MDH

MDH TRANSMISSION USER GUIDES

20.10 MAINFRAME DUAL HOST (MDH) SYSTEM USER GUIDE FOR PRELIMINARY SETTLEMENT STATEMENT RECORDS (OUTPUT)

Version 2

The Depository Trust Company May 2001

The Depository Trust Company
55 Water Street
New York, N.Y. 10041

Last Published: January 2, 2001

PREFACE

This document describes, specifically, the input and output requirements for obtaining the Preliminary Settlement Statement notification records via DTC's Mainframe Dual Host (MDH) system. These records (one per settlement 'activity-type') enable the participant to balance its system in an automated manner. The 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 whose mainframes 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 line (such as a telephone line). The procedure with which the participant can either request data to be transmitted to DTC or request data to be transmitted to it 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 Preliminary Settlement Statement Records (PSR)

Note: These records contain all the data that is shown on the PTS 'FFSD' inquiry by settlement-activity-type, one record each.

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

- Signon 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 PSR(168)	TRANS #2 LENGTH (4)	TRANS #2 DATA PSR (168)	\ \ / / \ \ / /	TRANS #N (UP TO 4085 TOTAL CHARACTERS

|←———MINIMUM 242——→|

|←——————————————— MAXIMUM 4085———————————————→|

Note: This data can be interspersed with other types of non-Preliminary Settlement Statement 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 242 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 Sign-on ID** must be a valid 8-character field
- **Individual user Number** - This 2-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: The 'starting seq. #' and 'total #' parameters are six-character numeric fields which must be right aligned and left zero filled.

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' signoff block.

Note:

- 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 requested, 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': DQF Recovery down 'S': Invalid signon ID 'X': Invalid password
RESPONSE-ERRROR-MESSIGE	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': 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': 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
RESPONSE-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', 'OP' or 'OD'
FILE-CONTROL-NUMBER	67	08	Format : YYYYDDDS
STARTING-SEQ-#	75	06	Numeric 'starting' sequence number desired
MAXIMUM-NUM-REQESTED	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
TRANSACTIONS-IN-BLOCK	63	04	Number of transactions in this block (Value = O)
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 - Response Block for 'SDFS' Prelim Settlement Statement

The '08' Response Block is variable length and consists of a 70-byte Block Prefix followed up by as many transactions as can fit in a 4,000-byte area. Each transaction is preceded by a 4-byte length attribute, which should be used for deblocking purposes. The exhibit below shows a block that contains a transaction for a single 'Preliminary Settlement Statement' record. It consists of the 70-byte block prefix, the 4-byte transaction length attribute, and 166 bytes of transaction data.

Response Block for SDFS Prelim Settlement Statement

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	38	Value spaces
RESPONSE-CODE	61	01	'A': Accepted 'R': Rejected
RESPONSE-REASON-CODE	62	01	'A': Not signed on 'B': Past cutoff 'C': Not in 'MDLU' 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
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 = '172'.
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
TYPE-08-RESPONSE	95	01	Value: 'E'
SET-FUNCTION-NAME	96	04	Values: 'SETA'

Field Name	Pos	Len	Field Attributes
SET-MDH-DATE	100	08	MDH processing date (CCYYMMDD)
SET-MDH-TIME	108	06	MDH processing time (HHMMSS)
SET-PARTICIPANT	114	08	Participant Number
SET-RECORD-TYPE	122	03	Values 'DET' (DETAIL RECORD)
SET-ACTIVITY-CODE	125	03	Activity code (Refer to Exhibit 9)
SET-ACTIVITY-CODE-SUFFIX	128	03	Activity code suffix (Refer to Exhibit 9)
SET-CREDITS	131	17	Credits to the participant's DTC account. Picture 9(15)V99
SET-DEBITS	148	17	Debits to the participant's DTC account. Picture 9(15)V99
SET-ADJUSTMENTS	165	18	Preliminary cutoff adjustments to the participant account. Picture 9(15)V99S
SET-NET-OBLIGATION	183	18	Preliminary cutoff net settlement obligation. Picture 9(15)V99S
SET-SHORT-DESCRIPTION	201	10	Short description of activity
SET-LONG-DESCRIPTION	211	30	Long description of activity
Filler	241	02	Value spaces

Figure 7. Response Block For SDFS Prelim Settlement Statement

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	**Pos**	**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 REQUEST		**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 ('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 (Automatic) LU6.2-to-LU6.2 system protocol message

Figure 11. Participant Request/MDH Response - Chart

Exhibit 9 - Settlement Activity/Sub-Activity Codes

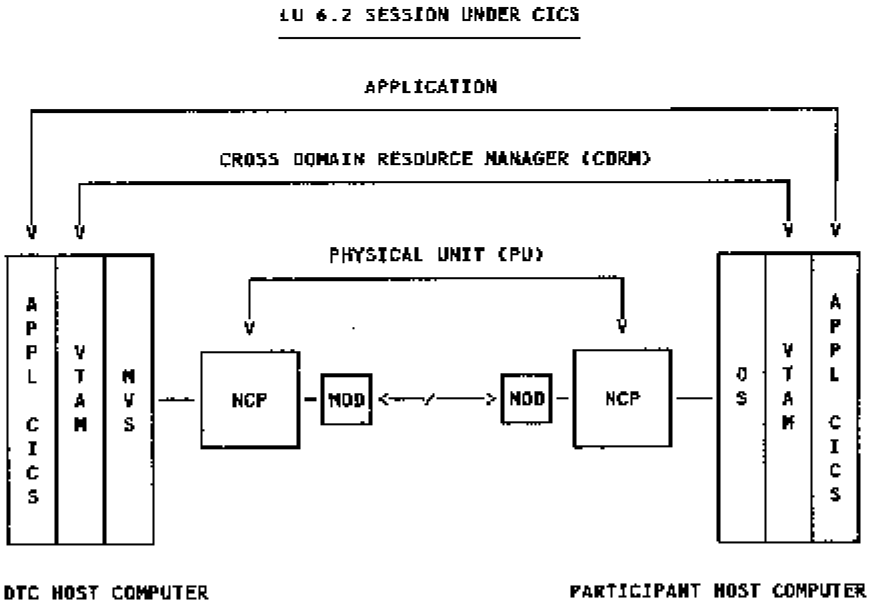
Activ	Sub-A	Description
019	000	ID Deliver Orders
019	001	MMI Night Deliver Orders
026	000	Deliver Orders SDFS
026	001	MMI Deliver Orders (Day)
026	003	REPO Deliver Orders
026	004	Issuance
026	005	Stock Loan Deliver Orders
026	006	Stock Loan Return DOs
026	007	Inter-Depository Reversals
028	000	Maturity Presentments
028	001	REORG Presentments
047	000	IPO/IPO Deliver Orders
048	000	IPO/Free Deliver Orders
049	000	Free/IPO Deliver Orders
052	000	Valued IPO Pledge
054	000	Valued Pledge
055	000	Valued Release
060	000	Income Presentments
061	000	Principal Presentments
070	000	Billing/Refunds
070	001	Part Fund Contributions
070	002	Controllers Adjustments
078	000	Security Payment Orders
078	001	MMI/Security Payment Orders
082	000	Premium Payment Orders
084	000	Reorganizations
086	000	CMO Refactoring
087	000	Dividend Allocations
088	000	Redemptions
089	000	Short Penalties
094	000	Yesterday's Suspense

Activ	Sub-A	Description
095	000	Funds Transfer Manual
095	001	Funds Transfer SPP
095	002	Funds Transfer Set
095	003	Fund Tran P/I Withdrawal
096	000	Fund Tran Cross-Endorsement
140	000	Fund Serv
141	000	Network Dividends
250	000	Late P/I Allocations

IV. MDH TECHNICAL DOCUMENTATION

This section describes the Communications and Systems Programming Requirements for participants who 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 LU 6.2 session:



A. General Communications Definitions

Listed below are the characteristics of the DTC test and production systems. For a participant to use MDH, his 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	DTCPO3
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 Requirement (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=SNASVCU,         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, i.e., '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 which 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.
```

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.

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

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

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

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

Where 'PROC-NAME' it 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' or '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 Black 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.

Purpose This code describes the LU 6.2 participant processing needed to receive transactions from DTC.

Note: Recovery logic is not included here.

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, i.e., 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 which is 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. Build and send a '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-ID)
                              FROM(BLOCK-07-AREA)
                              LENGTH(BLOCK-07-LENGTH)
                              INVITE
                              WAIT
                              END-EXEC.
```

10. Issue a RECEIVE for the 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 CICS ISSUE CONFIRMATION
                              CONVID(LU62-ID)
                              END-EXEC.
```

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

14. Determine if 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.