MDH

MDH TRANSMISSION USER GUIDES

20.07 MAINFRAME DUAL HOST (MDH) SYSTEM USER GUIDE FOR COD, W/T, AND DEPOSIT OUTPUT NOTIFICATIONS

Version 1

The Depository Trust Company May 2001

The Depository Trust Company
55 Water Street
New York, NY 10041

Last Published: January 21 2001

PREFACE

This document describes, specifically, the input and output requirements for receiving COD, W/T, and/or Deposit activity via DTC's Mainframe Dual Host (MDH) system.

A complete description of the MDH System and the functions it supports are contained in *The MDH. User Guide - Version 4*.

TABLE OF CONTENTS

I. OVERVIEW

Criteria for Developing MDH

The Mainframe Dual Host (MDH) system is designed to converse with those participants with mainframes that can support real-time (for example, CICS) processing via the LU6.2 communications protocol. This protocol is described in the next section of this document.

The MDH system provides the following advantages over older systems at DTC:

- Allows two-way traffic activity between DTC and the participant in a real-time environment.
- Eliminates the need for intermediate hardware/software.
- Provides backup through redundant mainframes and lines.

MDH will control the data flow between the DTC host computer and the participant host computer via a dedicated point-to-point communication, that is, telephone, line. The procedure with which the participant can either request data to be transmitted to DTC or request data to be transmitted to him from DTC is described in other sections of this document.

II. DTC-TO-PARTICIPANT DATA TRANSMISSION

A. General

This section describes in detail the procedure that the participant uses to receive data transmission blocks from MDH.

The types of data available are:

- COD Notification - Type '020'
- Deposit Notification - Type '030'
- W/T Notification - Types '080' and '081'

To review the sequence of transmissions required to receive data, the participant's host will:

- Sign on to MDH by transmitting a Type '01' logon block.
- Request the MDLS function by transmitting a Type '03' function-request block.
- Transmit a Type '07' transmission-request block.
- Receive one or more Type '08' data blocks until all the available data has been transmitted.

B. Receiving Data From DTC

Type '08' Transmission Block Format

The Type '08' transmission block format is shown in the diagram below:

BLOCK PREFIX (66)	BLOCK DATA LENGTH (4)	TRANS #1 LENGTH (4)	TRANS #1 DATA COD(376) DEP(376) W/T(376)	TRANS #2 LENGTH (4)	TRANS #2 DATA COD(376) DEP(376) W/T(376)	TRANS # N (UP TO 4085 TOTAL CHARACTERS)

|←MINIMUM 450→|

|←MAXIMUM 4085→|

Note: This data can be interspersed with other types of Non-COD/WT/Deposit activity.

Each block contains a 66-byte prefix followed by one to 10 transactions in any combination of the data record types described above. The minimum block length is 450 bytes and the maximum is 4085 bytes.

Transmitting the Request Block

The participant will transmit a Type '07' block containing the following fields:

- **Block Type** must be '07'
- **Time Stamp** is provided by MDH and used for cutoff-time checking.
- **Participant Signon ID** must be a valid eight-character field.
- **Individual user Number** - This two-digit field is provided by MDH in the Type '02' logon response block. It must be inserted in the prefix of every transmission block sent to MDH to uniquely identify the transmission.

- **Function Requested** must be 'MDLS'
- **Request Code** (should be 'AD' for 'all data').
- **File Control Number** (YYYYDDDs). (Julian date + session).
- **Starting Sequence Number** Desired ('nnnnnn').
- **Total Number Desired** ('nnnnnn' - optional).
  - Note that the 'starting seq. #' and 'total #' parameters are six-character numeric fields, which must be right aligned and left zero filled.

Note: See Exhibit 3 for this block's format

Receiving the Response Block

The participant host will receive one of the three following responses as a result of the Type '07' request.

1. A 'No Data Available' condition. The Type '08' block will contain:

- A 70-byte block prefix.
- A transaction length attribute with a value of 8.
- A message 'NONE' in the data portion of the transaction, indicating that there was no data available for the requested range. (See Exhibit 4).

  The participant can now submit a Type '03' function-request block or a Type '90' signoff block.

2. A 'Data Sent' condition. The Type '08' block will contain:

- A 70-byte block prefix indicating the number of transactions sent in the block.
- One or more occurrences of intermixed transactions in the sequence that they were chronologically processed at DTC.

  Note: The first block will be followed by additional '08' blocks until either no more data is available or the desired range has been satisfied.

3. A 'No More Data' condition. The Type '08' block will contain:

- A 70-byte block prefix.
- A transaction length attribute with a value of 8.
- A message 'END ' in the data portion of the transaction indicating that no more data is available or that the desired range has been satisfied. (See Exhibit 4).
- The participant can now submit a Type '03' function-request block or a Type '90' sign-off block.

Notes:

- The transaction length attributes shown above always contain the length of the data that follows plus four for the length of the length attribute field itself.
- The use of the 'Starting Seq. #' and 'Total #' parameters allows the end-user to receive the same data more than once if he so requests, similar to the 'RPNT' function on PTS. It does not, however, mark this data as being sent as an 'original transmission'
- As discussed earlier, if a system problem occurs at DTC and is caused, for example, by a program ABEND, unavailability of files or tables, or other reasons, MDH will send a Type '99' response transmission block instead of the Type '08' block currently being transmitted. When the problem is resolved at DTC, the participant will be informed and should attempt to reestablish the session in the normal manner.

III. EXHIBITS

Exhibit 1 - Logon Request

Logon Request Block from Participant - Length 68 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Numeric - Value is '01'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
PASSWORD	61	08	DTC-assigned User Password

Figure 1. Logon Request Block from participant

Logon Response Block From MDH - Length 142 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Numeric - Value is '02'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Returned by MDH. Must be copied by participant into all blocks sent to MDH.
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
RESPONSE-CODE	61	01	Values: 'A': Logon accepted 'R': Logon rejected
RESPONSE-REASON-CODE	62	01	Code indicating reason for rejection: 'B': Invalid block type 'C': Invalid connection I.D. 'D': Already logged on 'H': PTS is in 'Halt' mode 'P': PTS is down 'Q': DBF Recovery down 'S': Invalid signon ID 'X': Invalid Password
RESPONSE-ERROR-MESSAGE	63	80	Message explaining why the Logon was rejected

Figure 2. Logon Response Block from MDH

Exhibit 2 - Function Request

Function Request Block from Participant - Length 65 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Numeric - Value is '03'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Copied from Type '02' logon response.
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
FUNCTION-REQUESTED	61	04	Possible values: 'MDLS': DO/PO output to participant
Filler	65	01	Value space

Figure 3. Function Request Block from participant

Function Response Block from MDH - Length 146 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Numeric - Value is '04'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
FUNCTION-REQUESTED	61	04	Possible values: 'MDLS': DO/PO output to participant
RESPONSE-CODE	65	01	Values: 'A': Function request accepted 'R': Function request rejected
RESPONSE-REASON-CODE	66	01	Code indicating reason for rejection: 'A': Not signed on 'B': Past cutoff time 'C': Function does not exist 'D': User not eligible for function 'E': Function quiescing 'F': Function mismatch for block type '05' 'G': Function not for LU6.2 (MDH) 'H': Previous function not completed 'P': PTS is down 'Q': Recovery not available
RESPOHSE-ERROR-MESSAGE	67	80	Message explaining why Response Code is 'R'

Figure 4. Function Response Block from MDH

Exhibit 3 - Data Request Block (for MDLS Function)

Data Request Block from Participant - Length 86 Bytes

Field Name	**Pos**	**Len**	**Field Attributes**
TYPE-OF-BLOCK	01	02	Value '07'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Entered by Sender from Type '02' logon response
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
FUNCTION-REQUESTED	61	04	Value 'MDLS'
REQUEST-TYPE	65	02	Value of 'AD', 'OF' or 'OD'
FILE-CONTROL-NUMBER	67	08	Format : YYYYDDDS
STARTING-SEQ-#	75	06	Numeric 'starting' sequence number desired
MAXIMUM-NUM-REQUESTED	81	06	Numeric number of transactions desired

Figure 5. Data Request Block from participant

Exhibit 4 - Data Response Block from MDH with 'END' or 'NONE'

Data Response Block from MDH - Length 78 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Value '08'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
INDIVIDUAL-USER-NUMBER	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	30	Value spaces
FILE-CONTROL-NUMBER	53	08	Format : YYYYDDDS
RESPONSE-CODE	61	01	'A' or 'R'
RESPONSE-REASON-CODE	62	01	'A': Not signed on 'B': Past cutoff 'C': Not in 'MDLS' function 'D': Invalid range request 'E': Function incorrect 'F': Invalid Request-Type (MDLS) 'G': Wrong File-Control-# (MDLS) 'M': Message Delivery is down 'N': File-Ctl# vs Request-Type is invalid (MDLS) 'P': PTS is down
TRANSACTION5-IN-BLOCK	63	04	Number of transactions in this block (Value = 0)
BLOCK-DATA-LENGTH	67	04	Length of the data following this field plus 4 Value 12
TRANSACTION-LENGTH	71	04	Length of the transaction following this field plus 4 Value 8
REQUEST-END-MESSAGE	75	04	Values: 'END ' = All data requested has been sent 'NONE' = No data has been found for this request

Figure 6. Data Response Block from MDH

Exhibit 5 - Data Response. Block for COD, Deposit, and W/T Notifications

The '08' Response Block is variable length and consists of a 70-byte Block Prefix followed by up to 10 transactions, each preceded by a 4-byte length attribute. The exhibit below shows a block that contains a single COD, Deposit, or W/T transaction. It consists of the 70-byte prefix, the 4-byte length attributer and 376 bytes of the transaction.

Data Response Block For COD, Deposit, and W/T Notifications from MDH

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Value '08'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
Individual-User-Number	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	30	Value spaces
FILE-CONTROL-NUMBER	53	08	Format : YYYYDDDS
RESPONSE-CODE	61	01	'A': Accepted 'R': Rejected
RESPONSE-REASON-CODE	62	01	'A': Not signed on 'B': Past cutoff 'C': Not in 'MDLS' function 'D': Invalid range request 'E': Function incorrect 'F': Invalid Request-Type (MDLS) 'G': Wrong File-Control-# (MDLS) 'M': Message Delivery is down 'N': File-Ctl # vs Req-Type inval 'P': PTS is down
TRANSACTIONS-IN-BLOCK	63	04	Number of transactions in this block
BLOCK-DATA-LENGTH	67	04	Length of the data following this field plus 4
TRANSACTION-LENGTH	71	04	Length of the transaction following this field plus 4 Value = '380'
Filler	75	02	For DTC internal use only
Filler	77	01	Value space
DEST-PARTIC-ACCOUNT	78	08	Individual participant # or Group user #
DEST-SYMBOL	86	02	Destination Symbol – Numeric
Filler	88	01	Value '–'
DEST-ACCOUNT-SEQ-#	89	06	Sequence # of the transaction – unique for each account destination
RECORD-TYPE	95	01	Value 'M'
ORIGIN-CODE	96	01	DTC System origination Values: 2 = CCF

Field Name	Pos	Len	Field Attributes
			3 = PTS 4 = ID 5 = MDH 6 = AWT 7 = IVP 8 = SAC 9 = STS 0 = Other
Filler	97	01	Value space
ACTIVITY-CODE	98	03	Values: '020' = COD '030' = Deposit '080' = W/T '081' = W/T Adjustment
Filler	101	01	Value space
PARTICIPANT-#	102	08	Participant number for the affected account
Filler	110	02	Value spaces
CUSIP-#	112	09	CUSIP number
Filler	121	01	Value space
SHARE-OTY	122	09	Share quantity
Filler	131	02	Value spaces
ACCOUNT-TYPE	133	03	Account type (value: 010)
ACTION-CODE	136	01	Values: 1 = Add (for Deposit only) 2 = Subtract (for COD and W/T only)
Filler	137	01	Value space
MDH-TIME-STAMP	138	06	Time processed by MDH System (HHMMSS)
Filler	144	01	Value space
DATE	145	08	Current business date (CCMMDDYY)
Filler	153	01	Value space
DAY/NITE-IND	154	01	Values: 'D' = Day 'N' = Night
ATP-TIME-STAMP	155	06	Time processed by DTC (HHMMSS)
Filler	161	01	Value space
TAX-CODE	162	01	Tax code (for COD only)
CNS-CODE	163	01	CNS code (not currently available)
FAST-INDIC	164	01	Values: 'F' = FAST 'V' = Non-FAST

Field Name	Pos	Len	Field Attributes
BOND-INDIC	165	01	Values: ' ' = Not Bond 'B' = Bond 'M' = Muni-Bond
Filler	166	01	Value space
SDFS-INDIC	167	01	Values: ' '= Not SDFS 'S' = SDFS 'C' = SDFS-CP (SubI=525) 'M' = SDFS-MTN (SubI=530)
Filler	168	01	Value space
JOURNAL-CODE	169	01	Values: 0 = Not Journal 1 = Journal
Filler	170	10	Value spaces
ATP-RBN-#	180	08	Internal # on DTC Database
Filler	188	01	Value space
COMMENTS	189	80	Comments
Filler	269	01	Value spaces
SHARE-QUANTITY-NEW	270	09	Share quantity (new format where 1 =1 for all issues)
Filler	279	02	Value spaces
SUB-ISSUE-TYPE	281	03	Sub-Issue type for SDFS Cusips. Values: '120' = PFD-CP '320' = MUNI VRDO/CP '330' = MUNI CP '502' = Bas '504' = CDs '510' = IAM CP '520' = DISCOUNT CP '525' = CORPORATE CP '526' = CORP. VRDO/CP '530' = MTNs '531' = Deposit Notes '532' = M/T Bank Notes '550' = S/T Bank Notes '560' = DISCOUNT NOTES
Filler	284	167	Value spaces

Figure 7. Logoff Request Block from participant

Exhibit 6 - Logoff Request Block

Logoff Request Block From Participant - Length 60 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Value '90'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g., G0000123)
Individual-User-Number	17	02	Entered by sender from Type '02' logon response
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces

Figure 8. Logoff Request Block from participant

Logoff Response Block From MDH - Length 142 Bytes

Field Name	Pos	Len	Field Attributes
TYPE-OF-BLOCK	01	02	Value '91'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
Individual-User-Number	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
Response-Code	61	01	Values: 'A' = Logoff accepted 'R' = Logoff rejected
Response-Error-Code	62	01	Code indicating reason for rejection Values: 'A' = Not Logged on 'B' = Wrong Signon-ID 'P' = PTS is down
Error-Message	63	80	Error message if logoff has been rejected

Figure 9. Logoff Response Block from MDH

Exhibit 7 - System Error Block from MDH

System Error Block From MDH - Length 145 Bytes

Field Name	**Fee**	**Len**	**Field Attributes**
TYPE-OF-BLOCK	01	02	Value '99'
TIME-STAMP	03	06	Time received (HHMMSS)
USER-ID	09	08	Numeric for individual user (e.g. 00000161); Alphanumeric for group user (e.g. G0000123)
Individual-User-Number	17	02	Internal to MDH
LU6.2-TERMID	19	04	Internal to MDH
Filler	23	38	Value spaces
Function	61	04	Function in Progress at time of error
Error-Code	65	01	This error-code field is currently not used, but will eventually contain the CICS ABEND code at the time of system failure.
Error-Message	66	80	System error message

Figure 10. System Error Block from MDH

Exhibit 8 - Participant Request/MDH Response

Participant Request/MDH Response - Chart

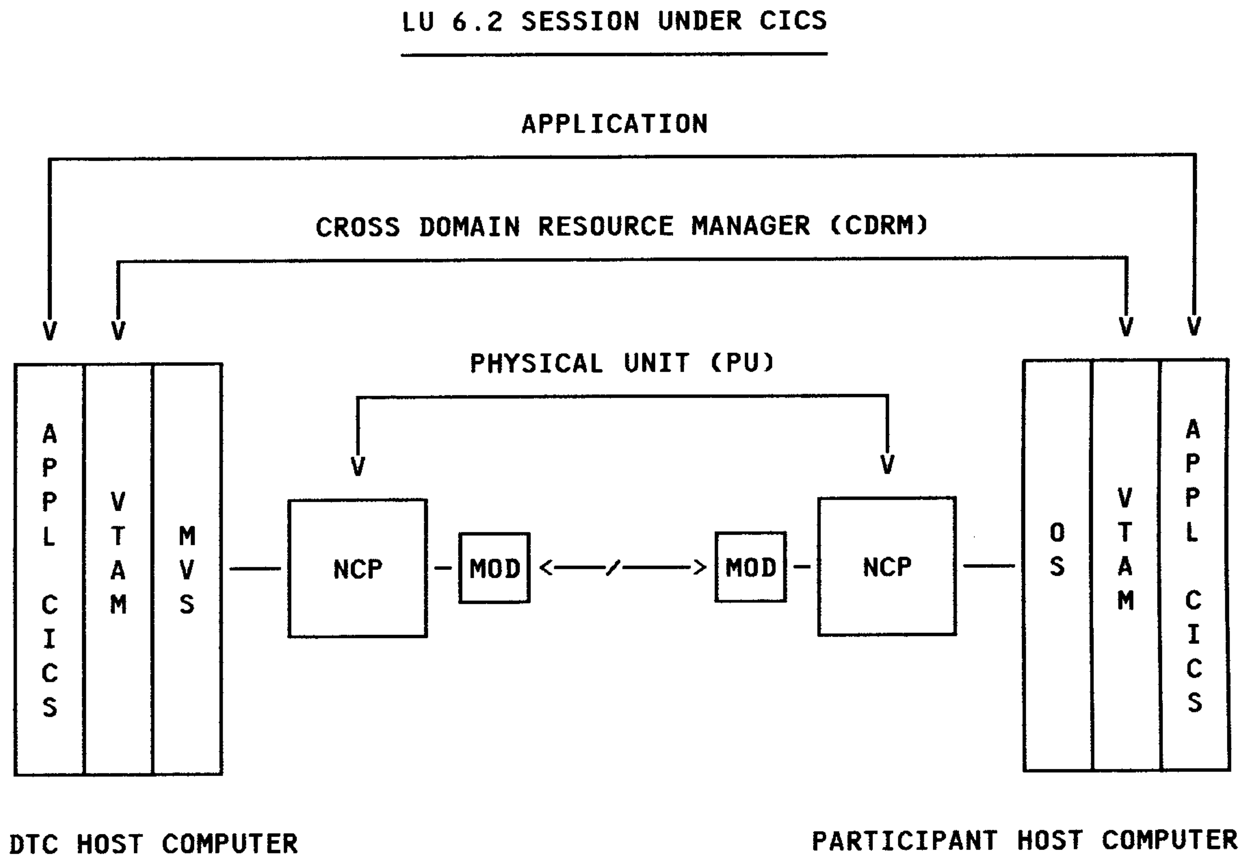
PARTICIPANT REOUEST		**MDH RESPONSE**	
Block type	**Description**	**Block type**	**Description**
'01'	Logon	'02' '06' '99'	Logon (accepted/rejected) - OR - Status of last good transmission ('recovery' after abnormal session termination) - OR - Logon rejected (System error)
'03'	Function	'04' '99'	Function (accepted/rejected) - OR - Function rejected (System error)
'05'	Data to DTC (Or 'Change of Function')	'06' '99'	Status of Data Block - OR - Data rejected (System error)
'07'	Data from DTC (via 'Range Request' or 'ALL')	'08' '99'	Data transmission - OR - Data rejected (System error)
'90'	Logoff	'91' '99'	Logoff (accepted/ rejected) - OR - Logoff rejected (System error)
NONE	'Time-out' (automatic)	NONE	Session terminated via LU6.2-to-LU6.2 system protocol message

Figure 11. Participant Request/MDH Response - Chart

IV. MDH TECHNICAL DOCUMENTATION

This section describes the Communications end Systems Programming Requirements for participants that wish to use the Mainframe Dual Host (MDH) System. The test and production environments at DTC are described along with guidelines for the environment at the participant's location.

The diagram below represents the 'layers' of communications that comprise an LU6.2 session:



A. General Communications Definitions

Listed below are the characteristics of the DTC test and production systems. For a participant to use MDH, its system must have matching characteristics at the same or a higher level and this information will be exchanged with DTC as soon as it is known:

No	Feature	DTC Test Frame	DTC Prod Frame
1	VTAM Level	CSV2R8	CSV2R8
2	NCP Level	V7 R5	V7 R5
3	NETID	DTCT	DTCT
4	NULL NETID **		
5	GWNCP	Yes	Yes
6	MAXSUBA	31	31
7	NULL NET MAXSUBA **		
8	HOST SUBAREA (NON GATEWAY)	Not applicable	Not applicable
9	NCP SUBAREA (NON GATEWAY)	Not applicable	Not applicable
10	NULL NET NCPSUBAREA **		
11	CDRM NAME	DTCT02	DTCP03
12	CDRM ADDRESS, ELEMENT (GATEWAY)	(,2)	(,2)
13	SSCPID	1025	1027
14	Transmission Group for Line	1	1
15	ERS, VRS	0,0 and 1,1	0,0 and 1,1
16	MAX RU SIZE	3840	3840
17	Application Name **	UTOR	PLCICS
18	Line Name **		
19	Line Station Name **		

Figure 12. Communication Requirements

** To be determined at time of installation by agreement between DTC and participant.

Note: The production link will run at 9600 Baud Full Duplex

Note: Items 8 and 9 do not apply if participant's installation is Gateway capable.

B. Controller 'Sysgen' Definitions

The following parameters must be included in 3705 or 3725 gens:

1. For Both 3705 and 3725 Group or Line Macros:

- NRZI= No
- NEWSYNC = No
- DUPLEX = Full

2. For 3725, code the Line Address as follows:

- Address = (XXX,Full) for Full Duplex
- Address = (XXX,Half) for Half Duplex

Note: IBM Informational APAR II01803 is very useful for Link Station definitions.

C. VTAM Requirements

1. Mode Table Definitions

The required Mode Table entry for use with LU6.2 is shown below:

MODELU62 TITLE MVS/XA SYSTEM MODE TABLE FOR LU 6.2 USE

MODULE NAME = MODELU62

```
MODELU62    MODETAB
SNASVCMG    MODEENT     LOGMODE=SNASVCMG
SNASVCUS    MODEENT     LOGMODE=SNASVCUS
            MODEEND
            END
```

2. VTAM CICS Application Definition

```
*
      XXX     APPL        EAS=160,                    ESTIMATED CONCURRENT SESSIONS
                          ACBNAME=XXX,                APPLID FOR ACB
                          SONSCIP=YES,
                          VPACING=3
                          MODETAB=MODELU62,
                          PARSESS=YES,
                          AUTH=(ACQ,VPACE,PASS)
*
```

D. CICS Requirements (General)

The participant's system must have the following:

1. At least CICS Release 1.6.1.

2. At least a PUT Level 8601 (with Release 1.6.1).

3. A TCT entry defining the CICS/LU6.2 line as suggested below. This entry defines the link for the LU6.2 Communications Facility and will be allocated by the LU6.2 Participant Application Region.

```
LU62    DFHTCT    TYPE=SYSTEM,          DEFINE IRC
                  ACCMETH=VTAM,         USE VTAM
                  TRMTYPE=LUTYPE62,     LOGICAL UNIT 6.2
                  FEATURE=SINGLE,       SINGLE SESSION
                  SYSIDNT=LU62,         NAME OF THIS LINK (ANY NAME)
                  NETNAME=PLCICS,       APPLID OF REMOTE SYSTEM AT DTC –
                                        (NOTE: USE TQCICS FOR TESTING)
                  MODENAME=SNASVCUS     LOG MODE ENTRY NAME –
                                        MATCH TO MODETAB ENTRY
                  BUFFER=1024,          OUTBOUND RUSIZE
                  RUSIZE=1024,          INBOUND RUSIZE
                  TCTUAL=172            OPTIONAL TCT USER AREA
```

Participants must provide DTC with the NETNAME, that is, 'APPLID', of *their* systems for inclusion in the DTC DFHTCT.

Note: It is recommended that the participant consider maintaining a separate CICS Region for the LU6.2 link with DTC. This will facilitate the coordination of PTF upgrades at each location and avoid incompatible versions of CICS. A policy paper discussing this issue is available upon request.

E. CICS/LU6.2 Application Requirements

Shown below are two skeleton programs that highlight the key activities required to establish an LU6.2 session with the MDH system and send and receive data.

The code is a combination of actual CICS commands, mainly related to establishing the session and conversing, and pseudocode which indicates the sequence of block types that will be transmitted back and forth over the communication line.

1. Initialization

```
EXEC    CICS HANDLE ABEND LABEL(LU62-EXIT) END-EXEC.
EXEC    CICS HANDLE CONDITION SYSIDERR(ALLOC-FAIL) END-EXEC.
```

2. Allocate an LU6.2 session with the MDH System.

```
EXEC    CICS ALLOCATE SYSID(LU62-SYSTEM) END-EXEC.
```

Where the label 'LU62-SYSTEM' should be the TCT ID of the remote facility, that is, DTC.

Note: When the resource is not available (DTC System is down or the session has already been taken), the program will wait at this point until the session becomes available.

```
MOVE    EIBRSRCE TO LU62-ID.
```

Where the label 'LU62-ID' is a storage area for the Session-ID required in subsequent code.

```
EXEC      CICS        CONNECT        PROCESS
                                     PROCNAME(PROC-NAME)
                                      PROCLENGTH(4)
                                     SYNCLEVEL(1)
                                     CONVID(LU62-ID)
                                     END-EXEC.
```

Where 'PROC-NAME' is a 4- byte constant 'LU62:

3. Format a Type-01 signon block in Working-Storage including Signon-ID and Password.

4. Send the block to DTC and receive the response (Type-02 block).

```
EXEC      CICS        CONVERSE
                                     CONVID(LU62-ID)
                                     FROM(SIGNON-BLOCK-AREA)
                                     FROMLENGTH(BLOCK-01-LENGTH)
                                     SET(BLL-CELL-2)
                                     TOLENGTH(BLOCK-02-LENGTH)
                                     END-EXEC.
```

5. Validate block Type-02 returned by LU6.2 and that the signon was accepted

6. Determine which function, 'DO', 'PO', etc., to request.

7. Build and send a 'DO',. 'PO', etc., Function Request (Type-03 block) and wait for the response (Type-04 block).

```
EXEC    CICS    CONVERSE
                            CONVID(LU62-ID)
                            FROM(BLOCK-03-AREA)
                            FROMLENGTH(BLOCK-03-LENGTH)
                            SET (BLL-CELL-4)
                            TOLENGTH(BLOCK-04-LENGTH)
                            END-EXEC.
```

8. Validate block Type-04 and response.

9. Get the next transactions to be sent (up to 10) and build a Type-05 data block.

10. Send the block and wait for the Type-06 block response.

```
EXEC    CICS    CONVERSE
                            CONVID(LU62-ID)
                            FROM(BLOCK-05-AREA)
                            FROMLENGTH(BLOCK-05-LENGTH)
                            SET(BLL-CELL-6)
                            TOLENGTH(BLOCK-06-LENGTH)
                            END-EXEC.
```

11. Validate block Type-06 and response.

12. If more transactions, go to send more data (Step 9).

13. Otherwise, build and send an 'END' Type-05 data block and go to process the next function (Step 6).

14. When no more input, build and send a signoff (Type-90) block and wait for the response (Type-91) block.

15. Free the session.

```
EXEC   CICS   FREE   SESSION(LU62-ID)   END-EXEC.
```

16. Terminate the program.

Important Notes:

- Coding should be included after every 'CONVERSE' instruction to test for a Type-99 block. This block will be returned if there is any MDH system failure at DTC.
- The 'HANDLE ABEND' Routine must contain:

  ```
  EXEC   CICS   FREE   SESSION(LU62-ID)   END-EXEC.
  ```

  as its first statement.
- In the 'HANDLE ABEND' Routine, a 'USER ABEND' is acceptable only after the 'FREE SESSION' has been requested. This is required in order to keep the LU6.2 session synchronized.

1. Initialization

```
EXEC   CICS   HANDLE   ABEND       LABEL(LU62-EXIT)        END-EXEC.
EXEC   CICS   HANDLE   CONDITION   SYSIDERR(ALLOC-FATL)    END-EXEC.
```

2. Allocate an LU6.2 session with the MDH System.

   EXEC CICS ALLOCATE SYSID(LU62-SYSTEM) END-EXEC.

   Where the label 'LU62-SYSTEM' should be the TCT ID of the remote facility, that is, DTC.

   Note: When the resource is not available (DTC System is down or the session has already been taken), the program will wait at this point until the session becomes available.

   MOVE EIBRSRCE TO LU62-ID.

   Where the label 'LU62-ID' is a storage area for the Session-ID required in subsequent code.

```
EXEC    CICS    CONNECT    PROCESS
                           PROCNAME(PROC-NAME)
                           PROC LENGTH(4)
                           SYNCLEVEL(1)
                           CONVID(LU62-ID)
                           END-EXEC..
```

   Where 'PROC-NAME' is a four-byte constant 'LU62'

3. Format a Type-01 signon block in Working-Storage including Signon-ID and Password.

4. Send the block to DTC and receive the response (Type '02 block).

```
EXEC    CICS    CONVERSE
                           CONVID(LU62-ID)
                           FROM(SIGNON-BLOCK-AREA)
                           FROMLENGTH(BLOCK-01-LENGTH)
                           SET(BLL-CELL-2)
                           TOLENGTH(BLOCK-02-LENGTH)
                           END-EXEC..
```

5. Validate block Type-02 returned by LU6.2 and that the signon was accepted.

6. Build and send an 'MDLU' Function Request (Type-03 block) and wait for the response (Type-04 block).

```
EXEC    CICS    CONVERSE
                           CONVID(LU62-ID)
                           FROM(BLOCK-03-AREA)
                           FROMLENGTH(BLOCK-03-LENGTH)
                           SET(BLL-CELL-4)
                           TOLENGTH(BLOCK-04-LENGTH)
                           END-EXEC.
```

7. Validate block Type-04 and response.

8. Build a Type-07 block containing 'ALL' to receive all messages or a range of message numbers.

9. Send the block.

```
EXEC      CICS      SEND
                                   CONVID(LU62-TD)
                                   FROM(BLOCK-07-AREA)
                                   LENGTH(BLOCK-07-LENGTH)
                                   INVITE
                                   WAIT
                                   END-EXEC.
```

10. Issue a RECEIVE for the Type-08 Response block, test whether a CONFIRMATION is required and, if so, send the CONFIRMATION.

```
EXEC      CICS      RECEIVE
                                   CONVID(LU62-ID)
                                   INTO(BLOCK-08-RESPONSE-AREA)
                                   LENGTH(BLOCK-08-LENGTH)
                                   END-EXEC.

IF EIBCONF EQUAL HIGH-VALUES

EXEC      CICS      ISSUE  CONFIRMATION
                                   CONVID(LU62-ID)
                                   END-EXEC.
```

11. Determine whether the Type-08 block contains 'NONE', 'END', or data:

- If 'NONE', go to end the session (Step 15).
- If 'END', go to end the session (Step 15).
- If data, go to process the transactions (Step 12).
-

12. Process the block of data.

13. Issue a RECEIVE for another Type-08 response block, test if a CONFIRMATION is required and, if so, send the CONFIRMATION.

```
EXEC      CICS      RECEIVE
                                   CONVID(LU62-ID)
                                   INTO(BLOCK-08-RESPONSE-AREA)
                                   LENGTH(BLOCK-08-LENGTH)
                                   END-EXEC.

IF EIBCONF EQUAL HIGH-VALUES

EXEC      CIC       ISSUE     CONFIRMATION
                                       CONVID(LU62-ID)
                                       END-EXEC.
```

14. Determine whether the Type-08 block contains 'END' or data:

- If 'END', go to end the session (Step 15).
- If data, go to process the transactions (Step 12).

15. Build and send a Signoff (Type-90) block and wait for the Response (Type-91) block.

16. Free the session.

```
EXEC    CICS FREE SESSION (LU62-ID) END-EXEC.
```

17. Terminate the program .