A53

Time To Get Acquainted With IMS Connect 1.1, Part 2

Gerald Hughes (Jerry)
ghughes@us.ibm.com

IMS Technical Conference

Miami Beach, FL October 22-25, 2001

★ ADDITIONAL IMS e-BUSINESS SESSIONS

- A12 - Application Design Considerations when Web-Enabling Your IMS
- A17 - IMS Connector for Java: A Perspective on Web-enabling Your IMS Transactions
- A20 - IMS Connector for Java: Developing Web Appcations for Accessing IMS Transactions
- A40 - The Base of IMS e-business Connectors-OTMA
- A41 - Exploring OTMA user exits
- A43 - Exploring IMS Connect User Exits
- A49 - WWW.IMS or WebSphere Working with IMS
- A52 - Time to Get Acquainted With IMS Connect (Part I)
- A53 - Time to Get Acquainted With IMS Connect (Part II)
- B04 - An Intro to IMS e-business Solutions
- S67 - IMS e-business Offering Performance Test Results
- W01- IMS Connectivity Workshop

IMS CONNECT DISCUSSION ITEMS

- ★ WHAT'S NEW WITH IMS CONNECT
- ★ IMS CONNECT CHANGES THAT MAY EFFECT THE CUSTOMER
- ★ WHY SHOULD THE CUSTOMER DEFINE THE CLIENT ID
- ★ TIMEOUT USAGE
- ★ NON-RESP TRANSACTIONS
- ★ ERROR CONDITIONS RETURNED TO CLIENT
- ★ HOST INITIATED CLIENT APPLICATION
- ★ USER MESSAGE EXIT ASYNCHRONOUS OPTIONS
- ★ SAMPLE MESSAGE FLOW
- ★ THINGS TO WATCH OUT FOR
- ★ TCP/IP TIPS
- ★ TCP/IP AND IMS CONNECT TIPS
- ★ CLIENT/HOST APPL APPEARS TO HANG
- ★ IMS CONNECT REQUIREMENTS LIST

★ WHAT'S NEW WITH IMS CONNECT

- PROGRAM PRODUCT
- SUPPORTS IMS 6.1, AND/OR 7.1
- HOLD ASYNCHRONOUS OUTPUT
- PERSISTENT SOCKETS
- INITIALIZATION EXIT
- DATA STORE TABLE
- FORMATTED DUMP
- ENHANCED TRACE
- CLOSEHWS FORCE/QUIESCE
- UNI CODE SUPPORT
- IMS CONNECTOR FOR JAVA LOCAL SUPPORT
- SMP/E INSTALLABLE
- SMP/E MAINTENANCE
- ACK/NAK RESPONSE REQUIRED
- Other small items

★ IMS CONNECT CHANGES THAT MAY EFFECT THE CUSTOMER

- Six fullwords have been added to OTMA User Data Header following OMUSR_USTAT field
  - If you've modified HWSSMPL0
  - If you've written your own User Message Exit
  - If you've modified HWSOMPFX macro
- CSM_ZZ and RSM_ZZ have been redefined, byte one provides flag byte's to communicate whether or not asynchronous output is available in IMS and if ACK/NAK response required
  - If you've modified the CSM or RSM in HWSIMSCB macro
  - If you want to provide support for ASYNCHRONOUS output
  - If you want to provide support of ACK/NAK response req'd
- IRM_RSV has been redefined, byte now provides a flag byte to communicate Socket type
  - If you've modified the IRM in HWSIMSCB macro
  - If you want to provide support for Socket type
- IRM_RSV02 has been redefined, byte now provides a flag byte to communicate Uni-code Schema type
  - If you've modified the IRM in HWSIMSCB macro
  - If you want to provide support for Uni-code
- IRM_F1 definition has been extended to include flag bytes for Uni-code support
- The TPIPE name for Commit Mode 1 has been changed from using the HWS ID= name (config file) to the PORT number. This was done to eliminate a bottleneck with OTMA. IMS Connect can now have multiple Commit Mode 1 TPIPEs.

★ WHY SHOULD THE CUSTOMER DEFINE THE CLIENT ID

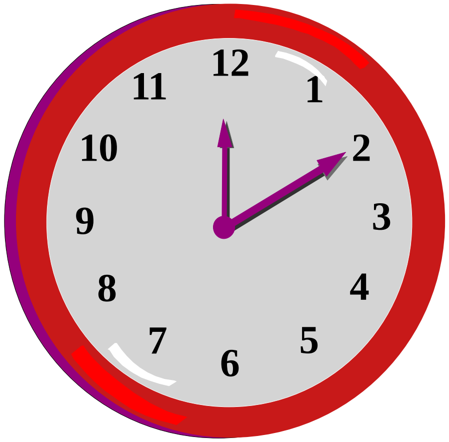
- CLIENTID IS USED AS LTERM IN IMS CONNECT
- UNIQUE CLIENTID IS REQUIRED TO BE ABLE TO ISRT TO AN ALTERNATE PCB AND BE ABLE TO DELIVER THE OUTPUT MESSAGE (ASYNCHRONOUS OUTPUT) TO THAT DESTINATION.
- IF PERSISTENT SOCKET THEN A NEW CLIENT ID IS GENERATED BY THE USER MESSAGE EXIT FOR EACH TRANSACTION AND IF NO DISCONNECT THEN THE CONTROL BLOCKS WILL BUILD UP IN IMS CONNECT AND OTMA.
- ONLY A DISCONNECT WILL ELIMINATE THE CLIENTID CONTROL BLOCKS IN IMS CONNECT, AND IT WILL ONLY ELIMINATE THE LAST IMS CONNECT CONTROL BLOCK USED FOR THAT CLIENT.

★ TIMEOUT USAGE

- TIMEOUT IN IMS CONNECT CONFIGURATION FILE

  Represents the timeout value that IMS Connect will wait for IMS to return data once the Client data has been passed to IMS. Which represents how long to wait:

  - **in case the IMS application is looping.**
  - **in case the transaction program is stopped.**
  - **in case there is no Dependent Region started.**
  - **before the Client must release the connection**



- **TIMEOUT IN IRM**

  This represents the timeout value that IMS Connect will wait on a Read request to IMS following:

  - **sending the Client ACK to IMS**
  - **sending the Client requested RESUME TPIPE**

★ NON-RESPONSE TRANSACTIONS

OTMA ONLY SUPPORTS RESPONSE MODE TRANSACTIONS.

OPTIONS TO GET AROUND THIS RESTRICTION

- USE SEND ONLY, OR
- CLIENT ISSUE READ TO RECEIVE DFS2082 MSG

OTHERWISE THE CLIENT WILL REMAIN HUNG UNTIL,

- THE CLIENT TIMEOUT OCCURS, IF ONE SET, OR
- THE CLIENT ISSUES A DISCONNECT

IMS CONNECT

★ ERROR CONDITIONS RETURNED TO CLIENT

- (4) - INPUT DATA EXCEEDS BUFFER SIZE
- (9) - CONTENT OF BUFFER INVALID
- (10) - OUTPUT DATA EXCEEDS BUFFER SIZE
- (12) - INVALID MESSAGE
- (16) - DO NOT KNOW WHO CLIENT IS
- (20) - OTMA SEGMENT LENGTH ERROR
- (44) - MESSAGE INCOMPLETE
- (48) - INCORRECT MESSAGE LENGTH

(NN) - Decimal REASON CODE for HWSSMPL0, HWSSMPL1, HWSIMSO0, and HWSIMSO1

● INPUT DATA EXCEEDS BUFFER SIZE

The following condition has occured:

- **The output buffer passed to the User Message Exit is to small to hold the message received from the Client.**

  This can happen when the User Message Exit request to small of an increase in the buffer size. This should only happen if the User Message Exit adds additional data to the message or rebulds the message incorrectly.

  No HWS Error Message
  RSM Return Code = X'04' Reason Code = X'04'

● CONTENT OF BUFFER INVALID

One of the following conditions has occured with the input buffer message passed to the User Message Exit:

- The IRM length field is zero
- The len field of a data segment is zero
- The len of the READ output buffer is not large enough to hold the rebuilt message structure

The Client has built the message incorrectly.

No HWS error message
RSM Return Code = X'04' Reason Code = X'09'

OUTPUT DATA EXCEEDS BUFFER SIZE

The following condition has occured:

- The output buffer passed to the User Message Exit is to small to hold the message being sent to the Client

  This can happen when the User Message Exit request to small of an increase in the buffer size. This should only happen if the User Message Exit adds additional data to the message or rebulds the message incorrectly.

  No HWS error message
  RSM Return Code = x'04' Reason Code = X'0A'

● INVALID MESSAGE NO DATA

The following condition has occured:

– The input message passed to the User Message Exit has encountered the EOM as the first data element following the IRM.

  The Client has built a message with IRM, and EOM only.

  No HWS error message
  RSM Return Code = x'04' Reason Code = X'0C'

● DO NOT KNOW WHO CLIENT IS

The following condition has occured:

– The input message from the client contains an invalid IRM_ID. For example the IRM_ID that is in the IRM has not been identifed by any of the User Message Exits defined to IMS Connect.

This can occur when the required User Message Exit was not included in the IMS Connect Config File Parm EXIT= of the TCPIP Statement, or the ID passed to IMS Connect by an exit at INIT call time was different than the Client is sending.

HWSP1445 error message
RSM Return Code = X'08' Reason Code = X'0F'

● OTMA SEGMENT LENGTH ERROR

One of the following conditions has occured:

- The application data length is zero, or
- The accumulated data element exceed the total message length.   As IMS Connect is processing each of elements of the message the data processed has exceeded the total message length returned to IMS Connect from the User Message Exit.

This can happen when the User Message Exit has built the message incorrectly.

HWSP1450 error message
RSM Return Code = X'08' Reason Code = X'0F'

MESSAGE INCOMPLETE

One of the following conditions has occured when IMS Connect was expecting to receive data:

- The Client has issued a disconnect, or
- The Client issued a SEND followed by a Disconnect and the Disconnect was received before the SEND of data completed.

This can happen when:

- SO_LINGER=Y,VALUE=0, or
- SO_LINGER=N

HWSP1435 error message
RSM Return Code = X'08' Reason Code = X'2C'

INCORRECT MESSAGE LENGTH

One of the following conditions has occured

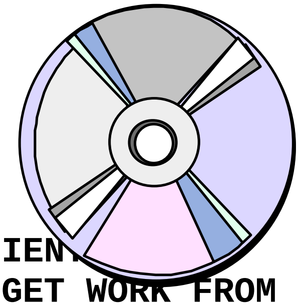
- LLLL contains Zeros, or
- LLLL is negative, or
- LLLL is less than 20 bytes, or
- LLLL is greater than 1 Gig

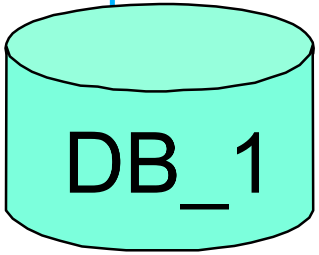
The client has built the message incorrectly.

HWSP1440 error message
RSM Return Code = x'08' Reason Code = X'30'

★ HOST INITIATE CLIENT APPLICATIOIN

```
   CLIENT02  (SEND RESUME TPIPE)----------------->

   CLIENT01 (SENDONLY/SEND)---------------------->APPL_A
                                               - PROCESS
                                               - ISRT ALT PCB APPL_B
                                               - ISRT ALT PCB CLIENT02
                                               - RETURN
|->CLIENT02  (READ)<-----------------------------
|  - PLACE ON LOCAL FILE
|<--------

                                                 APPL_B
                                               - PROCESS
                                               - GU TO DB_1 SEG_1 <---
                                               - GU OK               |
                                                  - NO, THEN WAIT    |
                                                        -------------|

   CLIEN
--->-GET WORK FROM LOCAL FILE
|   -PROCESS
|   -SEND TRANCODE/DATA---(SENDONLY/SEND)-------->APPL_C
|      |                                       - PROCESS
| -----|                                       - ISRT DB1_SEG_1
                                               - ISRT TO IO/PCB (IF SEND)
                                               - RETURN


                                                  - YES, ISRT TO IOPCB (IF SEND)
                                               - RETURN (IF SEND)
  CLIENT01<----------------------------------------
```





★ USER MESSAGE EXIT ASYNCHRONOUS OPTIONS

- SINGLE
- NOAUTO
- NOOPTION
- AUTO

➤SINGLE (RESUME TPIPE)

SET IRM_F5 TO IRM_F5_ONE

1. Client issues CONNECT function
2. Client issues RESUME TPIPE function
3. Client issues RECEIVE function
4. Client issues ACK to IMS Connect
5. Client issues DISCONNECT function

NOAUTO (RESUME TPIPE)

SET IRM_F5 TO IRM_F5_NOAUTO

1. Client issues CONNECT function
2. Client issues RESUME TPIPE function
3. Client issues RECEIVE function
   - if zero length received, go to event 6
4. Client issues ACK to IMS Connect
5. Client repeats events 3 and 4 until event 6 (below) occurs.
6. Client issues DISCONNECT function.

WHEN LAST MESSAGE RECEIVED FROM IMS, A NEW RESUMETPIPE MUST BE ISSUED

➤NO OPTION (RESUME TPIPE)

SET IRM_F5 TO IRM_F5_NOOPT

1. Client issues CONNECT function
2. Client issues RESUME TPIPE function
3. Client issues RECEIVE function
   - if zero length received, go to event 6
4. Client issues ACK to IMS Connect
5. Client repeats events 3 and 4 until event 6 (below) occurs.
6. Client issues DISCONNECT function.

WHEN LAST MESSAGE RECEIVED FROM IMS, A NEW RESUMETPIPE MUST BE ISSUED

➤AUTO (RESUME TPIPE)

SET IRM_F5 TO IRM_F5_AUTO

1. Client issues CONNECT function
2. Client issues RESUME TPIPE function
3. Client issues RECEIVE function
   - if zero length received, go to event 6
4. Client issues ACK to IMS Connect
5. Client repeats events 3 and 4
6. Client issues DISCONNECT function

To keep event 3 from getting a zero length received set the IRM_TIMER value to X'FF' to create dedicated output device.

★ SAMPLE MESSAGE FLOW

I. Persistent socket/Commit Mode 1

a. Conversation

b. Non-Conversation

II. Transaction socket/Commit Mode 1

a. Conversation

b Non-Conversation

III. Transaction socket/Commit Mode 0

a. Non-Conversation

I a. Persistent Socket/CM1/Conversation

```
                              IMS                 USER MSG
  CLIENT                    CONNECT                 EXIT              OTMA/IMS

  1.  connect--------->proc connect
  2.  send tc/data--->proc msg----------->rebuild msg
                                            build OTMA hdrs
                      proc msg<-----------------
                      send msg--------------------------------->proc tran
                      receive msg<----------------------------send output
                      proc msg----------->rebuild msg
                                            remove OTMA hdrs
                      proc msg<------------------
3.  read msg<--------send msg
4. *send ack--------->read ack
5. Repeat steps 2 (no tran code) through 4 to conversation completion
6. *Read resp<-------send RSM  (deallocate commit/abort
7. Repeat steps 2 through 6 above for next transaction or until you wish to
   terminate the persistent socket or wish to terminate the connection

8. disconnect------>proc disconnect                 * = required only if
               <--------issue disconnect              synch level=confirm
```

I b. Persistent Socket/CM1/Non-Conversation

CLIENT

IMS CONNECT

USER MSG EXIT

OTMA/IMS

1. connect--------->proc connect
2. send tc/data--->proc msg----------->rebuild msg

build OTMA hdrs

proc msg<-----------------

send msg-------------------------------------->proc tran

receive msg<---------------------------------send output

proc msg----------->rebuild msg

remove OTMA hdrs

proc msg<------------------

3. read msg<------send msg
4. *send ack-------->read ack
5. *Read resp<------send RSM (deallocate commit/abort
6. Repeat steps 2 through 5 above for next transaction or until you wish to terminate the persistent socket or wish to terminate the connection.

7. disconnect----->proc disconnect

<-------issue disconnect

* = required only if synch level=confirm

II a. Transaction Socket/CM1/Conversation

```
                              IMS              USER MSG
   CLIENT                   CONNECT              EXIT               OTMA/IMS

  1.  connect--------->proc connect
  2.  send tc/data--->proc msg----------->rebuild msg
                                           build OTMA hdrs
                       proc msg<-----------------
                       send msg------------------------------------->proc tran
                       receive msg<---------------------------------send output
                       proc msg----------->rebuild msg
                                           remove OTMA hdrs
                       proc msg<------------------
 3.  read msg<--------send msg
 4. *send ack-------->read ack
 5. Repeat steps 2 (no tran code) through 4 to conversation compeleltion
 6. *Read resp<-------send RSM  (deallocate commit/abort
 7. Disconnect------>proc disconnect                * = required only if
                <---------issue disconnect             synch level=confirm
```

Repeat steps 1 through 7 above for next transaction

II b. Transaction Socket/CM1/Non-Conversation

```
                      IMS            USER MSG
 CLIENT             CONNECT            EXIT            OTMA/IMS

 1.  connect--------->proc connect
 2.  send tc/data--->proc msg----------->rebuild msg
                                          build OTMA hdrs
                     proc msg<----------------
                     send msg------------------------------------>proc tran
                     receive msg<-------------------------------send output
                     proc msg----------->rebuild msg
                                          remove OTMA hdrs
                    proc msg<-----------------
3.  read msg<-------send msg
4. *send ack-------->read ack
5. *Read resp<------send RSM  (deallocate commit/abort
6. Disconnect----->proc disconnect                  * = required only if
             <--------issue disconnect                  synch level=confirm

Repeat steps 1 through 6 above for next transaction
```

III a. Transaction Socket/CM0/Non-Conversation

```
                    IMS               USER MSG
 CLIENT           CONNECT               EXIT              OTMA/IMS

 1.  connect--------->proc connect
 2.  send tc/data--->proc msg----------->rebuild msg
                                          transl/bld OTMA hdrs
                    proc msg<-----------------
                    send msg------------------------------------>proc tran
                    receive msg<--------------------------------send output
                    proc msg----------->rebuild msg
                                          trans/remove OTMA  hdrs
                    proc msg<------------------
3.  read msg<------send msg
4. send ack-------->read ack
5. Repeat steps 2 (no tran code) through 4 to conversation compleltion
6. disconnect----->proc disconnect
                <-----issue disconnect
```

Repeat steps 1 through 6 above for next transaction

★ THINGS TO WATCH OUT FOR

- DO YOU REALLY NEED A TIMEOUT AT THE CLIENT?????
  - ✓ IMS CONNECT TIMER > CLIENT TIMER
    - CAN RESULT IN DUPLICATE CLIENTid's
  - ✓ IMS CONNECT TIMER < CLIENT TIMER
    - CAN RESULT IN ZERO LENGTH READS AT THE CLIENT
  - ✓ TEST SYSTEM THAT REFRESHES THE MPP ON A FIXED TIME (LIKE HOURLY) CAN RESULT IN:
    - CLIENT HANG
    - ZERO LENGTH READS
    - DUPLICATE CLIENTS

★ TCP/IP TIPS

- TCP/IP "SO_LINGER = Y/N,VALUE=n"
  - SO_LINGER=Y,VALUE=0
    - Immediate return to Client Code
    - Socket Close can bypass data sent and the output will be lost
  - SO_LINGER=N
    - Immediate return to Client Code
    - Socket Close can bypass data sent and the output will be lost
  - SO_LINGER=Y,VALUE=10
    - Return to Client Code when ACK received from host, or wait for 10 sec before sending close.
    - Socket Close will not bypass data sent

★ TCP/IP TIPS (Continued)

- TCP/IP "TCPNODELAY=ENABLE/DISABLE "
  - TCPNODELAY=ENABLE
    - Data Transmitted per JAVA Write
    - Waits 1 Millisec before Transmission
    - Multiple Writes will result in Multiple Transmissions
  - TCPNODELAY=DISABLE
    - Data Collected from Writes when received from Client
    - Waits till Buffer Full before Transmission
    - Multiple Writes will result in 1 to n Transmissions

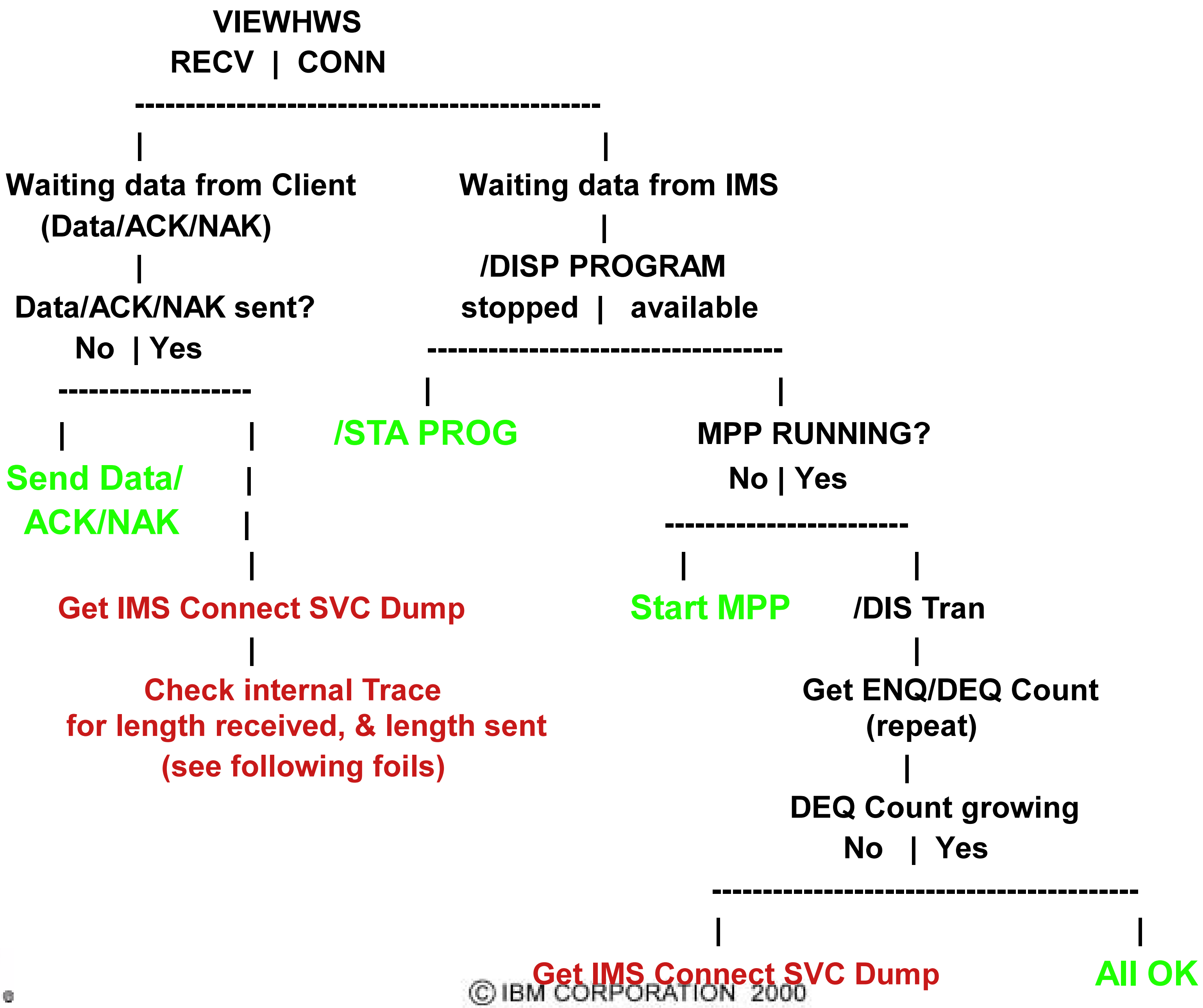
★ TCP/IP & IMS CONNECT TIPS

- TIMER USAGE IN CONFIG FILE, SO_LINGER VALUES AND CLIENT TIMEOUT SETTING RESULTS

  - SO_LINGER=v1,VALUE=v2
  - CONFIG FILE TIMEOUT=v3
  - Client timeout = v4

```
IF v1=(Y or ENABLED or ON) AND
     IF v2=0
         THEN possible DUPLICATE CLIENT

IF v1=(N or DISABLED or OFF)
    THEN possible DUPLICATE CLIENT

IF (v4 + v2) < v3    (if no v4 value use zero)
    THEN possible DUPLICATE CLIENT

IF v3 < (v2 + v4)
    THEN IMS Connect will disconnect if the v3 timer
    expires and cause the client READ to complete

    with a zero length READ
```

★ CLIENT/HOST APPL APPEARS TO HANG?



VIEWHWS

RECV | CONN

Waiting data from Client (Data/ACK/NAK)

Data/ACK/NAK sent?

No | Yes

Send Data/ ACK/NAK

Get IMS Connect SVC Dump

Check internal Trace for length received, & length sent (see following foils)

Waiting data from IMS

/DISP PROGRAM

stopped | available

/STA PROG

MPP RUNNING?

No | Yes

Start MPP

/DIS Tran

Get ENQ/DEQ Count (repeat)

DEQ Count growing

No | Yes

Get IMS Connect SVC Dump

All OK

★ REQUIREMENTS LIST

- REQUESTED REQUIREMENTS
  - #DISTRIBUTED SYNCHPOINT
  - #AUTO RECONNECT TO IMS
  - #AUTO RECONNECT TO TCP/IP
  - MODIFY COMMAND SUPPORT
  - PERSISTENT SOCKET FOR COMMIT MODE 0
  - RACF ENHANCEMENTS
    - #OVERRIDE RACF PER CALL FROM MSG EXIT
    - #PASSWORD NOT REQUIRED
    - #COMMAND TO CHANGE PASSWORD
  - IMS CONNECT COMMAND SUPPORT FROM CLIENTS
  - #DEALLOCATE RECORDER TRACE AT "RECORDER CLOSE" COMMAND TIME
  - DYNAMIC RE-LOAD OF MESSAGE EXITS
  - ALLOW USER MSG EXIT TO RETURN DATA TO TRACE
  - WIRELESS IP ADDRESS
  - #COMMAND AUTOMATION
  - #REMOVE MANDATORY EXITS (HWSJAVA0 and HWSINIT still required)
  - SOAP (XML) SUPPORT

CUSTOMER REQUIREMENTS

# DENOTE *OFFICAL* CUSTOMER REQUIREMENT

★ REQUIREMENTS LIST CONTINUED

- REQUIREMENTS THAT REQUIRE NO IMS CONNECT CODE CHANGES
  - # TCP/IP VIPA SUPPORT (In OS/390 R10, a Sysplex Distributor function is being shipped. This is a TCP/IP function that will distribute connect requests across the Syspelx. Sysplex Distributor requires VIPA).
- REQUIREMENTS SHIPPED WITH IMS CONNECT 1.1
  - #COMMAND TO ACTIVAT/DEACTIVATE IMS CONNECT TRACE (MODIFY COMMAND)
  - #TIMEOUT DETECTION (SEND RSM THEN DISCONNECT, RATHER THAN DISCONNECT)
  - #SUPPORT ALTPCB FULLY (ASYNCHRONOUS OUTPUT)
  - #SAMPLE DRU EXIT
  - #QUIESCE/FORCE CLOSEHWS OPTIONS
  - #PERSISTENT SOCKETS (COMMIT MODE 1)
  - #REMOVE AUTO LOAD OF HWSWEB00 EXIT
  - SIGNAL ACK/NAK RESPONSE REQUIRED TO CLIENT, SENT IN RSM/CSM (APAR)
  - S/390 NON-TCP/IP CONNECTION (APAR)
  - UNI-CODE SUPPORT (APAR)
  - #ON SEND FROM IMS CONNECT, SEND TOTAL MESSAGE LENGTH (APAR)
- REJECTED REQUIREMENTS
  - #RETURN MSG TO EXIT IF DATASTORE NOT AVAILABLE

DENOTE ***OFFICAL*** CUSTOMER REQUIREMENT

CUSTOMER REQUIREMENTS CONTINUED