of Data

Header: The header of CTKD is 16 bytes long. The header contains a record ID, a byte count indicating the length of the record, a four-byte record name, a four-byte file address, and a four-byte indexing factor.

Record Identification: The ID of CTKD is ‘CK’, indicating it is a Control Program keypoint record.

Byte Count: The byte count indicates the length of CTKD in bytes.

Record Name: The program name updating the keypoint is placed in the record name field.

File Address of Next Keypoint: CTKD contains the file address of Keypoint CTK1.

Indexing Factor: The indexing factor is used by the load sequence to define the displacement of the record relative to a base register.

Link Routing Table: This table is used by the TPF Synchronous Link Control programs to route Type A messages to or from a high level communications network (e.g., SITA HLN) attached to the system via a synchronous link. There are two types of eight-byte entries in the link routing table:

1. An entry for each indirect terminal interchange (TI) attached via the HLN.
2. An entry for each system with which we communicate via the HLN (e.g., Hotel Reservation or Fare Quotation).

Each entry consists of the following fields:

1. Four bytes of routing information comprising the symbolic link number, the HEX/HEN of the remote HLN center to which the associated TI/system is connected, and one byte used to identify the origin of the message at the remote HLN center.
2. One byte giving the pseudo line number that is substituted for the HLN routing details to enable processing by TPF applications programs. (This pseudo line number and all attached TIs must be contained in the tables referenced by 03-WGR.)
3. One byte to identify each TI attached to a remote HLN center (i.e., IA).
4. One byte of indicators:

   Bit 0: CXJPOL

   If on, this shows that a poll initiation service message has been sent from the Pseudo Line Start/Stop And Display program (03-CMP) to the HLN controlling the line to which this TI is attached.

   If off, this shows that a stop polling service has been sent to the HLN controlling this line.

   Bit 1: CXJNML

   If on, the initiate poll request from CMP was for polling at the normal rate.

   If off, the request from CMP was for polling at a reduced rate.

   Bit 2: CXJUII

   This bit controls the forward routing of type A messages associated with this entry by the TPF Synchronous Link Control component of the ECB Communications Source Program.

   If on, CILL will route the message to the common processing routines of the ECB Communications Source Program (03-CIAA).

   If off, CILL will route the message to either the Message Router (03-ROUT) or an application program specified by the user. The former is activated only when the Message Router function exists in the system.

   Bits 3–4: Unused

   Bits 5–7: CXJXCD

   These bits are used to control the line code in which messages associated with this LRT entry will be transmitted. The available codes are:

   000 Padded CCITT#2 (Padded Baudot)

   100 CCITT#5 (ASCII/ISO seven-bit code)

   110 Extended CCITT#5 (This enables a range of 256 data characters to be transmitted, rather than the 125 available within the CCITT#5 code).
5. One byte is spare.

Link Pool Availability Directory: This table is used by the Input Message Handler (03-CMR) to control the allocation and use of file addresses in a fixed file area defined by FACE. Each bit corresponds to a file address:

1 = file address available.
0 = file address in use.

The bit number within the field is used as a FACE ordinal number and the record type is #LKIBR.

Programming Aspects

Programming Areas: TPF Synchronous Link Control Component of Communications Source (03-CIAA) uses the Link Routing Table to determine the correct processing of input messages from a high level network.

TPF Synchronous Link Control Input Message Handler (03-CMR) uses the Link Pool Availability Directory to dispense file addresses to incoming message blocks for storing on file until a complete message is received.

TPF Synchronous Link Control CCP I/O Routines (03-CIOAI) are used. Send Intercept Routine CXKAII in CLQK uses the Link Routing Table to determine the additional routing characters for messages destined for an indirect TI.

TPF Synchronous Link Control Link Controller (03-CMC) segments CMCC/CMCD, CMR Restart, correct the Link Pool Availability Directory for any discrepancies caused by retrieval errors and the loss of chains of input message blocks.

TPF Synchronous Link Control Pseudo Line Start/Stop/Display program (03-CMP) uses the Link Routing Table to generate the appropriate polling message when a link is restarted and to maintain the polling status indicators according to commands.

Programming Techniques: The address of both tables is to be found in the SLC Common Area (CX#EV). The DSECT defining the fields of the Link Routing Table is in CAIEQ.

Data Macro Name

CK#LRT

Storage Factors

Record Size

The file copy of CTKD is stored in a 4K block. The size of the tables contained in CTKD are configuration-dependent. Care should be exercised to ensure that each table within CTKD is equal in length to the corresponding area reserved in the core-resident Control Program Segment CLQA.

File Requirements

1. CTKD belongs to the middle one-third of the Data Record types in TPF.
2. CTKD is assigned an absolute location:

Cylinder	X'00'
Head	X'02'
Record	X'11'

3. This record is active throughout the life of the online system.

Record Addressing

File Organization: CTKD has two backup copies on the file system. Due to the fixed nature of the Control Program keypoints, a special disk duplication exists. This duplication has a cylinder and head displacement of zero and, as such, may differ from the normal system duplication displacement (reference, MSP-PLM, label CK1DUP).

Retrieval Method: This record should be retrieved only by program MSP-PLM. No facility is available to update the file copy of this data record outside the keypoint update mechanism.

Data Control

The next keypoint address in the keypoint chain is obtained from this record.

Message Formats

In this section, the formats of Link Control Message Blocks and Data Message Blocks will be described.

- Link Control Message Block (LCB)

  Operational control of the link is through the use of fixed format fixed length link control blocks. These blocks are used to indicate correct or incorrect reception of data message blocks, to provide information regarding the status of the link and for fault recovery. The link control block format is:

  ```
  SYN SYN DLE LSI XXX ETB BCC
  ```

- Data Message Block

  Message text is exchanged between centers in variable length data message blocks. Within each block, the text is preceded by a sequence of control characters. The format of a data block, whether Type A or Type B, will be one of the following:

  ```
  SYN SYN DLE TSI MBI LCI ACI Text ETB BCC
  SYN SYN DLE TSI MBI LCI Text ETB BCC
  ```

Control Character Definition

SYN Synchronization Character 1/6

SYN characters must precede each message block to establish and maintain character framing. The number of SYN characters required to establish character framing is a function of the hardware but each message block on a line will be preceded by at least two SYN characters.

The IBM 3705EP transmit side automatically generates two SYN characters to precede each message block.

The receive side does not pass incoming SYNs to the CPU. The first non-SYN character signals the start of an incoming message block. At least two SYN characters are required to establish character framing, but the software can provide up to 12 additional SYN characters on output in multiples of 4. SYN is a reserved character and may not be used elsewhere within a message block.

(The above described function should not be confused with bit synchronization which is solely a function of the communication hardware (e.g., modem) and has no relevance to the procedure described in this document).

DLE Data Link Escape Character 1/0

The DLE character should be the first non-SYN character and indicates the start of a message block. It is a reserved character and may not be used elsewhere in a message block.

LSI/TSI

The character immediately following the DLE indicates the block type by the coding of its bits 7 and 6 as follows:

b7b6	Function	Type	Precedence
00*	None	–	–
01	LSI	LCB	1
10	TSI	Type A	2
11	TSI	Type B	3

Note: * The combination 00 cannot be used, as the LSI/TSI series could then include the reserved characters SYN, DLE, and ETB.

LSI Link Status Identifier

In a link control block, the five bits b5 through b1 identify the type of LCB as described in Section "Link Control Messages".

TSI Transmission Sequence Identifier

In a non-LCB, the five bits b5 through b1 are used as a Transmission Sequence Identifier. 31 combinations are used sequentially to serially number data message blocks on transmission, and to verify proper reception.

MBI Message Block Identifier

The MBI is used to permit chaining of Data Message Blocks.

The MBI combination 1/15 is used to indicate a single block message.

MBI bits b7b6b5 provide seven combinations for individual multi-block message labels.

The combination 001 is reserved to indicate a single block message.

MBI bits b4b3b2b1 provide sixteen combinations for sequential block numbering per message label. For a single block message, these bits will be coded 1111 to avoid any possible coincidence with the reserved character SYN, DLE, ETB.

LCI Link Characteristics Identifier

The LCI is mandatory in data message blocks. The bits of the LCI are used as follows:

b7=1 mandatory to avoid any possible coincidence with the reserved characters SYN, DLE, ETB

b6=0 indicates an original transmission

b6=1 indicates a possible duplicate message PDM or possible duplicate block, PDB

b5=1 indicates the presence of ACI

b5=0 indicates the absence of ACI from the control character sequence

b4=1 indicates inclusion of High Level Addresses

b4=0 indicates absence of High Level Addresses

b3=1 indicates maximum block protection not required

b3=0 indicates maximum block protection is required

b2 reserved

b1=0 not last block in a multi-block message

b1=1 last or only block

ACI Additional Characteristics Indicator

The ACI is optional in the TPF System

b7=1 mandatory to avoid any possible coincidence with the reserved characters SYN, DLE, ETB

b6 reserved

b5=0 must be zero – TPF Supports only one ACI

b4 reserved

b3b2b1
indicate code translation to be performed on message blocks as follows:

000	Padded CCITT No 2 code (Padded Baudot)
100	CCITT No 5 code (ASCII/ISO seven-bit code)
110	Extended CCITT No 5 code (this is a technique that enables a range of 256 data characters to be transmitted, rather than the 125 available within the CCITT No 5 code)

ETB End of Transmission 1/7

The ETB character indicates the end of the message block. It is a reserved character and may not be used elsewhere in the message block.

BCC Block Check Character

The BCC provides longitudinal even parity for each bit column (b7 to b1) excluding parity bits.

The BCC extends from the DLE to the ETB and includes both.

The BCC itself in common with all other characters has odd character parity.

On outgoing messages the BCC is generated by the IBM 3705EP. On incoming messages it is not passed to the CPU but is replaced by an Error Indicator Byte.

Link Control Messages

Positive Acknowledgment – ACK

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI	Positive acknowledgment 2/1
ATSI	Acknowledged TSI, the Transmission Sequence Indicator of the last correctly received block.

Negative Acknowledgment – NAK – IVB

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI Negative acknowledgment invalid block 2/2 indicating incorrect format, excessive block length, or parity error.

ATSI Transmission sequence indicator of the last correctly received block.

Negative Acknowledgment – NAK – OST

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI Negative acknowledgment out-of-sequence TSI, 2/3

ATSI Transmission sequence indicator of the last correctly received block.

Enquiry – ENQ

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI Enquiry 2/4. This message is sent during fault conditions and during restart. It requires a response.

ASTI Transmission sequence indicator of the last correctly received block.

Acknowledge Message Label – AML

Message: SYN SYN DLE LSI AMBI ETB BCC

Where:

LSI Acknowledge Message Label, 2/15 may optionally be sent when all blocks of a multi-block message have been correctly received and acknowledged.

AMBI Acknowledged MBI

b7b6b5 indicate the cleared message label

b4b3b2 are coded 000

b1 is coded 0 for a Type A message

b1 is coded 1 for a Type B message

Stop Sending – STP

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI Stop Sending on all channels, 3/0

Stop Sending on a specific channel 1 to 7, 3/1 to 3/7.

ATSI Transmission sequence indicator of the last correctly received block when a single channel is stopped. When the Stop Sending is for all channels the ATSI will be coded all ones.

Resume Sending – RSM

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI	Resume Sending on all channels, 3/8 Resume Sending on a specific channel 1 to 7, 3/9 to 3/15.
ATSI	Transmission sequence indicator of the last correctly received block when transmission is resumed on a single channel. When the Resume Sending is for all channels the ATSI will be coded all ones. Note that the Resume Sending on all channels is never generated and transmitted by the TPF Synchronous Link Control Package since it contains no ATSI information. Channels will be resumed individually.

Idle Line – ILB

Message: SYN SYN DLE LSI ATSI ETB BCC

Where:

LSI	Idle Line 2/14 sent during an idle line condition. It requires a response.
ATSI	Transmission sequence indicator of the last correctly received block.

Data Messages

The format of a data block, whether Type A or Type B, is either:
SYN SYN DLE TSI MBI LCI ACI Text ETB BCC or,
SYN SYN DLE TSI MBI LCI Text ETB BCC

The maximum length of the message block (excluding SYN characters) agreed by link governs the length of the text portion of the message.

Messages to and from a High Level Network (e.g., SITA) require two additional control fields to specify the HLN Entry and Exit Centers. The format of such a message will be:

SYN SYN DLE TSI MBI LCI HEX HEN ACI Text ETB BCC

or if the optional ACI character is not a component of the control character set:

SYN SYN DLE TSI MBI LCI HEX HEN Text ETB BCC

Where:

HEX	Address of Exit Center from the HLN
HEN	Address of Entry Center to the HLN

HEX and HEN comprise two characters each.

To identify the functional area to which input Type A messages should be routed, the first text character will be used as a Circuit Identity, and will not be translated.

Link Operational Procedures

For convenience, this section is divided into two parts: procedures for the Transmitting Center and procedures for the Receiving Center; however, it should be remembered that since the lines are full-duplex, every Transmitting Center is also a Receiving Center, and vice-versa.

Transmitting Center

1. Messages longer than the agreed maximum block size will be transmitted in several blocks.

   When a link consists of several lines, blocks of a multi-block message may optionally be spread (or ‘scattered’) over different lines of the link or may be sent on the same line. In either case, successive blocks of a multi-block message transmitted on a line may be interspersed with blocks of different messages with higher priority.

2. On a single line link, one Transmission TSI series and one Output MBI series per priority will be maintained. On a multi-line link a Transmission TSI series will be maintained for each line and one output MBI series per priority for the whole link.

   The Transmission TSI series will be advanced sequentially for each data message block transmitted without regard to message type. However, it will not be advanced more than a preset “TSI Exhaustion” value beyond the point to which it has been acknowledged and cleared.

   The Transmission TSI series will be reset on receipt of a link control block with ATSI bits b1 through b5 coded all zeros. It will also be altered on receipt of a link control block with an illogical ATSI (See Errors Section).

3. Message queues will be handled as follows:

   The Transmitting Center will queue messages for a link according to priority. Messages will be assigned an output MBI as follows:

   Single block messages: The message label will be 001 and the block chaining number 1111.

   Multi-block messages: The message label will be the first available combination of the possible seven, 000 and 010 through 111. Block chaining numbers will be allocated sequentially for each message label from 0000 through 1111.

   When a line becomes available, the queues will be checked in priority order to obtain the next block for transmission on that line and the next sequential Transmission TSI for that line will be allotted, in a series running from 1 to 31.

   Should the number of outstanding (unacknowledged) message blocks reach the TSI Exhaustion value, the Transmitting Center will cease transmitting data message blocks and send an Enquiry Link Control Block. The ATSI of the response (RSM or STP or optionally ACK) should enable the TSI Exhaustion to be relieved.

4. Message blocks are sent continuously with a block separation of at least two SYN characters. After transmission of a block has commenced it will not be interrupted either by a Link Control Block or a higher priority message block.

5. The Transmitting Center will bear the responsibility for data message blocks until the time they have been positively acknowledged. ACKn implies acknowledgment of all previously unacknowledged blocks up to and including the value of n. The following Link Control Blocks will also be interpreted as positive acknowledgment.
   STPn implies ACKn
   ENQn implies ACKn
   RSMn implies ACKn
   ILBn implies ACKn
   NAKn implies ACKn

   When all blocks of a multi-block message have been acknowledged and no AML agreement exists for the link, the message label is cleared for reuse immediately. However, when an AML agreement does exist for the link, the message label is cleared for reuse upon the receipt of an AML Link Control Block. If the AML has not been received within T1 or T6 seconds, one of two mutually exclusive processes are followed.

- When a message retransmission (N3) agreement does not exist for the link, an alarm message is generated and the message label is cleared for reuse.
- When a message retransmission (N3) agreement does exist for the link, the complete message shall be repeated using the same message label as previously and a possible duplicate message (PDM) shall be indicated. The message shall be retransmitted the agreed upon ‘N3’ times. If an AML still has not been received, the blocks of the multi-block message are retransmitted as a new message.

6. If the Transmitting Center receives a negative acknowledgment link control block, indicating invalid block NAK-INV or TSI sequence error NAK-OST, it will retransmit any outstanding unacknowledged blocks following the last one correctly transferred and acknowledged as identified by the ATSI of the NAK.
7. When the Transmitting Center has no data message blocks to send and there are outstanding unacknowledged blocks it will, if the idle line option allows, transmit an ILB link control block with the TSI of the last correctly received data message block in order to solicit an acknowledgment for the outstanding blocks. The Receiving Center will respond with RSM, or STP or optionally ACK indicating the TSI of its last correctly received data message block. If the above conditions continue, the ILB link control block will be repeated at time intervals T2. If no response is received for N2 successive ILBs, the line is declared out-of-service. If the option does not allow exchange of idle line ILB Link Control blocks, then the Receiving Center should exercise the Receive Idle procedure.
8. If the Transmitting Center receives a Stop Sending link control block for a specific line, it will immediately stop sending data blocks on the line specified by the LSI. All unacknowledged blocks for that line will be requeued for transmission on another line. Multi-block messages will retain their original MBIs. The Transmitting Center will continue to accept and process incoming data and link control blocks and transmit link control blocks. Idle line link control blocks (ILB or ACK) will be transmitted on the stopped line, at time intervals T2, intermixed with normal responses to traffic, until a Resume Sending is received.

   If the stopped line was the last active line on the link, then multi-block output messages which have not been cleared will be requeued for transmission, with the same output MBI, when a line becomes available.
9. If the Transmitting Center receives a Stop All link control block, the Enquiry procedure is entered for all lines.
10. If the Transmitting Center receives a Resume Sending link control block for a specific line, it will resume sending data message blocks on the line specified by the LSI commencing with the block following the last one correctly transferred and acknowledged as identified by the ATSI of the Resume Sending.
11. If the Transmitting Center receives a Resume All link control block, it will resume sending data message blocks on all lines of the link commencing in each case with the current value of the Transmission TSI series.
12. If the Transmitting Center receives an Enquiry link control block, it will respond with STP or RSM (and optionally ACK), and will then retransmit any outstanding unacknowledged blocks following the last one correctly transferred and acknowledged as identified by the ATSI of the ENQ.

13. If the Transmitting Center receives an Idle Line ILB link control block, it will respond with STP or RSM (and optionally ACK).

Receiving Center

1. On a single-line link, one Reception TSI series and one Input MBI series per priority will be maintained. On a multi-line link, a Reception TSI series will be maintained for each line and one input MBI series per priority for the whole link.

   The Reception TSI series will be advanced sequentially for each data message block received without regard to message type, and is used to facilitate the correct reception of data message blocks.

   When a link control block is generated and transmitted, the current value of the Reception TSI series is used as the ATSI of the LCB.
2. The Receiving Center will send a positive acknowledgment for every block of a multi-block message and at least once for every single-block data message, where the value of p is agreed by link. The ATSI of the ACK will indicate the last correctly received data message block and implies acknowledgment of all previously unacknowledged blocks.
3. Negative acknowledgment, NAK-IVB, will be sent by the Receiving Center to indicate incorrect format, excessive block length, or parity error in a non-LCB block. The ATSI of this link control block will indicate the last correctly received message block and constitutes a positive acknowledgment for that block and all preceding unacknowledged blocks. The error block will be dropped and all blocks received subsequently on that line will be dropped, without acknowledgment, until a block with next-in-sequence TSI has been correctly received.
4. Negative acknowledgment, NAK-OST, will be sent by the Receiving Center when the TSI of the received message is not the expected next-in-sequence TSI. The ATSI of this link control block will indicate the last correctly received message block and constitutes a positive acknowledgment for that block and all preceding unacknowledged blocks. The out-of-sequence block will be dropped and all blocks received subsequently on that line will be dropped without acknowledgment until a block with next-in-sequence TSI has been correctly received.
5. The NAK will be repeated at intervals of T1 seconds until either the correct block is received or the NAK has been repeated N2 times when the Receiving Center will go into the Enquiry procedure.
6. Incorrect reception of an LCB-length block (i.e., five characters) will result in the erroneous block being discarded without acknowledgment.
7. If the Receiving Center does not receive a data message block for a period of T1 seconds, it will transmit an idle line link control block (ILB or ACK) indicating the TSI of the last correctly received data message block. If an ILB is sent the Transmitting Center will respond with RSM or STP (or optionally ACK) indicating the TSI of its last correctly received data message block. The ACK or ILB link control block will be repeated at intervals of T2 seconds until a data message block is received. If no response is received for N2 successive idle line link control blocks, the line is declared out-of-service.
8. A Stop Sending link control block will be sent to stop the Transmitting Center from sending information on a line as specified by the LSI. It may be sent over any available line in a multi-line link. The ATSI will indicate the TSI of the last correctly received block on the specified line, and constitutes a positive acknowledgment of this and all preceding unacknowledged blocks. Any data

message blocks subsequently received on the stopped line will be discarded by the Receiving Center; incoming LCB's however, will continue to be accepted and processed.

9. A Resume Sending link control block will be sent to cause the Transmitting Center to resume the transmission of data message blocks on a specified line. This link control block will be sent only on the line to which it refers. The ATSI will indicate the TSI of the last correctly received block, and constitutes a positive acknowledgment for this and all preceding unacknowledged blocks.
10. When all blocks of a multi-block message have been correctly received, acknowledged, and assembled an AML link control block may optionally be sent to the Transmitting Center, indicating that this message label is cleared for further usage. On a spreading link, the AML may be sent over any available line. On a non-spread link, it will be sent on the line on which the block which caused the message to be complete was received.
11. If the Receiving Center discovers an overload condition, it will, if possible, send a Stop All down any available channel of the link, and subsequently respond to Enquiry link control blocks with Stop Sending.

    Two levels of overload are defined:

    a. At the primary level data blocks and link control blocks are accepted. (These data blocks would be 'inertia' blocks transmitted before the transmitting center could take action on the receipt of the Stop All.)
    b. At the secondary (lower) level, data blocks are discarded without response but link control blocks are accepted and processed.

    If it is not possible to send Stop All, the overload condition will be detected in one of two ways by the Transmitting Center:

    a. If the duration of the overload condition is greater than the link's time-out value, the Transmitting Center will go into the Enquiry procedure. If the overload condition still exists, the Receiving Center will either not respond to the ENQ, or respond with Stop Sending.
    b. If the duration of the overload condition is less than the link's time-out value but nevertheless, some data message blocks have been discarded, when the Receiving Center is able to accept data message blocks again, the first data message block accepted will have an out-of-sequence TSI and result in NAK-OST.

    When the overload condition is relieved, the Receiving Center will generate Resume Sending link control blocks on all lines and respond to Enquiry link control blocks with Resume Sending. Additionally, with each occurrence of a receive data time out (see paragraph 7 in "Receiving Center" Section), it will send a Resume Sending in addition to the idle line link control block, until such time as a data message block has been correctly received.
12. When a message discard (T7) agreement exists for a link, the receiving center, for each intermediate block of a multi-block message, will begin to time the message out. If the next or last block of the message is not received within T7 seconds, the partially received message is discarded. If the next block is received within T7 seconds, the message discard timer will be reset to T7 seconds to await subsequent block(s) of the message. When the last block of the message has been received, the message discard timer is set to preclude the occurrence of a time out and the complete message is then assembled and edited.

Link Cycling Procedures

Cycle-Up

To start the exchange of information on a line, the center will send an Enquiry link control block down the line.

Cycle-Down

To stop the exchange of information on a line, the center will send a Stop Sending and, if there are outstanding unacknowledged blocks, an Enquiry link control block down that line. To stop the exchange of information on a link, the above procedure is repeated for each line of the link. Type A messages due for output on the link and multi-blocked messages not yet assigned an MBI will be discarded. Type B messages which need to be secured by the link Control Software will be retained for transmission later.

Errors

Error Conditions

The Enquiry procedure is entered if a center detects any of the following conditions.

- The number of unacknowledged data message blocks exceeds a link defined value (TSI Exhaustion).
- A Link Control Block is received with an illogical ATSI, i.e., outside the range of the TSI's of the last block transmitted and the last block acknowledged.
- A link control block is received with an ATSI of zero.
- During the idle line condition (see paragraph 7 in “Receiving Center” Section), the idle link control block has been repeated N2 times without receiving a valid response. In addition, at this time, the line is declared out-of-service.
- Following the receipt of an incorrect data message block, the NAK has been repeated N2 times without receiving the correct block.
- STP ALL is received. The Enquiry procedure is entered for all lines of the link.

Enquiry Procedure

The center will stop transmitting data blocks and will send an Enquiry LCB indicating the TSI of its last correctly received data message block. Incoming data message blocks will be dropped but link control blocks will continue to be accepted and processed. If no valid response is received, i.e., RSM or STP (or optionally ACK), the Center will continue sending ENQs at time intervals T1. If a valid response with an illogical or reset ATSI is received, the ENQ will be repeated once more and the next response accepted. If the ATSI is still illogical or reset, the Transmission TSI series will be altered to the next expected TSI as indicated by the ATSI of the response. If after N2 repetitions of the ENQ no valid response has been received, the line will be considered defective and declared ‘out-of-service’ (unless already so; see “Error Conditions” Section).

Line Out Of Service

The Center will continue sending ENQs down the faulty line at intervals of T1 seconds. With each ENQ, an STP will be sent down any other surviving line with the LSI indicating the faulty line and its ATSI indicating the TSI of the last data message block correctly received on the faulty line. The Center will recover and resume normal operation on the defective line on the receipt of a valid response to an ENQ. If the line is out of service and the Center receives an ENQ, it will respond with Resume Sending (unless an overload condition exists; see paragraph 11 in “Receiving Center” Section) but will continue sending ENQs until it receives a valid response.

Line Invalidation

If hardware errors occur on a line, producing unit checks, and these persist on retry of the operation, then after a system defined number of retries a Stop Sending link control block will, if possible, be sent down any other surviving line with the LSI indicating the faulty line and its ATSI indicating the TSI of the last data message block correctly received on the faulty line. The line will be invalidated and an alarm message sent to an operator. The invalidated line can then only be restarted by operator action.

Items Subject to Bilateral Agreement

The following factors must be agreed by link between the parties concerned at each end.

1. The number of lines in the link.
2. The maximum size of data message blocks, bearing in mind the transmission requirements of the order of seconds for Type A messages, and the maximum message length of sixteen blocks.
3. The number of extra SYN characters required for character framing Two SYNs will automatically precede every block. SYNs will not be sent after a block, although it is possible to immediately follow one block with another, provided the second block begins with the required number of SYNs.
4. The frequency of acknowledgment of data message blocks (p). When p is greater than one, this does not preclude acknowledgment more frequently. In any event, acknowledgment will be sent by the TPF System for each block of multi-block messages.
5. T1 – the time between successive transmissions of ENQ or NAK without receiving a valid response.
6. T2 – the time between successive transmissions of the idle line link control block during the idle line procedure.
7. N2 – the number of ENQ, ILB or NAK repetitions without valid response before the line is considered impaired.
8. Whether an Acknowledge Message Label LCB is required to clear a message label for reuse.
9. The inclusion of additional message envelope control characters. The following envelope characters are required on all links:

   ```
   SYN SYN DLE TSI MBI LCI — ETB BCC
   ```

   The Additional Characteristics Indicator (ACI) is not a mandatory component of the message control character set, and its inclusion or omission from the message envelope must be agreed by link between the parties at each end.

   On connections to a High Level Network (e.g., SITA), two additional control fields are required following the LCI:

   HEX to specify the Exit Center from the HLN.

   HEN to specify the Entry Center to the HLN.
10. Whether ACK is an acceptable response to ENQ and ILB.

11. The number of unacknowledged data messages blocks before reaching “TSI Exhaustion”.
12. Whether blocks of multi-block messages may be transmitted down any line of a multi-line link (spread or scatter) or are confined to one line (non-spread or non-scatter).
13. The use of ILB link control blocks in the idle-line condition.
14. T6 – the time interval allowed between receipt of the last ACK of a multi-block message and the Acknowledge Message Label (AML) LCB before the entire message will be retransmitted. T6 is an alternate value to T1.
15. T7 – the time interval allowed between receipt of successive blocks of a multi-block message before the partially received message is discarded.
16. N3 – the number of times a multi-block message is retransmitted when no Acknowledge Message Label (AML) LCB has been received.

Note: T6, T7, and N3 are optional items; if these parameters are to be defined, they are subject to bilateral agreement between both parties on the link.

Communications Control Program Trace

The communications control program trace package is intended for use as a test tool in the development and maintenance of the communications control program (CCP).

It is designed to provide a sequential record of the activities (CCP macros, SIOSCs, RIOSCs and multiplexor interrupts) of a single line or one or more groups of lines. This record may optionally be written to the RTL/RTA tape and post processed by STPP, or may take the form of a *wrap-around* in core record containing at any time the trace information on the last 42 activities traced.

The trace operation is initiated, modified, and terminated via commands. These commands are also written to the RTL/RTA tape and listed with the CCP trace by the diagnostic output formatter (DOF).

Purpose

This package, when activated by a command, intercepts CCP processing of all CCP macros, SIOSCs, RIOSCs and multiplexor interrupts. If the line being operated on has been specified as a line to be traced, a snapshot of pertinent areas of core is taken and stored in a slot within a 1055-byte core block, before returning to normal CCP operation.

When all slots of this core block have been used, it is either reused or written to the RTL/RTA tape, depending on the option specified in the last CCP Trace command.

If the option of writing to the RTL/RTA tape is used, processing the RTL/RTA tape with DOF will produce a formatted listing of all items traced.

At any time while CCP Trace is active additional lines may be added to the list of lines being traced, and the option of writing to the RTL/RTA tape or reusing the core block may be varied via command. Also, CCP Trace may be deactivated at any time via command. When CCP Trace is deactivated, tracing is forced into ″wrap-around″ in core trace, and tracing will continue on the PRC and RO.

Functions

Activating CCP Trace

Activating CCP Trace involves:

- Formatting the two 1055-byte low core trace areas in copy segment CLVQ.
- Turning on a bit in the Lost Interrupt Table entry for the subchannel to be traced. If trace is being activated for either the PRC or RO, then the console trace bit for that subchannel is also turned on.
- Activating a *hook* at each point in the CCP that is to be traced. These ‘hooks’ or intercepts cause the CCP to link to the trace recording segment, CLVQ, instead of *falling through* the hook and continuing normally.

Deactivating CCP Trace

Deactivating CCP trace involves:

- Turning off all non-console trace bits in the lost interrupt table. If the console trace bit is on for a subchannel, neither bit is reset for that subchannel.

- If necessary, writing the partially filled core block to tape.
- Indicating the non-console trace area in low core will need to be reformatted before any non-console tracing is reactivated.
- Forcing trace into loop mode, so that in-core loop trace will continue on the console devices.

As part of CCP Trace operation, all CCP Trace commands are written to the RTL/RTA tape for DOF processing.

Interception

If CCP trace is active, whenever the CCP is processing CCP macros, SIOs, HIOs or multiplexor interrupts, the operation is intercepted. Pertinent information about the operation is recorded in the appropriate CCP trace low core area (console or non-console), then the operation is allowed to complete.

Output

If the option to write the trace blocks to tape was selected, as each low core trace area becomes full, a 1055-byte user storage block is obtained. Also, the information is copied from the low core trace area to the user storage block, and the user storage block is written to the RTL/RTA tape. In addition, when CCP tape trace is deactivated, the last two partially full trace areas are copied to user storage and written to the RTL/RTA tape.

Restrictions

- CCP trace may not be running when VFA is stopped or reinitialized; VFA will notify the operator to stop CCP trace before VFA processing is allowed to continue.
- In a Loosely-Coupled environment, CCP trace can only be activated on a non-EP processor by specifying the PRC or RO options. No other lines may be traced on a non-EP processor.
- The PRC option is only valid if the PRC is currently on symbolic line 01.
- The RO option is only valid if the RO is currently on symbolic line 00 or 01.

Programs

Programs In This Package

03-CCTR	CCP TRACE
	CCP Trace Restart/Shutdown (CIJG/CIJS) segments.
	This program accepts CCP Trace commands from the Line Control Editor (03-CVHD), performs extra validity checks on the parameters supplied, and processes the command. This involves activating or deactivating CCP Trace as previously described in the Functions section, or modifying the trace parameters. The process of deactivating trace involves forcing trace into in-core loop mode, and trace will continue tracing I/O activity to the PRC and RO.
	CCP Trace Tape Block Post Interrupt (CIJF) Segment
	If a trace core block becomes full and it is required to write it to tape, the block is attached to the Post Interrupt List for entry to this program.
	03-CCPNUC

CCP Trace Recording Segment CLVQ

This copy segment is part of CCCCP1 and is always present in memory. When entered, this segment determines if this line is to be traced, and if so, records the necessary information in the appropriate trace core area before returning from the CCP hook. It is entered from the CCP *hooks* as previously described in the Functions section through the CCP trace linkage routine (also in segment CLVQ). Also, this routine contains constants and variable fields necessary for the operation of CCP trace.

Associated Programs

03-CVHD Line Control Editor

This program receives and edits all line control commands, including CCP trace commands, and enters the appropriate message processing program. For CCP Trace commands, entry is made to (segment CIJG to activate CCP Trace, segment CIJS to deactivate it.

Input

CCP Trace is activated, deactivated, and has its parameters altered via the following commands:

- ZLTRF
- ZLTRN
- ZLTRL

See *TPF Operations* for more information about these commands.

Output

When CCP trace is active and the option of recording to the RTL/RTA tape is selected, the CCP trace recording segment CIJF will add each block to the PI list as it is filled. The block contains an enter expansion to CIJF Transfer Vector 5, which operates as a normal application program to write the core blocks to the RTL/RTA tape.

If the write to the RTL/RTA tape option was not selected, no output is generated, but on any dump that occurs, the *wrap around* trace core block will appear in the dump with the header CCP TRACE BLOCK. This block will contain a record of the last 40 activities traced.

Messages And Replies

The only console messages output by CCP trace are replies to input commands. For more information about these console messages, see *Messages (System Error and Offline)* and *Messages (Online)*.

Communications Control Program

All communication between the various external users of the system and the actual application programs is controlled by the communications control program (CCP). As such, the CCP is the interface between the application programs and the communication network. For SNA/NCP support, refer to the SNA overview in the *TPF ACF/SNA Data Communications Reference*. This chapter does address lines controlled by 3705 EP and locally attached 3270's.

Description

The purpose of the CCP is to provide an access method where by the application programs are relatively independent of the communications hardware operation.

Polling and Circuit Assurance

The CCP handles all polling (telling the Terminal Interchange (TI) to send in any messages it has in its buffer) on a time basis. Further, if the amount of available main storage is low, polling is postponed until enough main storage is again available. The CCP also handles the circuit assurance on all communications lines. This is to ensure that the lines are in an operable condition.

Execution of I/O Operations

The CCP executes all communication I/O operations for the application programs. The CCP sets up the CCWs, starts the I/O and handles the interrupts from the operation.

Main Storage Allocation For The Associated I/O Operations

The CCP obtains main storage on input operations and releases the output block.

Character Code Translation For Both Input And Output

The CCP translates all input messages except 3270 Local into an appropriate internal character code. On output, it translates the internal code to the appropriate line code for transmission except for 3270 Local. Both input and output for the 3270 Local are in EBCDIC, except for control characters, which are in hexadecimal.

Input Message Editing And Activation of the Application

This includes the assembly of HS input messages into a single main storage block, if needed, and checking for invalid characters.

CCP performs backspace editing, if required, and then activates the Router. The Router will activate the appropriate application program which will route the message to the proper application segment for processing.

Hardware Error Detection And Correction

The CCP performs all hardware error detection at interrupt time and attempts correction of the error condition. If error correction fails, notification is sent to operations and the line turned down.

Software Error Detection And Processing

The CCP detects all software errors associated with the communications lines and processes them by issuing an appropriate system error.

Communication Considerations for Loosely Coupled Support

The loosely coupled feature allows multiple processors running the same applications to concurrently access a common database. The balancing of the processing load across all processors in a complex is achieved by distributing the input message traffic across the complex. It is irrelevant which processor handles a terminal user's input, since each processor in a complex provides identical service. To a terminal user, say, a reservation agent, a loosely coupled complex has the appearance of a single TPF system. To the system operator, however, the system has the appearance of separate but connected processors. The system operators of a loosely coupled complex has the ability to run the complex from any or all loosely coupled processors.

There are three (3) considerations relative to communications support for the loosely coupled feature.

- Distribution of Message Traffic

  In order to balance the work load across the processors comprising a loosely coupled complex, the user must distribute the message traffic across those processors. The message traffic distribution technique available depends on the types of terminals in the loosely coupled network. The following describes the message distribution technique used relative to method or connection.

  3705 Emulator program (EP) – All terminals which access TPF through a 3705-EP must submit their input to a specific loosely coupled processor, called the EP Processor. Therefore, message traffic distribution techniques are not available for terminals in this group.

  3725 Network Extension Facility (NEF) –

  TPF Network Extension Facility (NEF) support depends on either:

  - The Network Extension Facility (NEF2 PRPQ P85025), or
  - Airlines Line Control Interface (ALCI) feature of ACF/NCP.

  All subsequent occurrences of NEF in the text of this publication are superseded by the previous information.

  All NEF terminals access the TPF loosely coupled complex through the 3725 NEF. Each 3725 with NEF is represented to each processor by one logical unit (LU). Message traffic from NEF terminals can be distributed across the complex by a user-written NEF exit routine. This routine resides in the 3725 and is link-edited with the NCP.

  SDLC NCP Terminals – Message traffic from these terminals can be distributed across the loosely coupled complex by two methods. The first method is conceptually identical to the NEF approach where multiple LUs are assigned to each cluster controller. Each LU represents a port into the loosely coupled complex which can be used to route messages to a specific processor. The user must write code in the programmable controller to distribute message traffic. The algorithm for selecting the specific processor is left to the user to develop. Some valid algorithms are:

  - Round-Robin

    Send the first input to the first available processor, the second input to the second available processor, etc.
  - Processor Preference

    Define to each cluster controller a list of preferred CPU-IDs. The user's algorithm would then send each input message to the most preferred active processor.

In both algorithms, it is assumed that when processors are added or deleted from the loosely coupled complex, the algorithm automatically redistributes the input message traffic across the complex. Automatic redistribution of input message traffic based on the processor composition of the complex may not be desirable. For example, assume an unplanned shutdown causes the loosely coupled complex to lose a processor and the algorithm automatically directs traffic to the remaining processor(s). In this case, the CPU utilization on the remaining processors may reach 100%, seriously affecting response time. Therefore, algorithms with the ability to automatically redistribute message traffic when a loosely coupled processor shuts down are viable only if the remaining loosely coupled processors can assume the full network message traffic load.

The second method for distributing message traffic from SDLC terminals is to assign, at network definition, the ownership of each cluster controller to a specific loosely coupled processor. With this procedure, the user must attempt to balance the message traffic across the loosely coupled complex through the assignment of cluster controller ownership. This is the only method available for devices that are not user-programmable, such as 3274/3276 cluster controllers.

- Routing Messages Between Loosely Coupled Processors

  In a loosely coupled complex, there are two classes of messages which are routed between processors. These classes are:

  - System operator input and output.
  - Output routed to a TPF application or terminal that is processor unique. Processor unique means that only one processor has the application or has a path to the terminal; SMPx is an example of a processor unique application; 3705 EP terminals are examples of processor unique terminals. In either case, when a message is routed to a processor unique resource in another processor, the “ROUTC” request is exported to the appropriate processor and then executed.

    The mechanism for exporting ROUTC requests to other processors within a loosely coupled complex is called *System Inter-Processor communication* (SIPC). The medium used for inter-processor communication is either a shared or dedicated file storage control unit.

- Routing CRAS Messages Between Processors.

  Terminals in the CRAS table logged to the System Message Processor can direct an input message to a specific application, or the ‘Z’ command processor in a Subsystem or Subsystem User. This method can also be used to direct an input message to another system. To accomplish this, the operator enters a message in the following format:

  ```
  aaaa/text
  ```

  ‘aaaa’ specifies a one- to four-character name (it may be an application name, a subsystem name or a subsystem user name) or ALL to indicate routing to SMP in all processors.

  ‘text’ is the data to be sent to the application and is forwarded without inspection.

  For example, the following input from processor B would result in a display of main storage location 10 in Processor A.

  ```
  SMPA/ZDCOR 000010
  ```

  This function could also be used to invoke a user application in any processor. For example:

  ```
  SMPB/RES0/OAA
  ```

input from processor A would result in OAA being sent to processor B for execution by application RES0.

Note: This message has been doubly prefixed to allow routing to SMPB in Processor B and then to RES0.

Component Programs

Packages Used by This Package

The CCP provides the I/O programming required to support four distinct types of communications media:

- 1052 or 3215 console connected directly (point-to-point) to the multiplexer channel
- Synchronous Link lines
- Binary Synchronous Communication (BSC) lines
- 3270 Local Lines and 3270 Native Console support.

Consequently, this package consists of four major main storage resident line control programs, and a collection of commonly used main storage resident subroutines. In addition, there are a vast number of file-resident support programs, each of which is associated with one or more of the above line control programs.

1052 or 3215 Console (Control Program Only)

This program performs all the major functions for the 1052 or 3215 and is composed of the following parts:

- 03-CCPNUC Communications Control Program Nucleus (1052/3215 I/O Program)

  This program controls I/O operations between the CPU and 1052/3215 operator's console if the 1052 option is selected.

- 03-CCPNUC Communications Control Program Nucleus

  FRACTIONAL CHANNEL CONTROL PROGRAM

  This area of the CCPNUC contains the routines which provide the overall control of the I/O operations on the multiplexor channel. Examples of routines in the FCCP are the General SIO and MPX interrupt analysis routines. The General Start I/O routine initiates all I/O operations for the main storage resident I/O initiation routines of the various line control programs (i.e., bi-synch). The MPX Interrupt Analysis Routine examines the Multiplexor I/O interrupt and directs the interrupt to the appropriate interrupt processing routine.

  COMMUNICATIONS MACRO ROUTING PACKAGE

  This area of the CCPNUC is responsible for directing macro requests to the proper macro routine based upon control unit type (i.e., 2703, 3270 Local, etc.). It also performs validity checks to insure proper macro inputs.

  COMMUNICATIONS KEYPOINT UPDATE MODULE

  This area of the CCPNUC contains the main storage resident code which is activated by the PKEY macro to initiate the update of a SLSTL entry or PKST entry.

  COMMUNICATIONS MULTIPLE USED ROUTINES

  This area of the CCPNUC contains routines and tables which are common to one or more of the line control programs.

  1052/3215 I/O PROGRAM

  This portion of the CCPNUC provides the support for the 1052 or 3215 console as an operator control station.

Of additional interest, the communications multiple used routines of the CCPNUC contains the macros (CRAS and SEND) which interface with the long message transmitter program and which output messages to the 1052/3215 I/O program.

Synchronous Link Lines (Control Program Only)

This program performs most of the major functions listed above (see "FUNCTIONS") for the Synchronous Link lines and is composed of the following parts:

- 03-CIOAI Synchronous Link Control CCP I/O Program

  This program controls input and output on the Synchronous Link line. It also provides, by means of macro support, an interface between application programs and the Synchronous Link Control Output Message Handler (03-CMS).
- 03-CCPNUC Communications Control Program Nucleus

  See "1052 or 3215 Console" for description.

BSC Lines (Control Program Only)

- 03-CIOBSC BSC I/O Routines

  This program controls the sending and receiving of data on the BSC lines.
- 03-CCPNUC Communications Control Program Nucleus

  See "1052 or 3215 Console" for description.

3270 Local Lines (Control Program Only)

- 03-CIOLC 3270 Local I/O Program

  This program controls the sending and receiving of data on the 3270 Local Lines.

  This program also controls the I/O operations between the CPU and the 3270 Local or natively attached 3270 operator's console.
- 03-CCPNUC Communications Control Program Nucleus

  See "1052 or 3215 Console" for description.

E-Type Programs Logically Continuous To The Control Program Resident Programs

Synchronous Link Lines (SLC), Binary Synchronous Communication (BSC), 1052/2315 Console, 3270 Local (LC) I/O Programs, and 3270 Native Console Support (NSC).

- 03-CIAA Communications Source

  The programs comprising Communications Source receive complete or partial message or data segments from the OPZERO functions of the relevant line discipline programs. Translation, editing and message assembly is performed as required. The message or data having been prepared for further transmission in the network or for a system or application editor is directed there by the construction of a Routing Control Parameter List (RC0PL) and the activation of the Message Router (03-ROUT).

E-Type Support Programs (1052/3215 Line Control)

The following file-resident support program is associated with the 1052 line control program.

- 03-CVKM Console Fallback

  When the prime CRAS device, 1052, 3215, or 3270 local CRT is inoperable, this program attempts to find a substitute. First, it will look for an alternate 1052/3215 in the first slot (line 00) of the Line Status Table. If an alternate 1052/3215 is not available, it will next look for a locally attached 3270 CRT device and its cross-referenced printer. If this fails, it will finally look for a valid 1977, or a valid CRT (2915, 4505, or 3270) with a cross-referenced RO printer, in the CRAS Status Table (CR0AT).

Support Programs (Synchronous Link Lines)

The following file resident support programs are associated with the Synchronous Link lines:

- 03-CMC Synchronous Link Control Link Controller

  This is a set of routines that controls the automatic handling of the link. It processes and generates LCB's as and when necessary.
- 03-CMR Synchronous Link Control Link Input Message Handler

  This program handles data blocks received on a link. It performs validity checks on each block, and causes an ACK or NAK to be sent in reply. When a complete message has been accepted, the program reformats it translates the text into EBCDIC, and passes the message to the appropriate message processing program.
- 03-CMS Synchronous Link Control Output Message Handler

  This program builds and queues output message blocks for transmission via synchronous link lines. These blocks may contain whole messages being transmitted for the first time, or whole or parts of messages that are being retransmitted due to reception of a NAK LCB.
- 03-CML1 Synchronous Link Control Restart Initializer

  This program sets up pointers to link and channel keypoints, and initializes queue control fields.
- 03-CML2 Synchronous Link Control Restart/Cycle Up.

  This program restarts the synchronous link lines.
- 03-CML3 Synchronous Link Control Start Lines/Links

  This program allows the selective starting of synchronous link lines.
- 03-CML4 Synchronous Link Control Stop Lines/Links

  This program allows the selective stopping of synchronous link lines.
- 03-CML5 Synchronous Link Control Display Link/Line Status

  This program displays synchronous link line status information.
- 03-CML6 Synchronous Link Control Line Invalidation

  This program invalidates a synchronous link line on which a non-correctable error has occurred. It then restores the tables and controls associated with the line to their initial settings.

Support Programs (BSC Lines)

The following file resident support programs are associated with BSC lines:

- 03-CRML BSC Long Message Transmitter

  This program is used to retrieve file segments of long messages on the BSC line queues. It also performs other functions requiring an ECB driven program such as releasing file pool records.
- 03-CRM0 Validate A BSC Line

  This program conditions the transmission control unit and initializes tables so that a BSC line may be started.
- 03-CRM1 Start BSC Lines

  This program starts the sending and receiving of data on the BSC lines.
- 03-CRM2 Stop BSC Lines

  This program stops the sending and receiving of data on a BSC line.

- 03-CRM3 Display BSC Line Status

  This program allows the system operator to determine the status of a BSC line.
- 03-CRM4 Alter BSC Send/Receive Controls

  This program allows the system operator to balance the load between the two CPU's on a BSC line.
- 03-CRMB BSC Online Test

  This program initiates and responds to request-for-test messages on a BSC line.
- 03-CRM5 Alter BSC Line Slow Poll/Poll Time Out Interval

Support Programs (3270 Local Lines)

- 03-CTLLC 3270 Local Functional Programs

  This document describes the routines that will invalidate, display status, start, stop and restart the 3270 local lines. They also are activated in the event of either correctable or uncorrectable I/O error correction procedures.

Support

The following file-resident support programs are associated with two or more line control programs.

- DCSLCRAS CRAS Support PRM

  This program allows the computer room operator to display and modify the contents of the CRAS Status Table.
- 03-CVFY Alter CCP Error Counts

  This program zeroes the CCP's error counts on file.
- 03-CVHD Line Control Editor

  This program analyzes 3270 Local line, SLC, BSC control commands and routes the messages to appropriate line control routines.
- 03-CVKD Display BSC Station Status

  This program will display, for computer room personnel, the status of the terminal interchanges on a given BSC line.
- 03-CVLL Communications Control Unit Status Display

  This program displays the current communications control unit configuration upon request from the operator.
- 03-CVMA Update CCP Error Counts

  This program periodically updates the CCP error counts on file from the error counts in main storage.
- 03-CVMB Display CCP Control Unit And Line Error Counts

  This program displays the CCP control unit or line error counts from file upon request from the operator.
- 03-CVMG Display/Alter Restart/Shutdown Polling Controls

  The CCP has the ability to discontinue polling the 3270 Local lines based on the size of the input list. This program will display or alter these polling controls. For 3270 Local devices which are attention driven versus polled, the concept applies to whether or not to service the Attention interrupt. It is not applicable for a 3270 Local logged to "System Message Processor" (03-CSMP).
- 03-CVMK Error Message For Communications Control Unit

  This program sends to the prime CRAS an error message identifying a failing control unit by symbolic control unit number when excessive control unit errors have been detected by the main storage resident CCP error routines.
- 03-CVPY Housekeep Line Controls

This program turns down (invalidates) a line, or pair of lines, on which an non-correctable error has occurred. It also restores all tables and controls associated with the line.

- 03-CVQP Communication Lines Restart

  This program restarts all the communications lines in the system. Appropriate programs are activated sequentially by this program to restart the BSC, SLC, and 3270 Local lines.
- 03-CVQS Error Messages For Communication Lines

  This program formats an error message, from information passed by the CCP, qualifying an error type that occurred on a communication line and sends it to the prime or RO CRAS.
- 03-CTIP Communication Table Initialization Program

  This program initializes the communication table in the restart schedule.

  The tables initialized are:
  - CCP CCW Area
  - Branch Vector Table
  - CCW Pointer Table
  - Line Status Table
  - Poll Status Table
  - Symbolic Line Status Table
  - Communication Control Unit Keypoint Status Record (Main Storage initialization).

  The program uses keypoint information in:
  - System Communication Keypoints
  - System Communication Configuration Keypoint
  - Communication Control Unit Keypoint Status Record.

Relationships to Other Packages

Packages Used by This Package

This package activates the 18-UI package or 03-CSMP dependent on the prime action code. The Router will be activated to rout the message to the appropriate editor for high-speed input. This package activates the synchronous link control package for input from synchronous link lines.

Packages Using This Package

This package is used by the 18-MR, 03-CSMP, 18-UI, and 18-SLC packages for output transmission on communications lines. It is also used by various other segments or packages for output transmission. Reference *TPF System Macros* (Send Macros) for user restrictions on the direct activation or use of the communication control program.

External Input and Output

Legend:

BCI	Block Check Indicator
CCC	Cyclic Check Character

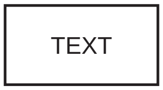
CCT	Character Count
CR	Carriage Return
ECB	Entry Control Block
EOM	End of Message
IA	Interchange Address
LF	Line Feed
LN	Line Number
NU	Not Used
SEG	Segment
SLN	Symbolic Line Number
TA	Terminal Address

Set Input/Output

Incoming Messages

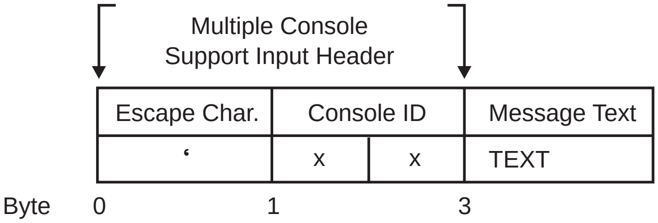
The high-speed message editors handle the following types of incoming messages:

1. EBCDIC input from 3270 local terminals:



Note: TEXT includes command byte first and can include 3270 cursor address and orders.

2. Automation gateway input message to TPF:



Where:

xx	EBCDIC hexadecimal digits ranging from 00 and 02 through the maximum terminal address specified in your IBM Extended Operations Console Facility/2 configuration.

Note: For additional information on the maximum terminal address, see the *IBM Extended Operations Console Facility/2 System Administrator's Guide*.

EBCDIC output to other packages for the previously illustrated formats:

HEADER	CCT	RESERVED	RESERVED	RESERVED	NOP	NOP	TEXT	EOM
0	16							

Note: TEXT can have terminal control characters included in it. All translation control characters are deleted during translation.

Outgoing Messages

EBCDIC input from packages or the Message Router at the point of transmission to 1977, 2915, 4505, and 3270 terminals, and the automation gateway:

HEADER	CCT	LN	IA	TA	TEXT	EOM

CIRY	CIRD	UC	TEXT	CIRB	EOM

Note: TEXT can have terminal control characters. For 3270 Local, the TEXT must start with the appropriate 3270 command followed by a Write Control Character (WCC) as a minimum.

This previously illustrated format is transformed upon output to one of the following formats:

1. EBCDIC output for 3270 local terminals:

WCC	TEXT

Note: WCC is the 3270 Write Control Character and the output message can contain only a WCC. If TEXT is included, it can contain 3270 orders.

2. Automation gateway output from TPF:

Multiple Console Support Output Header

	Escape Char.	Console ID		FSC Indicators				Message Text
	‘	x	x	b	b	b	b	TEXT
Byte	0	1		3		5		7

Where:

xx	TA converted to EBCDIC hexadecimal digits ranging from 00 and 02 through the maximum terminal address specified in your IBM Extended Operations Console Facility/2 configuration.
bbbb	EBCDIC hexadecimal digits representing functional support console (FSC) indicators used to route messages to specialized consoles. Multiple FSC indicators can be set.

Possible FSC indicators as defined in RTCEQ are shown in the following table.

Table 1. FSC Indicator by Type

FSC Type	FSC Indicator
RO CRAS (Receive Only CRAS)	8000
PRC (Prime CRAS)	4000
Tape	2000
DASD	1000
Communications	0800
AUDT (Audit Trail)	0400
Relational Database	0200
9 additional user-defined functional support consoles	0100-0001

Note: For additional information on the maximum terminal address, see the *IBM Extended Operations Console Facility/2 System Administrator's Guide*.

Synchronous Link Input/Output

For a complete description of message formats see "Synchronous Link Control Functional Specification" on page 1 and "Synchronous Link Control Link Trace Package" on page 33.

BSC Input/Output

See *TPF Operations* for a description of BSC message formats.

3270 Local Input/Output

See *TPF Operations* for a description of 3270 local message formats.

Internally Activated Processes

Error Processing

The CCP will activate appropriate error routines per line type when errors are detected on the lines.

Line Fallback and Recovery

(03-CVQS) Error Messages For Communication Lines

This program is activated on the initial error but not on subsequent retries for that error.

(03-CVPY) Housekeep Line Controls

This program is activated for all lines except synchronous link lines when the line error is serious enough to turn the line down (invalidate the line).

(03-CML6) Synchronous Link Control Line Invalidation

This program is activated when the line error is serious enough to turn the line down (invalidate the line).

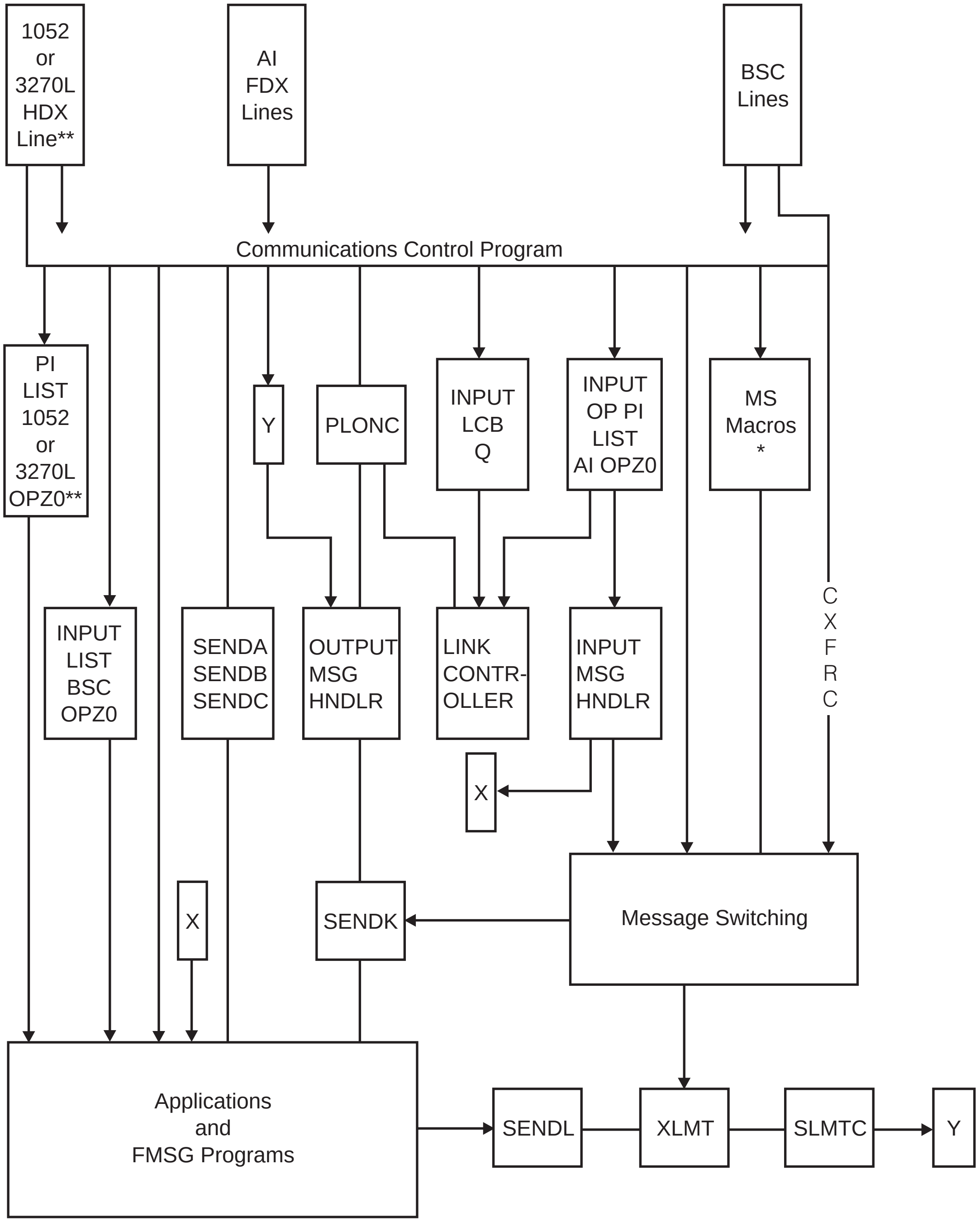
(03-CVMK) Error Message For Communications Control Units

This program is activated to send a message when an error is attributed to a control unit.

(03-CVKM) Console Fallback

When the 1052 or 3270 local as prime CRAS becomes inoperative, this program is activated in an attempt to switch the prime CRAS set.

Message Flow Overview



Synchronous Link Control Link Trace Package

The synchronous link control (SLC) link trace package is a diagnostic tool you can use to maintain SLC communication lines and to provide statistics about them. It provides a sequential record of all data blocks and line control blocks (LCBs), both input and output, on all links. For each data block or LCB, a record containing the following information is written to the link trace (LKT) tape:

- Time stamp
- Link status identifier (LSI) and ATSI for LCBs
- Transmission status identifier (TSI), message block identifier (MBI), automatic line identifier (ALI), character count, and error information block (EIB) for messages
- Various fields from the link and channel keypoints that indicate the status of the link
- Symbolic link number and link channel number.

The LKT tape is processed by the SLC link trace postprocessor, which formats the output.

Use the ZLKTN and ZLKTF commands to start and stop the SLC link trace. For more information about these commands, see *TPF Operations*.

The SLC link trace can be active across a state change.

Using the SLC Link Trace Package

To use the SLC link trace package, do the following:

1. Initialize and mount the LKT tape.
2. Issue the ZLKTN command to start the SLC link trace.
3. Issue the ZLKTF command to stop the SLC link trace.
4. Use the SLC link trace postprocessor to process the LKT tape.

After a Hardware Switchover

Always stop the SLC link trace before a hardware switchover. Before you can start the SLC link trace again, you must mount a new LKT tape.

After an Initial Program Load (IPL)

The TPF system stops the SLC link trace during an IPL, but the LKT tape remains open. To start the SLC link trace again, do the following:

1. Leave the original LKT tape mounted.
2. Enter the ZLKTN command to start the SLC link trace again.

   Note: The trace block before the IPL occurred is not written to the LKT tape. You can find this trace block in the catastrophic dump.
3. Enter the ZLKTF command to stop the SLC link trace.
4. Use the SLC link trace postprocessor to post-process the LKT tape.

After a Timer Alteration

If the basic subsystem (BSS) or time of day (TOD) clock is changed while you are using the SLC link trace, the trace records are not sorted correctly. Do the following to correct the sorting of the trace records:

1. Enter the ZLKTF command to stop the SLC link trace.
2. Enter the ZLKTN command to start the SLC link trace again. The correct sequence of traced link activity is restored.

Synchronous Link Control Package

See “Synchronous Link Control Functional Specification” on page 1 for more information about the terminology and protocols unique to SLC that are used in this chapter.

Support for synchronous link control is in three major areas. The CCP includes the logic directly concerned with communications I/O handling on synchronous link lines. This is subdivided into the core-resident CCP segments and supporting file-resident CCP segments.

Secondly, a group of ECB controlled programs forms an interface between system programs and the CCP for processing input messages (input message handler and SLC communications source) and output messages (output message handler).

Thirdly, another group of ECB controlled programs maintains overall control of the link via keypoint records and other tables and handle the special status messages unique to synchronous link control, that is, link control blocks (LCBs). These programs comprise the link controller.

Each of these groups are described in more detail in subsequent sections.

The link trace function is described in “Synchronous Link Control Link Trace Package” on page 33.

In addition, there are the following facilities:
- Input LCB Queues
- Output LCB Queues
- Input Block Queues
- Output Message Queues
- Applications Message Queues (AMQ)
- Message Control Records (MCR)
- SLC Common Area
- Link and Channel Keypoints
- Link Routing Table
- Macro Support.

The AMQ and MCR records are discussed under output message handler and the link routing table under input and output message handlers and SLC source communications. The other facilities are discussed in “General Facilities” on page 36.

A system may have more than one link, with differing options as described in “Synchronous Link Control Functional Specification” on page 1. In general, control of a Link is effected through the link and channel keypoint records. The link controller programs establish operational conditions on a link when cycling to CRAS state, or above. They are also driven by file-resident CCP and incoming link control blocks (LCBs), and generate and time-out routine monitors conditions (it is activated through the timer and CPU loop) and adds pseudo LCBs to the input LCB queue for processing by the link controller. file-resident CCP performs transient CCP functions and acts on command entries. Input message handler receives all incoming information blocks, through CCP post interrupt and link OPZO. SLC communications source manipulates them to suit applications requirements and passes them to the appropriate programs. Output message handler accepts applications messages for transformation into link message blocks and queues them for transmission using the macro PLONC-Place on Queue.

Core-resident CCP performs the normal I/O functions, adding incoming LCBs other than ENQs to the input LCB queue. ENQs and all message blocks are added to an input list (which may be the ready or input list as a system option) and subsequently passed to link OPZO that adds them to the appropriate channel input block queue for eventual processing by the input message handler. It also includes logic for queueing output applications messages to be processed by output message handler.

General Facilities

Macro Support

SENDCK – Send To Link

This is used by system programs to route messages to the output message handler for eventual transmission across the link.

PLONC – Place On Queue

This is used by output message handler to pass messages and by link controller to pass link control blocks to the CCP for transmission across the link.

Communications I/O Utility Macro

Since the synchronous link comprises free running full duplex lines and the CPU must always be in a position to receive information, a channel program is always active on the receive line. The CIOUC is used by the file resident CCP to initiate the channel program when a line is started and to terminate it when the line is stopped. In addition, if SLC has been implemented using a 3705 communications controller, line enabling and disabling is activated through CIOUC.

SENDC C/A SLMTC

These macros are used by system programs to send messages to high speed display (CRT) and printer terminals. Support for indirectly connected terminals (through a high level network, that is, SITA-HLN) is in the macro support to *intercept* messages destined for the indirect terminals and to pass them to the Output Message Handler.

Link and Channel Keypoint Records (LK4KC)

These consist of a keypoint record containing information pertaining to a link, and a keypoint record for each channel of the link containing information relevant to each channel. The link keypoint contains bilateral agreement values and option indicators, output message block and LCB queue control fields, and input and output MBI information. The channel keypoint contains channel status, procedure-in-progress indicators, error counts and input and output TSI information.

These keypoint records are generated by coding CTKL macros which are assembled as program segments. These special program records are defined in the Global 3 Directory. See *TPF General Macros* for more information about the CTKL macro.

The link keypoint records are not retrieved by IPLP. Instead, program CIJH retrieves them on restart. They are filed by time-initiated keypoint copy (CVZA) and by update keypoint records (CVFD) like any other core-resident record. They are also filed on system cycle Down and on a catastrophic systems Error (unless a program check occurs in CPSE).

The current TPF system is being shipped with keypoints CKAA to CKAO. These keypoints must be modified, because they are system dependent. They are included with the package so that they may be used as examples. If the user's requirements are less than those of the package shipped and if the defaults of the parameters shipped are taken, assembly errors may arise in the keypoints shipped that are not being used.

Keypoint Record D

Keypoint record D, which is used solely by the synchronous link control package, consists of two tables, the link routing table and the link pool availability directory. The former contains information necessary to route type A messages to or from a high level communications network (that is, SITA-HLN) attached to the system by a synchronous link. Each entry within the LRT is comprised of a symbolic link number, the HEX/HEN of the remote HLN center to which the associated TI/system is attached, and a message identifier. In addition, a pseudo line number, a remote center TI identifier, an HLN polling status indicator, and an associated line code associated with the transmitted text are included. The link pool availability directory is used to control the allocation and usage of file addresses in a fixed file area defined by FACE. These file addresses are temporarily assigned to incoming message blocks for the purpose of saving each intermediate message block on file until the complete message is received. The table consists of a bit string, each bit of which equates to a file address. The relative position of the bit within the string is the FACE ordinal number; the record type is #LKIBR.

Input LCB Queue

An area of core is reserved below working storage to contain the input LCB queue. This area comprises an integral number of four byte slots which are used in a continuous cyclic manner. CTIN sets up, in the SLC common area, the address of this area as well as a number of other fields to enable BXH and BXLE instructions to be used by those programs adding items to or deleting items from the queue. There is only one input LCB queue for the system irrespective of the number of links or lines.

Output LCB Queues

Following the input LCB queue an area of core is reserved for output LCB queues. A queue exists for each link and each channel of each link in the same format as described previously for the input LCB queue. The link queue control fields are in the link keypoint while those for the channel queues are in the corresponding CCW areas.

Note: On a link comprising only one channel, only a link queue is maintained.

Output Message Queues

These are simply chains of working storage blocks in system virtual memory containing messages awaiting transmission. Separate chains exist for each link, with control fields in the link keypoint, and for each channel of each link with control fields in the corresponding CCW areas. Separate queues are maintained for type A and type B messages.

Note: On a link comprising only one channel, only the link queues are maintained.

Input Block Queues

ENQ LCBs and message blocks are passed to OPZO from the CCP. OPZO maintains a queue of such blocks, one queue per channel, and the blocks are removed and processed one by one by the input message handler. By this means, LCBs and message blocks can be processed in strict sequential order. The pointers to the top and bottom of the queues are maintained in the keypoints.

SLC Common Area (CX#EV)

The SLC common area, which resides in the SLC core-resident CCP, contains pointers and switches which are system-wide, regardless of the number of links. In summary, they consist of input LCB queue controls, indicators to show when certain programs are active, core level controls, CCP subroutine addresses, and translate table addresses.

System Error Routine

Because the ECB controlled Input Message Handler is sometimes manipulating core blocks without any references in the ECB core block Reference Words, CPSE has to be specially cognizant of it.

Also, when a catastrophic systems error occurs, it is necessary to gather message references for some messages on the output message queues and to file link and channel keypoints in the same manner as some other keypoints.

Input Message Processing

Input Message Handler

Input CCP adds all LCBs except for ENQ to the input LCB queue and activates link controller if it is not already active. ENQ LCBs and message blocks are added either to the input list or post interrupt (Ready) list dependent upon a system parameter. The branch vector is set to direct them to link OPZO, which adds them to the channel input block queue. Input message handler directs ENQs to link controller and assembles a complete message which is then passed to the SLC component of the communications source program. Link controller is entered to send ACK or NAK according to the state of the block received by input message handler. Blocks are checked for an ETB and if none is found a system error is generated. Link or channel conditions which prevent acceptance of message blocks are tested and, if necessary, the blocks are discarded. Blocks are then checked for parity or TSI sequence errors and the link controller is entered to generate and transmit a NAK LCB if either has occurred. Blocks of multi-block messages are filled with the file address of the previous block placed in it, that is, they are chained together in reverse order to that in which they were received. When the last block is received (that is, the message is complete), the constituent blocks are retrieved from file and reassembled in the correct order according to the MBI prior to entering communications source.

Communications Source Message Handler

SLC communications source translates the message text from line transmission code to EBCDIC according to the translate code that is indicated in the ACI or in the link keypoint and then reformats the text for type A or type B messages. Type B messages are placed in XM0RL format and passed to XIMA. Type A messages are placed in AM0SG format, and a routing control parameter list (RC0PL) is constructed and placed in the ECB. Type A traffic containing a pseudo line number is passed to the common component of the Communications Source Program. Type A traffic not containing a pseudo line number is passed to the message router, if it exists in the system, or a user defined application program.

Note: When a system is generated without a world trade or domestic message switching package, type B messages are passed to a user exit routine, CIM2. It is your responsibility to create a message switching function, starting with a segment entitled CIM2, and to inform SLC when it is to release the link pool file addresses assigned to the message. (See the source listing of CIM2 for the mechanism used to accomplish the release.)

Output Message Handler

Output queues are maintained both for the link and for each channel of the link (on multichannel links), for LCBs and message blocks. LCBs will be added by the link controller program. Type A and B messages may be passed by any non-terminal oriented application program by use of the macro SENDCK – Send to Link. Terminal oriented system programs (that is, reservations) may pass type A messages, through SENDCA, SENDCC or SLMTC, to send to indirectly coupled sets; if a SENDCA, SENDCC or SLMTC specifies a pseudo line, the macro routine will search the link routing table using the pseudo line number and Interchange Address (IA) as the search argument in order to find the additional routing characters required to send the message across the high level network. It then effects the processing of the SENDCK macro. Since system programs pass messages, but the output queues require specially formatted link blocks, an output interface is needed. This is effected as follows:

- SENDCK or SENDCA/SENDCC/SLMTC intercept will detach the first (or only) block of the message from the ECB to form an in-core applications message queue (AMQ) by type (A or B) and by link.
- Two output sender programs (one for each type) process the AMQs, assign MBIs and perform the necessary functions to convert messages into link blocks. The blocks are passed to the CCP output routines and placed on the output queues through a common deblock and send program that issues the macro PLONC (Place on Queue).
- The core-resident CCP output routine services the queues as channels become available in the priority sequence of LCBs, type A blocks, and type B blocks. Single block messages are transmitted down the first available channel. As blocks are sent, the next-in-sequence TSI is assigned and information is recorded in the channel keypoint in case a NAK (or implied NAK) is received.
- The link controller picks up information from the channel keypoint when processing NAK LCBs and passes it to the NAK processor to retransmit the affected blocks.

Message Output From Applications

The SENDCK macro and SENDCA/SENDCC/SLMTC Intercept messages to the appropriate applications message queue (AMQ). Information concerning the message Type, and the means of recreating NAK'd link blocks is set up by the SENDA,SENDCC and SLMTC intercept coding from its own internal logic, while that for SENDCK is provided by you. In all cases, this *status* is stored in the message before attaching it to the AMQ. Following the creation (or extension) of the AMQ, the macro then determines if the appropriate output sender (type A or B) is active and, if not, activates it by means of the Control Transfer mechanism. The normal macro exit logic then returns control to the original output program, (for example, message router, XMLT).

Output Sender (TYPE A) (CMS1)

This takes messages one by one from the AMQ, assigns an MBI and then enters the *deblock and send* segment in order to add the message to the output queue. The following conditions may arise:

- Link down – messages are dropped.

Since type A messages require short transit times and are normally associated with a conversational transaction, there is no requirement to hold on to messages when the link is down.

- MBI exhaustion, that is, all output MBIs in use. This should be only a transient condition and in order to give some short time to recover, the following technique is adopted.
  - The current top of AMQ is added to the bottom of AMQ and the new top of AMQ is processed. This allows single (link) block messages (MBI+1/15) to be transmitted.
  - When the whole AMQ has been processed, the output sender CRETs to itself with a link defined time interval of the order of seconds and exits.
  - If the MBI exhaustion condition is detected again by the CRET entry, it is assumed that a fault exists somewhere in the system. All messages on the AMQ for this link are dropped and an alarm message is sent to the supervisory position.

Output Sender (Type B) (CMS2, CMS5, CMS8)

In the normal case, messages are removed one by one from the AMQ, assigned an MBI and passed to *deblock and send* to add the message to the output queue. However, since type B messages have the lowest priority, they are transmitted only when others on the link allow. A means of queueing at an earlier stage than the output queue may, therefore, be required to avoid core depletion and is provided by the MCR – message control record (MC0MC). This contains the file address of the prime block of the message and status information. If the link is down, or the type B link queue already contains a number of link blocks exceeding a link defined maximum, or MBIs are exhausted, the message reference (file address) and status information is stored in the MCR for that link and the core block released. In addition to the type B output sender being activated when the SENDCK macro address messages to the AMQ and the sender is not *active*, it is also activated by the communications control program when a link is restarted, the link queue is depleted or MBI exhaustion is relieved providing that a *wake-up* switch is on and the *active* switch is off.

Deblock And Send (CMS3, CMS6, CMS9)

An output message in OMSG, AMSG, or XMSG format is converted into link blocks, translated and the envelope of TSI, MBI, LCI and so forth, inserted. The blocks are passed one by one to the communications control program by means of the macro PLONC – Place on Queue. NAK'd blocks are handled by deblocking according to the original logic, but issuing the PLONC macro only for those link blocks that require retransmission.

NAK Processor (CMS4, CMS7)

Based on the *status* byte that contains information about the origin of the message in the TPF system, it is first determined whether it is possible to regenerate the original text. If not, a standard text is transmitted instead. Otherwise, it may be possible to regenerate the text by retrieving from file the original output message, or by means of repeat transmission logic in UIO or XLMT. Having retrieved, by one of these means, the original message, Deblock and Send, is entered to retransmit the NAK'd blocks.

Message Retransmission Processor (CMSD)

When an AML time out has occurred and the message has been previously retransmitted a specified number of times (N3) with the current message label, an attempt is made to regenerate and retransmit the entire message as an original message. When reference to the message has been saved, the message is retrieved from file and then is passed to UIO, UIS, or LMT to effect *repeat message* logic or is passed directly to the output message handler to process. Either alternative effectively will cause a new message label to be obtained for this message. The *old* message label in the link keypoint is indicated as being available. If it is not possible to regenerate the original message, the *old* message label is made available and this segment exits with no further processing.

Link Controller

The link controller consists of the following routines:
- CCP Linkage Routine (CMCK)
- Input LCB and Time-Out Block Receiver (CMC0)
- Common Routines (CMC1)
- A set of incoming LCB Handlers (CMC2, CMCW, CMCX, CMCY, CMCZ)
- A set of incoming LCB Handlers (CMC2, CMCX, CMCY, CMCZ)
- Time-Out Handler (CMC3)
- Output LCB Generator (CMC4)
- Stop/Invalidate Lines/Links (CMC5)
- Start Lines/Lines (CMCA)
- CMS Restart (CMCB)
- CMS Restart (CMCC, CMCD).

General System Description

- Incoming LCBs except ENQ are added by input interrupt to the input LCB queue, which resides just below working storage. They are processed by the link controller that makes appropriate adjustments to the link and channel status, formatting any replies (outgoing LCB) that may be necessary. In addition to these responses, link controller handles channel out-of-service and recovery (for example, error conditions which do not invalidate a line or require operator intervention), and maintains message assurance by keeping track of all blocks sent until acknowledgment is received. Interfaces exist to output message handler to handle any retransmissions which may be necessary. See “NAK Processor (CMS4, CMS7)” on page 40 and “Message Retransmission Processor (CMSD)” on page 40.
- To enable the link controller to give the most up to date channel status, input ENQs are passed to link control from OPZO. Should this LCB have been processed in the normal manner (that is, as quickly as possible through the input LCB queue), then those message blocks queued to input message handler would have been missed by the response and thus caused unnecessary retransmission of data message blocks over the link (since an ENQ response is treated as a NAK).
- Every second CLQM monitors for time out conditions and TSI exhaustion and adds *pseudo LCB* blocks to the input LCB queue for handling by the link controller.
- Basically, a link channel will be in one of two states: functioning or non-functioning. This distinction is important, because if a STP LCB is received, for example, the channel will still be functioning (link controller will be sending idle line link control blocks); contrast this with the non-functioning state produced when a line is invalidated, or stopped by the operator, or the system is cycled below CRAS state.
- A functioning channel will at any given instant be performing one of four procedures as described below.

Procedures

Data Procedure	The channel is functioning normally and data blocks are being exchanged. The line time-out routine is monitoring the time since the last data block was received or transmitted to detect if the Receive or Send Idle Line Procedure should be initiated.
Idle Line Procedure	
Receive Idle	If no data has been received over a certain time, then Idle Line LCBs (ILB or ACK) are transmitted over the line. If responses are not received, then the line is suspect and the Enquiry Procedure is initiated. During the receive idle procedure, data blocks may still be transmitted.
Send Idle	The system also monitors for Send Idle, that is, no data blocks transmitted for at least T1 seconds. Should this occur, and there are outstanding unacknowledged blocks, and the bilateral agreement allows the exchange of ILB (rather than ACK) idle line link control blocks, then the Idle Line Procedure is initiated and ILBs transmitted with the sole function of eliciting from the remote center a response containing the TSI of their last correctly received block. Without the Send Idle procedure, if the remote center sends ACK only every *p* blocks and there are no blocks to send, we may be required to maintain references to messages sent until some time later when the remote center has accumulated *p* blocks.
NAK Procedure	This procedure is initiated when an invalid data message is received. The NAK is repeated a number of times at set intervals and if the error has not been corrected, then the enquiry procedure is initiated. A *discard until* switch in the channel keypoint is turned on, that causes input message handler to discard data message blocks until it correctly receives a block with correct TSI sequence.
Enquiry Procedure	This procedure is initiated when the status of a channel is unknown or suspect. It implies that there is (or was) a fault in transmit, receive, or both directions. ENQs are transmitted and should no response occur within a certain time period, after a fixed number of repetitions, the channel is declared *out-of-service* and the enquiry procedure continues. If a response is received then the status of the channel is known again and transmission can proceed normally (unless the response was a STP). Entry to the ENQ procedure is normally caused by transient line errors (repeated hardware errors will, of course, cause the line to be invalidated thus preventing any I/O operations). *Out-of-service* means a condition less serious than invalidation and does not involve operator intervention. The

receipt of STP ALL will also cause the enquiry procedure to be initiated for all channels of the link. Since the STP ALL contains no ATSI information this is done to elicit individual responses (presumably STP) on each channel containing ATSs appropriate to each channel.

CCP Linkage Routine (CMCK)

This routine is activated by the CCP using the control transfer mechanism (which is an implied ENTNC) when it requires to activate functions of the link controller, for example, an LCB or time-out pseudo LCB is added to the input LCB queue and link controller is inactive). It activates such functions using ENTRC and on return issues an EXITC.

Input LCB And Time-Out Block Receiver (CMC0)

This routine handles the system queue of input LCBs and time-out blocks. It determines the link and channel of the incoming item and then passes control to the input LCB or time-out handler.

Common Routines (CMC1)

Link and Channel Base Address Calculator

This routine is entered by many programs to calculate the address of the appropriate link and channel keypoint records for a given line.

Good/Bad Block Handler

When a data message block is correctly received an entry is made by input message handler. The output LCB generator is entered to generate the appropriate acknowledgment ACK). An entry is also made by input message handler for NAK to be sent, if a block with bad parity or out-of-sequence TSI is received.

Line Out-of-Service Routine (LOS)

This is entered when a line is declared *out-of-service* to set indicators and send an alarm message.

Link Alarm Routine

This routine sets up the necessary conditions required to send an alarm message through the alarm/message generator to a specified terminal. See “Alarm/Message Generator” on page 49 for more information.

Input LCB Handlers (CMC2, CMCW, CMCX, CMCY, CMCZ)

As each LCB is processed, the status of the link and channel is checked and any appropriate changes to the keypoints and responses are made.

ACK Handler

An ACK received implies that the remote processor has accepted responsibility for one or more data blocks we have sent and we no longer have to keep any reference to these blocks. Single block message references are dropped as soon as the ACK is received. For multi-block messages, the reference is held until all the blocks of the message have been acknowledged if no AML agreement exists or until an AML is received.

NAK Handler

Reception of a NAK implies (a) that all blocks transmitted up to the NAK ATSI have been correctly received and (b) all blocks transmitted after have been rejected. Action is taken as described in “ACK Handler” for all blocks correctly received. For those blocks not acknowledged parameters are generated indicating which need to be retransmitted and control is then passed to output message handler that will handle the regeneration and requeuing of the rejected blocks.

AML Handler

After a multi-block message has been transmitted and all of its constituent blocks acknowledged an AML may be expected to indicate that the message label (MBI) may be reused and the message reference dropped.

ENQ Handler

Reception of an ENQ means that the remote processor is unsure of the state of the channel. The ATSI of the ENQ provides both an implied ACK (n) and an implied NAK (n+). After the ATSI has been acted upon the appropriate response is made depending upon the state of the link. A STP (and possibly an ACK) is returned on the same channel if we are cycling down or, due to a high input list condition, wish the remote processor to stop sending data. A RSM (and possibly an ACK) is returned in the case where the channel had been stopped and resumed by us but no data has yet been received. Otherwise a RSM or ACK is returned to indicate that our system is prepared to continue receiving data.

RSM Handler

Reception of a RSM indicates that the channel is serviceable. If a STP had previously been received or if the channel had been declared *out-of-service*, then it is restarted automatically. If the RSM is not in response to an ENQ, then it is treated as simply an implied ACK. If it was a response to an ENQ then it is also an implied NAK.

STP Handler

A STP indicates the status of a channel and invokes both implied ACK and implied NAK. Since no more messages are to be sent down the affected channel, any blocks on the channel queue are added to the link queue. If STP ALL is received, since it contains no ATSI the enquiry procedure is initiated to elicit individual STPs for each channel.

ILB Handler

ILBs are used to maintain circuit continuity in *no data* conditions. The ATSI of the ILB is treated as an *implied ACK* and the appropriate response is generated as described under ENQ Handler.

Time-Out Handlers

Every second CLQM interrogates the link and channel keypoints to see if a Time Out or TSI exhaustion situation has occurred. If so, a *pseudo LCB* is added to the Input LCB queue and subsequently passed by the input LCB and time-out block receiver to the time-out block receiver to the time-out handler for processing. The action taken depends on the procedure-in-progress as defined in the channel keypoint. When a fixed number of ENQs have timed out, the line is declared *out-of-service* and ENQs are repeated regularly until a response is received. In addition at each alternate ENQ a STP for that channel is sent down any other functioning channel. An ILB time out or a fixed number of NAK time outs means a possible impairment of the channel and the enquiry procedure is initiated. A data Time Out means that the line is idle and the idle procedure is initiated. The time-out handler also processes TSI exhaustion (a link defined of blocks have been transmitted without an ACK or implied ACK being received) which causes the enquiry procedure to be initiated. It also processes AML time outs, that is, all blocks of a multi-block message have ACK'd, but no AML has been received. The message is retransmitted in its entirety up to N3 times with the current message label and then retransmitted as an original message with a new message label.

This process continues until an AML is received. When no N3 agreement exists on the link, an AML is simulated in order to activate the appropriate action.

Output LCB Generator (CMC4)

This is a set of routines to generate any LCBs required. The created LCBs are then passed to the CCP output routines and placed on an output queue using the PLONC macro. Those LCBs (for example, STP ALL) which may be transmitted over any channel of a link are placed on the link queue whereas those that are channel dependent (for example, ACK) are placed on the appropriate channel queue. CCP Output Initiate will transmit channel-queued LCBs on the appropriate channel and link-queued LCBs on the first available channel.

Stop/Invalidate Lines/Links (CMC5)

This segment, which contains transfer vectors CMC6, CMC7 and CMC8, consists of routines to stop the exchange of information on any or all the lines of a link, for any or all links, and is activated as a result of commands or by the CCP in the case of line hardware errors. This involves the transmission of STP and possibly ENQ LCBs in order to determine the status of message blocks already exchanged, the possible retransmission of message blocks down other functioning channels and in the case of stopping or invalidating the last functioning channel of a link, that is, link cycle down, the filing of all single block messages on the type B output message queues and the clearing of all output queues. Since type B single block messages have no reference by which they may be retrieved when the link is cycled up they are chained together and the address of the chain saved in the link keypoint before filing it.

Start Lines/Links (CMCA)

This handles not only the starting of a line (which is effected simply by transmitting an ENQ and allowing the response to be handled in a normal manner) but also the starting of a link in three different conditions:

1. Link Cycled Down

   This is a normal situation in which the link was previously cycled down and is now being cycled up.

2. Unplanned Restart

   This occurs when a catastrophic systems error had occurred whilst a link was operating and we are attempting to restore the system into its working condition.

CMR Restart (CMCC)

A special fixed-file record area (FACE type #LKIBR) is maintained for use by Input Message Handler to file all Type B input message blocks and all except the last received block of multi-block Type A messages. A bit matrix is maintained in Keypoint D which reflects the use of these file addresses. CMR Restart, which is activated at system cycle above Utility state, checks the bit matrix and ensures that any messages which had been completely received across the link but were lost at the time the system went down are reentered into the system for processing.

Core-Resident CCP

The synchronous link control section of the core-resident CCP and receipt of information on a line, the control of LCB and message queues, the activating of the input LCB and message processing programs and the monitoring of time out situations.

The following members contain logic entirely associated with control of Synchronous Links.

CLQA SLC Common Area – see “SLC Common Area (CX#EV)” on page 38.

Also, it contains keypoint D and translate tables used by SLC communications source and the output message handler.

CLQC Link OPZO

When a data message block or ENQ is taken from either the input or ready (PI) list by the CPU loop, it is passed to this routine by means of the branch vector stored in the block. OPZO adds the blocks to the channel’s input block queue and if input message handler is not already active, activates it.

CLQE This contains two routines:

1. Add to Input LCB Queue
2. Get Core and Initialize READ CCWs

These routines are used by other members. CLQG uses function (1) to add actual LCBs whilst CLQM uses it to add “pseudo LCBs” reflecting time out or TSI exhaustion situation. Function (2) is used by CLQG as part of its initialization logic.

CLQG Input Initiate, Interrupt and Error Interrupt Routines

These routines set up and control the channel command words (CCW) associated with reading information, and handle the queueing of input LCBs and data message blocks. Input LCBs, except for ENQ, are added to the Input LCB queue, and ENQs and all data message blocks are added either to the input or ready (PI) list, according to the setting of a system parameter (in CLQA).

CLQI Output Initiate, Interrupt and Error Interrupt Routines

These routines set up and control the channel command words associated with writing information and handle the queueing and dequeueing of output LCB and message blocks (PLONC macro support). Output initiate includes the assignment of TSIs.

CLQK SENDCK and SENDCA/SENDCC/SLMTC Intercept

Contains the code for handling application messages destined for transmission across a link, adding them to an AMQ and, if necessary, activating the appropriate output sender.

CLQM Contains code to check various conditions on a regular one second basis. In particular it monitors each line and link for time-out conditions or TSI exhaustion adding “pseudo LCBs” to the Input LCB Queue if any arise.

The following modules have been modified, to a greater or lesser extent, to cater to the link.

CCIT	Modified by CPSE when dumping to allow only the clock update switch to be set (in the CPU loop) when an external interrupt, caused by the timer, occurs.
CLYM	Contains a linkage to the synchronous link line invalidation program.
CICR	Additions to the CINF Table.
CLVU	Additional branch addresses for link input and output core-resident CCP routines (that is, CLQG, CLQI).

CLXA/CLXM	The macro routing routine recognizes and routes SENDCA/SENDCC/SLMTC messages with pseudo line numbers to the send intercept code in CLQK. It also contains support for SENDCK.
CLVY	Contains code concerned with maintaining control unit error counts. Also code to direct an SWB (due to control unit error) to the synchronous link line invalidation program.
CLYE	Contains code to cater for a *hung PCIL* condition on a line.
CLYK	Contains code to route the SWB for line errors to the appropriate error routines in CLQG/CLQI.
CLVK	Contains code in the CIOUC macro support.
CPSE/CPSF	Contains logic necessary to save as much relevant information about links as possible in the event of a catastrophic systems error. Also needs to be aware of the input message handler in the case of a program error since there may be core block references embedded in the ECB work area.
CTIN	Contains the code to set up necessary tables and table addresses. For example, input and output LCB queues.

File Resident CCP

File resident communications control program segments perform the following functions:

- Restart Initialization
- Validate a line or link
- Start one or more lines or links
- Stop one or more lines or links
- Display the status of one or more lines or links
- Invalidate a line or link
- Initialize, update and display control unit and line error counts
- Alter and display CCP restart/shutdown levels
- Alarm/message generator
- Start/stop and display pseudo lines
- Cycle up/down links.

Restart Initialization

Activated by the restart scheduler CTKS to initialize the poll status table with pointers to the channel keypoints and set pointers to the link keypoints in the CCWA for each line of the link.

Restart/Line Validation

This program may be activated in three ways:

- By the restart schedule through CML1 to initialize the internal line number table and to complete the initialization of the CCWA for any lines that had previously been valid. In addition, if SLC has been implemented using a 3705 Communications Controller, enable commands are also issued to those lines.
- As a result of the following command entries:
  ```
  ZLVAL AI XX
  ZLVAL AI LK YY
  ZLVAL AI ALL.
  ```

Where:

AI indicates the type of line as belonging to the synchronous link category,

LK
indicates that the function requested applies to all lines of the link,

XX
indicates the symbolic line number of a line

YY
the symbolic line number of the first line of a link.

- This routine is activated by 03-RFIG to facilitate the assignment or replacement of individual AI lines, or communications control units that include AI lines.

Start Lines/Links

Activated by the following command entries:
- ZLSTA AI XX
- ZLSTA AI LK YY
- ZLSTA AI ALL
- ZLSTA ALL.

Its purpose is to start a single line, all lines of a link, or all lines. It sets the line as started in the line status table and activates the link controller through CREMC informing it which lines are to be started.

In addition, this routine is also activated internally from either the cycle schedule in response to a ZCYCL command to restart any AI lines that had previously been started, or from 03-LRFIG to facilitate the assignment or replacement of individual AI lines, or communications control units that include AI lines.

Stop Lines/Links/Cycle Down Links

Activated by the command entries:
- ZLSTP AI XX
- ZLSTP AI LK YY
- ZLSTP AI ALL.

Its purpose is to stop a single line, all lines of a link or all lines. It is also entered by the cycle schedules on system cycle down. This program enters the link controller to stop activity on the appropriate channels. On return, when I/O activity has ceased, it updates the line status table and releases any core blocks attached to the CCWAs.

This routine is also activated internally by 03-LRFIG to facilitate the idling or assignment of individual AI lines, or communications control units that include AI lines.

Display Link Status

Link and line status is stored in the link and channel keypoint records and line status is also contained in the line status table.

The command entries:
- ZLDLS AI XX
- ZLDLS AI LK YY
- ZLDLS AI ALL.

cause selected status indicator bytes to be displayed for a single line, all lines of a link or all lines.

Invalidate Line

This program is entered to turn down a line on which an uncorrectable I/O error has occurred. It will enter the link controller to stop the affected line. On return, it will set the line as stopped in the line status table and will housekeep all associated table entries and controls, so that on restart they are in an initial condition.

This routine is also activated internally by 03-LRFIG to facilitate the idling or assignment of individual AI lines, or communications control units that include AI lines.

Line And Link Error Counts

Line and link error counts are accumulated and stored in file resident records.

Initialize Error Counts

This program is entered by the command entry:

- ZLAEC ALL

to initialize the file resident records setting all error counts to zero.

Update Error Counts

This program is entered at regular intervals by the communications timer program to update the file resident records with the line error counts from the CCW areas and the link error counts from the channel keypoint records.

Display Line and Link Error Counts

This program is entered by the command entries:

- ZLDLE AI XX
- ZLDLE AI LK YY
- ZLDLE AI ALL

to display the line and link counts for a single line, all lines of a link or all lines.

Display Or Alter Restart/Shutdown Level

The CCP has the ability to inhibit the receipt of data message blocks on all links based on the size of the input list. This program is entered by the commands:

- ZLDCL
- ZLACL AI

to display or alter the control levels.

Alarm/Message Generator

This program handles the generation and transmission of alarm and error messages and is activated as a result of the Link Alarm macro, KARMA or by direct enter (ENTNC/ENTRC).

Start/Stop And Display Pseudo Lines

This program is entered by the command entries:

- ZLSTA PS
- ZLSTP PS
- ZLDLS PS

to start, stop or display the polling status of the high speed lines controlled by satellite processors attached to a SITA high level network exit center. It is also activated when a link is restarted to restore the polling status automatically.

SLC SITA P1024 Test Driver and Monitor

These programs are designed to perform those test functions described in the SITA P1024 Test Guide (P.Z.1885.3). They also provide facilities for monitoring data on the SLC link and to display the statistics of input and output data.

Two commands are provided for initiating the two functions implemented in CMD:

ZLTST To perform SITA P1024 tests.

Command format:

```
ZTLST  OMM(/.../X(YY)
       FMM
       DSS
       ANN
       BNN
       W
       E
```

Where:

O On Command

F Off Command

D Defer Command

A Generate Type-A Series NN

B Generate Type-B Series NN

W Wait Until Good Data Message Received

E Terminate Wait

MM
1–2 Digit For Command Number

NN
1–2 Digit For Test Series Number

SS
1–2 Digits For Time In Seconds

X(YY) Optional Parameters

Where:

OM
To Turn On Monitoring On Printer

FM
To Turn Off Monitoring On Printer

ZLKST To monitor data or to display statistics

Command format:

```
ZLKST  AI NN CL
       AI LK NN CL
```

Where:

NN
Symbolic Line Number

CL
Optional Parameter: To Reset Counts To Zero

Data Collection/Reduction

The collection and reporting of the status of various link control queues as well as the number of messages coming in through the link is done by the online message collector. This data is then reduced offline by the message reduction program. See *TPF System Performance and Measurement Reference* for more information.

Recoup

Recoup descriptor records BKDF through BKDT and BKDX are used by the Recoup package to control the chain of pool records that have references in the link and channel keypoint records.

Index

File Number: S370/30XX-30
Program Number: 5748-T14

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH31-0161-02