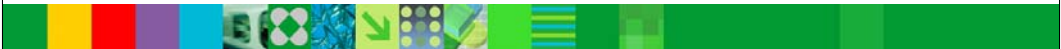




IMS Version 12

IMS to IMS TCP/IP Connectivity

Information Management software

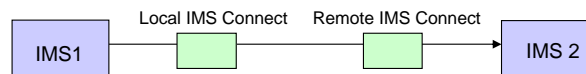


IMS to IMS TCP/IP Connectivity

- Enhancements to leverage TCP/IP networks for communications between IMS systems for:

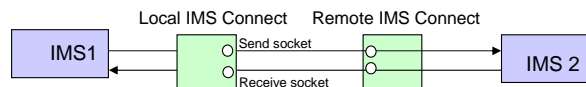
- **OTMA Support for Asynchronous IMS-IMS Communications**

- Uses one-way message communications (ALTPCB)



- **TCP/IP-Type Physical Links (MSC)**

- Request and response message communications
 - IMS Connect processes both the request and response messages as one-way asynchronous messages



379

IMS version 12 provides the ability to send messages from a local IMS to a remote IMS over a TCP/IP network. The messages are sent over the TCP/IP network by a local IMS Connect and received by a remote IMS Connect. The local IMS communicates with the local IMS Connect and the remote IMS communicates with the remote IMS Connect. IMS can send messages using OTMA or MSC protocols.

For OTMA messages, one-way asynchronous messaging is supported. IMS OTMA communicates with IMS Connect using XCF.

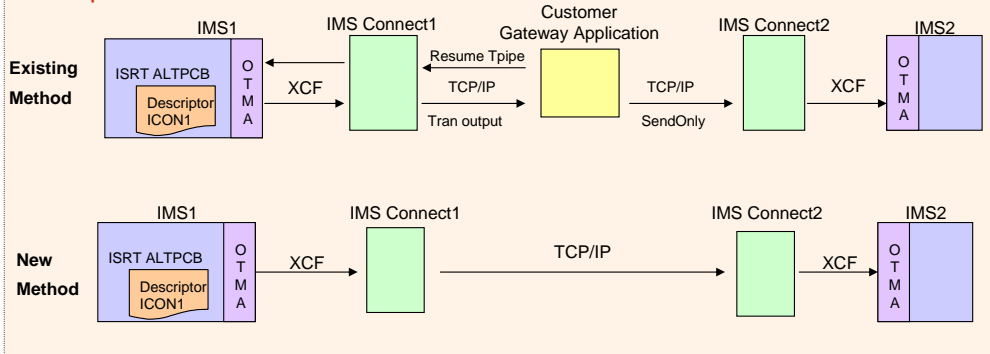
MSC supports request and response messages. IMS Connect treats those messages as one-way asynchronous messages. IMS MSC communicates with IMS Connect using SCI.

OTMA Support for Asynchronous IMS to IMS Communications

Asynchronous IMS-IMS TCP/IP Support

- TCP/IP connections between the local and remote IMS systems
 - Are managed by IMS Connect to IMS Connect communications
 - Without having to write client code or invoke additional gateways
 - The goal is simplification and ease of use

Example:

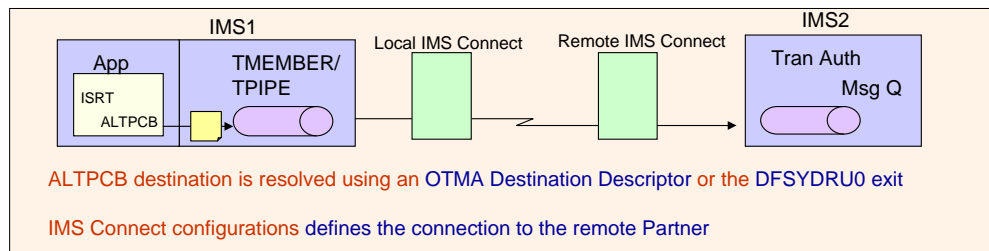


381

IMS Connect to IMS Connect messaging is a new function that sends messages from IMS to a remote IMS through a local IMS Connect and a remote IMS Connect. Existing methods typically require the creation of customer gateway applications that implement Resume Tpipe code to retrieve the message from one IMS along with code to establish a connection and send the message to another IMS system. With this new capability, IMS and IMS Connect provide the necessary communication path and code that eliminate the need for a gateway application.

Asynchronous IMS-IMS TCP/IP Support ...

- OTMA
 - Sends OTMA remote ALTPCB messages to IMS Connect using new destination information
 - OTMA destination descriptors or DFSYDRU0 exit Routine
- IMS Connect
 - Receives OTMA ALTPCB messages from a local IMS and sends them to the remote IMS Connect for processing in the remote IMS
 - Enhanced IMS Connect configuration specifications



382

The IMS application in IMS1 makes an ISRT ALTPCB call. If an OTMA descriptor is used, then the application specifies a descriptor name as the destination on the change call. IMS1/OTMA looks up the descriptor name to retrieve the values that describe the remote connection and inserts them into the OTMA headers of the message. IMS1/OTMA then sends the message to the local IMS Connect specified in the T MEMBER parameter of the descriptor. If the OTMA descriptor is not specified, the OTMA DFSYDRU0 user exit can be coded to provide the appropriate values for the remote connection.

IMS1 waits for an ACK.

IMS Connect1 receives the message, removes the OTMA headers and builds an IRM and sends the message to the remote IMS Connect specified in the OTMA headers using SendOnly with ACK. IMS Connect1 waits for an ACK.

IMS Connect2 receives the message, sends the message to the remote IMS.

When the remote IMS2 receives the message and queues the message, an ACK is sent back to IMS Connect2. IMS Connect2 sends the ACK back to IMS Connect1 which sends the ACK back to IMS1. IMS1 dequeues the message from its tpipe.

The remote IMS processes the message as a transaction. Responses, if any, from the IMS application will be sent to the remote IMS's tpipe.

OTMA Support – DFSYDTx Descriptors and DFSYDRU0

▪ New destination descriptor parameters (DFSYDTx member of proclib)

- RMTIMSCON= specifies the remote IMS Connect (required)
- RMTIMS= specifies the target remote IMS (required)
- RMTTRAN= specifies the transaction name to use on the remote IMS (optional)
- USERID= specifies the RACF userid name to use on the remote IMS for transaction authorization (optional)
- Existing parameter (required)
 - TMEMBER= specifies the local IMS Connect

OR

▪ DFSYDRU0 exit

- Set output flag bit to indicate the message is an OTMA remote ALTPCB message.
- Set the required and optional information in the OTMA prefix area (mapped by HWSOMPFX macro).
 - New specifications in the user data section:
 - OMUSR_RMTICON – remote IMS Connect
 - OMUSR_RMTIMSID – remote IMS
 - OMUSR_RMTTRAN – remote transaction
 - OMUSR_RMTUID – userid for transaction authorization

– If a descriptor is used, the DFSYDRU0 exit is not called

383

ALTPCB messages that are to be sent to another IMS are sent to the target environment using information that is specified in either an OTMA Destination Descriptor or in an OTMA Destination Resolution (DFSYDRU0) exit routine. If a descriptor is coded, the DFSYDRU0 exit routine is not called.

The four new parameters that have been added to the OTMA destination routing descriptor are: RMTIMSCON, RMTIMS, RMTTRAN, and USERID.

- RMTIMSCON is the name of the remote IMS Connect connection to be used by the local IMS Connect. This value must match a value in the ID parameter of the RMTIMSCON statement in the local IMS Connect. If specified, the parameter RMTIMS must also be specified.
- RMTIMS is the name of the remote IMS to which the message is to be sent. This value is the same value specified in the ID parameter of the DATASTORE statement in the remote IMS Connect. If specified, the parameter RMTIMSCON must also be specified.
- RMTTRAN is optional and defines the transaction code to use at the remote IMS. If a value is not specified, the transaction in the start of the message will be used.
- USERID is optional and is the SAF (RACF) userid name to use at the remote IMS. If not specified, the userid from the IMS application that did the ISRT will be used.

Alternatively, the user exit DFSYDRU0 can provide the connection values. A sample DFSYDRU0 exit provides details of the output flag bit indicator in the output parameter list. Information about the local IMS Connect continues to be specified as in previous releases, i.e., in the output parameter list of the DFSYDRU0 exit or in the OTMA pre-routing exit DFSYPRX0 which is called before the DFSYDRU0 exit.

OTMA Support – Type-2 Commands

- Descriptors can be created, queried, or modified dynamically
 - CREATE | QUERY | UPDATE OTMADESC
 - Type-2 commands support the new parameters
 - RMTIMSCON, RMTIMS, RMTTRAN, USERID

```
CREATE OTMADESC NAME(DESC1) SET(TYPE(IMSCON) TMEMBER(ICON1) RMTIMSCON(ICON2)
RMTIMS(IMS3) RMTTRAN(TRAN01) USERID(USER01))
```

Output:

DestName	MbrName	CC
OTMACL01	IMS1	0

```
QUERY OTMADESC NAME(OTMACL01)
```

Output:

MbrName	DestName	CC	Type	TMember	SMember	Tpipe	Adapter
IMS1	DESC1	0	IMSCON	ICON1			
Converter	Option	RmtIMS	RmtIMSCon	RmtTran	Userid		
		IMS3	ICON2	TRAN01	USER01		

```
UPDATE OTMADESC NAME(DESC1) SET(RMTIMS(IMS2) RMTTRAN(TRANABC))
```

Output:

DestName	MbrName	CC
DESC1	IMS1	0

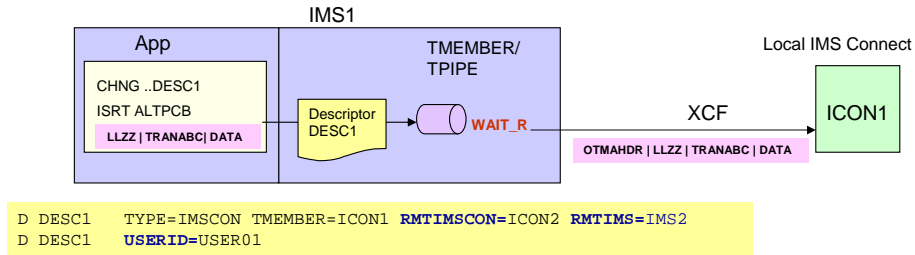
OTMA destination descriptors can be created dynamically as well as queried and updated as needed. This visual show an example of the Type-2 commands CREATE/QUERY/UPDATE and the new parameters that have been introduced for the OTMA TCPIP support.

Descriptors and Command Overrides

- Specifying a descriptor parameter requires an associated value
 - Not having value can result in an error
 - Example
 - D DESC1 TYPE=IMSCON TMEMBER=ICON1 **RMTIMSCON=ICON2 RMTIMS=IMS2 RMTTRAN=**
 - DFS2385E SYNTAX ERROR FOR DESCRIPTOR = TEST0001 RMTTRAN MUST BE 1 TO 8 CHARACTERS LONG IMS1

- The Type-2 UPDATE command can specify the parameter without a value to reset it
 - Example
 - Specify the RMTTRAN parm without any value in the type-2 update command:
 - UPD OTMADESC NAME(TEST01) SET(RMTTRAN())
 - Clears the RMTTRAN field

OTMA Support – Message Flow Details



IMS Application

- Issues CHNG call to set the destination name (descriptor)
 - IMS resolves this destination using an OTMA destination descriptor by the same name or through DFSYDRU0 specifications
- Issues ISRT ALTPCB to send the message to a remote IMS

OTMA

- Using information from either the descriptor or exit
 - Sends the message to the local IMS Connect from a regular TPIPE queue
 - Builds the OTMA header
- Waits for ACK/NAK
 - TPIPE queue is in a new **WAIT_R** status during this wait

The goal is to send a transaction message from an IMS application in the local IMS1 to the remote IMS2.

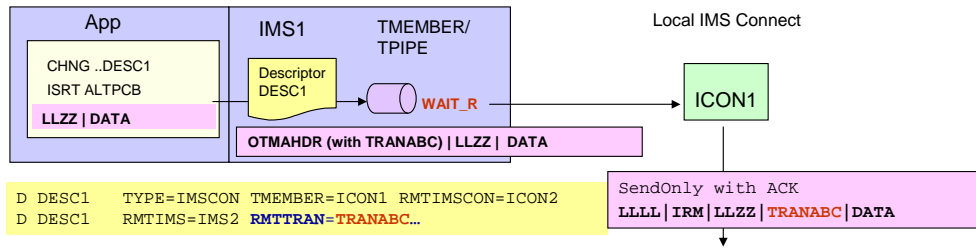
The local IMS application builds the message to send to the remote IMS. The message must follow the standard IMS transaction message format:

LLZZ | TRANCODE | DATA (if the receiving IMS application expects a trancode in the first part of the message) or LLZZ | DATA :

- LL - Halfword length of the message including the 4-byte length of the LLZZ fields.
- ZZ - Halfword of zero.
- Optional: TRANCODE - Transaction name up to 8 characters.
- DATA - Data expected by the IMS application that will process the transaction on the remote IMS.

The local IMS application issues a CHNG call to specify the descriptor if the descriptor name is different than that had coded in the ALTPCB, and then issues ISRT ALTPCB to send the message.

OTMA picks up the TMEMBER from the descriptor to determine which local IMS Connect to use for the transmission. The message is sent out from a regular TMEMBER/TPIPE queue and the queue then goes into a WAIT_R status.

Note: Message Destinations

- **RMTTTRAN is not specified**
 - The first 8 bytes (or first set of characters preceding a blank) in the IMS application message are considered to be the trancode
- **RMTTTRAN is specified**
 - IMS Connect **always** adds the value to the message with a corresponding increase in the length of the message
 - If the application does not specify a trancode
 - IMS Connect uses the RMTTTRAN value (e.g. TRANABC and increases message length)
 - Example: IMS app message: x'0010' X'0000' C'123456789012' (LLZZ|DATA)
After IMS Connect: x'0018' X'0000' C'TRANABC 123456789012'
 - If the application specifies trancode at the beginning of the message
 - The message built by IMS Connect will include both trancodes with an increase of the message length
 - Example: IMS app message: x'0010' X'0000' C'TRAN123 9012' (LLZZ|DATA)
After IMS Connect: x'0018' X'0000' C'TRANABC TRAN123 9012'
- **The remote IMS may reject the message if the trancode is invalid**
 - HWST3570E on local IMS Connect and the NAK'd message is sent to HWS\$DLQ on local OTMA

387

The RMTTTRAN specification in the OTMA descriptor is optional. If unspecified, then IMS Connect does not touch the message and the first 8 characters (or characters preceding a blank) at the front of the message are assumed to be a trancode. This is a consideration when determining whether or not to specify a RMTTTRAN value because IMS applications can choose whether or not to insert a trancode at the beginning of a message.

•Example 1: IMS app message: x'0010' X'0000' C'TRANABC 9012' (LLZZ|TRANCODE|DATA) and the descriptor has not specified RMTTTRAN. IMS Connect does not change the message and the trancode for the remote IMS system is assumed to be TRANABC.

•Example 2: IMS app message: x'0010' X'0000' C'123456789012' (LLZZ|DATA) and the descriptor has not specified RMTTTRAN. IMS Connect again does not change the message and the trancode for the remote IMS system is assumed to be 12345678 which may be invalid and will result in an error.

On the other hand, the RMTTTRAN specification in the OTMA descriptor can be used to set the destination trancode. If specified, it is always added to the front of the message with a corresponding increase in the length field.

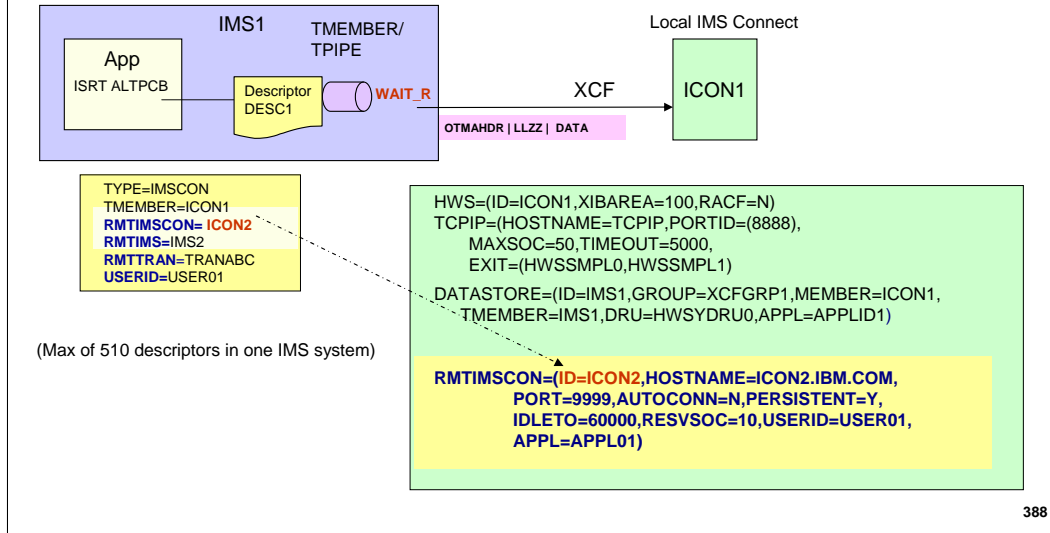
•Example: IMS app message: x'0010' X'0000' C'123456789012' (LLZZ|DATA) and the descriptor specifies RMTTTRAN(TRANABC). After IMS Connect processing: x'0018' X'0000' C'TRANABC 123456789012' which now includes the transaction value (from RMTTTRAN=TRANABC) padded with blanks.

If the application specifies a trancode at the front of the message and RMTTTRAN is also specified then the message built by IMS Connect will include both the transaction provided by the IMS application as well as the RMTTTRAN value. The message length will be increased accordingly.

•Example: IMS app message: x'0010' X'0000' C'TRAN123 9012' (LLZZ|TRANCODE|DATA) and the descriptor specifies RMTTTRAN(TRANABC). After IMS Connect processing: x'0018' X'0000' C'TRANABC TRAN123 9012' which now includes the TRANABC value padded with blanks. This might be an issue for the receiving system depending on whether the real IMS target transaction is TRANABC or TRAN123.

IMS Connect Support

- IMS Connect provides a new configuration statement **RMTIMSCON** to describe the remote IMS Connect TCP/IP environment



388

The RMTIMSCON value specified in the descriptor is sent to the local IMS Connect, e.g., ICON1, in the OTMA prefix headers. ICON1 uses the RMTIMSCON value to lookup the RMTIMSCON in its configuration member which contains the remote IMS Connect hostname to send the message.

IMS Connect Configuration – RMTIMSCON ...

```
RMTIMSCON=(ID=ICON2,HOSTNAME=ICON2.IBM.COM, PORT=9999,AUTOCONN=N,PERSISTENT=Y,
IDLETO=60000,RESVSOC=10,USERID=USER01,APPL=APPL01)
```

- New RMTIMSCON statement in the local IMS Connect identifies the remote IMS Connect partner information
 - **ID=** Identifier for a remote IMS Connect connection
 - **IPADDR=** specifies the remote IMS Connect IP address (IPADDR or HOSTNAME required, but not both)
 - Accepts a loopback IP address
 - Local and remote IMS Connects are on the same TCP/IP stack (same LPAR)
 - In the range 127.0.0.0 to 127.255.255.255
 - Likely used only in test environments
 - **HOSTNAME=** specifies the remote IMS Connect hostname
 - **PORT=** specifies the remote IMS Connect port
 - **AUTOCONN= Y|N**, specifies whether to automatically make socket connections to the remote IMS Connect on startup (optional - for OTMA only, MSC requires AUTOCONN=N)
 - **PERSISTENT= Y|N**, specifies whether the socket connections to this remote IMS Connect are persistent (optional - for OTMA only, MSC requires PERSISTENT=Y)

389

The RMTIMSCON statement is required for using both the OTMA remote ALTPCB output function and for MSC TCP/IP links.

Either an IP address or Hostname specification is required, but not both. For test environments, if both the Local and Remote IMS connects reside on the same TCP/IP stack (same LPAR) that the loopback IP address can be used.

•AUTOCONN defaults to No, if not specified. MSC links do not support AUTOCONN=Y. If an MSC statement references a RMTIMSCON statement in which AUTOCONN=Y is specified, AUTOCONN=N is used and a warning message is issued.

•PERSISTENT defaults to No, if not specified. MSC links do not support PERSISTENT=N. If an MSC statement references a RMTIMSCON statement in which PERSISTENT=N is specified, PERSISTENT=Y is used and a warning message is issued.

Note: If AUTOCONN=Y and PERSISTENT=N is specified, then AUTOCONN will be set to N and a warning message is issued.

IMS Connect Configuration - RMTIMSCON

```
RMTIMSCON=(ID=ICON2,HOSTNAME=ICON2.IBM.COM, PORT=9999,AUTOCONN=N,PERSISTENT=Y,
IDLETO=60000,RESVSOC=10,USERID=USER01,APPL=APPL01)
```

- New RMTIMSCON statement in the local IMS Connect identifies the remote IMS Connect partner information...
 - **IDLETO=** specifies the amount of time the socket connection is idle before terminating it due to inactivity (Optional, for OTMA only)
 - **RESVSOC=** specifies the number of sockets to reserve for use to this remote IMS Connect
 - **USERID=** specifies user ID for generating RACF Passticket to be use by the remote IMS Connect for authenticating the user (optional – if specified, then APPL must also be specified.)
 - **APPL=** specifies the application name for generating RACF Passticket to be use by the remote IMS Connect for authentication
 - optional – if specified, then USERID must also be specified

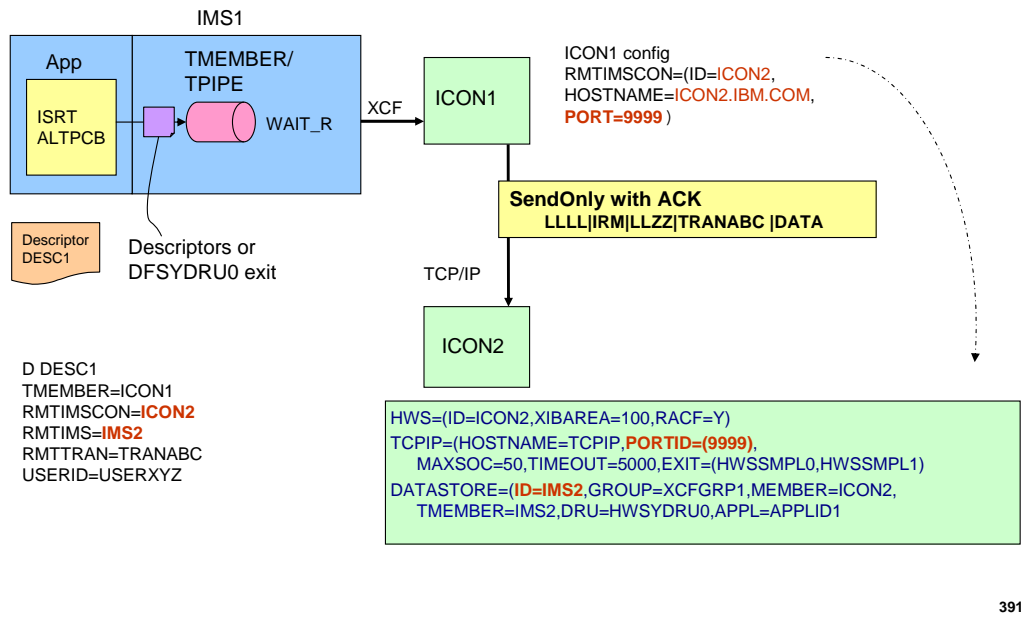
390

•IDLETO defaults to 0 if not specified. 0 means no timeout.

•RESVSOC defaults to 0 if not specified. For connections using for MSC, this value should be set to the number of parallel sessions. IMS Connect will reserve this number of sockets from the maximum number of sockets allowed for this instance of IMS Connect, as specified in the MAXSOC= parameter in the TCPIP statement. If additional connections are required for MSC above the RESVSOC count, IMS Connect will get them if possible and they will count towards the max sockets count. The sum of all the RESVSOC= values in the RMTIMSCON statements must not exceed the value specified in the TCPIP MAXSOC= parameter. If the sum exceeds the MAXSOC value, then IMS Connect will issue an error message and abend on startup.

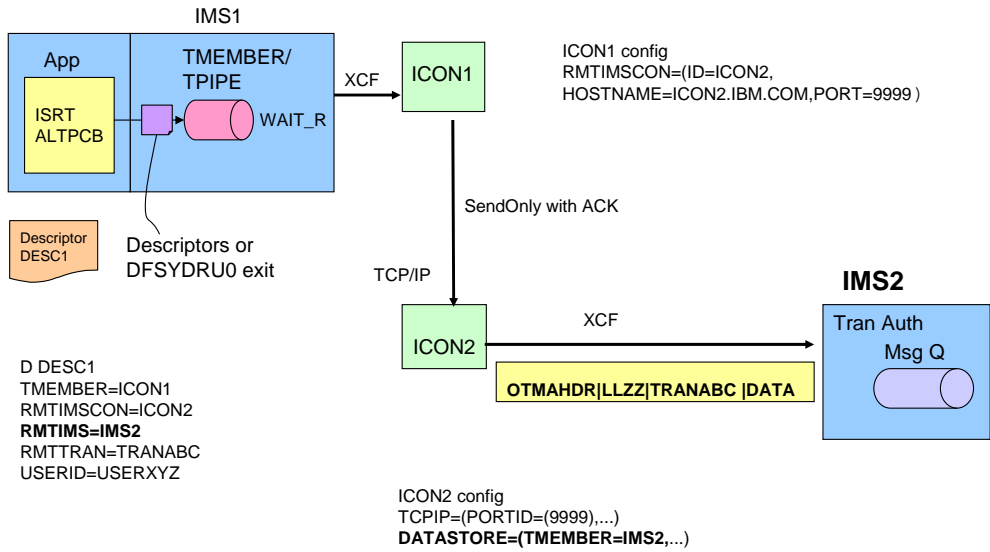
Note: The connections between the two IMS Connect systems have the option of leveraging the RACF Passticket security support. If specified, the local IMS Connect generates a PassTicket and sends it to the remote IMS Connect on the initial connection request. The remote IMS Connect calls RACF to authenticate the user with the PassTicket. The connection socket is then marked as authenticated. All subsequent messages on the socket are considered to be from a trusted user. To enable this security, set RACF=Y in the HWS statement of the remote IMS Connect and specify the USERID and APPL parameters in the RMTIMSCON statement of the local IMS Connect. RACF= is an existing parameter. This security can be used for both OTMA remote ALTPCB output and MSC TCP/IP link messages.

Remote IMS Connect Configuration

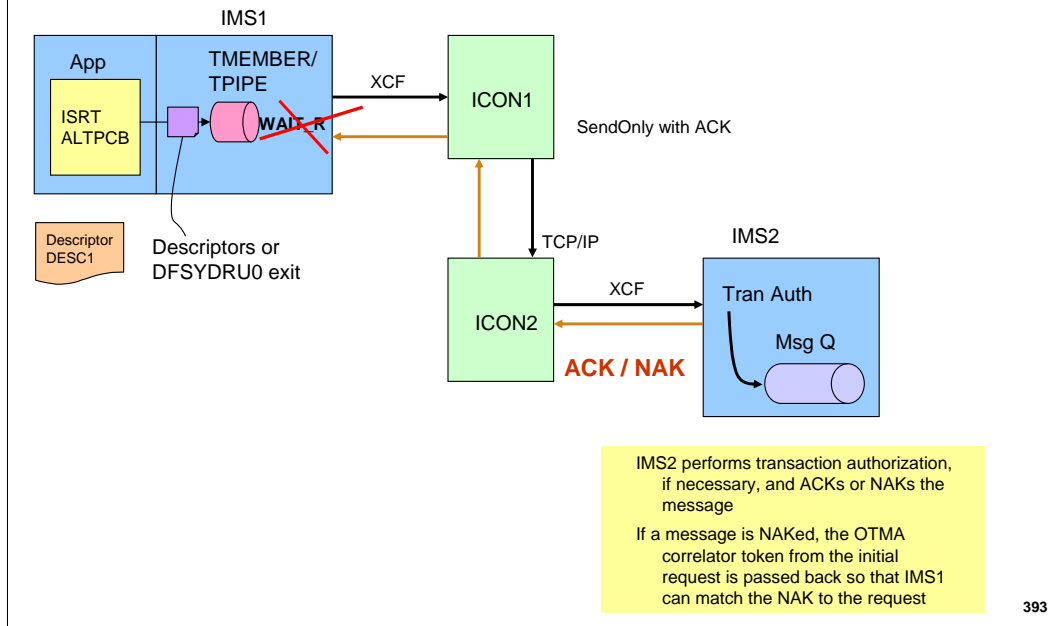


The local IMS Connect sends the message to the remote IMS Connect specified in the RMTIMSCON statement using SendOnly with ACK protocol and waits for acknowledgement that the message has been received. The message is in the format: LLLL | IRM | LLZZ | TRANCODE | DATA. The remote IMS Connect must be configured to listen for incoming connections on a specific port and to connect to the remote IMS datastore.

Send to remote IMS



And finally ... the Return ACK / NAK



Upon receipt of the message, the remote IMS Connect, e.g., ICON2, forwards the message on to the target IMS system. In the example on this visual, IMS2 receives the message, proceeds through transaction authorization and completes enqueue processing. IMS2 sends an ACK back to IMS Connect2 which forwards the acknowledgement to IMS Connect1 and IMS1 which releases the outstanding WAIT-R. IMS1 dequeues the message from its tpipe. If at any time during the aforementioned interaction IMS2 rejects the message then a negative acknowledgement or NAK is sent back to IMS1.

Note that when sending a NAK back to the local IMS Connect or local IMS, the remote IMS passes the OTMA correlator token that was included on the initial request. OTMA uses the correlator token to match the correct request message to NAK.

On the remote IMS2 system:

- The IMS application that issues a GU, IOPCB to retrieve the message can detect where the message came in using this support because the LTERMname in the IOPCB is an ICON1 generated clientid which is prefixed with the characters "OTM".
- Any IOPCB reply will need to be retrieved using a Resume TPIPE request.
- Any ALTPCB output will need to use another one way connection (RMTIMSCON) from ICON2 to ICON1.

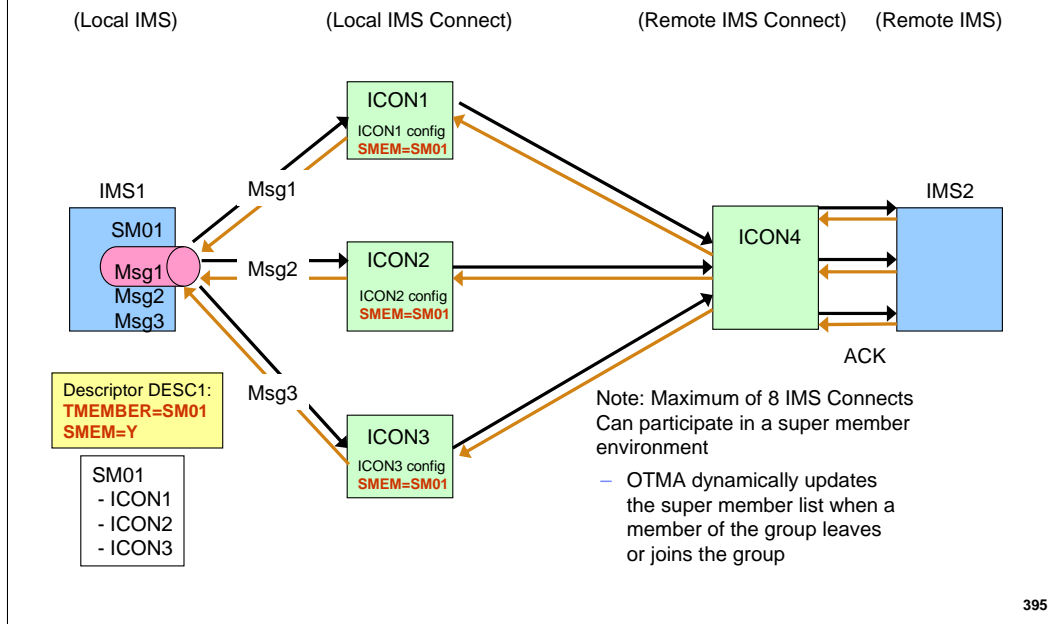
Application Considerations

- This support assumes the destination is a transaction
 - No support for messages being sent to an LTERM destination name
- On the remote IMS (e.g., IMS2)
 - IOPCB LTERMname is OTMxxxxx
 - Clientid generated by the sending IMS Connect, e.g., ICON1
 - IOPCB output is placed on the hold queue
 - Either retrieved using a Resume TPIPE request or purged via command
 - If the application issues an ALTPCB for a destination back in the other IMS (e.g., IMS1)
 - Definitions for OTMA descriptor and RMTIMSCON going from IMS2 to IMS1

On the remote IMS2 system:

- The IMS application that issues a GU, IOPCB to retrieve the message can detect where the message came in using this support because the LTERMname in the IOPCB is an ICON1 generated clientid which is prefixed with the characters "OTM".
- Any IOPCB reply will need to be retrieved using a Resume TPIPE request.
- Any ALTPCB output will need to use another one way connection (RMTIMSCON) from ICON2 to ICON1.

Super Member support



395

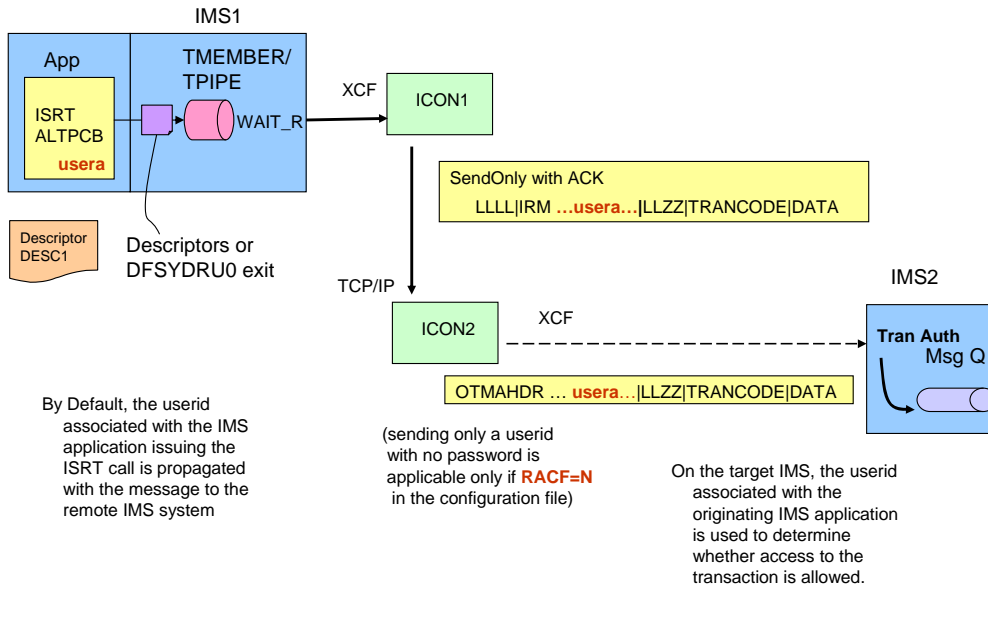
In an OTMA super member environment, OTMA sends the OTMA remote ALTPCB messages to the IMS Connects in the super member using a round robin algorithm.

In the example on this visual, IMS1 has a descriptor DESC1 which specifies TMEMBER=SM01 and SMEM=YES. Three local IMS Connect systems ICON1A, ICON1B, ICON1C all have SMEMBER=SM01 specified in their IMS Connect configuration members. When an IMS application in IMS1 issues an ISRT ALTPCB using the descriptor DESC1, OTMA looks up the IMS Connects registered to the super member group SM01 and selects one of them in a round robin order to send the message to. For example, the first ISRT ALTPCB message using descriptor DESC1 will be sent to ICON1. A second ISRT ALTPCB message using descriptor DESC1 will be sent to ICON2. A third ISRT ALTPCB message using descriptor DESC1 will be sent to ICON3. A fourth ISRT ALTPCB message using descriptor DESC1 will be sent to ICON1.

Note: the super member support for OTMA remote ALTPCB output is limited to 8 IMS Connects participating in the same super member group. If more than 8 IMS Connects register to the same super member group, then only the first 8 will be used in the round robin algorithm.

Security: Transaction Authorization

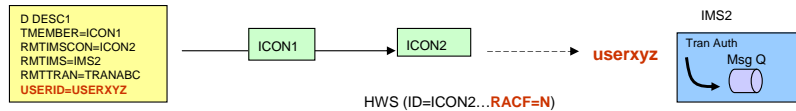
Security: Transaction authorization – Default userid



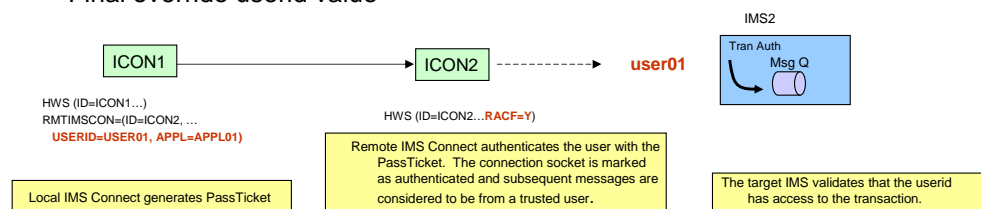
Transaction authorization is performed on the remote IMS/OTMA system. When the local IMS/OTMA sends a message to the remote IMS/OTMA, it includes the userid of the IMS application that issued the ISRT call. The ability to send a userid with no password is applicable only if RACF=N on the remote IMS Connect (ICON2). For the local IMS Connect (ICON1), outbound messages do not invoke security so the RACF configuration value can be either Y or N.

Security: Transaction authorization – Alternate userid

- Alternatively, the OTMA descriptor can be used to specify a userid
 - Overrides the userid associated with the sending IMS application



- And optionally, “connection security”
 - RMTIMSCON configuration in IMS Connect provides USERID and APPL
 - Supports generation of a PassTicket for a trusted user connection
 - Final override userid value



398

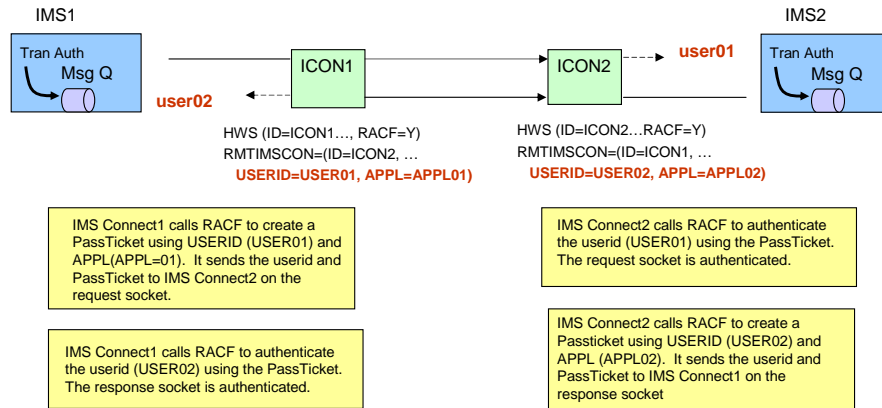
Alternatively, the USERID parameter in the OTMA descriptor can specify a value that is to be sent with the message and used for transaction authorization on the remote side. If the USERID parameter is defined in the OTMA destination descriptor, then it overrides the default of using the IMS application userid. The same considerations as discussed on the previous visual apply for the RACF specifications.

The IMS Connect to IMS Connect messaging function also introduces a new optional security feature that verifies the local IMS Connect connecting to the remote IMS Connect is a trusted user by using RACF PassTicket. This security can be used for both OTMA and MSC messages. The local IMS Connect generates a PassTicket, based on the values in the USERID AND APPL parameters, and sends it to the remote IMS Connect on the initial connection request. The remote IMS Connect calls RACF to authenticate the user with the PassTicket. The connection socket is then marked as authenticated. All subsequent messages on the socket are considered to be from a trusted user. To enable this security, RACF=Y must be set in the HWS statement of the remote IMS Connect and valid values specified in the USERID and APPL parameters in the RMTIMSCON statement of the local IMS Connect.

As a review, a RACF PassTicket is a cryptographically-generated, single-use (cannot be replayed), short-lifespan (approximately 10 minutes) password substitute and is, therefore, inherently more secure than a password. The algorithm that generates the PassTicket is a function of: the USERID of the client, the APPLID, a secured sign-on application key which is known to both sides and used for the encryption/decryption, and a time and date stamp.

Security: Transaction authorization - Alternate userid...

- For two-way “connection security”
 - RMTIMSCON security specifications must be made on both systems



Note: Only security that is applicable to MSC connection

For two-way IMS to IMS communications, as well as for all MSC TCP/IP configurations, the security has to be enabled on both local and remote IMS Connects for each of the request and response sockets.

PassTicket Connection Security

- Applicable to persistent connections
 - Setup
 - Sending (Local) Side
 - RACF: Define PTKTDATA Class entry
 - IMS Connect: Specify USERID and APPL in the RMTIMSCON definition
 - Receiving (Remote) Side
 - RACF: Define matching PTKTDATA Class entry
 - Define APPL Class entry
 - Authorize (PERMIT) valid connection userids
 - Results
 - Sending IMS Connect generates the PassTicket and sends the connection credentials
 - once per socket on first request
 - Receiving IMS Connect with RACF=Y specified authenticates and authorizes the connection
 - sets socket as trusted for subsequent transactions

Not recommended
for short-lived, non-
persistent
connections

PassTicket connection security is of value to persistent connections. RACF setup is required on both ends of the connection.

For non-persistent connections, using Passticket connection security is not recommended because it creates high overhead at each connection. Additionally, if the remote or target IMS Connect is configured with RACF=N, then USERID and APPL on the local IMS Connect should not be set so that the local IMS Connect system does not generate the Passticket.

PassTicket Connection Security - Details

▪ Sending or Local IMS Connect

- RMTIMSCON statement must specify both USERID and APPL
 - If only one is specified, IMS Connect issues an error message and abend at startup

```
HWSX0909E ERROR IN PROCESSING CONFIG MEMBER HWSMSC01: M=XCFG
HWSX0909E THE APPL IS NOT SPECIFIED BUT THE USERID IS; R=68, S=NOAPPLID
BPE0006I HWS TRSP TCB ABEND U3401-00000004 DIAG=1004000180
```

- Create PTKTDATA Class entry in RACF

- **Key** must match remote side

```
SETOPTS CLASSACT(PTKTDATA)
SETOPTS RACLIST(PTKTDATA)
RDEF PTKTDATA APPLI2I SSIGNON(KEYMASKED(E001193519561977)) UACC(N)
SETOPTS REFRESH RACLIST(PTKTDATA)
```

- IMS Connect generates Passticket from Userid and APPL
 - Only on first send, regardless of RACF= specification on the sending IMS Connect system
 - Userid does not have to exist, but PTKTDATA class resource must

401

The USERID and APPL parameters are paired in the IMS Connect configuration. If one is specified but not the other, IMS Connect abends at startup. Additionally, if both parameters are specified, IMS Connect will call the underlying security product, e.g., RACF, to create the PassTicket regardless of the RACF= specification. To receive the PassTicket and validation the connection security, the remote IMS Connect must specify RACF=Y.

Note that the PTKTDATA class entry in RACF has a key that must match the key in the remote side's PTKTDATA class entry.

If the request to generate a PassTicket fails, IMS Connect issues message HWST3520E. In this situation, no connection security credentials are passed.

PassTicket Connection Security – Details ...

- Receiving or Remote IMS Connect
 - Requires RACF specifications for the users, PTKTDATA, and APPL
 - Remember that the PTKTDATA Class entry **key** must match sending side

```

DELUSER USER001
ADDUSER USER001 PASSWORD(USER0001) TSO(ACCTNUM(D1001) PROC(TPROC02))
DELUSER USER002
ADDUSER USER002 PASSWORD(USER0002) TSO(ACCTNUM(D1001) PROC(TPROC02))

SETROPTS CLASSACT(PTKTDATA)
SETROPTS RACLIST(PTKTDATA)
RDEF PTKTDATA APPL02 SSIGNON(KEYMASKED(E001193519561977)) UACC(N)
SETROPTS REFRESH RACLIST(PTKTDATA)

SETROPTS CLASSACT(APPL)
SETROPTS RACLIST(APPL)
RDEFINE APPL APPL02 UACC(N)
PE APPL02 ACCESS(READ) CLASS(APPL) ID(USER002)
SETROPTS RACLIST(APPL) REFRESH
RLIST APPL APPL02 AU
  
```

PM32008

To prevent timeouts

```

RDEF PTKTDATA APPL02 SSIGNON(KEYMASKED(...)) UACC(N)+
APPLDATA('NO REPLAY PROTECTION')
  
```

402

The receiving IMS Connect checks to see if RACF is enable (RACF=Y) and, if so, issues a RACROUTE VERIFY to validate the userid and PassTicket.

If successful, then a trusted user connect is established and the userid is passed to IMS/OTMA. Otherwise, IMS Connect issues HWSP1507E CONNECTION SECURITY FAILURE; C=OTM097BC, U=USER001, A=APPLI2I, RC=8, RSN=52, M=OTMX. Sends a NAK to the partner IMS Connect system and closes the socket connection.

Operational and Usage Considerations

Considerations

Operational Characteristics

- Each RMTIMSCON statement results in a separate TCB
- If AUTOCONN=Y and PERSISTENT=N
 - AUTOCONN is internally reset to N
 - HWSX0902W message is issued
- AUTOCONN=Y and PERSISTENT=Y connections are protected from timing out with DELDUMMY
 - For socket connection that have been established
 - Local IMS Connect sends PING message with OTMxxxx client name to keep connection from timing out on remote IMS Connect
- After receipt of ACK/NAK on a non-persistent connection
 - Associated socket connection and control blocks are cleaned up
- Persistent sockets are closed if idle for the IDLETO timeout value

404

If AUTOCONN=Y and PERSISTENT=N then internally AUTOCONN is reset to N. The reasoning behind this decision is that there is no point in automatically connecting if the connection is not going to be persistent. IMS Connect will issue the HWSX0920W message: HWSX0920W VALUE OF PARAMETER parm IN STATEMENT statement=id CHANGED TO newvalue FROM oldvalue BECAUSE reason; M=mc.

To prevent AUTOCONN and PERSISTENT connections from timing out, the local IMS Connect sends PING messages with OTMxxxx client name.

For non-persistent connections, the local IMS Connect closes the socket connection and cleans up the associated control blocks after receipt of the ACK/NAK.

When a persistent socket connection is idle for the IDLETO value amount of time, IMS Connect closes the socket connection

Considerations

- **Restrictions**
 - Cannot be used from IMS in a shared queues environment that is not connected to IMS Connect
 - Resolution is to establish connectivity directly from each IMS
 - Super member support only allows up to 8 IMS Connects or 8 IMS Connect datastores in a super member group

- **Wait-R can result in a NAK due to error situations**
 - Can cause message reroutes to the dead letter queue, e.g., HWS\$DLQ

In certain instances, if OTMA receives a NAK response from IMS Connect, OTMA stores the undelivered message on the IMS Connect dead letter queue. The IMS Connect dead letter queue is identified by the pipe name HWS\$DLQ. This capability was introduced in IMS version 11. With the new OTMA remote ALTPCB output support in IMS version 12, the WAIT-R that is issued in the local IMS system introduces other situations that could cause messages to be rerouted to this queue.

Considerations ...

▪ Timeouts

- When a message is sent to the remote IMS, the tpipe from which the message is sent is in WAIT_R state
 - If a timeout occurs based on CMO ACK timeout value (IMS 11)
 - OTMA reroute message to the timeout queue and issues DFS3494E

▪ IMS Connect Auto Retry

- If a connection cannot be established to send a message to the remote IMS Connect
 - NAK is sent to the local OTMA to keep the message at the front of the queue
 - Connection retry occurs every 2 minutes if failure is due to
 - TCP/IP is not available (local or remote)
 - Remote IMS Connect is not available
 - When connection is established, OTMA is notified

406

When OTMA sends out a message to the remote IMS through IMS Connects, the tpipe that the message was sent from is set to WAIT_R state. OTMA waits for an ACK/NAK for a specified time based on the CMOACK timeout value. When this timeout occurs, OTMA will reroute the message to the timeout queue and issue DFS3494E message. The CMOACK timeout value can be set in the following ways. These methods are existing functions prior to IMS V12:

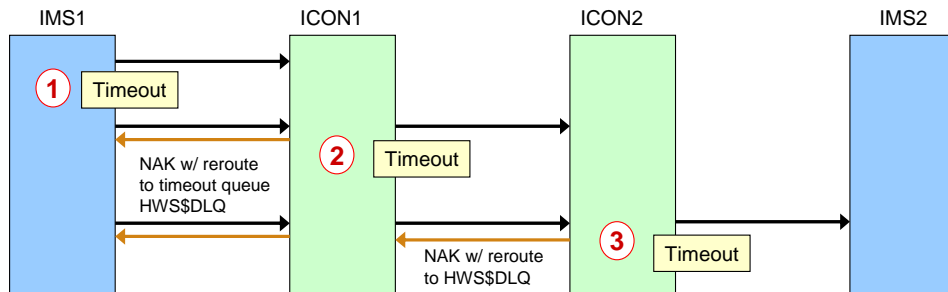
- /STA TMEMBER TIMEOUT command
- ACKTO parameter of the DATASTORE statement in the IMS Connect configuration member.
- T/O parameter in the type M descriptor of the DFSYDTCx proclib member.

When IMS Connect receives an OTMA remote IMS message request, it tries to create a connection to the remote IMS Connect to send the message. If IMS Connect fails to create the connection, it will send a NAK to OTMA to leave the message in front of the queue and then continually retry to connect to the remote IMS Connect every 2 minutes. However, it will only retry connecting to the remote IMS Connect for one of these failures:

- local TCP/IP is not available
- remote TCP/IP is not available
- remote IMS Connect is not available

When the connection to the remote IMS Connect is established, IMS Connect notifies OTMA to start sending remote IMS messages that uses this specific remote IMS Connect RMTIMSCON.

Possible Timeout Scenarios



1. OTMA ACK timeout: message is moved to dead letter queue, e.g., HWS\$DLQ
2. Local IMS Connect ACK timeout: send NAK to local IMS with reroute to dead letter queue
3. Remote IMS Connect ACK timeout: send NAK to local IMS Connect and local IMS with reroute to dead letter queue

(Late Reply ACKs are NAKed by OTMA with RC=2B which is ignored by IMS Connect)

Messages on the dead letter queue can either be dequeued via command, or retrieved by issuing a Resume TPIPE request against the dead letter queue

407

As the visual shows, a timeout can occur in OTMA, or in either of the IMS Connect systems, local or remote, possibly resulting in messages being rerouted to the dead letter queue.

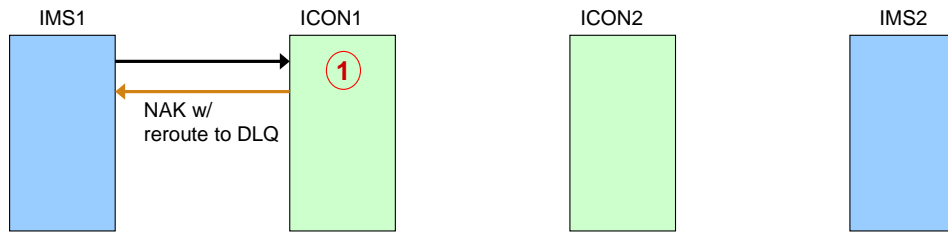
The queue counts for HWS\$DLQ can be viewed by issuing the command `/DISPLAY TMEMBER tmembername TPIPE HWS$DLQ`

Messages can be retrieved by specifying the dead letter queue, e.g., HWS\$DLQ as the alternate client ID on a RESUME TPIPE call. Alternatively, a message can be dequeued from HWS\$DLQ by:

- Stopping HWS\$DLQ: issue the command `/STOP TMEMBER tmembername TPIPE HWS$DLQ`
- Dequeuing the message: issue the command `/DEQUEUE TMEMBER tmembername TPIPE HWS$DLQ PURGE1`
- Starting HWS\$DLQ: issue the command `/START TMEMBER tmembername TPIPE HWS$DLQ`

Any Late reply ACK that is received by OTMA is rejected with a RC=2B which is sent back to IMS Connect and ignored.

Error Scenarios

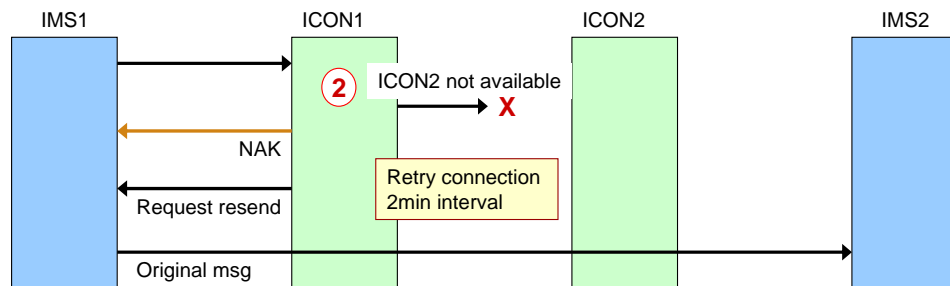


Example 1: Error in local IMS Connect , e.g., RMTIMSCON is not defined or internal error occurs.

- Local IMS Connect, NAKs with a reroute to the dead letter queue HWS\$DLQ
- OTMA creates 6701 log record with ICON's RC/RSN.
- OTMA copies IMS Connect's RC/RSN into the original rerouted message, and routes the message to HWS\$DLQ
- Local IMS Connect issues WTO error message.

Operator or Administrator action: To retrieve the message from the dead letter queue for analysis, invoke previously prepared Resume TPIPE process against the dead letter queue

Error Scenarios ...

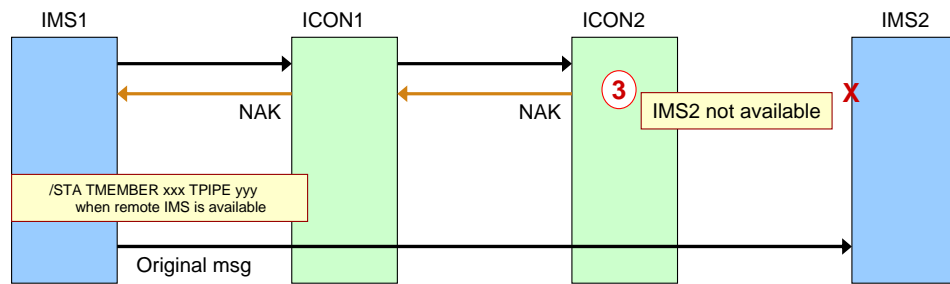


Example 2: Remote IMS Connect (ICON2) is unavailable

- Local IMS Connect sends NAK to local IMS
- OTMA leaves message in the front of the local IMS queue
- OTMA creates 6701 log record with IMS Connect's RC/RSN
- Local IMS Connect retries connection to remote ICON every 2 min.
 - > If successful, local IMS Connect sends protocol to OTMA to resend the message
 - > OTMA sends the original message to the local IMS Connect

Operator or Administrator action: None. Automatic retry and resend.

Error Scenarios ...



Example 3: Remote IMS is not available

- Remote IMS Connect sends NAK to the local IMS Connect which forwards on the NAK to the local IMS
- OTMA leaves the original message in front of queue.
- OTMA creates 6701 log record with IMS Connect's RC/RSN.
- Local IMS Connect issues a WTO error message.

Operator or Administrator action: When the remote IMS becomes available, issue a `/STA TMEMBER TPIPE` on the local IMS – this causes OTMA to resend the message

Considerations, Usage, and Benefits

▪ **Operational Characteristics**

- Each RMTIMSCON statement results in a separate TCB
- Restrictions
 - Cannot be used from IMS in a shared queues environment that is not connected to IMS Connect
 - Super member support only allows up to 8 IMS Connects or 8 IMS Connect datastores in a super member group

▪ **Usage**

- IMS applications: ISRT ALTPCB
- IMS environment: destination descriptor or a DFSYDRU0 exit routine
- IMS Connect: configuration specifications

▪ **Benefits**

- Supports TCP/IP communications to invoke transactions between IMS systems without having to create or maintain a separate gateway solution
 - IMS-provided and supported solution



MSC TCP/IP

MSC TCP/IP

- Support for MSC communications across a TCP/IP network
 - A new physical link MSPLINK TYPE=TCPIP
 - Provides a mechanism to
 - Take advantage of TCP/IP networks
 - Complement or backup existing SNA/VTAM links
 - Take advantage of potentially higher bandwidths
 - Supports operational compatibility with other link types (CTC, MTM, VTAM)
 - Starting, stopping, updating, displaying, and assigning resources
 - Only between IMS 12 systems

Environments that primarily use TCP/IP networks for communications, but require SNA/VTAM to sustain their MSC links, now have the option of also migrating their MSC connections entirely to TCP/IP. A new TCPIP link type has been added to the MSPLINK macro. These new links can be used to replace, complement or backup existing MSC VTAM connections. Operationally, the new support is compatible with other link types, e.g., CTC, MTM, VTAM, using the same command structure.

From an MSC perspective, the operation of TCP/IP physical links and VTAM physical links is very similar. Apart from some differences in system definition and buffer size requirements, the IMS and z/OS components that support each connection type present the most significant operational differences. Both TCP/IP and VTAM physical link types can be used either as your primary physical link type, or as a backup link type in case the other link type fails for any reason.

Depending on various factors, such as network traffic and the distance between the two connected IMS systems, a TCP/IP physical link is likely to provide better performance than a VTAM physical link.

MSC TCP/IP requires no additional requirements over the basic version 12 requirements but is only supported between two version 12 IMS systems.

Highlights

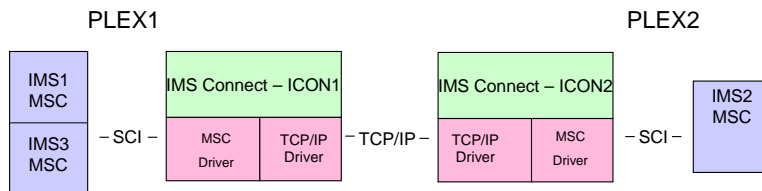
- **MSC TCP/IP leverages IMS Connect and the Common Service Layer**
 - IMS Connect sends/receives messages via the TCP/IP network
 - IMS Connect manages the TCP/IP communications
 - IMS MSC manages the message processing
 - CSL provides the Structured Call Interface (SCI) for communications between IMS components including IMS Connect
 - Each IMS and its local MSC-routing IMS Connect system must be part of the same IMSplex
 - IMSPLEX= plexname parameter in the Common Layer Section of the DFSDFxxx of IMS proclib
 - The Operations Manager (OM) is not required but recommended
 - For type-2 command support

IMS Connect is a key component of the new MSC support since it is IMS Connect that provides the TCP/IP communications capability. IMS Connect manages the TCP/IP socket connections and IMS continues to manage the message processing.

A basic Common Service Layer (CSL) configuration is required to support MSC TCP/IP. This minimum configuration includes the Structure Call Interface (SCI) and SCI address space for communication between IMS components such as IMS to IMS Connect. Although not required, the Operations Manager (OM) is recommended in order to provide type-2 command support. The new MSC support does not impose any change on the existing CSL definition structure in the DFSDFxxx member of IMS proclib. The IMSPLEX parameter value in DFSDFxxx is the same value that is referenced in the setup definitions for MSC.

Highlights ...

- IMS to IMS Connect functionality
 - Isolates TCP/IP from the IMS Control Region
 - Uses the existing IMS Connect TCP/IP support
 - Provides a new MSC driver as well as TCP/IP driver for MSC
 - Supports communication with IMS via the Structured Call Interface (SCI)



415

Each IMS or group of IMS systems and IMS Connect address space are members of an IMSplex. (e.g., PLEX1 and PLEX2). There may be multiple IMS systems within the same IMSplex (i.e. IMS1 and IMS2 in PLEX1) as well as multiple IMS Connects.

The IMS systems and IMS Connects within the same IMSplex use the structure call interface (SCI) to communicate (i.e. send and receive IMS messages). IMS connect uses an IMS driver to communicate to its local IMS. IMS connect uses a TCP/IP driver to communicate to the remote IMS connect via the TCP/IP network.

IMS messages are sent back and forth between the local IMSplex and its associated remote IMSplex, via the IMS connects, and the TCP/IP network

IMS Definition: MSPLINK

- MSPLINK macro defines the physical link (new parameters)

```
PLNK12TA MSPLINK TYPE=TCPIP,NAME=IMS2,LCLICON=ICON1,LCLPLKID=MSC12,
          BUFSIZE=8192,SESSION=2
```

- **TYPE =**
 - Defines the physical link type as TCP/IP
- **NAME =**
 - IMSID of remote IMS (for VTAM links this value is APPLID)
- **LCLICON =**
 - Identifies the local IMS Connect member within the IMSplex
- **LCLPLKID =**
 - The ID of the MSC statement within the local IMS Connect configuration member
 - The associated MSC statement contains the IMS Connect configuration definitions for the physical link.

416

This visual shows the new MSPLINK operands that are required to define a TCP/IP link to IMS. Note that BUFSIZE (4096 to 65535) and SESSION (1 to 675) are existing parameters.

Additionally, the associated MSLINK (logical link) and MSNAME (logical link path) macros remain unchanged. One or more logical links (MSLINKs) are defined and assigned to the MSPLINK to process messages, in the same manner as is done for the other physical link types (i.e. MTM, CTC and VTAM). For TCP/IP links, each TCP/IP logical link that is defined will result in the IMS Connect address space creating a pair (2) of socket connections to communicate when the logical link is started. One socket connection is used to send messages from the local IMS Connect to the remote IMS Connect (via TCP/IP), which sends it to the remote IMS. The other socket connection is used to send messages in the other direction, from the remote IMS Connect to the local IMS Connect (via TCP/IP), which sends it to the local IMS.

The new TCP/IP logical link may coexist with MTM, CTC, or VTAM logical links. There are four new parameters for the TCPIP support

A total of 675 logical links of all types may be defined in each IMS/MSC address space.

IMS Connect Configuration : MSC Statement

- New MSC statement in IMS Connect supports the MSPLINK connection

```
MSC=(LCLPLKID=MSC12,RMTPLKID=MSC21,LCLIMS=IMS1,RMTIMS=IMS2,
      IMSPLEX=(MEMBER=ICON1,TMEMBER=PLEX1),RMTIMSCON=CONNECT2)
```

- **LCLPLKID =**
 - Identifies the name of this MSC statement for the local IMS
- **RMTPLKID =**
 - Identifies the name of the matching MSC statement in the remote IMS
- **LCLIMS =**
 - Identifies the local IMS IMSID
- **RMTIMS =**
 - Identifies the remote IMS IMSID
- **IMSPLEX =** identifies the plexname information for SCI communications
 - MEMBER= IMS Connect's name,
 - TMEMBER = the IMS plexname as also defined in DFSCGxxx
- **RMTIMSCON (must be different than the one for async IMS-IMS support)**
 - Specifies the associated remote IMS Connect identifier statement

417

The MSC statement defines a one-way send path for an MSC physical link between a local IMS system to a remote IMS system. The physical link send path uses a TCP/IP socket connection established by this IMS Connect instance with a remote IMS Connect instance. The connection used by the physical link must be defined to this IMS Connect instance by an RMTIMSCON statement. Correspondingly, the remote IMS Connect instance must also define corresponding RMTIMSCON and MSC statements. Each MSC logical link assigned to a physical link requires two socket connections: one established by the local IMS Connect to send transaction messages and one established by the remote IMS Connect instance to return the reply messages.

- **LCLPLKID =** The name of this MSC statement which associates the subsequent definitions with the physical link definitions in an MSPLINK macro statement on the local IMS system.
- **RMTPLKID=** The remote name of the MSC statement as specified on the LCLPLKID parameters of both the MSC statement of the remote IMS Connect and the MSPLINK macro of the remote IMS system.
- **LCLIMS=** IMS ID of the local IMS system as registered with SCI in the IMSplex.
- **RMTIMS=** IMS ID of the remote IMS system as registered with SCI in the remote IMSplex.
- **IMSPLEX=()** Enables IMSplex communications between IMS Connect and MSC.
- **MEMBER=** A name that identifies IMS Connect in the IMSplex. For MSC communications, this name must match the name specified on the LCLICON parameter of the MSPLINK macro definition of the local IMS system.
- **TMEMBER=** The name of the IMSplex that IMS Connect is joining, as specified on the IMSPLEX(NAME=) statement of the CSLSIxxx PROCLIB member of the SCI instance that is managing communications between IMS Connect and the IMSplex. Restriction: The name specified on the TMEMBER parameter cannot be the same as a name specified on the ID parameter of any DATASTORE statement.
- **RMTIMSCON=** The remote IMS Connect connection to use for MSC messages. The value must match the value of the ID parameter of one of the RMTIMSCON statements specified in the local IMS Connect configuration.

IMS Connect Configuration: RMTIMSCON Statement

- New RMTIMSCON statement identifies the remote IMS Connect partner information

```
RMTIMSCON=(ID=CONNECT2, HOSTNAME=ICON2.IBM.COM, PORT=9999, RESVSOC=2)
```

- **ID=**
 - Identifier for a remote IMS Connect connection
 - **Different** than the one used for asynchronous IMS-IMS support
- **IPADDR=**
 - specifies the remote IMS Connect IP address (IPADDR or HOSTNAME required, but not both)
- **HOSTNAME=**
 - specifies the remote IMS Connect hostname
- **PORT=** specifies the remote IMS Connect port
- **RESVSOC=** number of sockets to reserve
 - Match with MSC parallel sessions across the link definitions
 - Coordinate with overall MAXSOC ★

418

The new IMS Connect RMTIMSCON configuration statement is used to define how the two IMS Connect's are to communicate with each other. At a minimum, you must define:

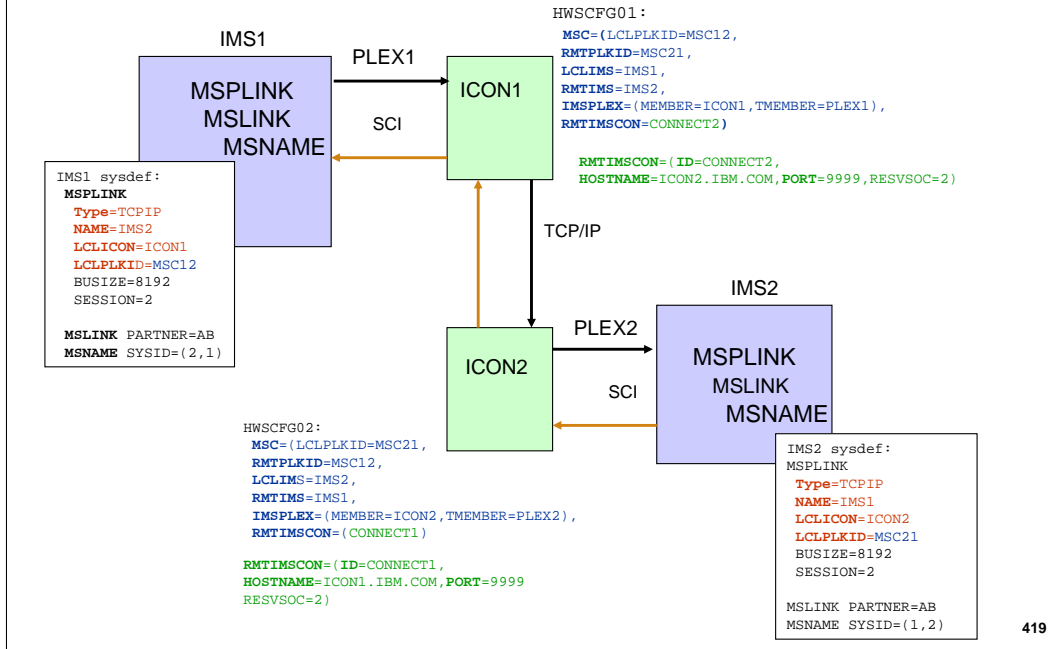
- **ID** - Uniquely identifies this connection. The ID value must also be specified on the RMTIMSCON keyword on the IMS Connect MSC configuration statement.
- **IPADDR or HOSTNAME** - Identifies the internet address of the receiving IMS Connect instance
- **PORT** – Port for the receiving IMS Connect instance which must match the value specified on either the PORT or PORTID keyword in the remote IMS Connect TCPIP configuration statement.

Optionally, the following additional parameters can be specified:

- **APPL and USERID** – If the partner IMS Connect is configured to authenticate RACF PassTickets from incoming MSC TCP/IP connections, you must specify both the APPL and USERID parameters.
- **AUTOCONN** (defaults to No). MSC links do not support AUTOCONN=Y. If an MSC statement references a RMTIMSCON statement in which AUTOCONN=Y is specified, AUTOCONN=N is used and a warning message is issued.
- **PERSISTENT** - Not required for MSC but code PERSISTENT=Y to avoid receiving a warning message when IMS Connect changes the default value of N during each start up.
- **RESVSOC** - Reserves send sockets for the MSC logical links that use this connection. The corresponding receive sockets that are required by MSC logical links are not reserved. The number specified in this parameter cannot exceed 50% of the value specified on the MAXSOC parameter in the TCPIP configuration statement.

A note on the socket specifications: If, for example, the MSPLINK macro specifies SESSION=2 meaning that the link can support two parallel sessions, i.e., two logical links, then the recommended minimum in the RMTIMSCON statement should be RESVSOC=2 (which will be used as sending sockets). Additionally, two receiving sockets will be allocated from the pool of sockets that are part of the current MAXSOC specification in IMS Connect. The current MAXSOC specification, already used for all other types of interactions, may need to be increased if the value specified is close to being reached.

Example: Configuration Definitions

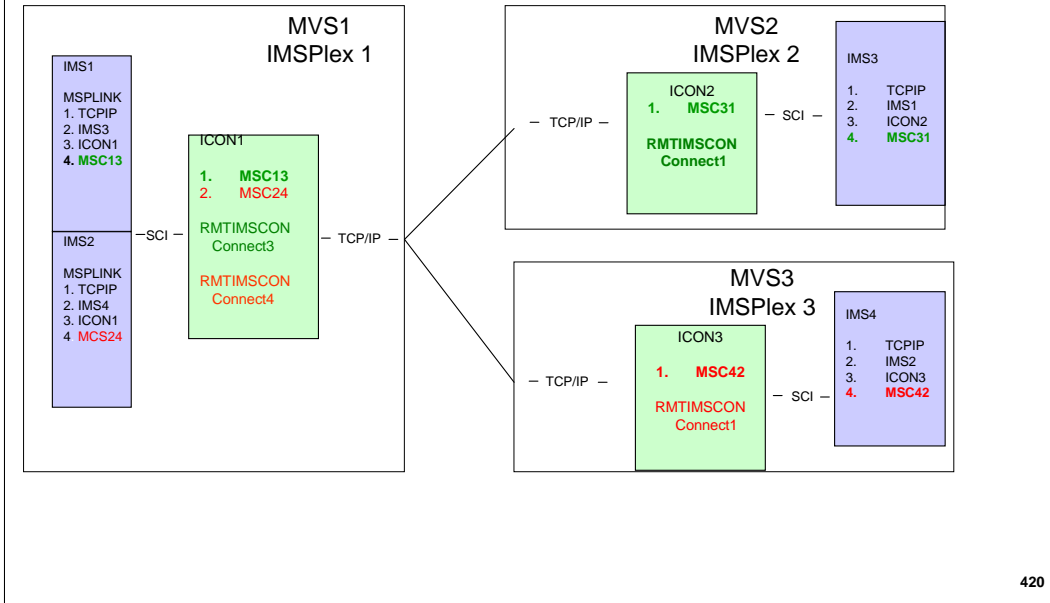


This visual shows a simple MSC TCP/IP configuration along with the appropriate IMS system definition and IMS Connect configuration member statements.

In IMS, the physical link is defined by the MSPLINK macro followed by the logical link (MSLINK) and the link SYSID path (MSNAME). The MSLINK and MSNAME macro specifications remain unchanged.

In the IMS Connect configuration specification, the physical link path is defined via an MSC statement. The RMTIMSCON configuration statement connects the two IMS connects together via the TCP/IP network.

Example: 3-way IMSplex configuration



420

This figure shows four IMS systems in three separate PLEXes. Each physical path between two IMS systems has an MSPLINK macro and a corresponding MSC configuration statement in the local IMS connect to define the path within the PLEX.

IMS Considerations

- **Use of TCP/IP MSC links**
 - Requires IMSplex architecture
 - SCI interface to IMS Connect
 - Relies on Type2 command support for **optimally** managing the environment
 - Operations Manager
 - Only uses bandwidth (BW) mode
 - BW mode was introduced in IMS 10 as optional for CTC, MTM, and VTAM links
- **Installing at least one MSC link of any type also installs MSC**
 - If MSC is defined in one IMS within a shared queues IMSplex, it must also be installed in the other IMS systems
 - No change from previous releases

421

There are no new architectural changes needed for MSC TCP/IP however there are some first time exploitations of the existing IMS architecture. IMSplex is needed for the SCI interface and operations management (OM) support for the first time. Type2 command support (Query and Update) are recommended. Also, MSC TCP/IP only supports bandwidth (BW) mode.

As is the case for any MSC link type in a shared queues IMSplex, if MSC is defined in one IMS within a shared queues group, then it must be installed in all the IMS systems in that same group. This means that each IMS must have at least one MSPLINK, MSLINK and MSNAME set of macros even if they are not active and not used in the other IMS systems. The IMS systems in the shared queues group exchange SYSIDs and MSNAMEs at initialization. During this SYSID exchange, each IMS system creates dynamic MSNAMEs for any MSNAMEs that are defined in other IMS systems that it does not define itself. The dynamic MSNAMEs result in paths to all of the IMS systems. The SYSIDs are merged to create a common SYSID routing table. The table is the same in each IMS. Therefore, any local SYSIDs are local in all IMS systems and override any remote or undefined SYSIDs. Remote SYSIDs override only undefined SYSIDs.

Operational Considerations

■ Type-1 Commands

- /DIS ASSIGNMENT MSPLINK command
 - Displays: Type = TCPIP and Name = the remote IMSID.
- Other MSC type-1 commands, no changes (i.e. /RST link, /PSTOP link, etc)

■ Type-2 Commands

- Query MSPLINK shows all the new MSPLINK parameters
 - Type, Name, LCLICON, LCLPLKID
- Other MSC type-2 commands, no changes
- **Enhanced** IMS Connect Type-2 query and update command support
 - Documented in the IMS Connect Type-2 Commands presentation
- **New** IMS Connect WTOR command support (ViewMSC, StartMSC, etc)
 - Documented in the Reference information at the end of this section

It is possible to operate TCP/IP links with type-1 command support (/RSTART, /PSTOP, /DISPLAY, etc.) but the functionality is limited. For example, the new values for LCLICON and LCLPLKID are not displayed by the type-1 /DISPLAY command.

To fully manage the new MSC TCPIP links, type-2 command (QUERY and UPDATE) support can be used. Additionally, IMS version 12 introduces new IMS Connect type-2 commands and also provides updates to the existing IMS Connect command support (refer to the following sections of the class: IMS Connect Type-2 Command Support and IMS to IMS TCP/IP Communications).

Operational Considerations...

Example: Type-1 Commands

```
/DIS ASMT MSPLINK ALL
```

Response:

LINK	PLINK	TYPE	ADDR	MAXSESS	NODE
3	PLNK12M	MTM	****		
2	PLNK12C	CTC	0511		
1	PLNK12V	VTAM	02000006	3	L6APPL1
4	PLNK12TA	TCPIP	****	2	IMS2

Example: Type-2 Commands

```
QUERY MSPLINK TYPE(TCPIP) SHOW(ALL)
```

TSO SPOC output:

MSPLink	MbrName	CC	Type	RmtImS	LclImSCon	LclPlkID	LclStat
PLNK31TA	SYS3	0	TCPIP	IMS1	ICON3	MSC31	

Type-2 command	Equivalent type-1 IMS Connect command
UPDATE MSPLINK NAME(msplinkname) SET(ICONPLKID(<i>iconplkid</i>))	No similar type-1 IMS command exists.
UPDATE MSPLINK NAME(msplinkname) SET(IMSCON(<i>imscconname</i>))	No similar type-1 IMS command exists.
UPDATE MSPLINK NAME(msplinkname) SET(RMTIMS(<i>rtims</i>))	No similar type-1 IMS command exists.

Operational Considerations...

▪ **MSC TCP/IP Error Messages**

- DFS3175E An MSC CSLSCMSG request failed
- DFS3176E ICON error message received
- DFS3177E MSC detected an error
- DFS3178E MSC detected an internal logic error

▪ **Security**

- IMS and IMS Connect both register with SCI at initialization
 - Using a CSLREG request which issues a RACROUTE REQUEST=AUTH
 - The associated address space userid must have UPDATE authority
 - Note: CSL SCI Security is documented in the IMS System Administration Guide

424

DFS3175E, DFS3176E, DFS3177E, and DFS3178E are MSC TCP/IP error messages. Each message applies to an error detected at a different point in the MSLINK processing as identified by the message text immediately following the message number.

- Messages DFS3175E and DFS3176E are errors detected by the IMSplex SCI and IMS Connect members respectively. They each have a unique RETCODE and RSNCODE table.
- Messages DFS3177E and DFS3178E are errors detected by IMS within the MSC TCP/IP Device Dependent Module (DDM). These messages share a common RETCODE/RSNCODE table.
- For all messages, the MODULE name is the routine that detected the error condition and issued the message.

All the above messages share a LOSTSESS table. The cause of a lost session can include any of the following possible values: NONE, ICONABE, ICONSTOP SCIABE, SCISTOP, or FORCESTOP.

Security for using IMS and IMS connect to communicate via MSC is established at initialization when IMS and IMS connect each register with SCI by issuing the CSLSCREG request. For each request, SCI issues a RACROUTE REQUEST=AUTH call to RACF to see if the address space has the authority to register. The address space (IMS or IMS Connect) userid is used as the registration ID and must be defined to RACF in the RACF® FACILITY class profile for the IMSplex with at least UPDATE authority.

Operational Considerations...

- **IMS shutdown or abend**
 - For *MSC connections*
 - IMS cleans up links for IMS shutdown and sends link shutdown directive to IMS Connect
 - IMS Connect disconnects the send and receive clients associated with the MSC physical links for that IMS
 - IMS Connect's MSC status is set to DISCONNECT
 - IMS Connect action for *OTMA connections*
 - For connections in CONN state
 - On IMS shutdown and abend, no action for IMS Connect
 - For connections in RECV state
 - When ACK/NAK is received
 - IMS Connect cannot deliver it so it throws the message away and issues an error message

Operational Considerations...

- **IMS shutdown or abend ...**
 - OTMA action for TPIPEs in WAIT_R state
 - On IMS shutdown
 - When IMS processes a warm restart, OTMA places messages that are in the tpipe into a timeout queue
 - On IMS abend
 - When IMS ERE restarts, OTMA places messages that are in the TPIPE into a timeout queue

- **SCI shutdown or abend**
 - IMS Connect disconnects the send and receive clients for MSC physical links that use the IMSplex
 - IMS Connect's MSC status is set to DISCONNECT
 - IMS cleans up MSC physical links.
 - IMS issues DFS3177E message.
 - MSC link state PSTOPPED IDLE ERE

After a normal IMS shutdown with TPIPEs in WAIT_R state, when IMS warm starts, OTMA places the messages in those TPIPEs into the timeout queue. After an abnormal IMS termination with TPIPEs in WAIT_R state, when IMS emergency restarts, OTMA places the messages in those TPIPEs into the timeout queue.

On normal and abnormal SCI termination, IMS Connect terminates all MSC logical links on the MSC physical link for that IMS. All send and receive sockets used by the MSC logical links are closed as well. The MSC physical link status in IMS Connect is set to DISCONNECT state.

Operational Considerations

- **IMS Connect shutdown or abend**
 - For MSC connections
 - IMS Connect disconnects send and receive clients for all MSC physical links
 - On shutdown, IMS Connect sends MSC level shutdown directive to MSC
 - On abend, IMS gets notified by SCI
 - IMS MSC cleans up physical link
 - Issues DFS3176E or DFS3177E message
 - For OTMA connections
 - For connections in CONN state, IMS Connect closes connections
 - For connections in RECV state
 - On IMS Connect force shutdown (CLOSEHWS FORCE), a NAK is sent to OTMA with reroute to timeout queue.
 - On IMS Connect shutdown (CLOSEHWS), IMS Connect waits for the ACK/NAK indefinitely.
 - On IMS Connect abend, OTMA reroutes messages to the timeout queue

On normal IMS Connect termination, IMS Connect terminates all send and receive sockets and notifies IMS that it is shutting down.

When IMS Connect terminates through the CLOSEHWS command, it closes all send socket connections that are in CONN state and quiesces all send socket connections in RECV state. IMS Connect will wait indefinitely for the ACK/NAK on the send socket connections that are in RECV state. Once all RECV state connections receive the ACK/NAK, IMS Connect proceeds to terminate IMS Connect.

When IMS Connect terminates through the CLOSEHWS FORCE command, it closes all send socket connections. Additionally, for send socket connections in RECV state waiting for ACK/NAK, IMS Connect sends NAKs to OTMA with a reroute to the timeout queue.

When IMS Connect terminates abnormally, messages waiting for an ACK/NAK will be rerouted to the timeout queue.

Setup Scenario and Switching from VTAM to TCP/IP

- All IMS systems must be at V12 to use MSC TCP/IP
 - Consider a VTAM to TCP/IP migration and TCP/IP to VTAM fallback scenario
- Example of switching an MSC link from VTAM to TCP/IP
 1. Existing VTAM link in stage 1 sysgen

```
PLNK1  MSPLINK  TYPE=VTAM,NAME=L6APPL1,SESSION=1,BUFSIZE=8192
LNK12  MSLINK  PARTNER=AK,MSPLINK=PLNK1,OPTIONS=FORCSESS
MSN12  MSNAME  SYSID=(20,10)
```

2. Add a TCP/IP physical link to stage 1

```
PLNK2  MSPLINK  TYPE=TCP,NAME=IMS2,LCLICON=ICON1,
        LCLPLKID=MSC12,SESSION=1,BUFSIZE=8192
```

The next two visuals shows an example of the steps needed to create a TCP/IP physical link for IMS and IMS Connect, and assigning existing VTAM logical links over to it for migration and testing purposes.

Setup Scenario ...

3. Add the MSC and RMTIMSCON configuration statements to the local IMS Connect configuration proclib member

```
MSC=(LCLPLKID=MSC12,RMTPLKID=MSC21,LCLIMS=IMS1,RMTIMS=IMS2,  
      IMSPLEX=(MEMBER=HWS1,TMEMBER=PLEX1),RMTIMSCON=CONNECT2)  
RMTIMSCON=(ID=CONNECT2, HOSTNAME=ICON2.IBM.COM,PORT=9999,RSVSOC=2)
```

4. Add an SCI and OM (optionally) if not already defined
5. Stop the physical and logical links and move the logical link from VTAM to TCP/IP

(Repeat steps 1 through 5 for the remote IMS)

6. Restart the links

Falling Back from TCP/IP to VTAM

- If a fallback is needed
 1. Stop the physical TCP/IP links and associated logical links
 2. Move the logical links from the TCP/IP physical links back to the MSC VTAM physical links
 3. Restart the MSC VTAM links and logical links

MSC TCP/IP Error Handling

- Error handling technique for errors detected by MSC TCP/IP is to:
 - Issue an error message(s) to the MTO when then error is detected
 - DFS3175E AN MSC CSLSCMSG REQUEST FAILED
 - DFS3176E ICON ERROR MESSAGE RECEIVED
 - DFS3177E MSC DETECTED AN ERROR
 - DFS3178E MSC DETECTED AN INTERNAL LOGIC ERROR
 - Snap 6701-MSS1, MSS2, and MSS3 log records to the IMS log
 - Terminate the link connection (shutdown the link)
 - Set the link status to PSTOPPED IDLE on both sides of the link
 - send a shutdown immediate message to the partner
 - If the partner cannot be notified, it will remain in active status
 - /PSTOP LINK FORCE or UPDATE MSLINK STOP(COMM) OPTION FORCE command can be issued to disconnect the partner side of the link and place it in PSTOPPED IDLE status.

431

This visual describes the actions that occur when the MSC TCP/IP support detects an error condition.

This is true also when certain STOP commands are issued in IMS Connect which cause termination of the connection. In these cases the error message that is issued is a DFS3176E. The associated return and reason codes differ based on the command that was issued.

```
STOPLINK = RETCODE = x'00000018'  RSNCODE = x'00006400'
STOPMSC  = RETCODE = x'00000018'  RSNCODE = x'00006404'
STOPRMT  = RETCODE = x'00000018'  RSNCODE = x'00006408'
STOPIP   = RETCODE = x'00000018'  RSNCODE = x'0000640C'
```

In the situations when either SCI or IMS Connect terminate or abend, a notify exit in the local IMS is driven. This exit notifies the affected active logical links with a LOSTSESS code. This code is stored as a one byte code in the CLBLOST byte of the Link LLB control block and the link is posted to handle the error. Error message DFS3177E is issued for each active MSC link affected (i.e. active logical links assigned to the affected MSPLINK are notified). The RETCODE, RSNCODE, and LOSTSESS for each condition are included in the error message as well as in the snapped6701-MSS1, MSS2, and MSS3 log records. The cause of a lost session, can include any of the following possible values: NONE, ICONABE, ICONSTOP SCIABE, SCISTOP, or FORCESTOP.

Performance Considerations (sample performance test)

- **Benchmark both VTAM and TCP/IP**
 - Query the MSC statistics for comparison numbers
 1. **Queue up 10,000 messages to the link**
 2. **UPDATE MSLINK NAME(name) START(STATISTICS) OPTION (NORESET, CHKPT)**
 - IMS will not reset the query statistics at each IMS checkpoint
 3. **UPDATE MSLINK NAME(name) START(STATISTICS) OPTION (RESET)**
 - IMS will reset the link statistics to zero and set the start time to the current time
 - (Only for VTAM): 4. **UPDATE MSLINK NAME(name) SET(BANDWIDTH(ON))**
 - IMS will set bandwidth on for VTAM (TCP/IP always uses BANDWIDTH)
 5. **UPDATE MSLINK NAME(name) START(COMM)**
 - IMS will start the link and let all the messages process (Msgs sent one way)
 6. **QUERY MSLINK NAME(name) SHOW(STATISTICS)**
 - Look at Send_Msg_Time - Interval of time between the first and last message sent. Used for benchmark testing of a block of messages. Can also use Rec_Msg_Time.
 7. **Repeat steps 1-6 with larger buffer sizes**

These are the steps needed to run a benchmark test of a MSC links bandwidth performance. This can be done for a VTAM and a TCP/IP link to compare the performance between the two links.

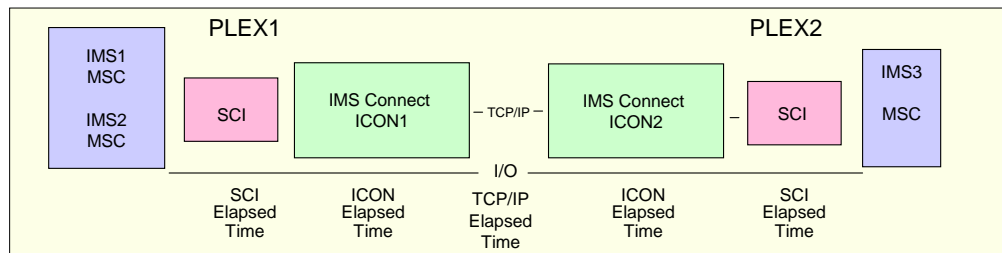
Performance Considerations

- Expect TCP/IP Bandwidth to improve over VTAM
- Verify results using the QUERY MSLINK SHOW (STATISTICS)
Current statistics support includes:
 - General statistics
 - Total ITASK dispatch count, Total processing time. Hi, Lo, Avg processing times per dispatch, check write counts and rates
 - Send statistics
 - Total messages sent, total byte count sent, Hi, Lo, Avg send message sizes, queue manager get (QGET) counts (GU, GN, DEQ calls,), and Hi, Lo, Avg QGET times, and Hi, Lo, Send I/O times
 - Receive statistics
 - Total messages received, total byte count received, Hi, Lo, Avg receive message sizes, QMGR ISRT (QPUT) counts (ISRT, and ENQ calls), Hi, Lo, Avg QPUT and times, and Hi, Lo, Receive I/O times

The type-2 command QUERY MSLINK NAME(linkname) SHOW (STATISTICS) can be used during a benchmark run of messages across a VTAM and TCP/IP link.

Enhanced I/O Statistics

- Enhanced Query MSLINK Show (Statistics)
 - Added elapsed times for SCIs, ICONs, and TCPIP
 - I/O statistics reported in SCI, ICON, and TCP/IP elapsed times
 - SendIO_Times
 - Tot_SCI_SendIO_Time, Tot_ICON_SendIO_Time, Tot_TCPIP_SendIO_Time ...
 - Hi_SCI_SendIO_Time, Hi_ICON_SendIO_Time, Hi_TCPIP_SendIO_Time ...
 - Low_SCI_SendIO_Time, Low_ICON_SendIO_Time, Low_TCPIP_SendIO_Time ...
 - ReceiveIO_Times
 - Tot_SCI_ReclO_Time, Tot_ICON_ReclO_Time, Tot_TCPIP_ReclO_Time...
 - Hi_SCI_ReclO_Time, Hi_ICON_ReclO_Time, Hi_TCPIP_ReclO_Time...
 - Low_SCI_ReclO_Time, Low_ICON_ReclO_Time, Low_TCPIP_ReclO_Time...



434

TCP/IP statistics have been enhanced to give more detailed I/O statistics and to provide a breakdown of elapsed times for SCI, IMS Connect and TCP/IP. Each side of the link will display the enhanced statistics for its associated send socket request. Statistics appear in the order of the Send Flow: Local_SCI, Local_ICON, TCPIP, RMT_ICON, RMT_SCI.

Note that the TCPIP elapsed time value requires use of an external time reference (ETR) to be accurate. IMS cannot detect if ETR is being used and therefore the value will be zero if the calculation result is a negative number.

Query MSLINK Enhanced I/O Statistics

QUERY MSLINK NAME(LNK12T01) SHOW(ALL)

MSLink	MSLink#	MbrName	CC	MSPLink	PID	RecdCnt	SentCnt	BufSize	Bandwidth	Affin
LNK12T01	22	IMS1	0	PLNK12TA	TA	0	2	65536	ON	

QUERY MSLINK NAME(LNK12T01) SHOW(STATISTICS)

MSLink	MSLink#	MbrName	CC	Option	Start_Time	Tot_Dispatch	Tot_Proc_Time	Hi_Proc_Time
LNK12T01	22	IMS1	0	RESET,CHKPT	2010.209 11:52:59.39	16	0.020195	0.004190

MSLink	MSLink#	MbrName	Low_Proc_Time	Avg_Proc_Time	Chkw_CT	ChkwIO_CT	Chkw_Rate	Tot_Send_CT	Tot_Msg_Send_CT
LNK12T01	22	IMS1	0.000235	0.001262	5	2	0.000000	5	2

MSLink	MSLink#	MbrName	Tot_MsgByte_Send_CT	Tot_Byte_Send_CT	Hi_Msg_Send_SZ	Low_Msg_Send_SZ	Avg_Msg_Send_SZ
LNK12T01	22	IMS1	1,040	2,816	538	502	520

MSLink	MSLink#	MbrName	Send_Msg_Time	Send_MsgCT_Rate	Send_MsgByte_Rate	Tot_Qget_CT	Tot_Qget_Time	Hi_Qget_Time
LNK12T01	22	IMS1	47.917786	30.472011	15.845.445957	6	0.000129	0.000043

MSLink	MSLink#	MbrName	Low_Qget_Time	Avg_Qget_Time	Tot_SendIO_Time	Hi_SendIO_Time	Low_SendIO_Time	SendIO_Req_Rate
LNK12T01	22	IMS1	0.000004	0.000021	0.052853	0.016744	0.011833	94.602009

MSLink	MSLink#	MbrName	SendIO_Byte_Rate	Tot_Loc_SCI_SendIO_Time	Hi_Loc_SCI_SendIO_Time	Low_Loc_SCI_SendIO_Time
LNK12T01	22	IMS1	53,279.851664	0.001503	0.000988	0.000103



Query MSLINK Enhanced I/O Statistics ...

MSLink	MSLink#	MbrName	SendIO_Byte_Rate	Tot_Loc_SCI_SendIO_Time	Hi_Loc_SCI_SendIO_Time	Low_Loc_SCI_SendIO_Time		
LNK12T01	22	IMS1	53,279.851664	0.001503	0.000988	0.000103		
MSLink	MSLink#	MbrName	Tot_Loc_ICON_SendIO_Time	Hi_Loc_ICON_SendIO_Time	Low_Loc_ICON_SendIO_Time			
LNK12T01	22	IMS1	0.023818	0.008290	0.004861			
MSLink	MSLink#	MbrName	Tot_TCPIP_SendIO_Time	Hi_TCPIP_SendIO_Time	Low_TCPIP_SendIO_Time	Tot_Rmt_ICON_SendIO_Time		
LNK12T01	22	IMS1	0.003279	0.001210	0.000506	0.019013		
MSLink	MSLink#	MbrName	Hi_Rmt_ICON_SendIO_Time	Low_Rmt_ICON_SendIO_Time	Tot_Rmt_SCI_SendIO_Time			
LNK12T01	22	IMS1	0.005403	0.004312	0.019013			
MSLink	MSLink#	MbrName	Hi_Rmt_SCI_SendIO_Time	Low_Rmt_SCI_SendIO_Time	Tot_Rec_CT	Tot_Msg_Rec_CT		
LNK12T01	22	IMS1	0.005403	0.004312	5	3		
MSLink	MSLink#	MbrName	Tot_MsgByte_Rec_CT	Tot_Byte_Rec_CT	Hi_Msg_Rec_SZ	Low_Msg_Rec_SZ	Avg_Msg_Rec_SZ	
LNK12T01	22	IMS1	1,595	3,419	549	502	531	
MSLink	MSLink#	MbrName	Rec_Msg_Time	Tot_Qput_CT	Tot_Qput_Time	Hi_Qput_Time	Low_Qput_Time	Avg_Qput_Time
LNK12T01	22	IMS1	235.933123	0	0.000000	0.000000	0.000000	0.000000
MSLink	MSLink#	MbrName	Tot_RecIO_Time	Hi_RecIO_Time	Low_RecIO_Time	RecIO_Req_Rate	RecIO_Byte_Rate	
LNK12T01	22	IMS1	0.053281	0.014511	0.012323	93.842082	64,169.216043	

And also some information regarding **UK54729(V10) UK54732 (V11) UK54730 (V12)**
 Added fields **Send_Msgct_Rate, Send_MsgByte_Rate** to compliment **Send_Msg_Time**
 to **QUERY MSLINK NAME(XXX) SHOW(STATISTICS)** for

- VTAM Link

```
cmd input. . . . . : QUERY MSLINK NAME(LLK78V26) SHOW(STATISTICS)
```

MSLink	MSLink#	MbrName	CC	Option	Start_Time	Tot_Displ_CT	Tot_Proc_Time	Hi_Proc_Time
LLK78V26	50	IMM7	0	NORESET,CHKPT	2010.020 14:54:25.39	1,893	0.865719	0.003529
Low_Proc_Time	Avg_Proc_Time	Chkw_CT	ChkwIO_CT	Chkw_Rate	Tot_Send_CT	Tot_Msg_Send_CT	Tot_MsgByte_Send_CT	
0.000001	0.000456	938	938	0.000000	938	9,999	5,639,436	
Tot_Byte_Send_CT	Hi_Msg_Send_SZ	Low_Msg_Send_SZ	Avg_Msg_Send_SZ	Send_Msg_Time	Send_Msgct_Rate	Send_MsgByte_Rate		
7,303,426	564	564	564	120.512383	82.974488	46,797.611290		
Tot_Qget_CT	Tot_Qget_Time	Hi_Qget_Time	Low_Qget_Time	Avg_Qget_Time	Tot_SendIO_Time	Hi_SendIO_Time	Low_SendIO_Time	
19,999	0.123673	0.001091	0.000000	0.000006	103.628331	22.389010	0.001128	
SendIO_Req_Rate	SendIO_Byte_Rate	Tot_Rec_CT	Tot_Msg_Rec_CT	Tot_MsgByte_Rec_CT	Tot_Byte_Rec_CT	Hi_Msg_Rec_SZ		
9.051578	70,477.116918	938	0	0	1,154,038	0		
Low_Msg_Rec_SZ	Avg_Msg_Rec_SZ	Rec_Msg_Time	Tot_Qput_CT	Tot_Qput_Time	Hi_Qput_Time	Low_Qput_Time	Avg_Qput_Time	
0	0	0.000000	0	0.000000	0.000000	0.000000	0.000000	
Tot_RecIO_Time	Hi_RecIO_Time	Low_RecIO_Time	RecIO_Req_Rate	RecIO_Byte_Rate				
8.192409	1.811653	0.000105	114.496236	140,866.746276				

437

This is an example of a benchmark test run for APARs PK85645/PM06331 and associated PTFs UK54729(V10), UK54732/UK54733 (V11) UK54730 (V12) which added two new fields to MSC statistics. These fields do not require a specific benchmark set of messages such as field **Send_Msg_Time** requires. Consequently, they can be used dynamically during normal link processing.

Note: **Send_Msg_Time** is the total elapsed time from the GU of the first message sent, to the GU of the last message sent.

Send_MsgByte_Rate - The send message byte rate which displays the MSC link's message bytes sent per second. Send time per each message sent is the time from get unique to get the message to send, to dequeue the message when the response is received. Link idle time in between sending messages is ignored.

Send_MsgCT_Rate - The send message count rate which displays the MSC link's Messages sent per second. Send time per each message sent is the time from get unique to get the message to send, to dequeue the message when the response is received. Link idle time in between sending messages is ignored.

Generic MSC name Support

TCP/IP Generic name support (similar to MSC VTAM VGR)

- Allows IMSs in a shared queues IMSplex, or XRF active/alternate pair to be represented by a common (generic) IMSID
 - Systems outside the IMSplex can use this generic IMSID (GENIMSID) to establish MSC link sessions with the IMSplex
 - Instead of using the IMSID of a specific IMS system in the IMSplex
- Simplifies moving MSC link sessions between IMS systems within the IMSplex
 - No change to the remote IMSID name on the NAME= parameter in the MSPLINK definition for each IMS system outside the IMSplex
- Works similarly to the MSC VTAM generic resources (VGR) support
 - And can coexist with VGR

439

A generic name support for TCP/IP, similar to that of MSC VTAM generic support is another enhancement to the TCP/IP support for MSC. With this capability, MSC IMS systems outside of an IMSplex can use a generic IMSID (GENIMSID) to establish MSC link sessions with the IMSplex instead of using the IMSID of a specific IMS system in the IMSplex.

Additionally, with the use of a TCP/IP GENIMSID, MSC link sessions can be moved between IMS systems within the IMSplex without having to change the remote IMSID name on the NAME= parameter in the MSPLINK stage 1 system definition macro for each IMS system outside the IMSplex.

Generic IMSID name support works similarly to MSC VTAM generic resources (VGR). The difference is VTAM VGR uses a common generic APPLID name and TCP/IP uses a common generic IMSID name.

TCP/IP Generic name support (similar to MSC VTAM VGR)

■ Implementation

– IMS

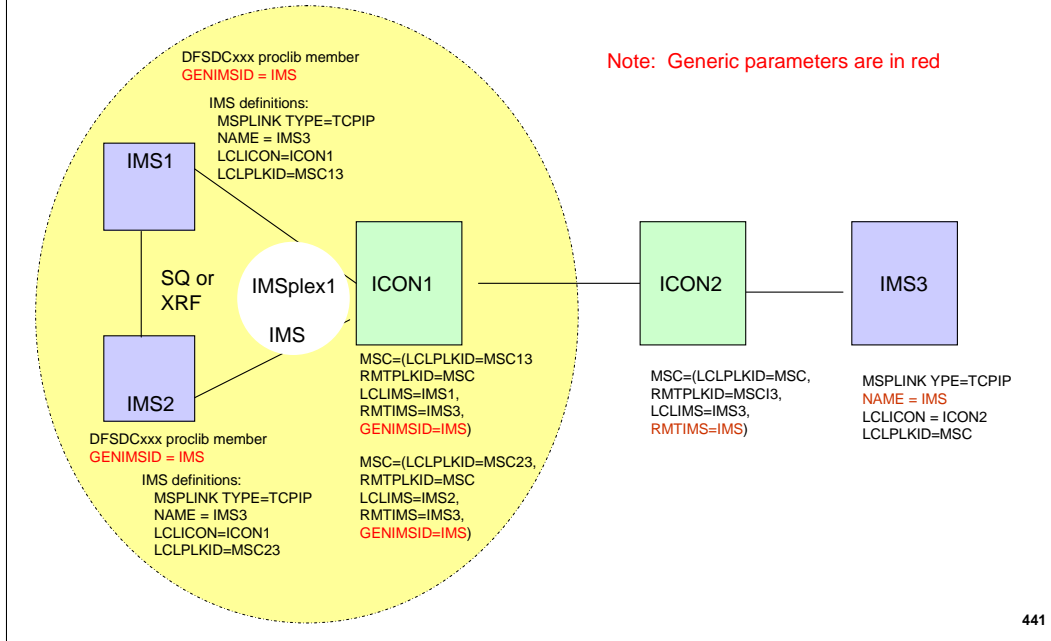
- Code **GENIMSID=generic_IMSID** on IMS DFSDCxxx proclib member for each IMS in the IMSplex.
- In the remote IMS's MSPLINK macro, use the *generic_IMSID* name as the IMSID in the NAME= parameter.

– IMS Connect

- Code **GENIMSID=generic_IMSID** in each MSC statement of configuration member.

A generic name support for TCP/IP, similar to that of MSC VTAM generic support is another enhancement to the TCP/IP support for MSC. To invoke the function, a new startup parameter that specifies the generic IMSID via GENIMSID = imsid. The corresponding support in IMS Connect can be requested in the configuration member MSC statement with a similar specification of GENIMSID = imsid.

TCP/IP Generic name Example: IMS1/IMS2 Generic Name = IMS



This is a pictorial view showing the Generic Name support for Shared queues (SQ) and XRF.

Affinities

- Characteristics (similar to affinities in the MSC VGR environment)
 - All the TCP/IP parallel sessions have affinity to the same IMS in the IMSplex
 - Affinity is removed when the last TCP/IP parallel session terminates normally
 - Affinities and GENIMSID can be displayed
 - /DIS ACTIVE and QUERY MSC commands display the GENIMSID
 - /DIS LINK and QUERY MSLINK commands display the affinity status

When using the generic IMSID name support, all the TCP/IP parallel sessions between the same MSPLINK partners, must be between the same pair of IMS systems. The first link that is established between partners, establishes the IMS pairs. On the IMSplex side, this link, and all the other parallel sessions, are said to have affinity to this IMS system and AFFIN will be displayed as a status on the MSLINKs (e.g., through the /DISPLAY LINK or QUERY MSLINK commands) within IMS, and on the MSC statement in IMS Connect (e.g., the VIEWMSC command). The generic IMSID name can be displayed using the QUERY MSC or /DIS ACTIVE commands.

Affinities remain until the last parallel MSC link is disconnected normally (i.e. IDLE NRE or IDLE COLD status).

Affinities ...

▪ Characteristics ...

- Affinities can be controlled when the link is first started by stopping the MSC path to the systems where an affinity is not desired
 - /STOP MSPLINK or UPDATE MSPLINK in IMS
 - UPDATE MSPLINK(name) START|STOP (GENIMSD) controls the generic IMSID
 - UPDATE MSPLINK(name) START|STOP (LOGON) controls the generic IMSID logons
 - STOPMSC in IMS Connect

- Optionally, establish affinity with a specific IMS system in the generic resource group by restarting the link from that specific IMS system

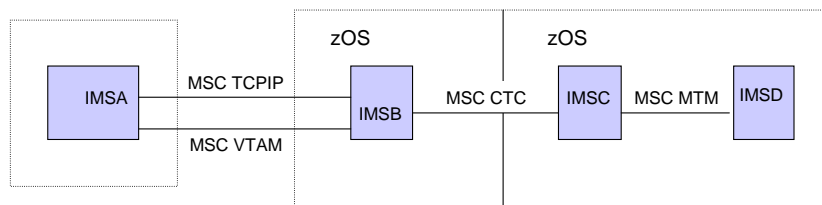
When the first link is started from the remote IMS, the affinity can be controlled by any one of the following methods:

- Stopping the MSC path to the other IMS system in the IMSplex where the link connection is not desired.
- Stopping the MSPLINK on the IMS system(s) where the connection is not desired. Those IMS systems will then not accept the restart message from the remote IMS.
- Have only one IMS up in the IMSplex when the first parallel link is started. If more than one IMS is up, and the MSC path has not been stopped in IMS connect, then the restart message will be sent to each IMS that is up. The first one to respond becomes the IMS that will own the affinity.
- Restarting the link from the generic resource group side.

MSC TCP/IP - Benefits

■ Benefits

- Takes advantage of TCP/IP networks for MSC
 - Can potentially provide for a higher MSC bandwidth
- Supports different configurations
 - Coexistence with or backup of VTAM/SNA links
 - Increases availability
 - Logical links can be moved between VTAM and TCP/IP
 - Flexibility



444

MSC TCP/IP provides an alternative to MSC VTAM and can also be used to backup an VTAM SNA network for availability purposes.



Reference Information

- IMS Commands
- IMS Messages
- IMS Connect Commands
- Trace records
- Additional Error Scenarios

IMS Commands

- /DISPLAY LINK and /DISPLAY LINK OPTION BUFSIZE
 - No change but TCP/IP links only support BANDWIDTH mode

```
/DIS LINK 4

Response:
LINK PARTNER RECD ENQCT DEQCT QCT SENT PSTOPPED IDLE COLD
  4 TA      0      0      0      0      0      0      0      0

/DIS LINK 4 OPTION BUFSIZE

Response:
LINK LINKNAME BUFSIZE STATUS
  4 LNK12T01   8192  PSTOPPED IDLE COLD BANDWIDTH
```

There is no change to these commands for TCPIP. However, since TCPIP links operate in bandwidth mode only, the /DISPLAY LINK OPTION BUFSIZE command will always show BANDWIDTH status.

IMS Commands ...

- **/DISPLAY ASSIGNMENT MSPLINK**
 - Now also includes TCP/IP physical links

```
/DIS ASMT MSPLINK ALL
```

Response:

LINK	PLINK	TYPE	ADDR	MAXSESS	NODE
3	PLNK12M	MTM	****		
2	PLNK12C	CTC	0511		
1	PLNK12V	VTAM	02000006	3	L6APPL1
4	PLNK12TA	TCPIP	****	2	IMS2

TCPIP physical links are displayable along with the other existing link types (i.e. CTC, MTM, and VTAM). TCPIP will display as the link type and the IMSID of the remote IMS will be displayed under the NODE column.

IMS Commands ...

▪ UPDATE MSPLINK

- New parameters for the TCP/IP environment

```

ÊÊUPDATEMSPLINK
  -UPD-
      @
      a-NAME(msplinkname)
      -msplinkname*
      -NAME(*)
Ê-----A-----
  a-START(-LOGON)
  -STOP(-LOGON)
A:
  ~
a-SET(-ASR(-ON))
  -OFF
  a-ICONPLKID(-TCP/IP_local_IMSconnect_MSC_configuration_statement_name)
  a-IMSCON(-TCP/IP_local_IMSconnect_member_name)
  a-MODETBL(-modetablename)
  a-MSPLINK(-new_msplinkname)
  a-NODE(-VTAM_node_name)
  -RMTIMS(-TCP/IP_remote_IMS_name)

```

448

New parameters have been added to the existing UPDATE MSPLINK command. These include:

- **ICONPLKID()** - For TCP/IP physical links, this specifies the 1- to 8-alphanumeric name of the local MSC configuration statement id in the local IMS Connect. This is the name of the MSC statement in the local IMS Connect configuration member. This member contains the parameters for IMS Connect to connect and communicate to its local IMS, and to the remote IMS via TCP/IP and the remote IMS Connect. This name may be changed to another IMS Connect MSC configuration statement.
- **IMSCON()** - For TCP/IP physical links, this specifies the 1- to 8-alphanumeric name of the local IMS Connect member name. This is the member name used by the local IMS Connect to join the IMSplex. This name may be changed to another local IMS Connect member name.
- **RMTIMS()** - For TCP/IP physical links, this specifies the 1- to 8-alphanumeric name of the remote IMS that this physical link connects to. This name may be changed to another IMS system's IMSid.

IMS Commands ...

■ QUERY MSPLINK

- Output shows the new parameters for the TCP/IP environment

```
/QUERY MSPLINK TYPE(TCPIP) SHOW(ALL)
```

Response:

MSPLink	MbrName	CC	Type	RmtIms	LclImsCon	LclPlkID	LclStat
PLNK31TA	SYS3	0	TCPIP	IMS1	ICON3	MSC31	
PLNK32TA	SYS3	0	TCPIP	IMS2	ICON3	MSC32	

The QUERY MSPLINK command output shows the new parameters associated with TCP/IP links.

IMS Connect Commands

- Enhanced WTOR and z/OS Modify commands
 - VIEWHWS and QUERY MEMBER TYPE(IMSCON)
 - VIEWPORT and QUERY PORT
 - STOPIP and UPDATE IMSPLEX NAME(*plexname*) STOP(COMM)
 - STARTIP and UPDATE IMSPLEX NAME(*plexname*) START(COMM)
- New WTOR and z/OS Modify commands
 - VIEWRMT and QUERY RMTIMSCON
 - STOPRMT and UPDATE RMTIMSCON STOP(COMM)
 - STARTRMT and UPDATE RMTIMSCON START(COMM)
 - STOPSCLN and DELETE RMTIMSCON SENDCLNT
 - VIEWMSC and QUERY MSC
 - STOPMSC and UPDATE MSC STOP(COMM)
 - STARTMSC and UPDATE MSC START(COMM)
 - STOPLINK and DELETE LINK

IMS Connect Commands ...

- **Enhancements to STOPIP and UPDATE IMSPLEX NAME(*plexname*) STOP(COMM)**
 - Disables communication with the IMSplex
 - If the IMSplex is used for **MSC**
 - IMS Connect disconnects the send and receive clients used on the MSC physical link
 - IMS Connect sends MSC level shutdown directive to MSC
 - IMS MSC cleans up physical link.
 - IMS issues DFS3176E message
 - IMS Connect's MSC status is set to DISCONNECT
- **Enhancements to STARTIP and UPDATE IMSPLEX NAME(*plexname*) START COMM)**
 - Enables communication with the IMSplex
 - If the IMSplex is used by **MSC**
 - IMS Connect's MSC status is set to ACTIVE

IMS Connect commands ...

- **VIEWRMT and QUERY RMTIMSCON**
 - Displays the new RMTIMSCON configurations values
 - RMTIMSCON status
 - ACTIVE – there is at least one socket connection to the remote IMS Connect
 - NOT ACTIVE – there is no socket connection to the remote IMS Connect
 - STOPPED – a STOPRMT command has stopped communications to the remote IMS Connect
 - Displays connection activity to the remote IMS Connect
 - Supports wildcards

IMS Connect commands ...

- VIEWRMT and QUERY RMTIMSCON example 1

IMS to IMS connections

MSC connections

```
nnVIEWRMT ALL
F HWS1,QUERY RMTIMSCON NAME(*)

HWSC0001I RMTIMSCON=ICON1 STATUS=ACTIVE
HWSC0001I IP-ADDRESS=009.030.221.055 PORT=5555
HWSC0001I HOSTNAME=ECSER14.VMEC.SVL.IBM.COM
HWSC0001I AUTOCONN=N PERSISTENT=N
HWSC0001I IDLETO=3000
HWSC0001I RESVSOC=4 NUMSOC=2
HWSC0001I SENDCLNT USERID STATUS SECOND SENDPORT
HWSC0001I OTM62A45 USER01 CONN 13 5421
HWSC0001I OTMD18BE USER02 CONN 5454 6792
HWSC0001I TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0
OTHER=0
```

```
nnVIEWRMT ALL
F HWS1,QUERY RMTIMSCON NAME(*)

HWSC0001I RMTIMSCON=ICON2 STATUS=ACTIVE
HWSC0001I IP-ADDRESS=009.030.221.055 PORT=5555
HWSC0001I HOSTNAME=ECSER14.VMEC.SVL.IBM.COM
HWSC0001I AUTOCONN=N PERSISTENT=Y
HWSC0001I IDLETO=0
HWSC0001I RESVSOC=2 NUMSOC=2
HWSC0001I SENDCLNT LCLPLKID STATUS SECOND SENDPORT
HWSC0001I MSC93D68 MSC12 CONN 1973 5555
HWSC0001I MSC3C970 MSC12 CONN 2026 5555
HWSC0001I TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0
OTHER=0
```

Example of RMTIMSCON used for OTMA remote messages as well as for MSC messages.

IMS Connect commands ...

- **STOPRMT and UPDATE RMTIMSCON STOP(COMM)**
 - Stop communications to the remote IMS Connect for the RMTIMSCON
 - All existing socket connections are closed
 - RMTIMSCON status set to STOPPED state
 - Supports wildcard

The STOPRMT OR UPDATERMTIMSCON STOP(COMM) commands request that communications with the remote IMS Connect be disabled.

From an OTMA perspective:

- IMS Connect disconnects the socket connections. If a connection is waiting for an ACK, then as a result of the command a NAK is sent back to OTMA with a reroute to HWS\$DLQ, the socket is disconnected and IMS Connect issues the HWST3570E message. A new sense code indicates the RMTIMSCON STOPPED state.
- When a new OTMA message is received, IMS Connect sends a NAK to OTMA to leave the message in the front of the queue, and issues an HWST3575W message.

For MSC:

- IMS Connect sends the MSC level shutdown directive to MSC. MSC then does an internal PSTOP and issues the DFS3176I message.
- IMS Connect disconnects the send and receive socket connections.
- If a new MSC message is received, IMS Connect sends an error message back to MSC.

On completion of the command, an HWST3505I message is issued: HWST3505I COMMUNICATIONS WITH REMOTE IMS CONNECT *rmtimscon* STOPPED; M=*mc*. If the command is issued on a RMTIMSCON that does not exist, then HWSC0114W message is issued: HWSC0114W COMMAND=STOPRMT FAILED; R=4, S=NFNDSCT , M=CVBC

Equivalent type-2 commands are available.

IMS Connect commands ...

- **STARTRMT and UPDATE RMTIMSCON START(COMM)**
 - Enable communications to the remote IMS Connect for the RMTIMSCON
 - RMTIMSCON status set to NOT ACTIVE from STOPPED state
 - Supports wildcard

The STARTRMT command and UPDATE RMTIMSCON START(COMM) z/OS modify command are new commands to enable the remote IMS Connect connection after it has been disabled by the STOPRMT and UPDATE RMTIMSCON STOP(COMM) commands.

- The RMTIMSCON status is set to NOT ACTIVE because there are no active socket connections at this point.
- If the AUTOCONN parameter has been set to Y, socket connections to remote ICON are automatically started based on the RESVSOC value.
- IMS Connect notifies OTMA to resend the OTMA remote messages for TPIPEs that uses the RMTIMSCON.

On completion of the command, an HWST3500I message is issued: HWST3500I COMMUNICATIONS WITH REMOTE IMS CONNECT *rmtimscon* STARTED; M=*mc*.

Equivalent type-2 commands are available.

IMS Connect Commands

- **STOPSCLN and DELETE RMTIMSCON SENDCLNT...**
 - Terminates a client connection to the remote IMS Connect
 - Only valid for connections used for OTMA remote ALTPCB messages
 - Supports wildcard
 - If issued when connection is in CONN state
 - IMS Connect disconnects send client socket and cleans up control blocks
 - Remote IMS Connect is notified of socket close by TCP/IP and cleanup of the control blocks occurs
 - If issued when connection is in RECV state
 - IMS Connect disconnects send client socket and cleans up control blocks
 - IMS Connect sends NAK with reroute to HWS\$DLQ to OTMA and issues HWST3570E message.
 - The Remote IMS Connect gets socket error when trying to send back ACK/NAK to local IMS Connect
 - Existing logic cleans up socket and associate control blocks

456

The STOPSCLN (WTOR) and DELETE RMTIMSCON SENDCLNT (z/OS modify) commands are new commands to terminate the send client sockets for a specified remote IMS Connect connection.

On completion of the command, an HWST3525I message is issued: HWST3525I THE SEND CLIENT *sendclient* WAS TERMINATED FOR REMOTE IMS CONNECT *rmtimscon*; M=*mc*. When issued on a non-existing send client name, the command fails with a HWSC0114W message.

When issued on a send client name used by MSC, the send client socket will be terminated and it will not be displayed in the VIEWRMT output. The send client name, however, will still be associated with its link in the VIEWMSC output. When a MSC message is sent on this logical link, a new send client connection is created and the VIEWMSC output will display the new send client name.

Equivalent type-2 commands are available.

IMS Connect commands

VIEWMSC and QUERY MSC

- Displays the new MSC configuration values
- MSC status
 - ACTIVE – communications to MSC for the physical link is enabled
 - NOT ACTIVE – communications to MSC for the physical link is disabled
- Displays MSC logical link activity of the MSC physical link
- Supports wildcard

```
nnVIEWMSC ALL
F HWS1, QRY MSC NAME(*)

HWSC0001I MSC=MSC12 STATUS=ACTIVE
HWSC0001I RMTPLKID=MSC21
HWSC0001I LCLIMSID=IMS1 RMTIMSID=IMS2
HWSC0001I IMSPLEX=PLEX1
HWSC0001I MEMBER=HWS1 TARGET MEMBER=PLEX1
HWSC0001I RMTIMSCON=ICON2
HWSC0001I IP-ADDRESS=009.030.221.055 PORT=5555
HWSC0001I HOSTNAME=ECSER14.VMEC.SVL.IBM.COM
HWSC0001I LINK PARTNERID STATUS SENDCLNT RECVCCLNT
HWSC0001I LNK12T01 TA ACTIVE MSC3C970 MSCBB237
HWSC0001I LNK12T02 TB ACTIVE MSC93D68 MSCC4B72
```

The VIEWMSC (WTOR) and QUERY MSC (z/OS modify) commands are new commands that display information about the MSC physical link defined in the MSC statement in the IMS Connect configuration member

IMS Connect commands

- **STOPMSC and UPDATE MSC STOP(COMM)**
 - Stops communications for the MSC physical link
 - All existing logical links on the MSC physical link are terminated and reported to IMS
 - MSC status for the physical link set to NOT ACTIVE state
- **STARTMSC and UPDATE MSC START(COMM)**
 - Enables communications for the MSC physical link
 - RMTIMSCON status set to ACTIVE state
 - Supports wildcard
- **STOPLINK and DELETE LINK**
 - Terminates MSC logical link and reports to IMS
 - IMS Connect disconnects the associated send and receive clients
 - Terminates the socket connections used by the MSC logical link
 - Should also be issued on the remote IMS Connect

458

After a STOPMSC command is issued, the MSC status is set to NOT ACTIVE state. When an MSC message is sent from the local IMS on the physical link that is NOT ACTIVE in IMS Connect, IMS Connect returns an error message to IMS indicating the MSC physical link is disabled.

- IMS Connect disconnects the send and receive clients used on this MSC physical link and sends an MSC level shutdown directive to IMS. IMS Connect's MSC status is set to NOT ACTIVE.
- IMS MSC cleans up the physical link and issues the DFS3176I message.

On completion of the command, an HWSF3305I message is issued: HWSF3305I COMMUNICATIONS ON MSC PHYSICAL LINK *lclplkid* STOPPED; M=*mc*

The STARTMSC or UPDATE MSC START(COMM) commands enable MSC communications for that physical link. IMS Connect's MSC status is set to ACTIVE. On completion of the command, an HWSF3300I message is issued: HWSF3300I COMMUNICATIONS ON MSC PHYSICAL LINK *lclplkid* STARTED; M=*mc*

The STOPLINK (WTOR) and DELETE LINK (z/OS modify) commands are used to terminate the MSC logical link resource type. IMS Connect disconnects send and receive clients and cleanup blocks. (Connections are no longer displayed in VIEWRMT and VIEWPORT output.) IMS Connect cleans up the logical link block and link is also no longer displayed in VIEWMSC output.

- IMS Connect sends a link level shutdown directive to the local MSC. IMS MSC cleans up the logical link and issues a DFS3176I message.
- On completion of the command, an HWSF3310I message is issued: HWSF3310I LOGICAL LINK *msclink* TERMINATED; MSC=*lclplkid*, M=*mc*
- A STOPLINK on the remote IMS Connect should also be issued.

Equivalent type-2 commands are available.



IMS Connect Recorder Trace

```

*...ICONRS...*
*.....*
*.....*IPB*
*OTMA prefix and Message from local IMS*
*.....*
*.....*OPB*
*IRM and Message from local IMS*
*.....*
    
```

Send message from local IMS to remote IMS Connect

```

*...ICONMS...*
*.....*
*.....*IPB*
*MSC Message from local IMS*
*.....*
*IRM and MSC Message from local IMS*OPB*
*.....*
    
```

Send message from local IMS to remote IMS Connect

```

*...ICONRR...*
*.....*
*.....*IPB*
*IRM and Message from remote ICON*
*.....*
*.....*OPB*
*OTMA prefix and Message from remote ICON*
*.....*
    
```

Receive message from remote IMS Connect to local IMS

```

*...ICONMR...*
*.....*
*.....*IPB*
*IRM and MSC Message from remote ICON*
*.....*
*MSC Message from remote ICON*OPB*
*.....*
    
```

Receive message from remote ICON to local IMS



Recorder Events for recorder exit HWSTECL0

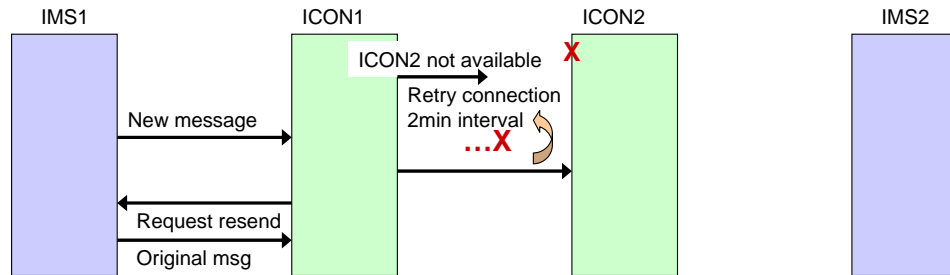
Event number	Event key	Event description
113	EVENT	Connected to remote IMS Connect.
114	EVENT, TPIPE, or PARTNERID	Disconnected from remote IMS Connect.
115	TPIPE or PARTNERID	Communications thread started for a remote IMS Connect connection.
116	TPIPE	Message received from OTMA for OTMA remote ALTPCB function.
117	TPIPE	Message sent to remote IMS Connect over TCP/IP for OTMA remote ALTPCB function.
118	TPIPE	Message received from remote IMS Connect over TCP/IP for OTMA remote ALTPCB function.
119	TPIPE	Message sent to OTMA for OTMA remote ALTPCB function.

Recorder Events for recorder exit HWSTECL0

Event number	Event key	Event description
120	PARTNERID	MSC message received from MSC.
121	PARTNERID	MSC message sent to remote IMS Connect.
122	PARTNERID	MSC message received from remote IMS Connect.
123	PARTNERID	MSC message sent to MSC.
124	TPIPE	Connection to remote IMS Connect timed out.

Additional Error Scenarios

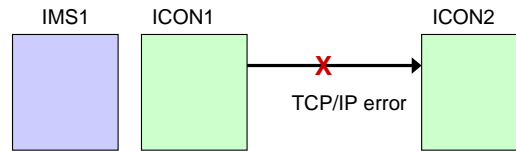
▪ New request message during connection retries



- In between connection retry attempts, a new message is received from OTMA
- Local IMS Connect attempts to make connection to remote IMS Connect
 - If successful, ICON1 cancels the previous ongoing retries and sends OTMA protocol to all connected IMS datastores to start sending messages that uses that RMTIMSCON

Additional Error Scenarios ...

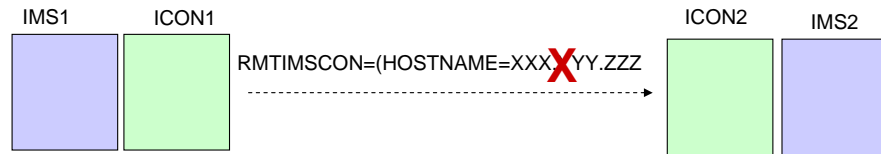
- Local IMS Connect cleanup on persistent socket failures



- Local IMS Connect detects that the socket is no longer valid when the remote IMS Connect goes down or a TCP/IP failure occurs
 - Cleans up the sockets
 - Connection is not displayed in VIEWRMT output.

Additional Error Scenarios ...

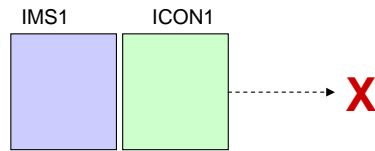
- IMS Connect startup detects an invalid name for the remote system



- The HOSTNAME parameter in the RMTIMSCON definition contains an invalid hostname
 - HWST3510E message is issued
 - HWST3510E COMMUNICATIONS WITH REMOTE IMS CONNECT rmtmscon FAILED TO START; R=rc, S=sc, M=mc
 - The thread and TCB of the RMTIMSCON is terminated
 - The RMTIMSCON status set in STOPPED state.
- To fix this problem, specify a valid hostname and restart IMS Connect.

Additional Error Scenarios ...

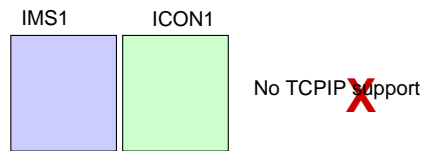
- Remote IMS Connect is unavailable during local IMS Connect startup



- In the RMTIMSCON statement of ICON1, the HOSTNAME parameter contains a valid hostname, but ICON2 is not started.
 - AUTOCONN=Y connections will fail and new connections will fail with this error message issued:
 - HWST3700E TCP/IP SOCKET FUNCTION CALL FAILED; F=CONNECT , R=-1, E=1128, N=76630291, RIC=ICON2B , M=TDCO
- When remote IMS Connect is available, new connections will be made properly

Additional Error Scenarios ...

- Local TCP/IP support is unavailable during local IMS Connect startup



- On startup, ICON1 issues the following messages:
 - HWSS0785W OPEN TCPIP COMMUNICATION ON HOSTNAME=TCPIP FAILED; R=-1, S=112, M=SOC1 (existing message)
 - HWST3700E TCP/IP SOCKET FUNCTION CALL FAILED; F=GETIPADR, R=-1, E=1, N=78AE1004, RIC=ICON2B, M=TDOT (new V12 message)
 - HWST3510E COMMUNICATIONS WITH REMOTE IMS CONNECT ICON2B FAILED TO START; R=8, S=OPENTHRD, M=TSCH (new V12 message)
 - Terminate thread and TCB of the RMTIMSCON
 - RMTIMSCON status set in STOPPED state
- When local TCP/IP is available, issue STARTRMT in the local IMS Connect to enable communications with that RMTIMSCON.

Additional Error Scenarios ...

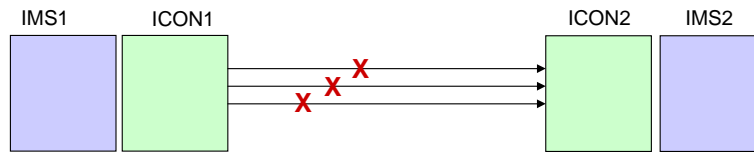
- Remote TCP/IP support is unavailable during local IMS Connect startup



- On Local IMS Connect startup, if AUTOCONN=Y is specified
 - ICON1 issues the following message after a 3 minute wait trying to connect to the remote ICON2:
 - HWST3700E TCP/IP SOCKET FUNCTION CALL FAILED; F=CONNECT , R=-1, E=1127, N=76630291, RIC=ICON2B , M=TDCO
 - RMTIMSCON status set in NOT ACTIVE state.
- When remote TCP/IP is available, new connections will be made properly.

Additional Error Scenarios ...

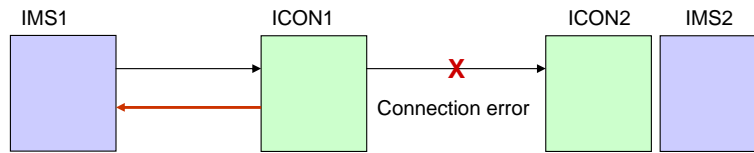
- Socket connection failures at startup for AUTOCONN=Y



- Local IMS Connect
 - Issues HWST3700E message for each connection failure
 - HWST3700E TCP/IP SOCKET FUNCTION CALL FAILED; F=fn, R=rc, E=ec, N=rsn, RIC=ri, M=mc
 - Retries connection to remote IMS Connect every 2 minutes
- When successful retry occurs
 - The local IMS Connect sends OTMA protocol signal to all connected IMS datastores to start sending messages that use that RMTIMSCON

Additional Error Scenarios ...

- Socket connection error

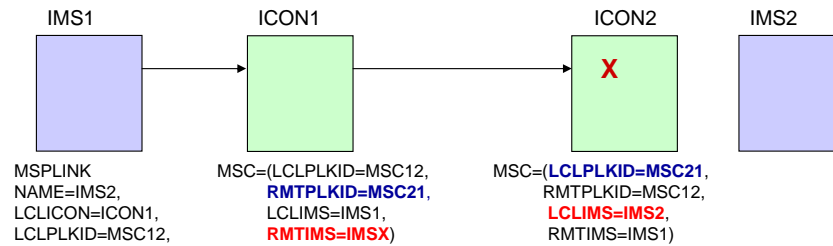


- Local IMS Connect

- Issues HWST3700E message for each connection failure.
 - HWST3700E TCP/IP SOCKET FUNCTION CALL FAILED; F=fn, R=rc, E=ec, N=rsn, RIC=ri, M=mc
- Sends NAK to OTMA and
 - If error is due to either a network problem or that the remote IMS Connect is not available
 - OTMA leaves message in front of queue.
 - Otherwise, for all other connection errors
 - OTMA reroutes the message to HWS\$DLQ queue.

Additional Error Scenarios ...

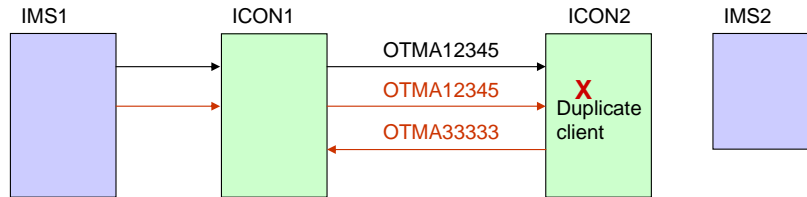
- RMTIMS validation error



- Mismatch on validation that the RMTIMS value in the local IMS Connect is the same as the LCLIMS value in the Remote IMS Connect
 - On validation error, issue HWSS0742W message
 - HWSS0742W MESSAGE FAILURE, RECEIVED FROM ORIGIN=portid clientid TO DESTID=did, MBR=tmname; **R=16, S=MISMATCH, M=mc**
 - Also send error response message to the requesting MSC

Additional Error Scenarios ...

- Duplicate clientID received on remote IMS Connect



- When the local IMS Connect sends a message to the remote IMS Connect with a clientid
 - The CIDREQ option in the IRM is set which tells the remote IMS Connect to generate a new clientid if the one in the message is discovered to be a duplicate clientid
 - If a new clientid is generated in the remote IMS Connect then the value is sent back to the local IMS Connect on the ACK/NAK
 - ACK message format: LLLL{LLZZCID}LLZZCSM
 - NAK message format: LLLL{LLZZCID}LLZZRSM
 - CID mapping contains the new clientid