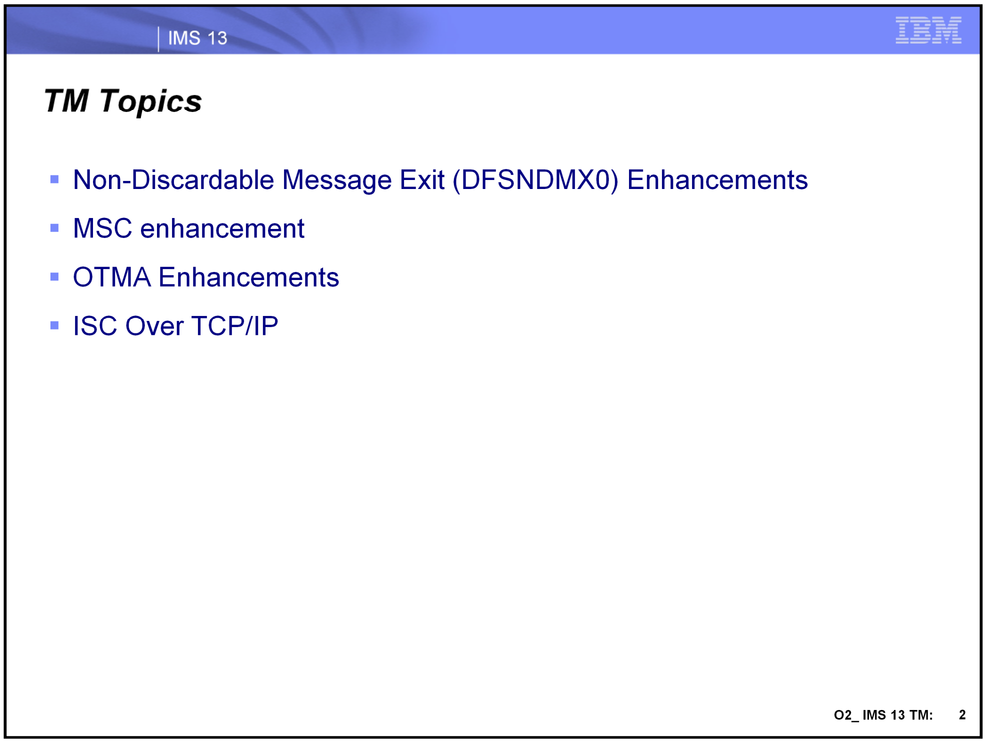
IMS 13

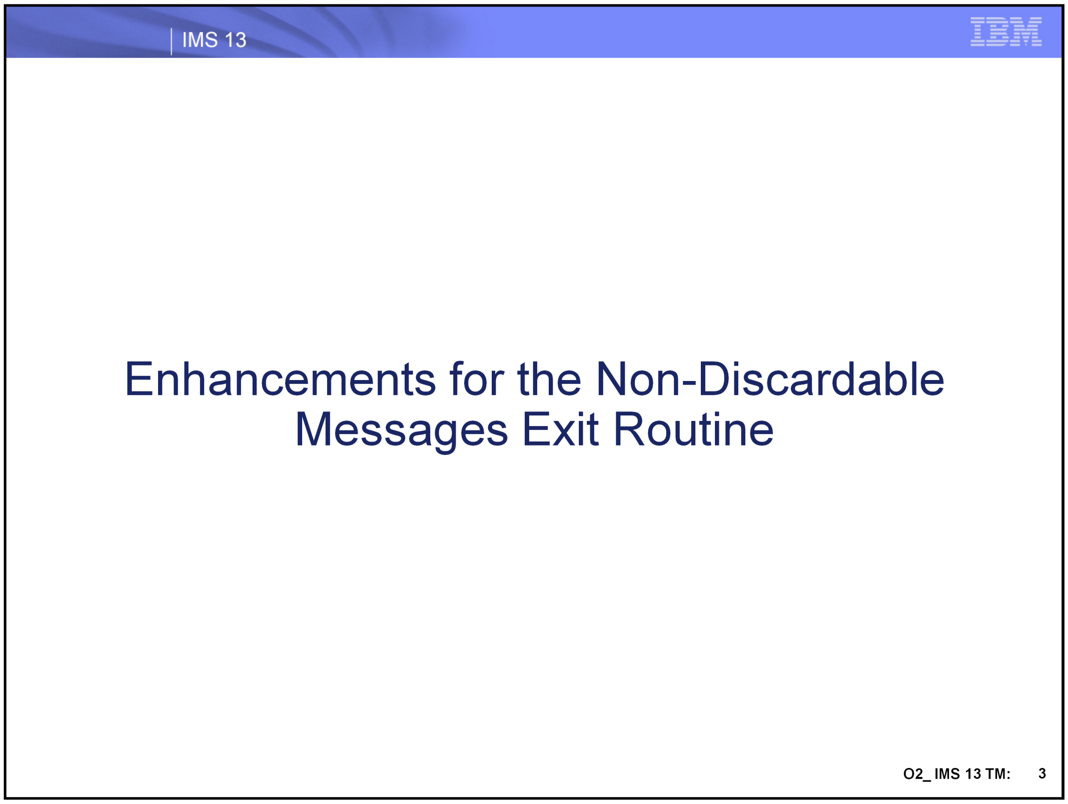
TM Enhancements

Information Management software



TM Topics

- Non-Discardable Message Exit (DFSNDMX0) Enhancements
- MSC enhancement
- OTMA Enhancements
- ISC Over TCP/IP

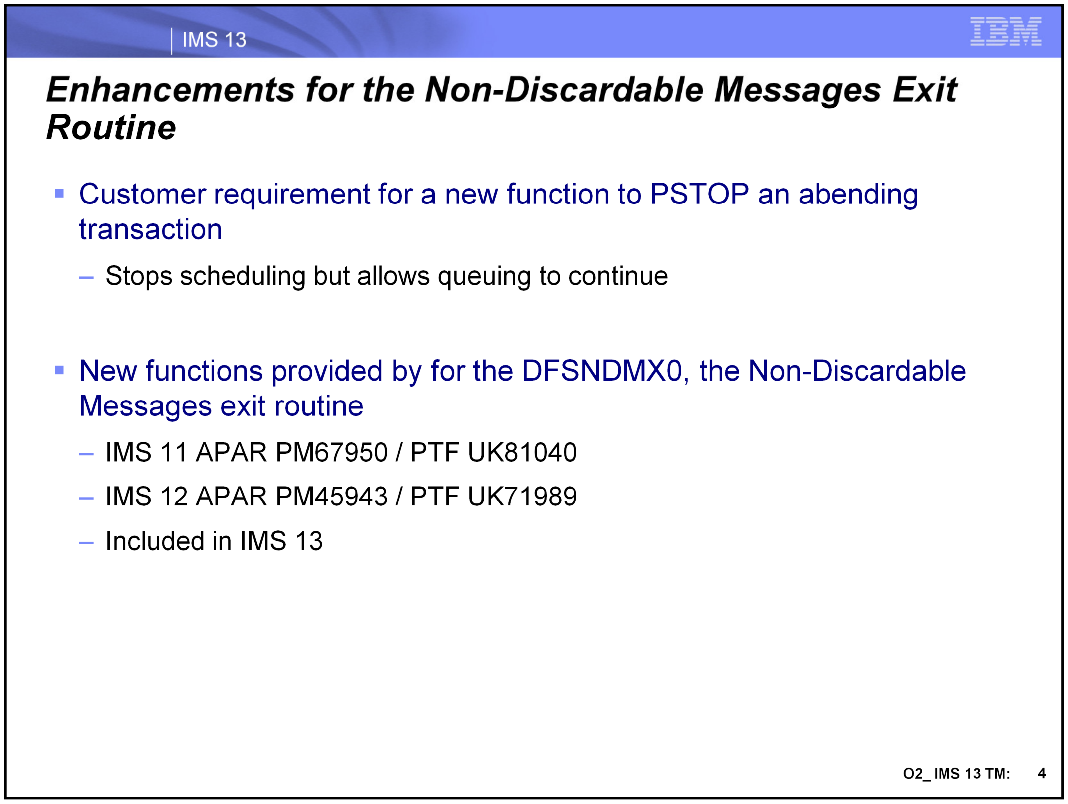


IMS 13

Enhancements for the Non-Discardable Messages Exit Routine

O2_ IMS 13 TM: 3

This topic summarizes some new enhancements for the non-discardable messages exit routine, DFSNDMX0.

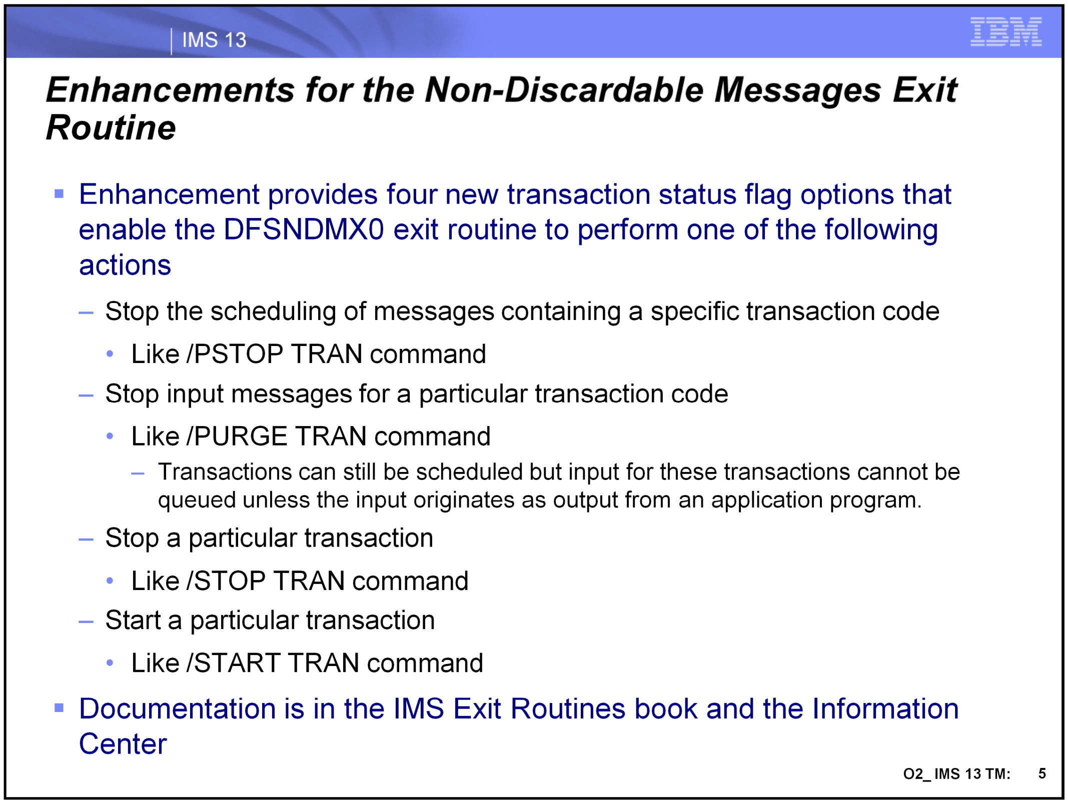


Enhancements for the Non-Discardable Messages Exit Routine

- Customer requirement for a new function to PSTOP an abending transaction
  - Stops scheduling but allows queuing to continue
- New functions provided by for the DFSNDMX0, the Non-Discardable Messages exit routine
  - IMS 11 APAR PM67950 / PTF UK81040
  - IMS 12 APAR PM45943 / PTF UK71989
  - Included in IMS 13

This enhancement provides new function for the DFSNDMX0, the non-discardable messages exit routine.

These enhancements are included in IMS 13 . They are also available for IMS 12 and IMS 11 via PTF.

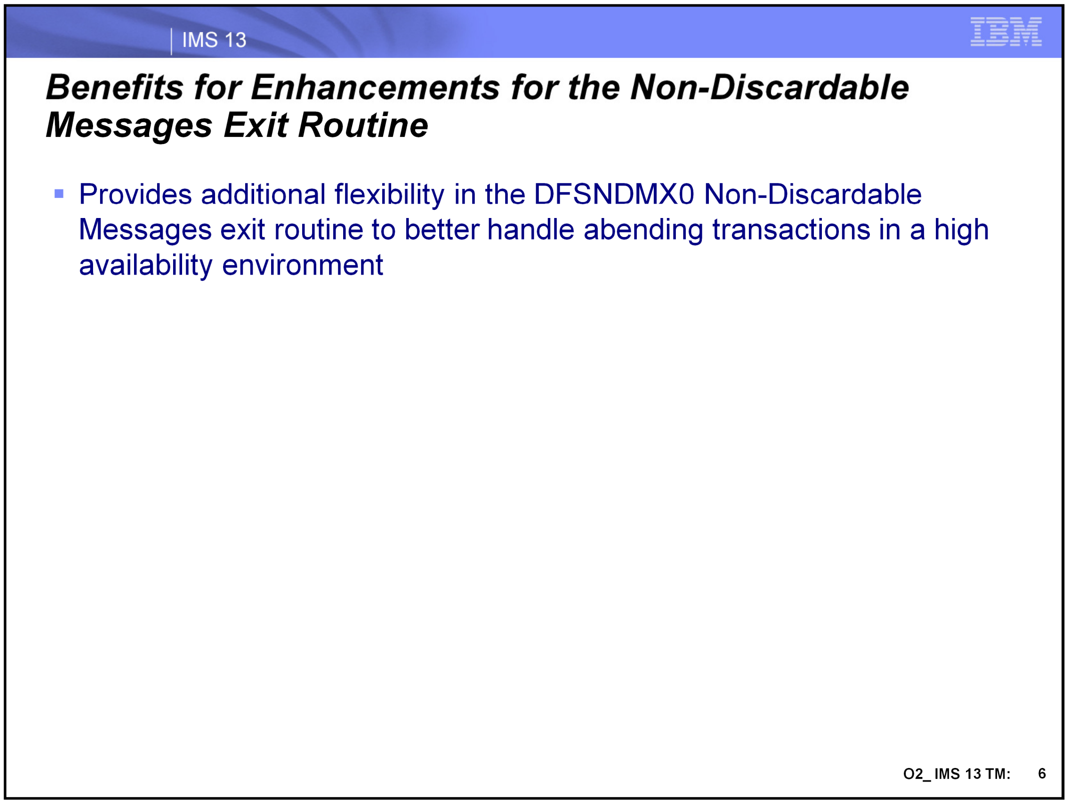


Enhancements for the Non-Discardable Messages Exit Routine

- Enhancement provides four new transaction status flag options that enable the DFSNDMX0 exit routine to perform one of the following actions
  - Stop the scheduling of messages containing a specific transaction code
    - Like /PSTOP TRAN command
  - Stop input messages for a particular transaction code
    - Like /PURGE TRAN command
      - Transactions can still be scheduled but input for these transactions cannot be queued unless the input originates as output from an application program.
  - Stop a particular transaction
    - Like /STOP TRAN command
  - Start a particular transaction
    - Like /START TRAN command
- Documentation is in the IMS Exit Routines book and the Information Center

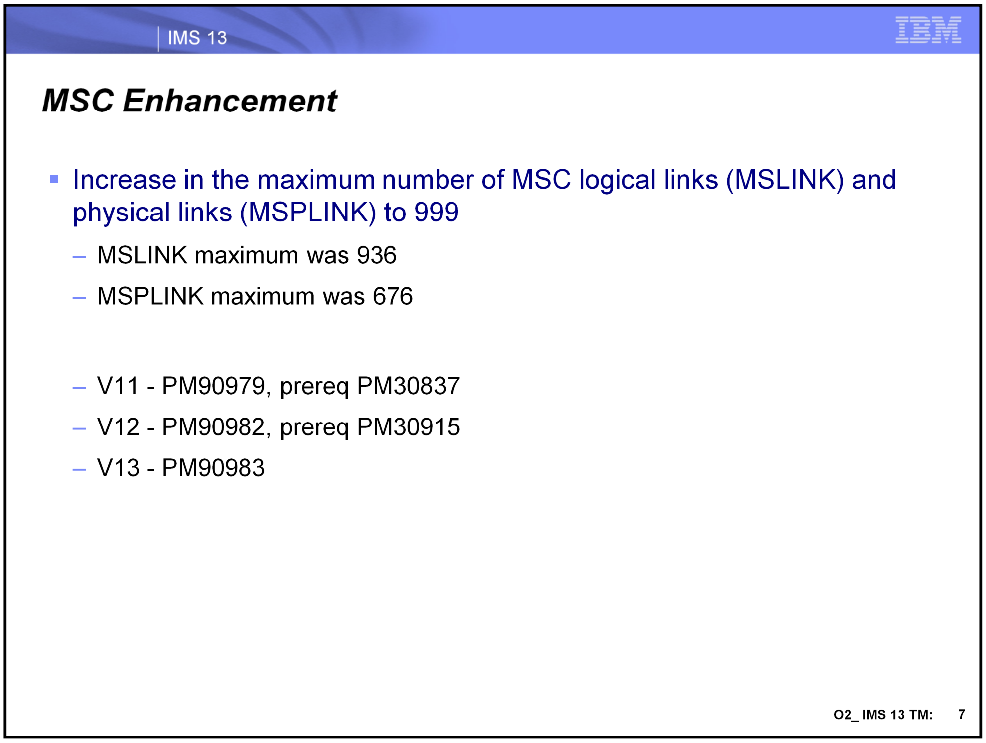
Four new enhancements for the transaction status flag are provided.

Additional documentation in the IMS Exit Routines book and the Information Center.



Benefits for Enhancements for the Non-Discardable Messages Exit Routine

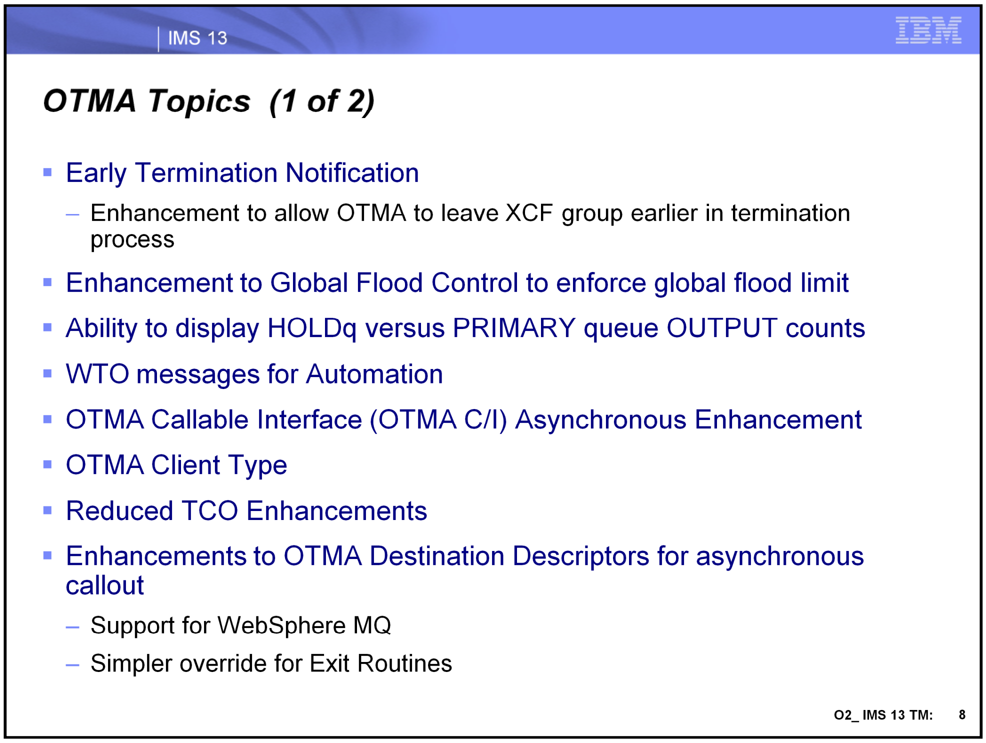
- Provides additional flexibility in the DFSNDMX0 Non-Discardable Messages exit routine to better handle abending transactions in a high availability environment



MSC Enhancement

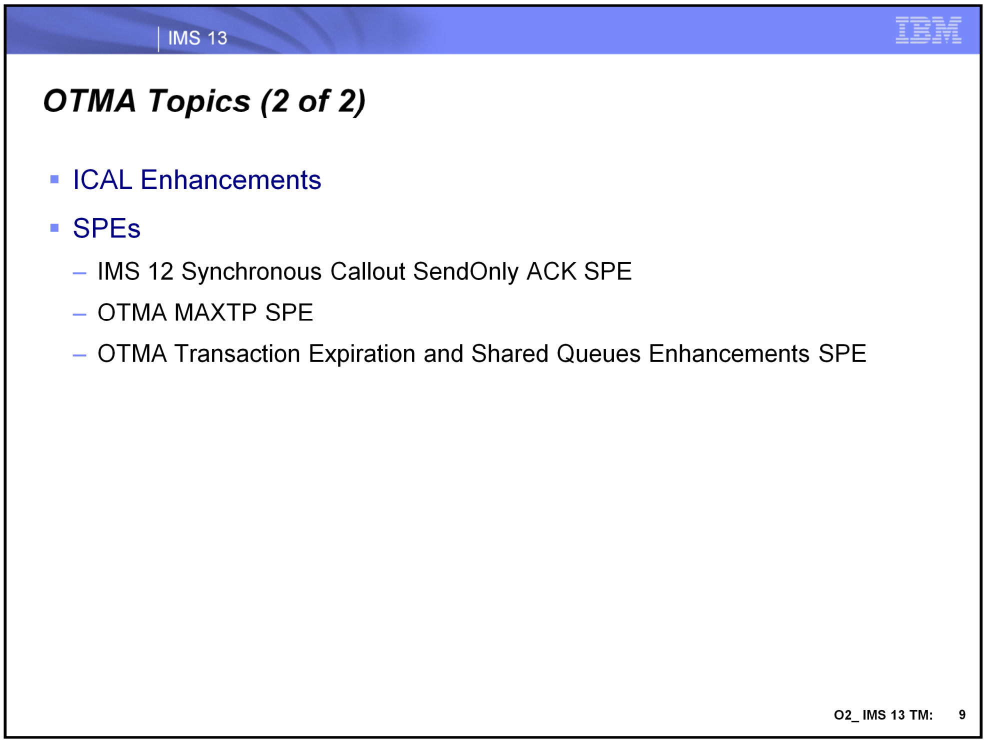
- Increase in the maximum number of MSC logical links (MSLINK) and physical links (MSPLINK) to 999
  - MSLINK maximum was 936
  - MSPLINK maximum was 676

  - V11 - PM90979, prereq PM30837
  - V12 - PM90982, prereq PM30915
  - V13 - PM90983



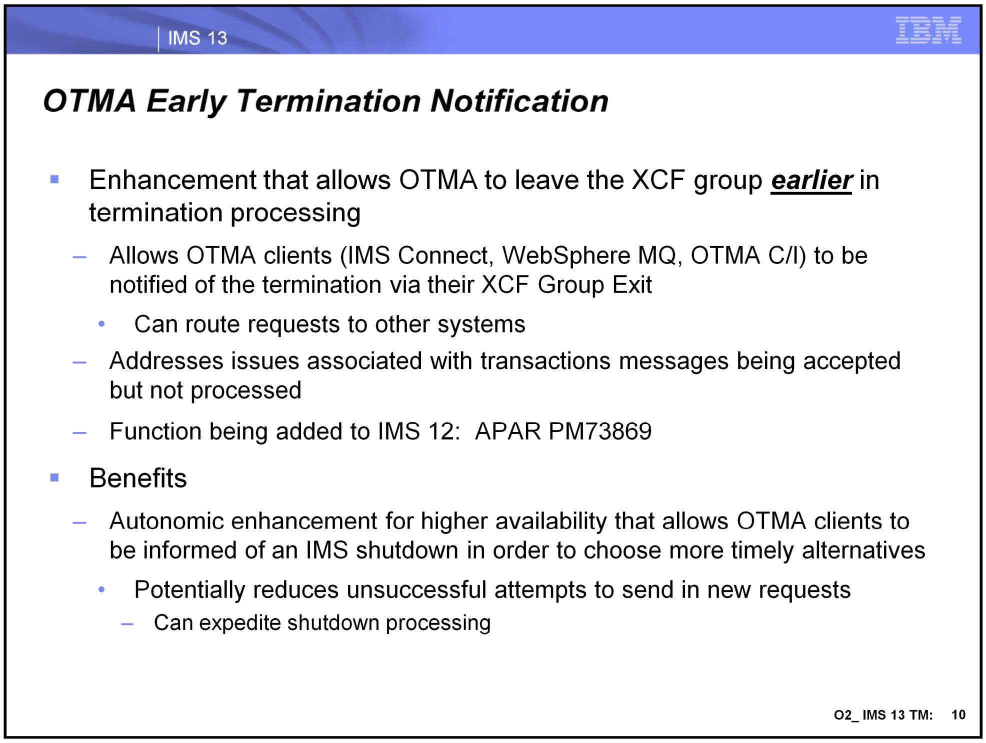
OTMA Topics (1 of 2)

- Early Termination Notification
  - Enhancement to allow OTMA to leave XCF group earlier in termination process
- Enhancement to Global Flood Control to enforce global flood limit
- Ability to display HOLDq versus PRIMARY queue OUTPUT counts
- WTO messages for Automation
- OTMA Callable Interface (OTMA C/I) Asynchronous Enhancement
- OTMA Client Type
- Reduced TCO Enhancements
- Enhancements to OTMA Destination Descriptors for asynchronous callout
  - Support for WebSphere MQ
  - Simpler override for Exit Routines



OTMA Topics (2 of 2)

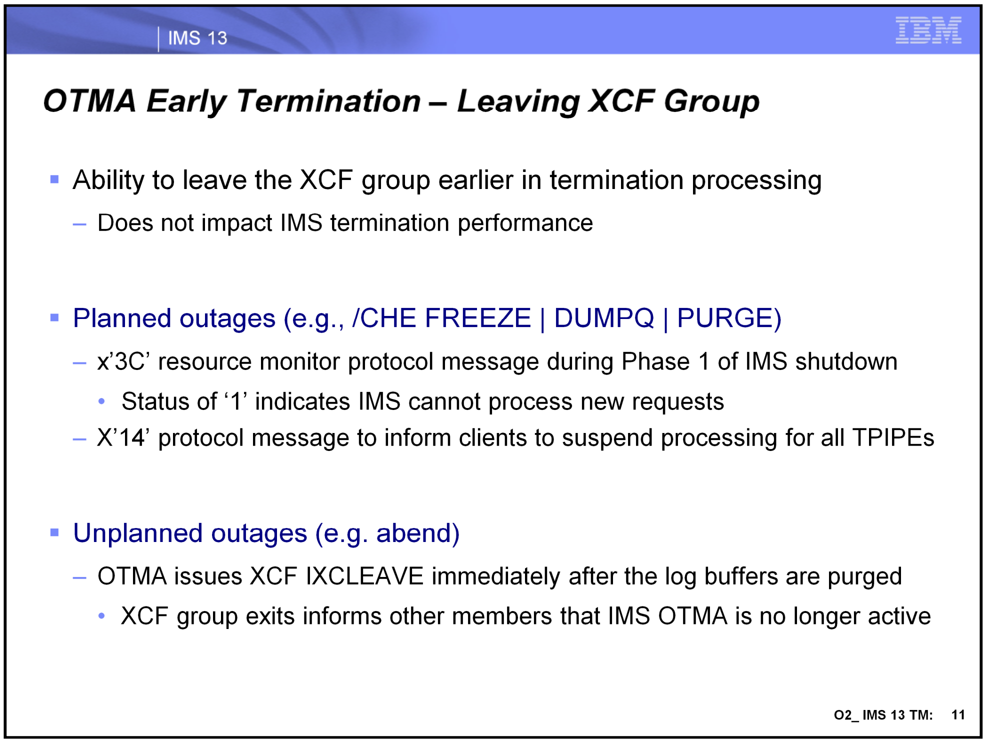
- ICAL Enhancements
- SPEs
  - IMS 12 Synchronous Callout SendOnly ACK SPE
  - OTMA MAXTP SPE
  - OTMA Transaction Expiration and Shared Queues Enhancements SPE



OTMA Early Termination Notification

- Enhancement that allows OTMA to leave the XCF group ***earlier*** in termination processing
  - Allows OTMA clients (IMS Connect, WebSphere MQ, OTMA C/I) to be notified of the termination via their XCF Group Exit
    - Can route requests to other systems
  - Addresses issues associated with transactions messages being accepted but not processed
  - Function being added to IMS 12: APAR PM73869
- Benefits
  - Autonomic enhancement for higher availability that allows OTMA clients to be informed of an IMS shutdown in order to choose more timely alternatives
    - Potentially reduces unsuccessful attempts to send in new requests
      - Can expedite shutdown processing

IMS 13 introduces an enhancement that allows OTMA clients (e.g.,. IMS Connect, WebSphere MQ, OTMA C/I) to be notified of IMS termination earlier in the termination process. This will be done for both scheduled and unscheduled terminations. The IMS OTMA clients can then route new requests to other IMS systems to be processed. This enhancement addresses environments that process high rates of transactions via IMS OTMA and experience some transactions being accepted but not processed by IMS during either scheduled or unscheduled IMS terminations.



IMS 13

OTMA Early Termination – Leaving XCF Group

- Ability to leave the XCF group earlier in termination processing
  - Does not impact IMS termination performance

- Planned outages (e.g., /CHE FREEZE | DUMPQ | PURGE)
  - x’3C’ resource monitor protocol message during Phase 1 of IMS shutdown
    - Status of ‘1’ indicates IMS cannot process new requests
  - X’14’ protocol message to inform clients to suspend processing for all TPIPEs

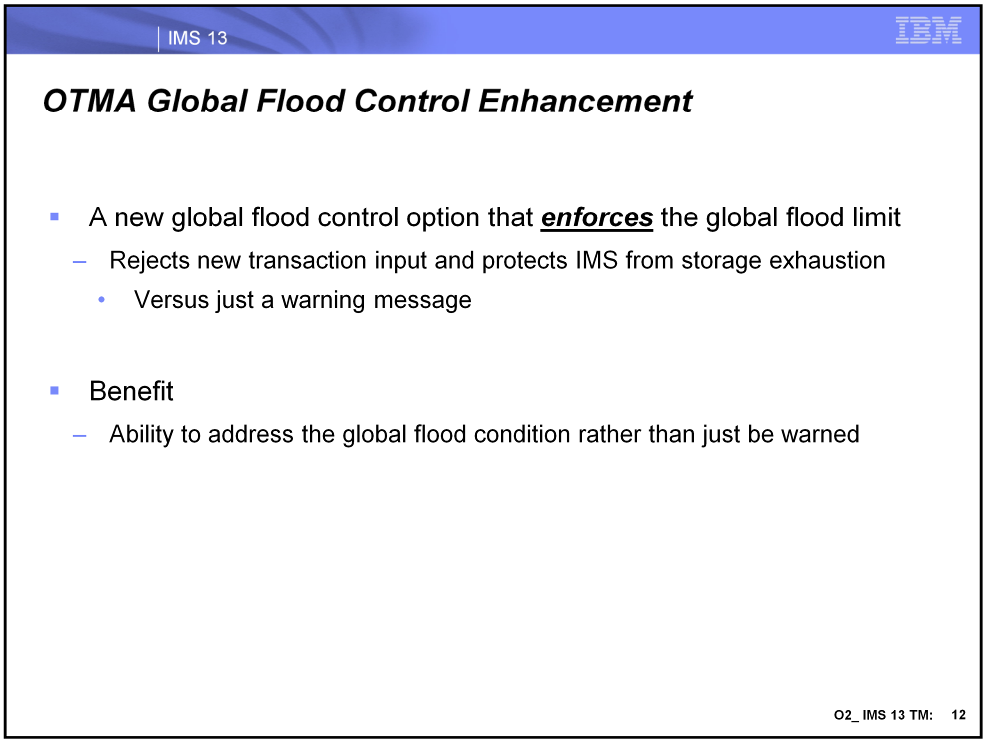
- Unplanned outages (e.g. abend)
  - OTMA issues XCF IXCLEAVE immediately after the log buffers are purged
    - XCF group exits informs other members that IMS OTMA is no longer active

O2_ IMS 13 TM: 11

OTMA Early Termination Notification is part of the base IMS processing and is autonomic meaning that there are no definition or configuration changes to take advantage of the enhancement.

When IMS has a planned shutdown (/CHE {FREEZE | DUMPQ | PURGE}) then OTMA adds notification to OTMA clients during Phase 1 of shutdown. OTMA sends a TMAMMNTR (x’3C’) resource monitor protocol message with TMAMRSIM_S1SHTDN indicating IMS shutdown with TMAMRSIM_STATUS set to ‘1’ indicating IMS cannot process new requests. OTMA also sends a TMAMCSPA (x’14’) protocol message informing clients to suspend processing for all TPIPEs. These protocol messages, sent at the start of shutdown, inform OTMA clients that IMS is no longer available and they should route new requests to another system. (note: the resource monitoring protocol was introduced in IMS 11 as a way for OTMA to inform its member clients of potential issues).

When IMS has an unplanned shutdown (abend) then OTMA will issue IXCLEAVE to leave the XCF group immediately after the log buffers are purged. This will inform the OTMA clients via their XCF group exits that IMS OTMA is no longer active in the XCF group. The OMTA clients can then route new requests to another system.



IMS 13

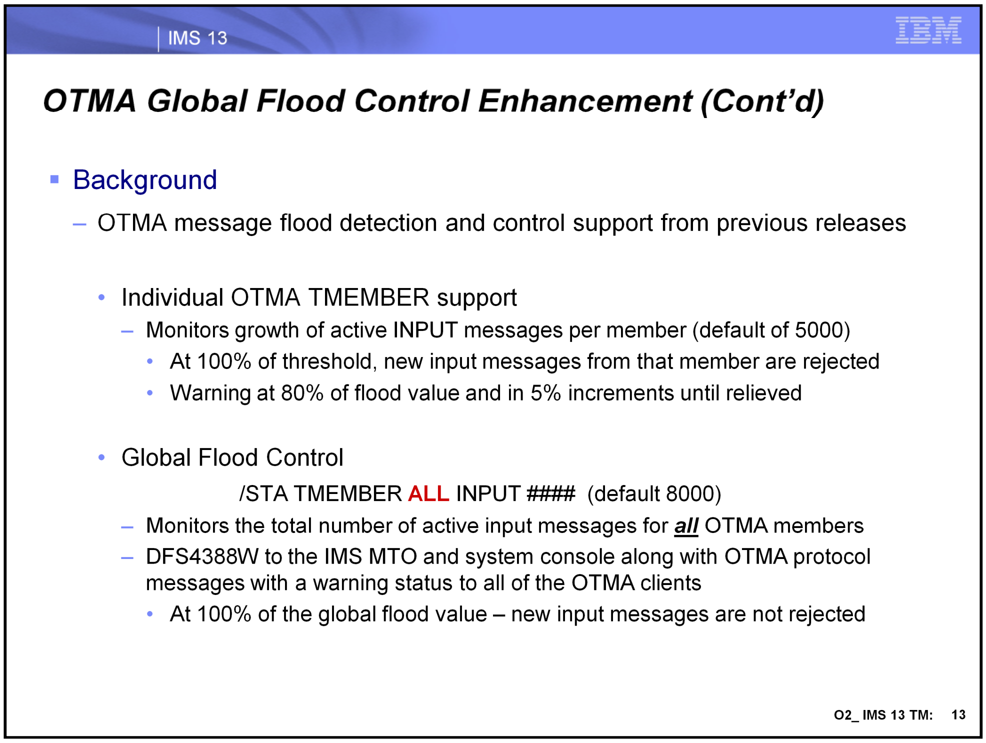
OTMA Global Flood Control Enhancement

- A new global flood control option that ***enforces*** the global flood limit
  - Rejects new transaction input and protects IMS from storage exhaustion
    - Versus just a warning message

- Benefit
  - Ability to address the global flood condition rather than just be warned

O2_ IMS 13 TM: 12

IMS OTMA also provides a new option to enforce the global flood limit to reject new requests and protect IMS from storage exhaustion. Previous releases protected flood conditions for individual OTMA clients (TMEMBERs) but only provided warning messages when the global flood limit was reached.



IMS 13

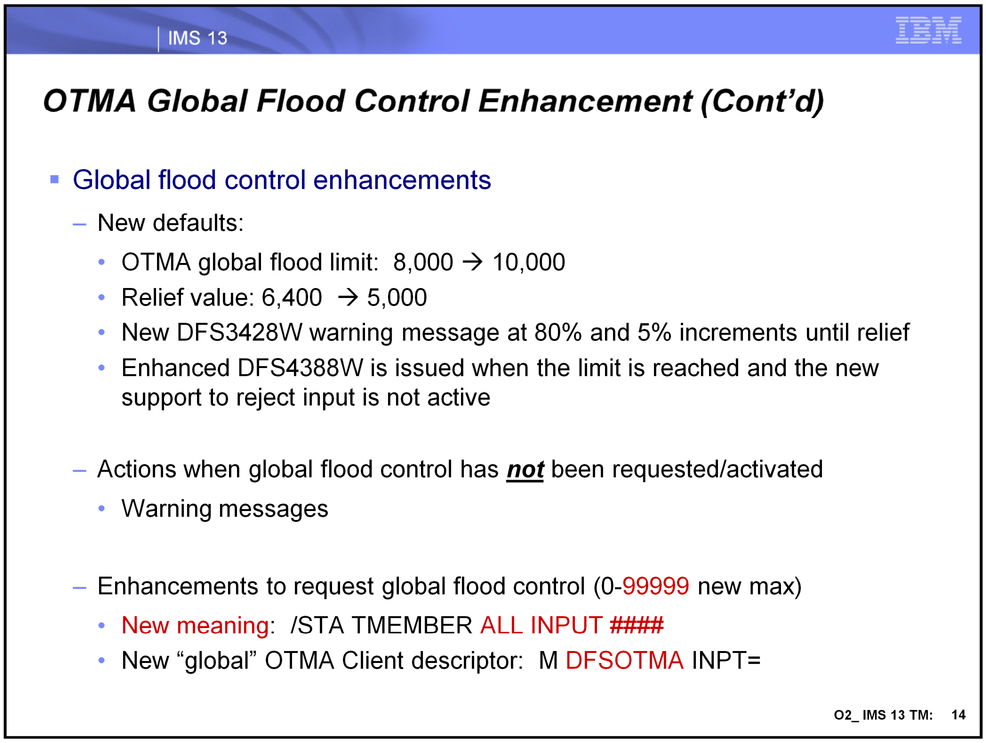
OTMA Global Flood Control Enhancement (Cont’d)

- Background
  - OTMA message flood detection and control support from previous releases
    - Individual OTMA TMEMBER support
      - Monitors growth of active INPUT messages per member (default of 5000)
        - At 100% of threshold, new input messages from that member are rejected
        - Warning at 80% of flood value and in 5% increments until relieved
    - Global Flood Control

      /STA TMEMBER ALL INPUT #### (default 8000)
      - Monitors the total number of active input messages for ***all*** OTMA members
      - DFS4388W to the IMS MTO and system console along with OTMA protocol messages with a warning status to all of the OTMA clients
        - At 100% of the global flood value – new input messages are not rejected

O2_ IMS 13 TM: 13

In previous releases, OTMA’s Message Flood Detection and Control capability provided a mechanism to automatically monitor the growth of active OTMA input messages and the control blocks associated with these input requests. Specifically, when an OTMA member or client sends a transaction to IMS, OTMA internally creates a control block called the TIB (Transaction Instance Block) to track each active input message. If several thousand OTMA input transactions are received from an OTMA member and waiting to be processed, thousands of control blocks representing the requests would take lots of IMS storage which could affect the overall IMS operations in the system. To prevent this type of OTMA message flood condition from an OTMA member, OTMA stops receiving the input transactions from this member based on a maximum value for the number of TIBs allowed for the OTMA member in the system. However, suppressing the input OTMA transactions only targets individual OTMA members that have the maximum value or threshold of TIBs defined. This type of OTMA flood control does not protect the IMS system when it has multiple OTMA members. In this case, each one of them may not reach the flood limit for the member. However, the total TIBs from all the OTMA members could exceed the global flood limit of the IMS system. When this happens, the IMS system (pre-IMS13) issues a warning signal with the Global Flood Control capability introduced in IMS 11 to the system console, the MTO and all the OTMA members without suppressing the input OTMA transactions. New transactions, however, can still come into IMS and flood the system. This could cause the IMS system to fail with a S878 abend.



OTMA Global Flood Control Enhancement (Cont'd)

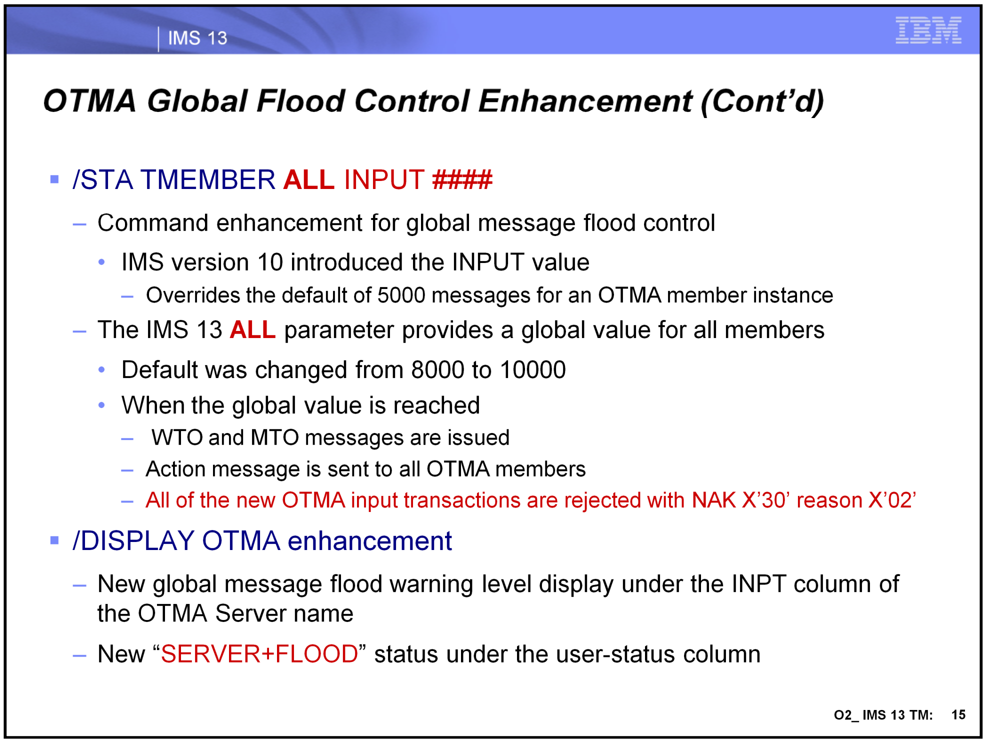
- Global flood control enhancements
  - New defaults:
    - OTMA global flood limit: 8,000 → 10,000
    - Relief value: 6,400 → 5,000
    - New DFS3428W warning message at 80% and 5% increments until relief
    - Enhanced DFS4388W is issued when the limit is reached and the new support to reject input is not active
  - Actions when global flood control has ***not*** been requested/activated
    - Warning messages
  - Enhancements to request global flood control (0-99999 new max)
    - New meaning: /STA TMEMBER ALL INPUT ####
    - New "global" OTMA Client descriptor: M DFSOTMA INPT=

The default OTMA global flood limit of the system has been changed from 8,000 to 10,000 along with a default relief level change from 6400 to 5000. New DFS3428W warning message has been introduced when the 80% of global flood limit has reached. This message will be issued every 5% increment thereafter until the global flood limit is reached. Once the global flood limit is reached, the enhanced warning message DFS4388W will be sent to the system console and MTO along with OTMA protocol messages reflecting a warning status to all the OTMA members *when the global flood control has not been activated.* This global flood status can be relieved when the input messages in the system have been processed and the number of TIBS has been reduced to 50% or less of the global limit. The enhanced relief message DFS0793I will then be sent to the IMS MTO and system console along with OTMA protocol messages reflecting a good status to all the OTMA member clients.

The default actions (above) will take place if IMS is activated without limits to request input rejection with global flood control (below). This means that an IMS system without any specification or changes will come with the default OTMA global flood monitoring and a default global limit of 10,000. IMS will send out warning messages DFS3428W and DFS4388W to the system console and MTO along with protocol warning messages to all the OTMA members. No input OTMA transactions will be suppressed. This default mode of OTMA global flood monitoring can be changed to suppress input transactions from all the OTMA members once a global flood control and limits are specified

IMS 13 allows the activation of "global flood control" with two methods: (a) issuing the /STA TMEMBER **ALL** INPUT #### command where #### is the global flood limit as in previous releases but will now cause a rejection of new input when the value is reached in IMS 13, or (b) specifying a global limit value in a new OTMA client descriptor (DFSOTMA).

The valid global flood limit has a new max value and can be specified between 0 and 99999. (The valid range for the member flood limit continues to be between 0 and 9999.) As in previous releases, a non-zero value activates the global flood control for OTMA for the system. A value of 0 deactivates the global flood support and a value between 1 and 200, is treated as though 200 had been specified.



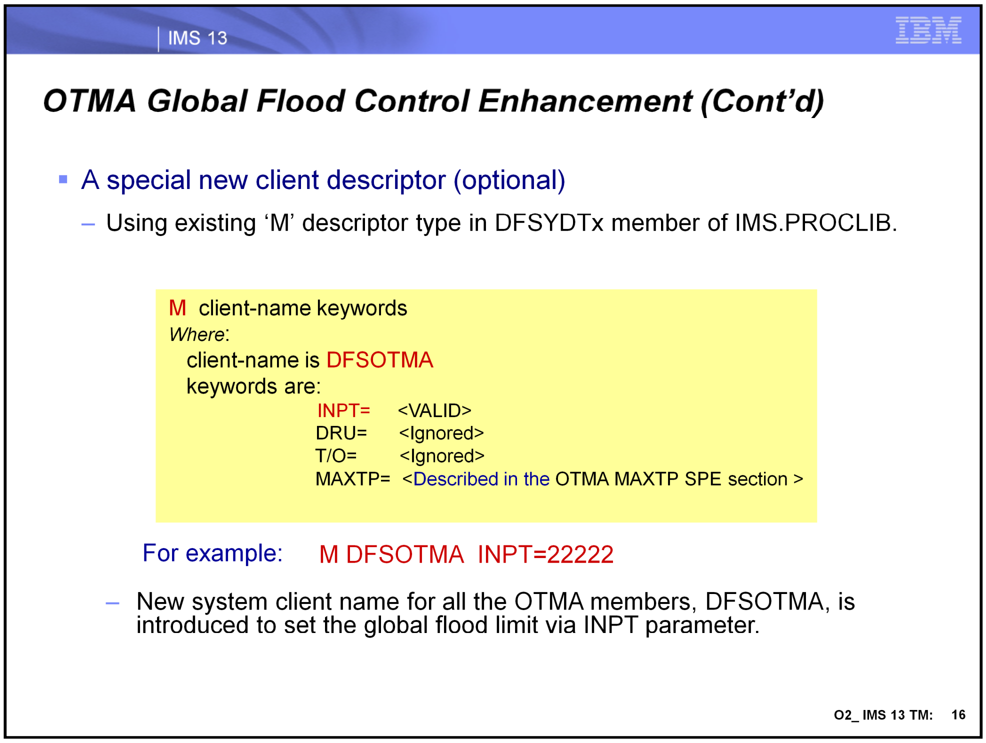
IMS 13

OTMA Global Flood Control Enhancement (Cont'd)

- /STA TMEMBER **ALL** INPUT ####
  - Command enhancement for global message flood control
    - IMS version 10 introduced the INPUT value
      - Overrides the default of 5000 messages for an OTMA member instance
  - The IMS 13 **ALL** parameter provides a global value for all members
    - Default was changed from 8000 to 10000
    - When the global value is reached
      - WTO and MTO messages are issued
      - Action message is sent to all OTMA members
      - All of the new OTMA input transactions are rejected with NAK X'30' reason X'02'
- /DISPLAY OTMA enhancement
  - New global message flood warning level display under the INPT column of the OTMA Server name
  - New "SERVER+FLOOD" status under the user-status column

In IMS 13 the meaning and actions associated with the /STA TMEMBER ALL INPUT #### command have changed. In IMS 11, when this command was issued, OTMA used the specified limit instead of the system default of 8000 to monitor the total active input message count for send-then-commit (CM1) transactions from all of the OTMA members. If the total number reached this specified limit, OTMA issued a DFS4388W to the IMS MTO and system console along with OTMA protocol messages with the warning status to all of the OTMA clients. In IMS 13, the limit is used to for more than a warning and allows OTMA to prevent any new message from being accepted until the flood situation is relieved.

The /DISPLAY OTMA command of a degraded system shows "SERVER+FLOOD" in the user status of the OTMA server member when the global flood control has been activated and the global flood limit is reached.



OTMA Global Flood Control Enhancement (Cont'd)

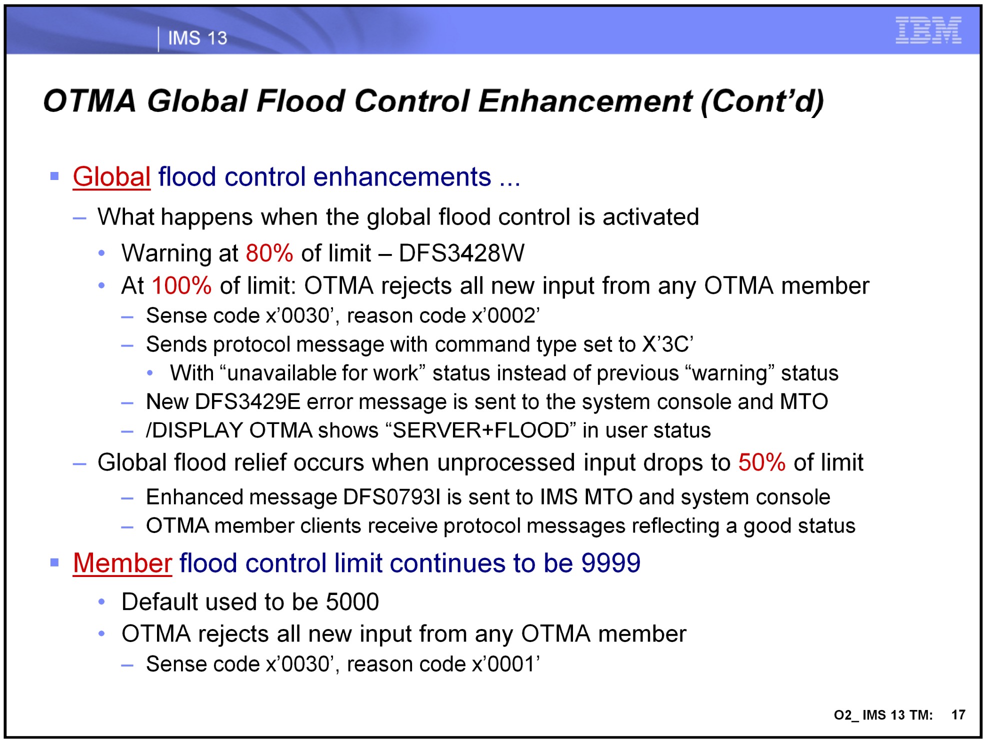
- A special new client descriptor (optional)
  - Using existing 'M' descriptor type in DFSYDTx member of IMS.PROCLIB.

```
M  client-name keywords
Where:
   client-name is DFSOTMA
   keywords are:
                        INPT=     <VALID>
                        DRU=      <Ignored>
                        T/O=      <Ignored>
                        MAXTP=  <Described in the OTMA MAXTP SPE section >
```

For example: M DFSOTMA INPT=22222

  - New system client name for all the OTMA members, DFSOTMA, is introduced to set the global flood limit via INPT parameter.

The option also exists to use a descriptor to define the global flood control limit using a client descriptor. Note that a special system client name, DFSOTMA, has been introduced for this purpose. Activation of this DFSOTMA client descriptor enables the global flood control and requests suppression of any new OTMA transactions when the limit is reached. The parameter that applies to this function in the DFSOTMA client descriptor is INPT=. Specification of other parameters, such as DRU=, T/O= are ignored. The MAXTP parameter capability is described in the OTMA MAXTP SPE later in this presentation.



IMS 13

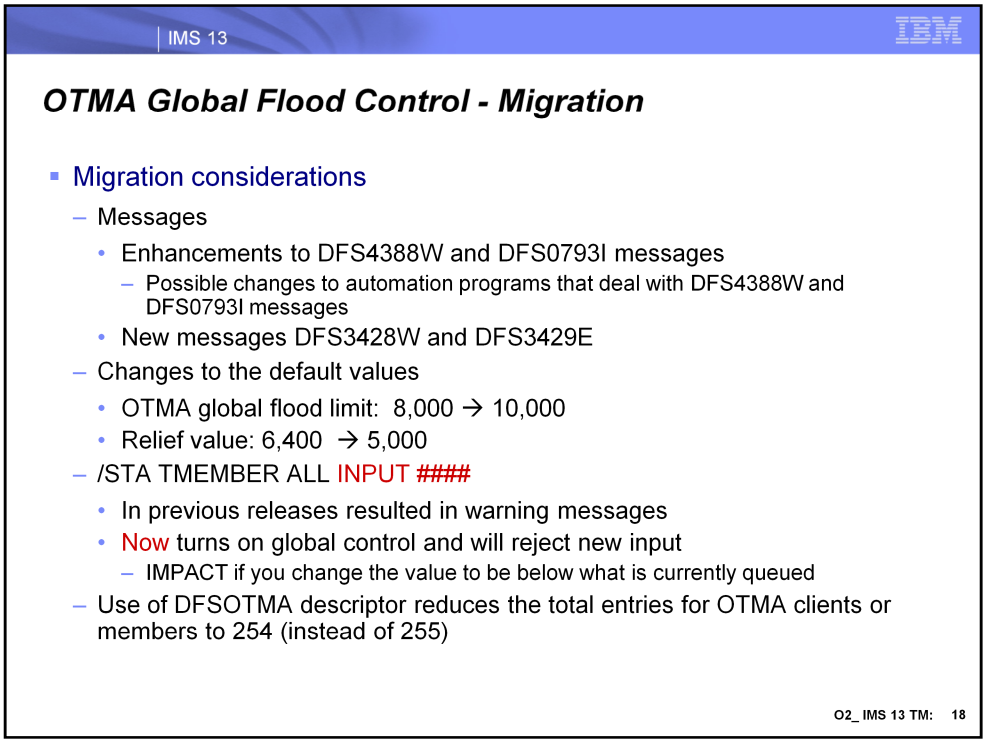
OTMA Global Flood Control Enhancement (Cont'd)

- Global flood control enhancements ...
  - What happens when the global flood control is activated
    - Warning at 80% of limit – DFS3428W
    - At 100% of limit: OTMA rejects all new input from any OTMA member
      - Sense code x'0030', reason code x'0002'
      - Sends protocol message with command type set to X'3C'
        - With "unavailable for work" status instead of previous "warning" status
      - New DFS3429E error message is sent to the system console and MTO
      - /DISPLAY OTMA shows "SERVER+FLOOD" in user status
  - Global flood relief occurs when unprocessed input drops to 50% of limit
      - Enhanced message DFS0793I is sent to IMS MTO and system console
      - OTMA member clients receive protocol messages reflecting a good status
- Member flood control limit continues to be 9999
    - Default used to be 5000
    - OTMA rejects all new input from any OTMA member
      - Sense code x'0030', reason code x'0001'

O2_ IMS 13 TM: 17

Global flood control has action points at 80% (warning), 100% (rejection of new messages), and 50% (relief) of the global flood limit. When "global flood control" has been activated and the global flood limit is reached, OTMA rejects all the **new** input transactions from all the members with OTMA sense code x'0030' and reason code x'0002'. Any synchronous program switch requests from the internal OTMA member DFSYICAL are also rejected. The new DFS3429E error message is then sent to the system console and MTO and a protocol message with the command type set to X'3C' is sent to all the OTMA members with an "unavailable for work" status unlike previous releases which only sent a "warning" status. When OTMA members receive this status, they can choose to take corrective action. One example would be the rerouting of all the **new** transaction requests from one IMS system to another. Once IMS reaches the 50% value of the global limit for unprocessed messages, the global flood status is considered to be relieved and IMS begins accepting new input messages. The enhanced relief message DFS0793I is then sent to the IMS MTO and system console along with OTMA protocol messages reflecting a good status to all the OTMA member clients.

Note that the MEMBER flood limit (versus GLOBAL for all members combined) remains at 9999 as in previous releases. When a member reaches the member flood limit, new transactions from this member will be rejected with OTMA sense code x'0030' and the new reason code x'0001'.



IMS 13

OTMA Global Flood Control - Migration

- Migration considerations
  - Messages
    - Enhancements to DFS4388W and DFS0793I messages
      - Possible changes to automation programs that deal with DFS4388W and DFS0793I messages
    - New messages DFS3428W and DFS3429E
  - Changes to the default values
    - OTMA global flood limit: 8,000 → 10,000
    - Relief value: 6,400 → 5,000
  - /STA TMEMBER ALL INPUT ####
    - In previous releases resulted in warning messages
    - Now turns on global control and will reject new input
      - IMPACT if you change the value to be below what is currently queued
  - Use of DFSOTMA descriptor reduces the total entries for OTMA clients or members to 254 (instead of 255)

O2_ IMS 13 TM: 18

Older releases of IMS provided the following messages for the input message flood condition: DFS1988W, DFS1989E, and DFS0767I.

IMS 11 introduced four new messages: DFS4380W, DFS4381I, DFS4388W, and DFS0793I.

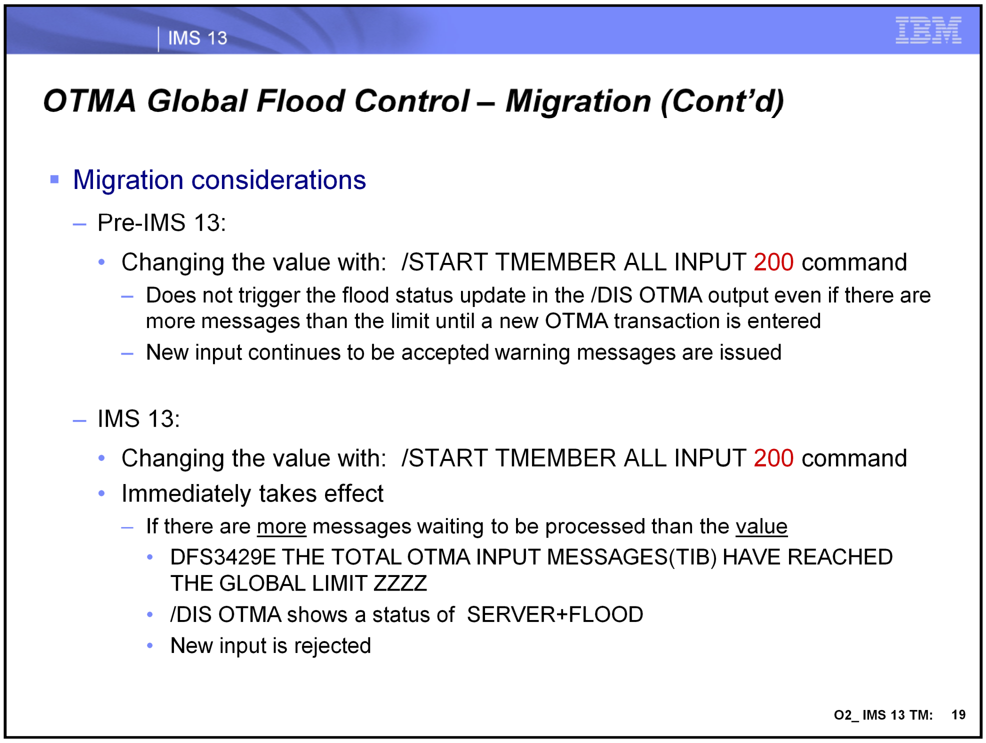
In IMS 13, the existing DFS4388W (when limit is reached and the new global support is not active) and DFS0793I (when flood is relieved) messages have been enhanced to make them clearer and to be consistent with the new messages:

•DFS3428W THE TOTAL OTMA INPUT MESSAGES(TIB) HAVE REACHED XX% OF THE GLOBAL LIMIT ZZZZ

•DFS3429E THE TOTAL OTMA INPUT MESSAGES(TIB) HAVE REACHED THE GLOBAL LIMIT ZZZZ

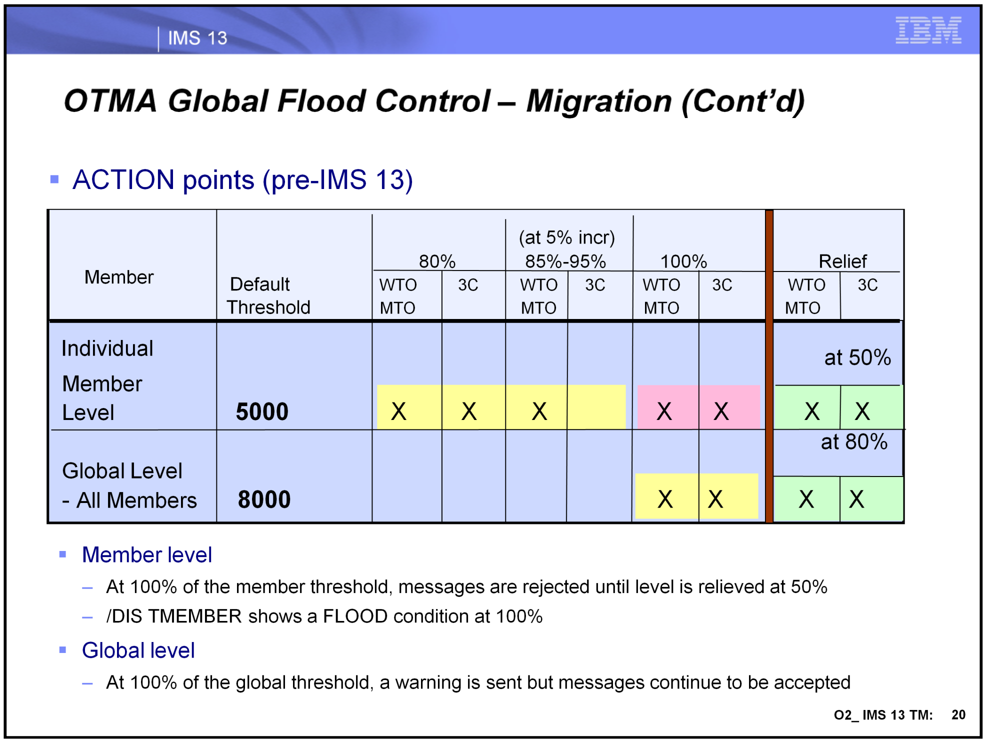
Prior to IMS 13, issuing the /STA TMEMBER ALL INPUT command turned on global flood notification. The same command in IMS 13 enforces the global limit and will cause any new input to be rejected.

If DFSOTMA is used, IMS considers it to be one entry toward the 255 max member entries. This means the total entries allowed for the OTMA clients or members become 254.



OTMA Global Flood Control – Migration (Cont'd)

- Migration considerations
  - Pre-IMS 13:
    - Changing the value with: /START TMEMBER ALL INPUT 200 command
      - Does not trigger the flood status update in the /DIS OTMA output even if there are more messages than the limit until a new OTMA transaction is entered
      - New input continues to be accepted warning messages are issued
  - IMS 13:
    - Changing the value with: /START TMEMBER ALL INPUT 200 command
    - Immediately takes effect
      - If there are more messages waiting to be processed than the value
        - DFS3429E THE TOTAL OTMA INPUT MESSAGES(TIB) HAVE REACHED THE GLOBAL LIMIT ZZZZ
        - /DIS OTMA shows a status of SERVER+FLOOD
        - New input is rejected



OTMA Global Flood Control – Migration (Cont'd)

- ACTION points (pre-IMS 13)

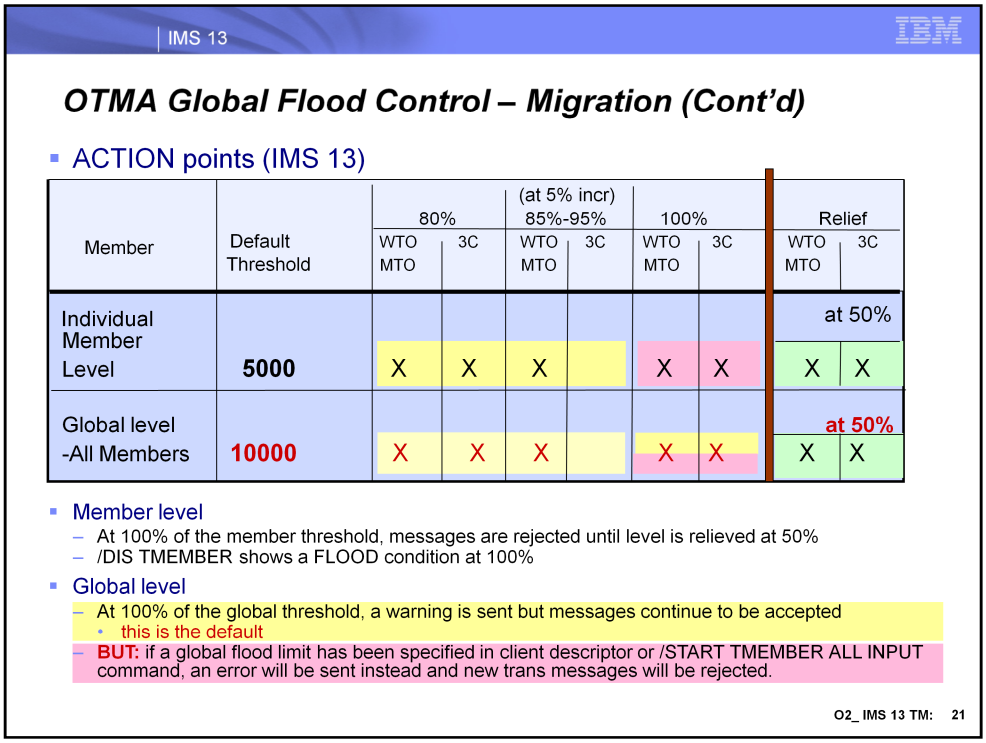
				(at 5% incr)					
Member	Default Threshold	80%		85%-95%		100%		Relief	
		WTO MTO	3C	WTO MTO	3C	WTO MTO	3C	WTO MTO	3C
Individual Member Level	**5000**	X	X	X		X	X	at 50% X	X
Global Level - All Members	**8000**					X	X	at 80% X	X

- Member level
  - At 100% of the member threshold, messages are rejected until level is relieved at 50%
  - /DIS TMEMBER shows a FLOOD condition at 100%
- Global level
  - At 100% of the global threshold, a warning is sent but messages continue to be accepted

The table in this visual shows the action points of the global flood support prior to IMS 13 and the percentages at which WTO and X'3C' protocol action messages are issued. The threshold values shown are the defaults of 5000 for an individual member and 8000 for all members globally.

At the Member Level, both the WTO message and the X'3C' message are sent at 80% as a warning as well as at 100% to indicate a severe condition. At increments of 5% from 85%-95%, only the WTO message is sent. At 50% after the threshold was previously reached, the WTO and X'3C' messages are issued to indicate a relief of the situation.

At the Global Level, the WTO and X'3C' messages are sent both at the 100% mark and subsequently at 80% to indicate a relief of the situation.



IMS 13

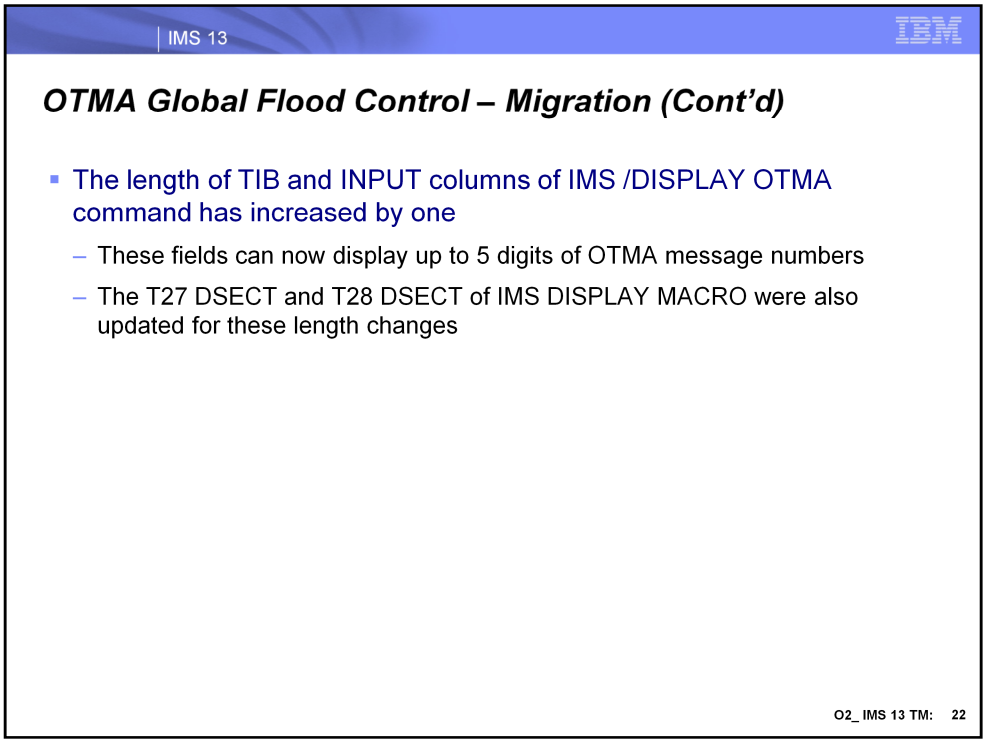
OTMA Global Flood Control – Migration (Cont'd)

- ACTION points (IMS 13)

Member	Default Threshold	80%		(at 5% incr) 85%-95%		100%		Relief	
		WTO MTO	3C	WTO MTO	3C	WTO MTO	3C	WTO MTO	3C
Individual Member Level	**5000**	X	X	X		X	X	at 50% X	X
Global level -All Members	**10000**	X	X	X		X	X	**at 50%** X	X

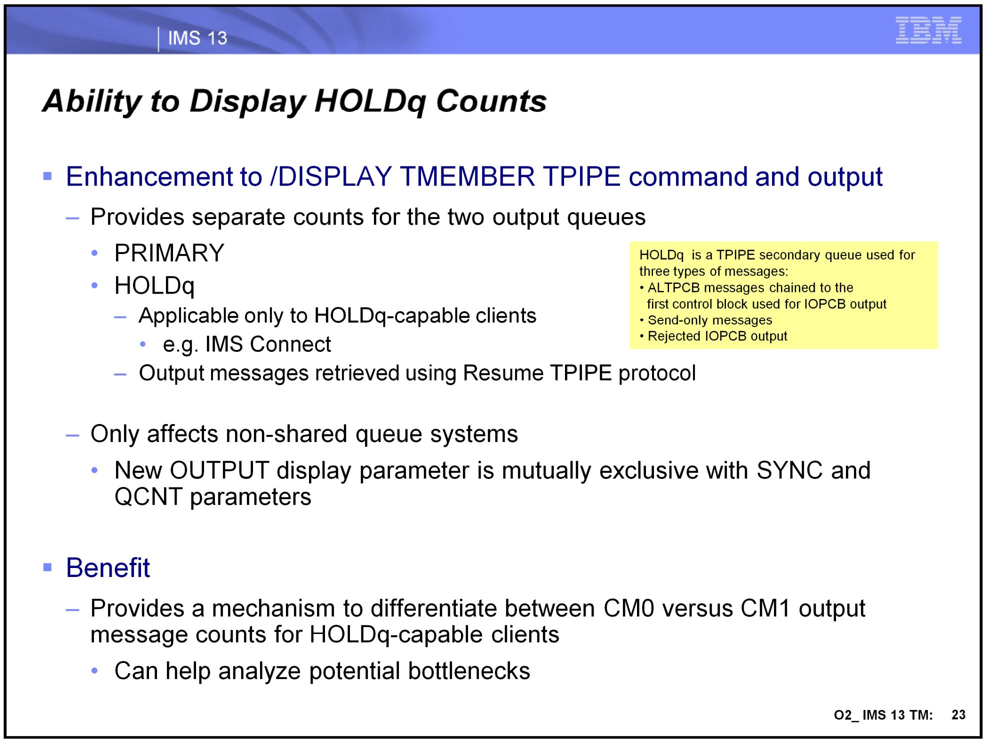
- Member level
  - At 100% of the member threshold, messages are rejected until level is relieved at 50%
  - /DIS TMEMBER shows a FLOOD condition at 100%
- Global level
  - At 100% of the global threshold, a warning is sent but messages continue to be accepted
    - this is the default
  - **BUT:** if a global flood limit has been specified in client descriptor or /START TMEMBER ALL INPUT command, an error will be sent instead and new trans messages will be rejected.

In IMS 13, the default for global flood controls continues to be a warning message but the default threshold is now at 10000 and the relief at 50%. However, if the global flood limits have been activated by either specification of the DFSOTMA client member descriptor or the /STA TMEMBER ALL INPUT command, reaching the 100% mark results in rejection of new transaction messages until the flood has been relieved at 50%. Additionally, warning messages will be issued at the 80% level.



OTMA Global Flood Control – Migration (Cont’d)

- The length of TIB and INPUT columns of IMS /DISPLAY OTMA command has increased by one
  - These fields can now display up to 5 digits of OTMA message numbers
  - The T27 DSECT and T28 DSECT of IMS DISPLAY MACRO were also updated for these length changes


IMS 13

Ability to Display HOLDq Counts

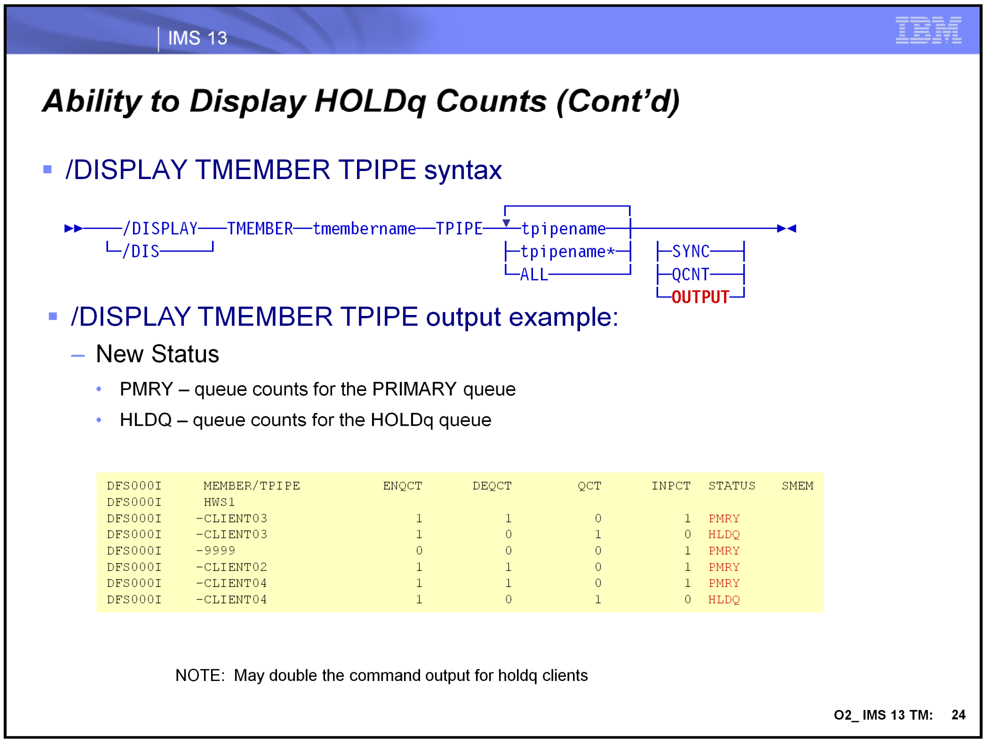
- Enhancement to /DISPLAY TMEMBER TPIPE command and output
  - Provides separate counts for the two output queues
    - PRIMARY
    - HOLDq
      - Applicable only to HOLDq-capable clients
        - e.g. IMS Connect
      - Output messages retrieved using Resume TPIPE protocol
  - Only affects non-shared queue systems
    - New OUTPUT display parameter is mutually exclusive with SYNC and QCNT parameters
- Benefit
  - Provides a mechanism to differentiate between CM0 versus CM1 output message counts for HOLDq-capable clients
    - Can help analyze potential bottlenecks

HOLDq is a TPIPE secondary queue used for three types of messages:
- ALTPCB messages chained to the first control block used for IOPCB output
- Send-only messages
- Rejected IOPCB output

O2_ IMS 13 TM: 23

IMS OTMA provides two queues for asynchronous output destinations that support “HOLDq”-capable clients (e.g.,. IMS Connect). Send/Receive responses for Commit Mode 1 (Commit-then-Send) transactions are queued to the PRIMARY queue. Asynchronous messages (from SendOnly transactions or Alternate IOPCB inserts) are queued to the HOLDq queue and are retrieved using Resume TPIPE protocol. Prior to IMS 13, the existing /DISPLAY command output reports the total messages queued to both queues. There is no external method to determine whether the messages are on the PRIMARY or HOLDq or both.

IMS 13 adds a new OUTPUT parameter to the /DISPLAY TMEMBER TPIPE command to report the PRIMARY and HOLDq counts separately. Note that is enhancement is valid only for HOLDq-capable OTMA clients and does not apply to non-HOLDq clients such as WebSphere MQ. This option is also only available for non-shared queues IMS systems and will be ignored when IMS has shared queues enabled. The OUTPUT parameter is mutually exclusive with the SYNC and QCNT parameters.



Ability to Display HOLDq Counts (Cont'd)

- /DISPLAY TMEMBER TPIPE syntax

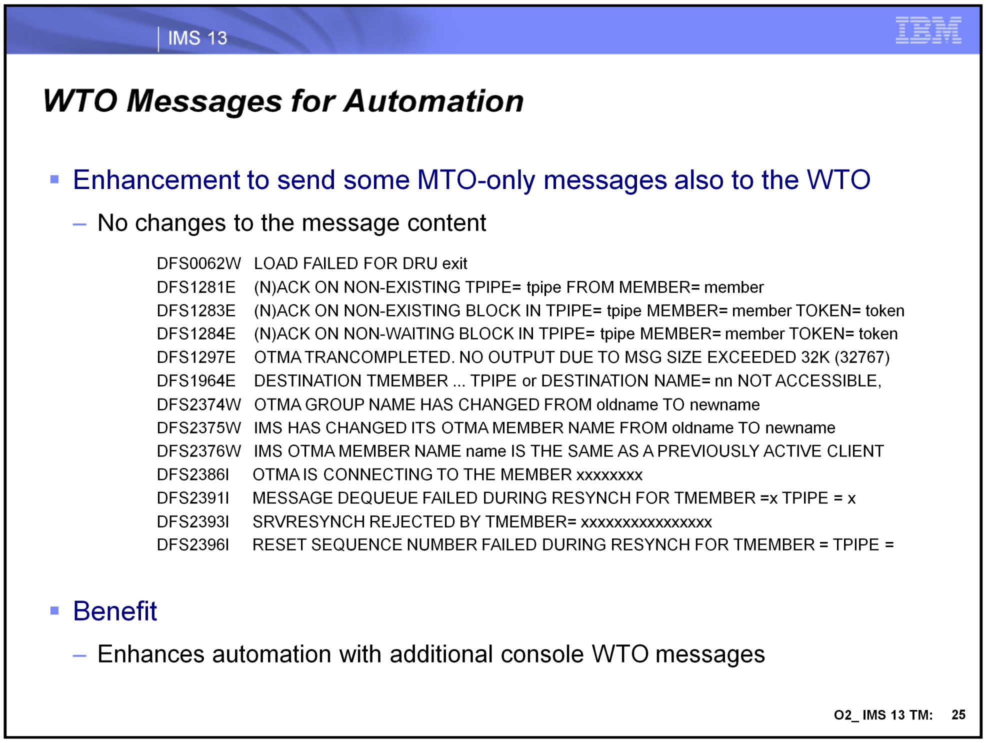
```
                                         ┌──────────────┐
                                         ▼              │
►►──/DISPLAY──TMEMBER─tmembername─TPIPE─┬─tpipename──┬─────────────►◄
  └─/DIS────┘                           ├─tpipename*─┤  ├─SYNC───┤
                                        └─ALL────────┘  ├─QCNT───┤
                                                        └─OUTPUT─┘
```

- /DISPLAY TMEMBER TPIPE output example:
  - New Status
    - PMRY – queue counts for the PRIMARY queue
    - HLDQ – queue counts for the HOLDq queue

```
DFS000I     MEMBER/TPIPE             ENQCT     DEQCT       QCT     INPCT  STATUS   SMEM
DFS000I     HWS1
DFS000I    -CLIENT03                     1         1         0         1  PMRY
DFS000I    -CLIENT03                     1         0         1         0  HLDQ
DFS000I    -9999                         0         0         0         1  PMRY
DFS000I    -CLIENT02                     1         1         0         1  PMRY
DFS000I    -CLIENT04                     1         1         0         1  PMRY
DFS000I    -CLIENT04                     1         0         1         0  HLDQ
```

NOTE: May double the command output for holdq clients

This visual gives an example of the new syntax of the /DIS TMEMBER TPIPE command with the new OUTPUT parameter along with an example of the OUTPUT display. Note that for HOLDq client output, the ability to report on both the PRIMARY and HOLDq queues could result in an increase of the display output. Specifying the OUTPUT parameter will double the number of lines for the display. There may be some impact when using the ALL or masked TPIPE names for TMEMBERs with a large number of TPIPEs.



WTO Messages for Automation

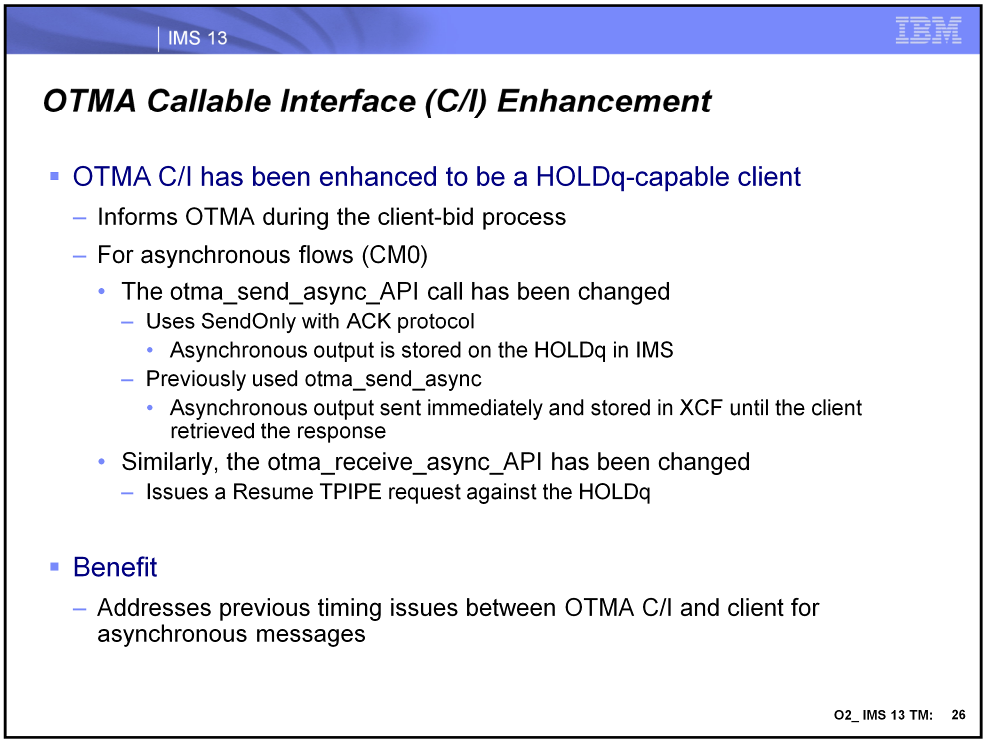
- Enhancement to send some MTO-only messages also to the WTO
  - No changes to the message content

DFS0062W	LOAD FAILED FOR DRU exit
DFS1281E	(N)ACK ON NON-EXISTING TPIPE= tpipe FROM MEMBER= member
DFS1283E	(N)ACK ON NON-EXISTING BLOCK IN TPIPE= tpipe MEMBER= member TOKEN= token
DFS1284E	(N)ACK ON NON-WAITING BLOCK IN TPIPE= tpipe MEMBER= member TOKEN= token
DFS1297E	OTMA TRANCOMPLETED. NO OUTPUT DUE TO MSG SIZE EXCEEDED 32K (32767)
DFS1964E	DESTINATION TMEMBER ... TPIPE or DESTINATION NAME= nn NOT ACCESSIBLE,
DFS2374W	OTMA GROUP NAME HAS CHANGED FROM oldname TO newname
DFS2375W	IMS HAS CHANGED ITS OTMA MEMBER NAME FROM oldname TO newname
DFS2376W	IMS OTMA MEMBER NAME name IS THE SAME AS A PREVIOUSLY ACTIVE CLIENT
DFS2386I	OTMA IS CONNECTING TO THE MEMBER xxxxxxxx
DFS2391I	MESSAGE DEQUEUE FAILED DURING RESYNCH FOR TMEMBER =x TPIPE = x
DFS2393I	SRVRESYNCH REJECTED BY TMEMBER= xxxxxxxxxxxxxxxx
DFS2396I	RESET SEQUENCE NUMBER FAILED DURING RESYNCH FOR TMEMBER = TPIPE =

- Benefit
  - Enhances automation with additional console WTO messages

There are many important IMS OTMA messages that have only been sent to the Master Terminal Operator (MTO) LTERM. Many environments that use automation for console Write To Operator (WTO) messages and could also benefit from these OTMA messages. They are now being sent to both the MTO and WTO to the console. There are no changes to the message content.

The auditlog function for the Operations Manager (OM) has not changed and continues to be controlled by the UOM parameter in the DFSDFxxx member of proclib for unsolicited messages.



IMS 13

OTMA Callable Interface (C/I) Enhancement

- OTMA C/I has been enhanced to be a HOLDq-capable client
  - Informs OTMA during the client-bid process
  - For asynchronous flows (CM0)
    - The otma_send_async_API call has been changed
      - Uses SendOnly with ACK protocol
        - Asynchronous output is stored on the HOLDq in IMS
      - Previously used otma_send_async
        - Asynchronous output sent immediately and stored in XCF until the client retrieved the response
    - Similarly, the otma_receive_async_API has been changed
      - Issues a Resume TPIPE request against the HOLDq
- Benefit
  - Addresses previous timing issues between OTMA C/I and client for asynchronous messages

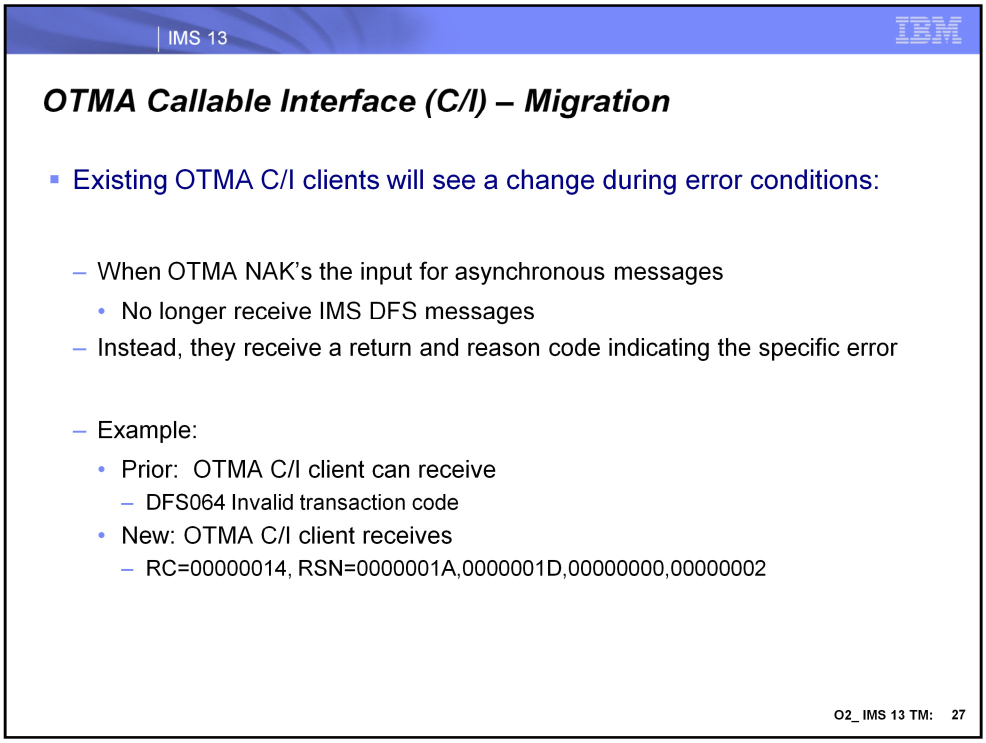
O2_ IMS 13 TM: 26

OTMA Callable Interface (OTMA C/I) provides a high level interface for host infrastructure applications to send and retrieve messages with IMS OTMA using XCF services. Prior to this enhancement, OTMA C/I ***asynchronous*** (CM0 – Commit-then-Send) messages that are sent using otma_send_async_API have the responses sent immediately by OTMA since there is no HOLDQ. OTMA C/I must “store” these messages in XCF until the client issues otma_receive_async_API. This has led to many timing issues between OTMA C/I and the client.

This enhancement changes OTMA C/I to use the OTMA HOLDQ support for asynchronous (CM0) messages. OTMA C/I specifies it is HOLDQ capable on the client-bid. The otma_send_async_API has been changed to use SendOnly with ACK protocol. The asynchronous output will be placed on the HOLDQ. The otma_receive_async_API will now use Resume TPIPE Single to retrieve the output.

These changes to OTMA C/I are internal with the minor impact to customers outlined in the migration section.

This enhancement does not change the synchronous Send-then-Commit (CM1) flows.



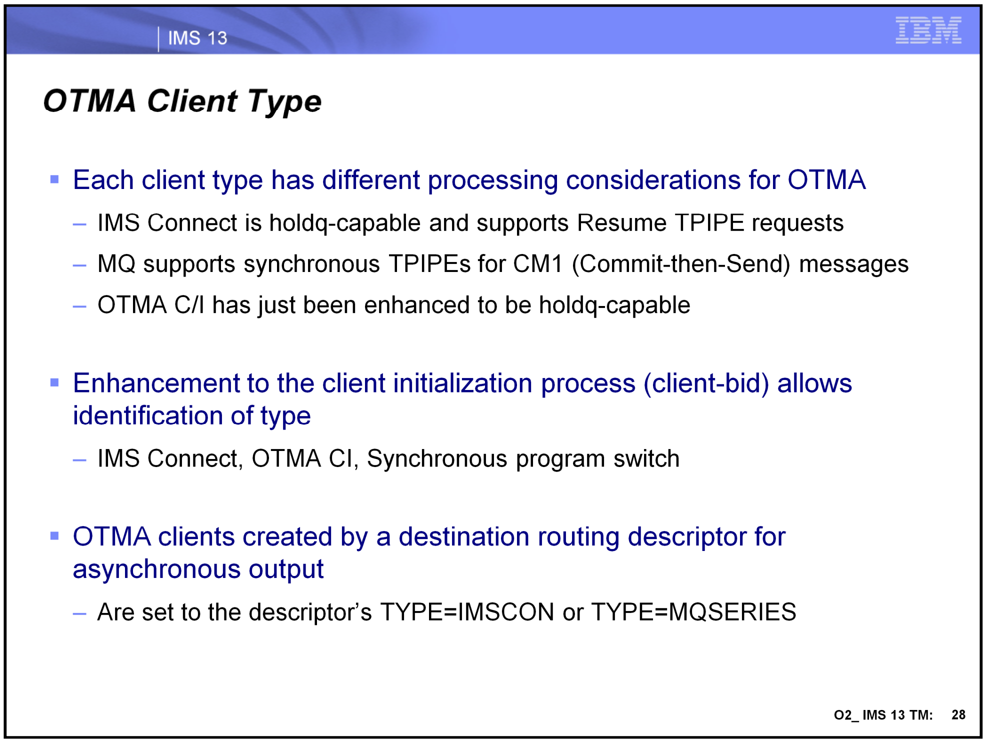
OTMA Callable Interface (C/I) – Migration

- Existing OTMA C/I clients will see a change during error conditions:
  - When OTMA NAK's the input for asynchronous messages
    - No longer receive IMS DFS messages
  - Instead, they receive a return and reason code indicating the specific error
  - Example:
    - Prior: OTMA C/I client can receive
      - DFS064 Invalid transaction code
    - New: OTMA C/I client receives
      - RC=00000014, RSN=0000001A,0000001D,00000000,00000002

The OTMA C/I client using asynchronous messaging will no longer receive IMS DFS messages when the input is NAK'd by OTMA. OTMA C/I will be provided return and reason codes to indicate the specific error.

Example: rather than DFS064 for an invalid transaction code, the OTMA C/I client will now receive RC=00000014, RSN=0000001A,0000001D,00000000,00000002.

RC=x'14' indicates OTMA NAK codes, RSN=x'1A' indicates message cancelled, x'1D' indicates SMB not found (invalid transaction code),



IMS 13

OTMA Client Type

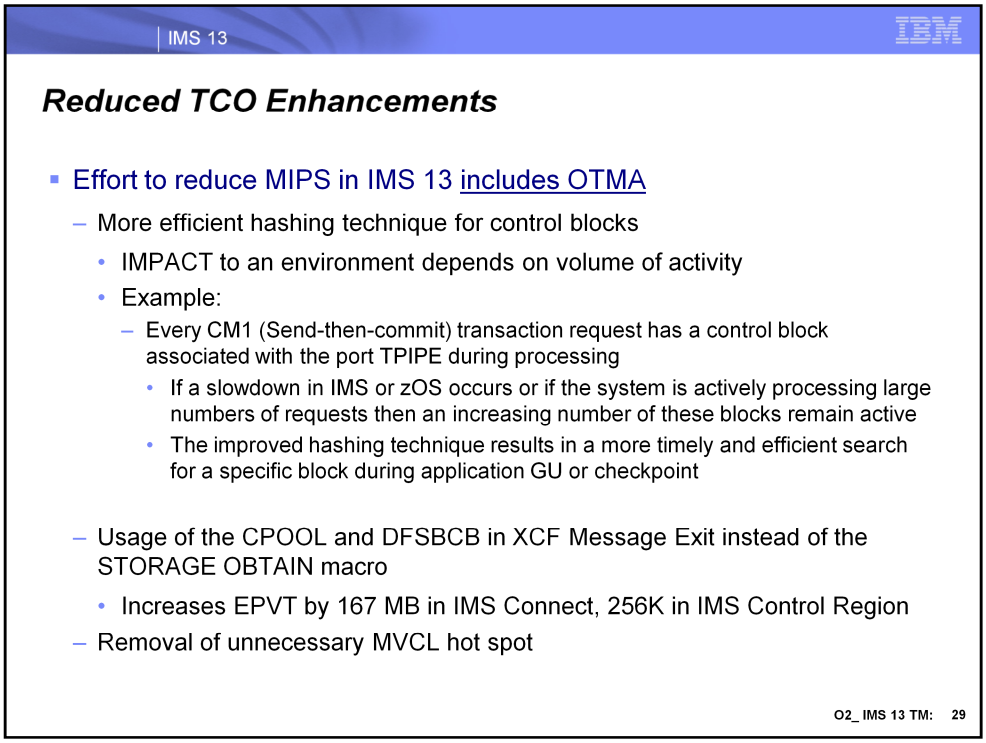
- Each client type has different processing considerations for OTMA
  - IMS Connect is holdq-capable and supports Resume TPIPE requests
  - MQ supports synchronous TPIPEs for CM1 (Commit-then-Send) messages
  - OTMA C/I has just been enhanced to be holdq-capable
- Enhancement to the client initialization process (client-bid) allows identification of type
  - IMS Connect, OTMA CI, Synchronous program switch
- OTMA clients created by a destination routing descriptor for asynchronous output
  - Are set to the descriptor's TYPE=IMSCON or TYPE=MQSERIES

O2_ IMS 13 TM: 28

IMS OTMA clients (eg. IMS Connect, MQ, OTMA C/I) each have unique processing considerations. IMS Connect is holdq-capable and supports the Resume TPIPE protocol. MQ supports synchronous TPIPEs for Commit Mode 1 (Commit-then-Send) messages. In many cases the differences in client type impact how OTMA processes client requests and messages. The client initialization process (client-bid) has been enhanced to allow clients to declare their "type" to OTMA. Knowing this will allow OTMA to optimize certain functions for greater efficiency.

IMS Connect in IMS 13, OTMA C/I, and the internal DFSYICAL for Synchronous Program Switch identify their client type to OTMA.

IMS OTMA Destination Routing Descriptors can be used to create OTMA clients (TMEMBER) for asynchronous output. If the TMEMBER did not previously exist, the client type is set for TYPE=IMSCON or TYPE=MQSERIES.



IMS 13

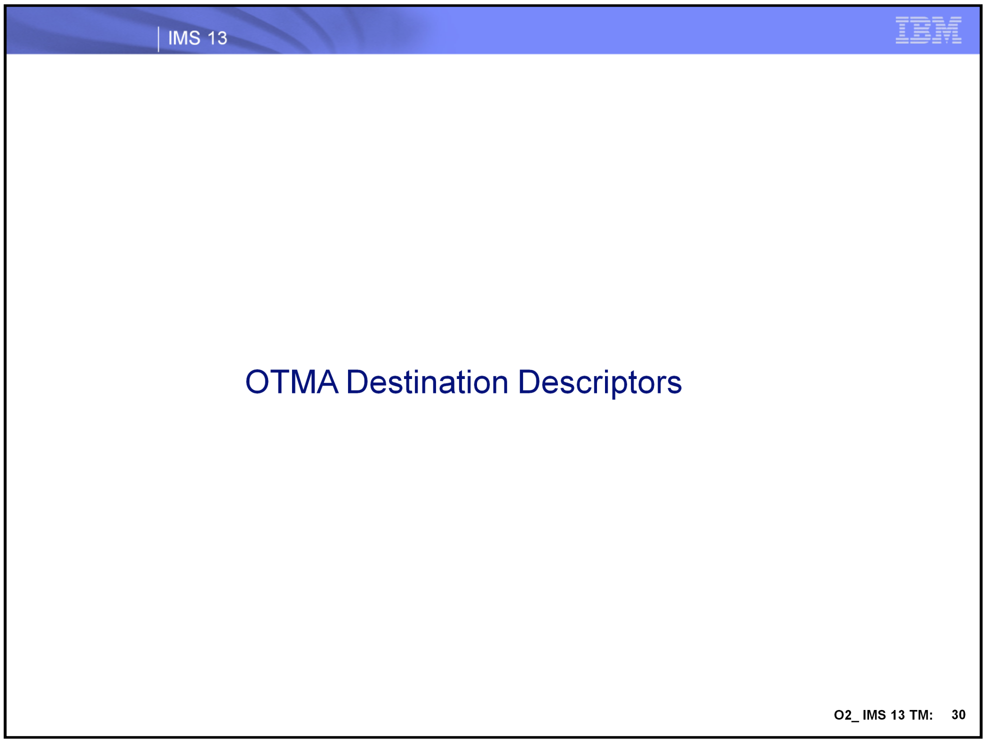
Reduced TCO Enhancements

- Effort to reduce MIPS in IMS 13 includes OTMA
  - More efficient hashing technique for control blocks
    - IMPACT to an environment depends on volume of activity
    - Example:
      - Every CM1 (Send-then-commit) transaction request has a control block associated with the port TPIPE during processing
        - If a slowdown in IMS or zOS occurs or if the system is actively processing large numbers of requests then an increasing number of these blocks remain active
        - The improved hashing technique results in a more timely and efficient search for a specific block during application GU or checkpoint
  - Usage of the CPOOL and DFSBCB in XCF Message Exit instead of the STORAGE OBTAIN macro
    - Increases EPVT by 167 MB in IMS Connect, 256K in IMS Control Region
  - Removal of unnecessary MVCL hot spot

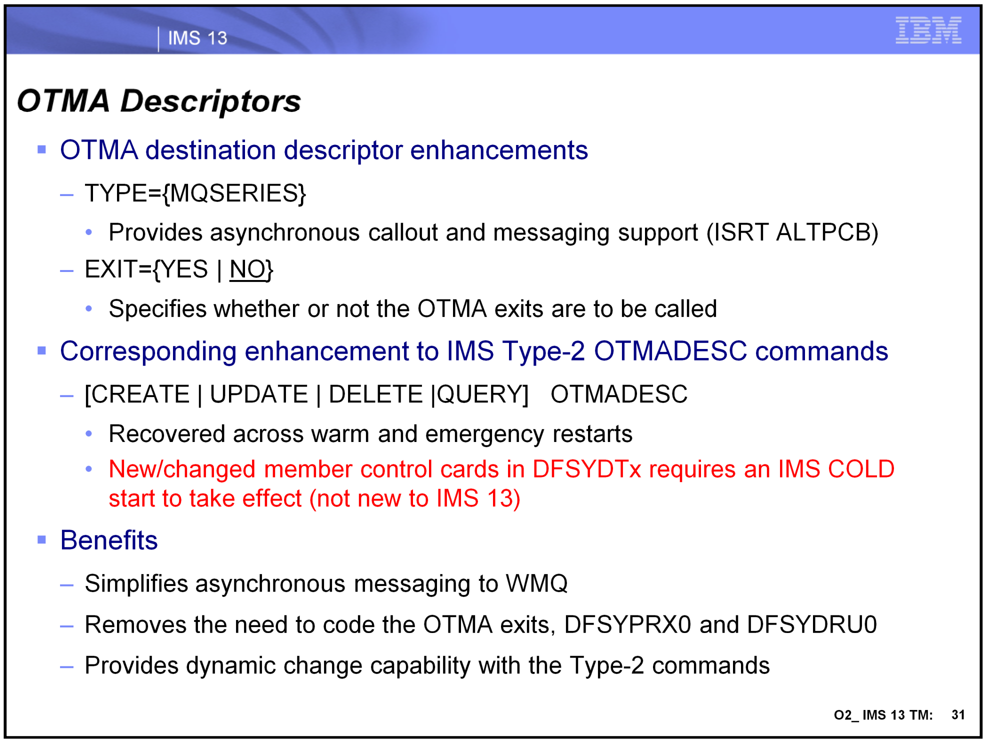
O2_ IMS 13 TM: 29

IMS 13 introduces a more efficient hashing mechanism for OTMA control blocks associated with transaction messages. The impact to a specific IMS environment depends on its use of OTMA.

The cell pool macro (CPOOL) provides programs with another way of obtaining virtual storage other than the STORAGE OBTAIN macro. CPOOL provides centralized, high-performance cell management services and DFSBCB is an IMS storage manager service which manages storage in specific storage pools.



OTMA Destination Descriptors



IMS 13

OTMA Descriptors

- OTMA destination descriptor enhancements
  - TYPE={MQSERIES}
    - Provides asynchronous callout and messaging support (ISRT ALTPCB)
  - EXIT={YES | NO}
    - Specifies whether or not the OTMA exits are to be called
- Corresponding enhancement to IMS Type-2 OTMADESC commands
  - [CREATE | UPDATE | DELETE |QUERY]   OTMADESC
    - Recovered across warm and emergency restarts
    - New/changed member control cards in DFSYDTx requires an IMS COLD start to take effect (not new to IMS 13)
- Benefits
  - Simplifies asynchronous messaging to WMQ
  - Removes the need to code the OTMA exits, DFSYPRX0 and DFSYDRU0
  - Provides dynamic change capability with the Type-2 commands

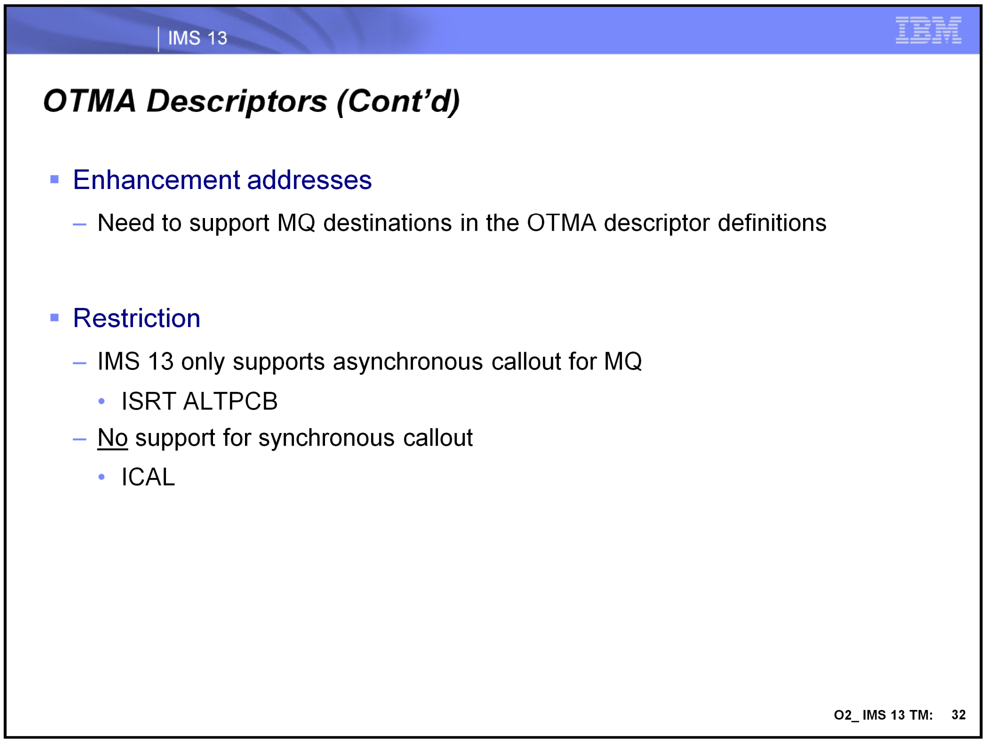
The destination descriptor for IMS asynchronous callout has been expanded to introduce a new type, MQSERIES, to allow customers to define message switch destinations (from ALT IOPCB) as WebSphere MQ. This new type of descriptor for WebSphere MQ externalizes the definitions in the DFSYDTx member of IMS.PROCLIB rather than forcing customers to code OTMA routing exits (DFSYPRX0 & DFSYDRU0). Thirteen new parameters for this new descriptor are introduced to prepare the MQMD structure in the OTMA user data prefix.

Dynamic addition, update, or deletion of the destination descriptors for WMQ can be performed using the type-2 commands:

CREATE OTMADESC

UPDATE OTMADESC

DELETE OTMADESC



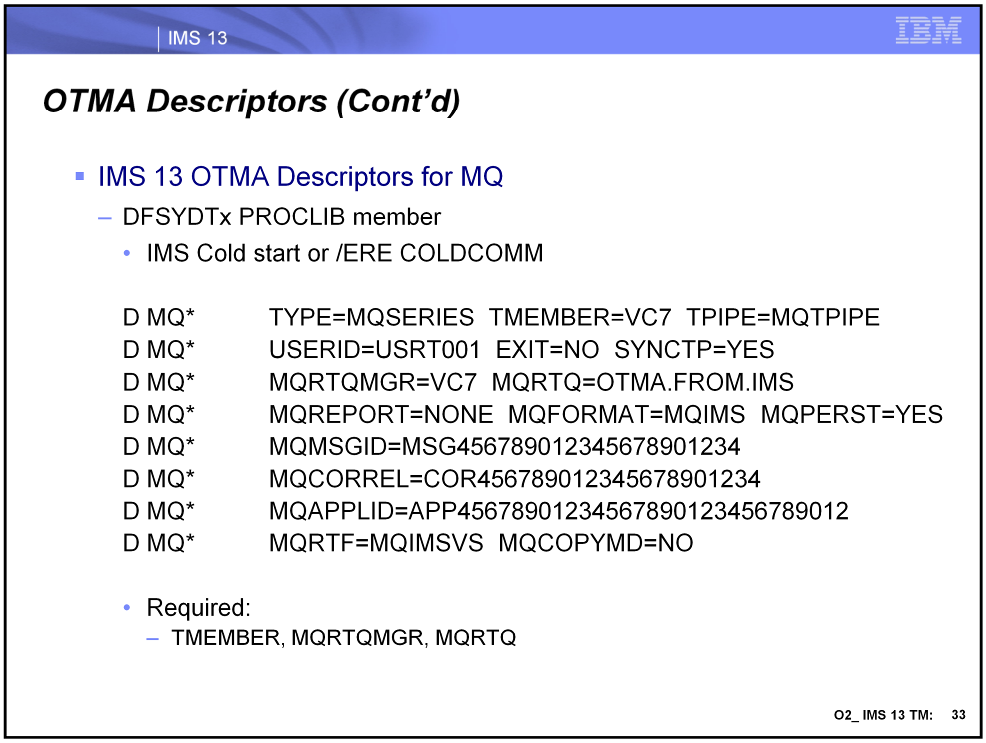
IMS 13

OTMA Descriptors (Cont'd)

- Enhancement addresses
  - Need to support MQ destinations in the OTMA descriptor definitions
- Restriction
  - IMS 13 only supports asynchronous callout for MQ
    - ISRT ALTPCB
  - No support for synchronous callout
    - ICAL

O2_ IMS 13 TM: 32

The new destination descriptor support for WebSphere MQ destinations cannot be used for synchronous callout ICAL calls.



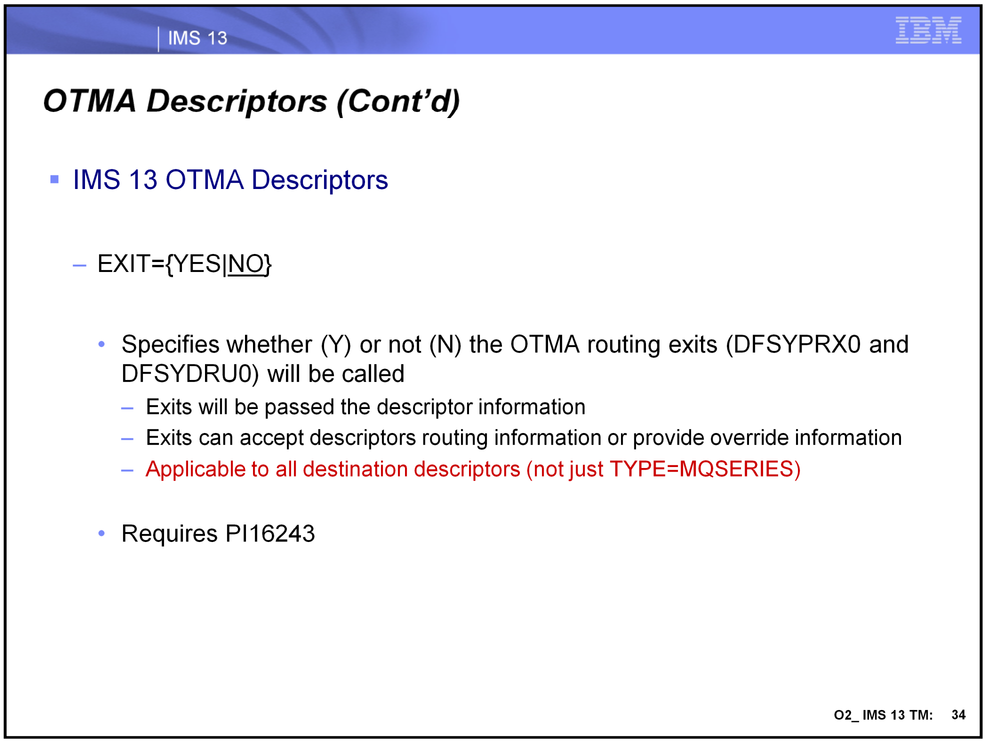
IMS 13

OTMA Descriptors (Cont'd)

- IMS 13 OTMA Descriptors for MQ
  - DFSYDTx PROCLIB member
    - IMS Cold start or /ERE COLDCOMM

```
D MQ*        TYPE=MQSERIES  TMEMBER=VC7  TPIPE=MQTPIPE
D MQ*        USERID=USRT001  EXIT=NO  SYNCTP=YES
D MQ*        MQRTQMGR=VC7  MQRTQ=OTMA.FROM.IMS
D MQ*        MQREPORT=NONE  MQFORMAT=MQIMS  MQPERST=YES
D MQ*        MQMSGID=MSG4567890123456789012 34
D MQ*        MQCORREL=COR456789012345678901234
D MQ*        MQAPPLID=APP45678901234567890123456789012
D MQ*        MQRTF=MQIMSVS  MQCOPYMD=NO
```

- Required:
  - TMEMBER, MQRTQMGR, MQRTQ



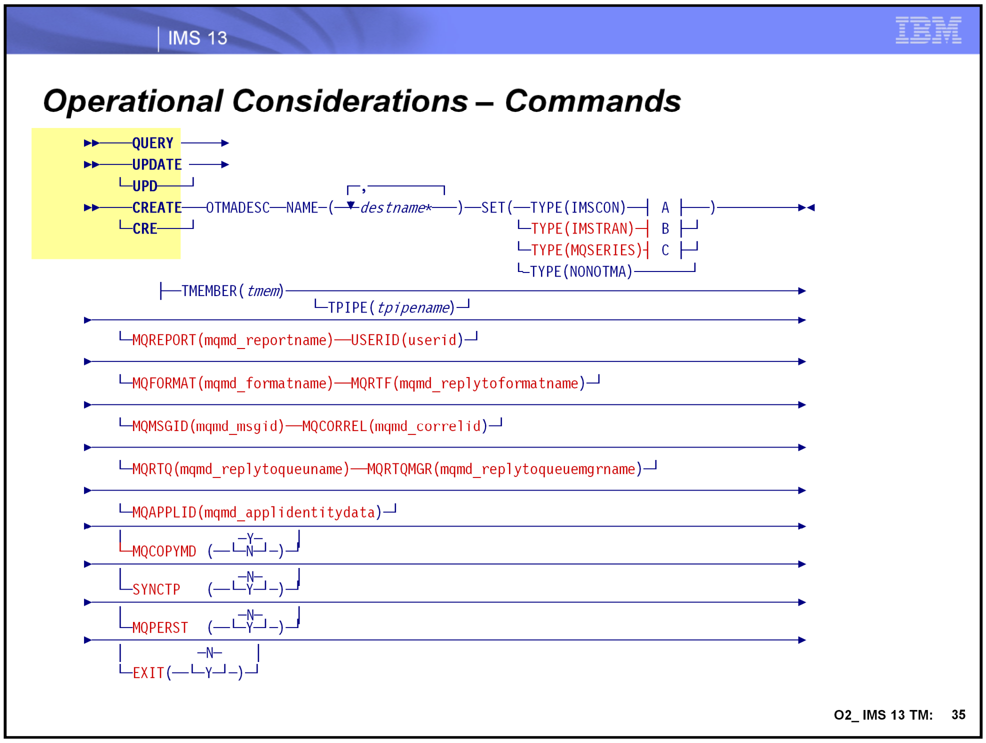
IMS 13

OTMA Descriptors (Cont'd)

- IMS 13 OTMA Descriptors
  - EXIT={YES|NO}
    - Specifies whether (Y) or not (N) the OTMA routing exits (DFSYPRX0 and DFSYDRU0) will be called
      - Exits will be passed the descriptor information
      - Exits can accept descriptors routing information or provide override information
      - Applicable to all destination descriptors (not just TYPE=MQSERIES)
    - Requires PI16243

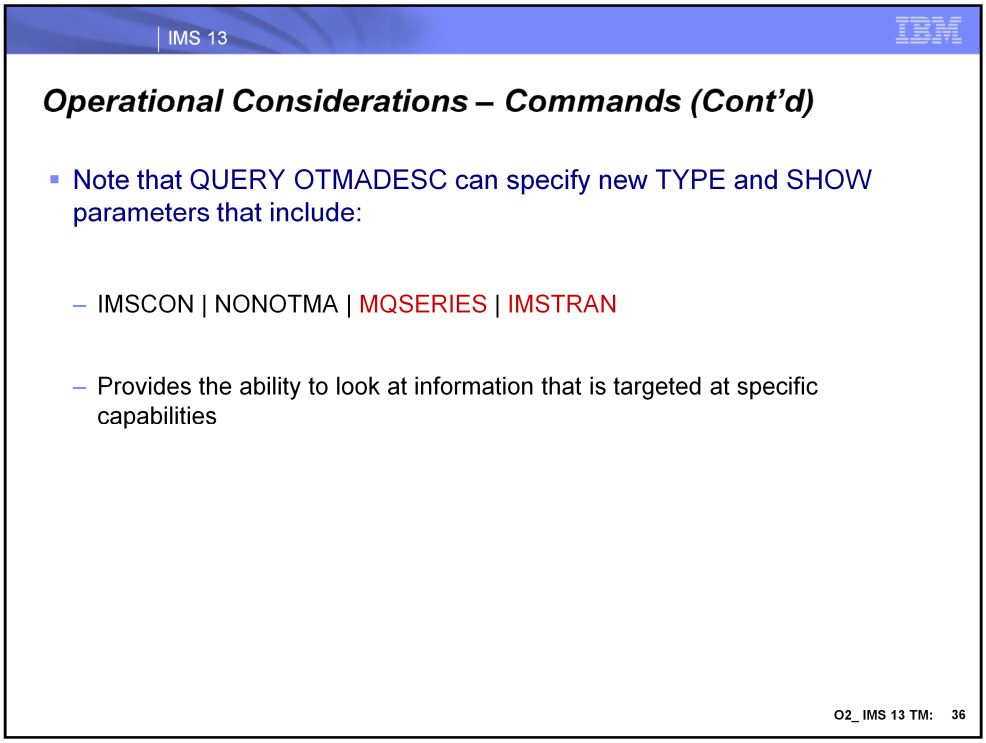
O2_ IMS 13 TM: 34

A new parameter EXIT= is added to the destination descriptor. When EXIT=YES is specified and OTMA routing exits (DFSYPRX0 and DFSYDRU0) are found in the IMS, the exits are allowed to override the routing information specified in the descriptor. Other IMS exits such as DFSYIOE0 are not affected by this new parameter. The default setting of EXIT is NO which means the OTMA routing exits, if found, cannot be called to override the routing information in the descriptor.



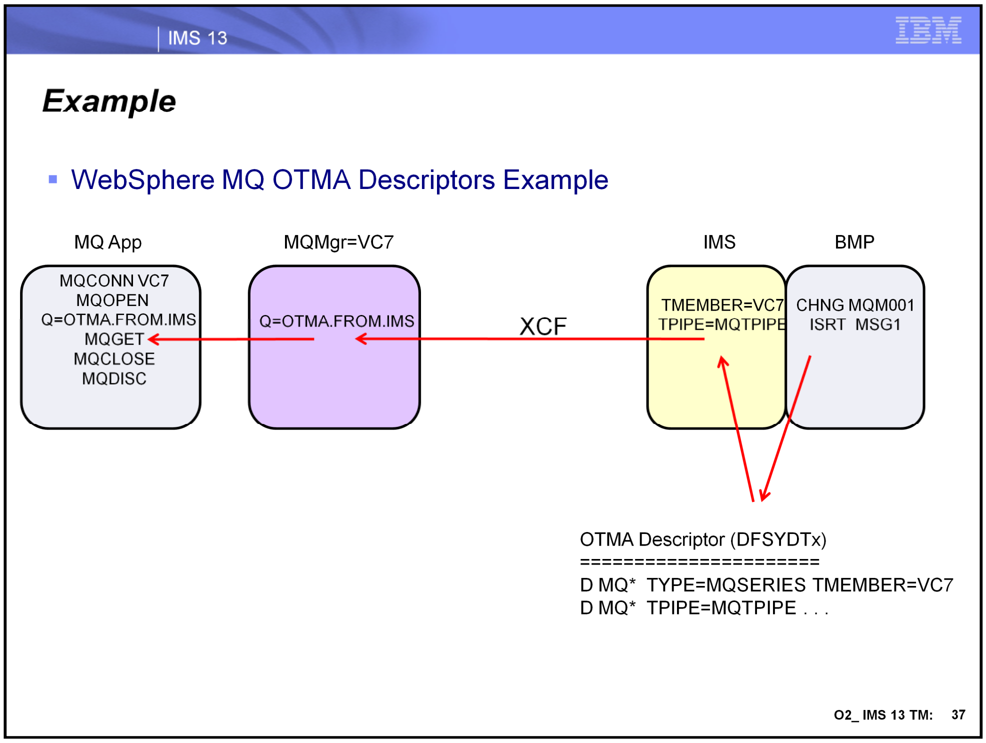
Operational Considerations – Commands

```
▶▶──QUERY ───▶
▶▶──UPDATE ──▶
    └UPD───┘
                                    ┌,──────┐
▶▶──CREATE──OTMADESC──NAME─(─▼─destname*──)──SET(──TYPE(IMSCON)──┤ A ├──)──────▶◀
    └CRE───┘                                       └TYPE(IMSTRAN)─┤ B ├┘
                                                   └TYPE(MQSERIES)┤ C ├┘
                                                   └TYPE(NONOTMA)──────┘

   ├──TMEMBER(tmem)─────────────────────────────────────────────────▶
                        └TPIPE(tpipename)─┘
▶──────────────────────────────────────────────────────────────────▶
  └MQREPORT(mqmd_reportname)──USERID(userid)─┘
▶──────────────────────────────────────────────────────────────────▶
  └MQFORMAT(mqmd_formatname)──MQRTF(mqmd_replytoformatname)─┘
▶──────────────────────────────────────────────────────────────────▶
  └MQMSGID(mqmd_msgid)──MQCORREL(mqmd_correlid)─┘
▶──────────────────────────────────────────────────────────────────▶
  └MQRTQ(mqmd_replytoqueuename)──MQRTQMGR(mqmd_replytoqueuemgrname)─┘
▶──────────────────────────────────────────────────────────────────▶
  └MQAPPLID(mqmd_applidentitydata)─┘
▶──────────────────────────────────────────────────────────────────▶
  │              ─Y─      │
  └MQCOPYMD  (──└─N─┘─)──┘
▶──────────────────────────────────────────────────────────────────▶
  │              ─N─      │
  └SYNCTP    (──└─Y─┘─)──┘
▶──────────────────────────────────────────────────────────────────▶
  │              ─N─      │
  └MQPERST   (──└─Y─┘─)──┘
▶──────────────────────────────────────────────────────────────────▶
  │         ─N─      │
  └EXIT(──└─Y─┘─)──┘
```



Operational Considerations – Commands (Cont'd)

- Note that QUERY OTMADESC can specify new TYPE and SHOW parameters that include:
  - IMSCON | NONOTMA | MQSERIES | IMSTRAN
  - Provides the ability to look at information that is targeted at specific capabilities



Example

- WebSphere MQ OTMA Descriptors Example

MQ App

MQCONN VC7
MQOPEN
Q=OTMA.FROM.IMS
MQGET
MQCLOSE
MQDISC

MQMgr=VC7

Q=OTMA.FROM.IMS

XCF

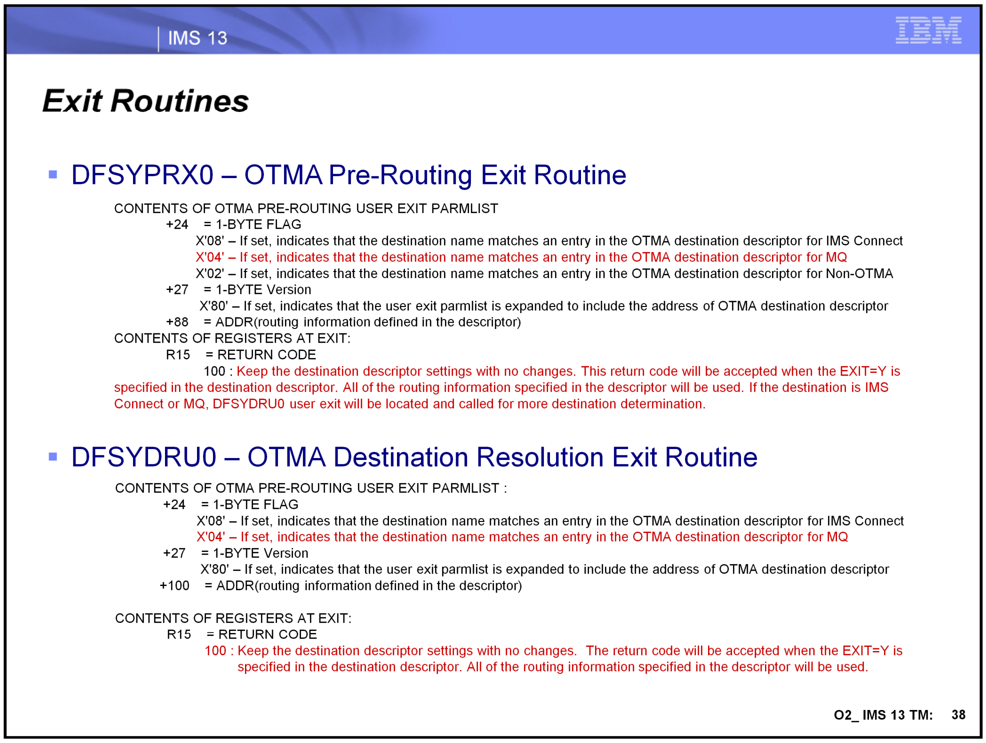
IMS

TMEMBER=VC7
TPIPE=MQTPIPE

BMP

CHNG MQM001
ISRT MSG1

```
OTMA Descriptor (DFSYDTx)
======================
D MQ*  TYPE=MQSERIES TMEMBER=VC7
D MQ*  TPIPE=MQTPIPE . . .
```



IMS 13

Exit Routines

- DFSYPRX0 – OTMA Pre-Routing Exit Routine

```
CONTENTS OF OTMA PRE-ROUTING USER EXIT PARMLIST
       +24    = 1-BYTE FLAG
               X'08' – If set, indicates that the destination name matches an entry in the OTMA destination descriptor for IMS Connect
               X'04' – If set, indicates that the destination name matches an entry in the OTMA destination descriptor for MQ
               X'02' – If set, indicates that the destination name matches an entry in the OTMA destination descriptor for Non-OTMA
       +27    = 1-BYTE Version
                X'80' – If set, indicates that the user exit parmlist is expanded to include the address of OTMA destination descriptor
       +88    = ADDR(routing information defined in the descriptor)
CONTENTS OF REGISTERS AT EXIT:
       R15    = RETURN CODE
                100 : Keep the destination descriptor settings with no changes. This return code will be accepted when the EXIT=Y is
specified in the destination descriptor. All of the routing information specified in the descriptor will be used. If the destination is IMS
Connect or MQ, DFSYDRU0 user exit will be located and called for more destination determination.
```

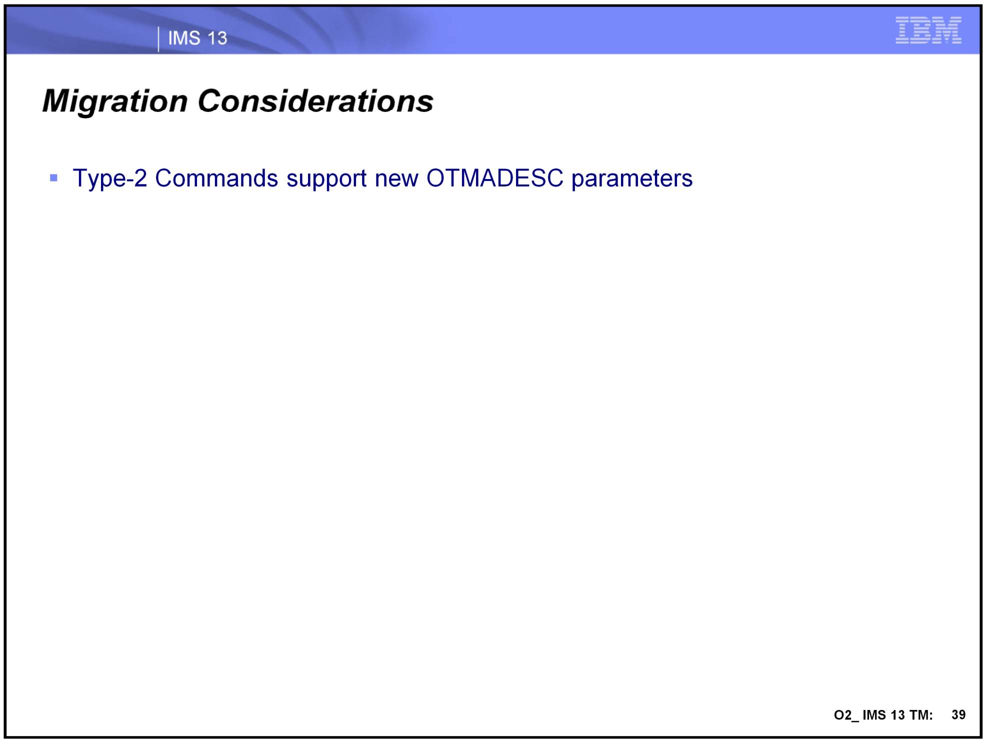
- DFSYDRU0 – OTMA Destination Resolution Exit Routine

```
CONTENTS OF OTMA PRE-ROUTING USER EXIT PARMLIST :
      +24    = 1-BYTE FLAG
              X'08' – If set, indicates that the destination name matches an entry in the OTMA destination descriptor for IMS Connect
              X'04' – If set, indicates that the destination name matches an entry in the OTMA destination descriptor for MQ
      +27    = 1-BYTE Version
               X'80' – If set, indicates that the user exit parmlist is expanded to include the address of OTMA destination descriptor
     +100    = ADDR(routing information defined in the descriptor)

CONTENTS OF REGISTERS AT EXIT:
       R15    = RETURN CODE
               100 : Keep the destination descriptor settings with no changes.  The return code will be accepted when the EXIT=Y is
                     specified in the destination descriptor. All of the routing information specified in the descriptor will be used.
```

O2_ IMS 13 TM: 38

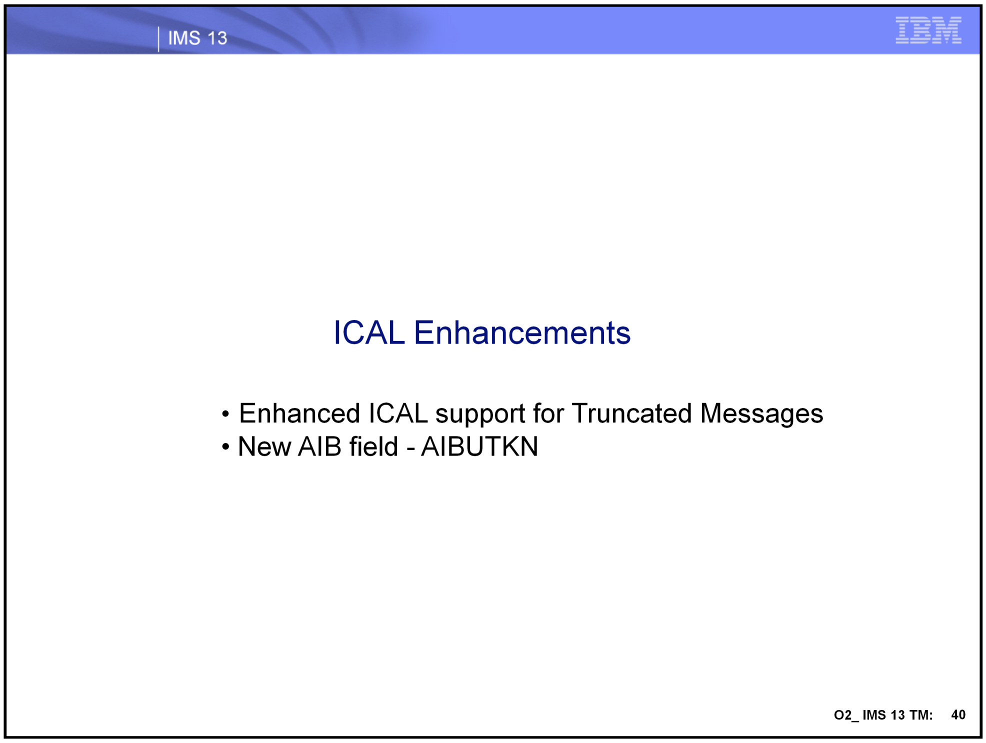
A new flag, new input field, new return code have been added into the user exit parameter list in the DFSYPRX0 exit routine. The address at +88 points to a location which stores the routing information defined in the descriptor for WebSphere MQ and IMS Connect. If the destination name matches a non-OTMA destination in the descriptor or it does not match any entry in the OTMA destination descriptor, this new field will be zero. For the destination as IMS Connect, please see the TMAMICON_DESCRIPTOR DSECT mapping for the detailed routing information. For the destination as MQ, please see the TMAMMQS_DESCRIPTOR DSECT mapping for the detailed routing information.

Similarly, DFSYDRU0 has added a new flag, new input field and a new return code into the user exit parameter list. The address at +100 points to a location which stores the routing information defined in the descriptor for WebSphere MQ and IMS Connect. If the destination name matches a non-OTMA destination in the descriptor or it does not match any entry in the OTMA destination descriptor, this new field will be zero.



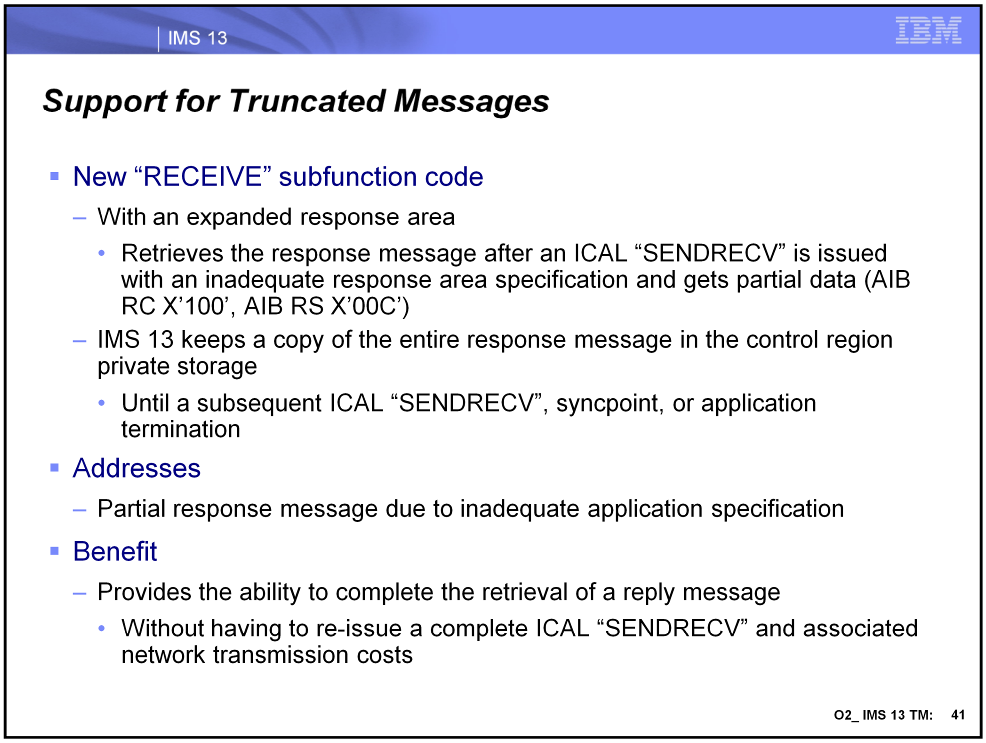
Migration Considerations

- Type-2 Commands support new OTMADESC parameters



ICAL Enhancements

- Enhanced ICAL support for Truncated Messages
- New AIB field - AIBUTKN

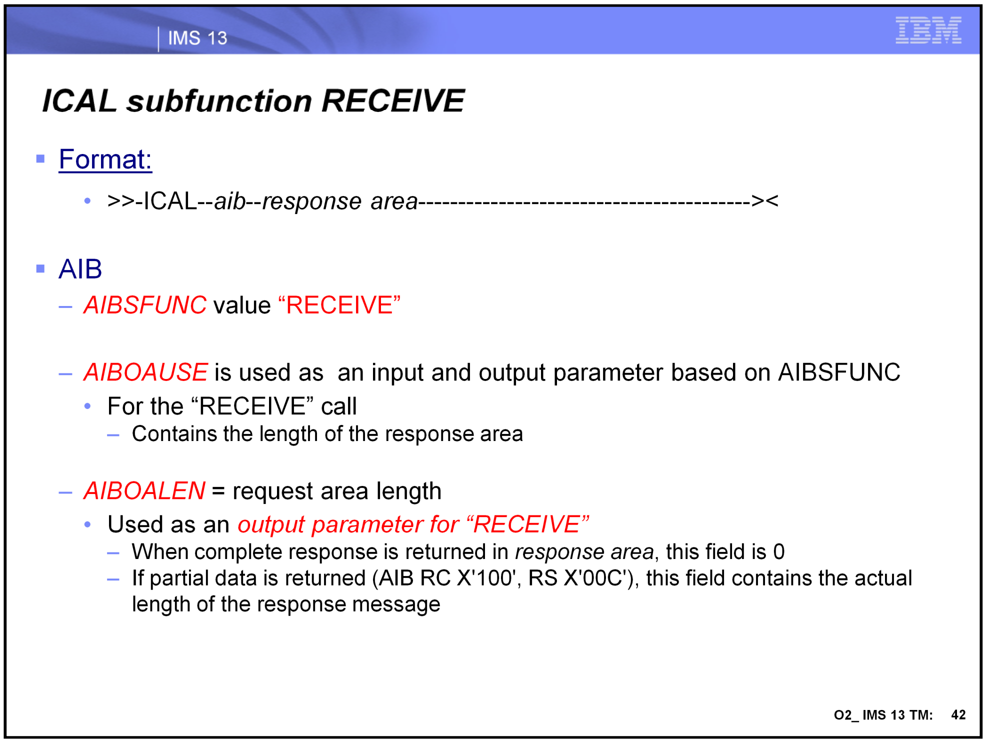


Support for Truncated Messages

- New “RECEIVE” subfunction code
  - With an expanded response area
    - Retrieves the response message after an ICAL “SENDRECV” is issued with an inadequate response area specification and gets partial data (AIB RC X’100’, AIB RS X’00C’)
  - IMS 13 keeps a copy of the entire response message in the control region private storage
    - Until a subsequent ICAL “SENDRECV”, syncpoint, or application termination
- Addresses
  - Partial response message due to inadequate application specification
- Benefit
  - Provides the ability to complete the retrieval of a reply message
    - Without having to re-issue a complete ICAL “SENDRECV” and associated network transmission costs

IMS 13 introduces a new subfunction of “RECEIVE” for the ICAL. This enhancement supports the retrieval of an entire response message if the original ICAL received partial data and an AIB return code of x’100’ and reason code of x’00C’. By taking advantage of the new subfunction and expanding the response area in the call, the IMS application is able to retrieve the copy of the message that is temporarily kept in the IMS control region’s private storage.

This enhancement relieves the burden of having to re-issue a full ICAL “SENDRECV” along with any associated network delays.



IMS 13

ICAL subfunction RECEIVE

- Format:
  - >>-ICAL--*aib*--*response area*--><
- AIB
  - *AIBSFUNC* value "RECEIVE"
  - *AIBOAUSE* is used as an input and output parameter based on AIBSFUNC
    - For the "RECEIVE" call
      - Contains the length of the response area
  - *AIBOALEN* = request area length
    - Used as an *output parameter for "RECEIVE"*
      - When complete response is returned in *response area*, this field is 0
      - If partial data is returned (AIB RC X'100', RS X'00C'), this field contains the actual length of the response message

O2_ IMS 13 TM: 42

The format of the ICAL with a subfunction of "RECEIVE includes specification of the ***aib*** and a ***response area*** that can contain the entire message. Not all the *aib* fields are used for this call. The following apply:

•AIBID = DFSAIB

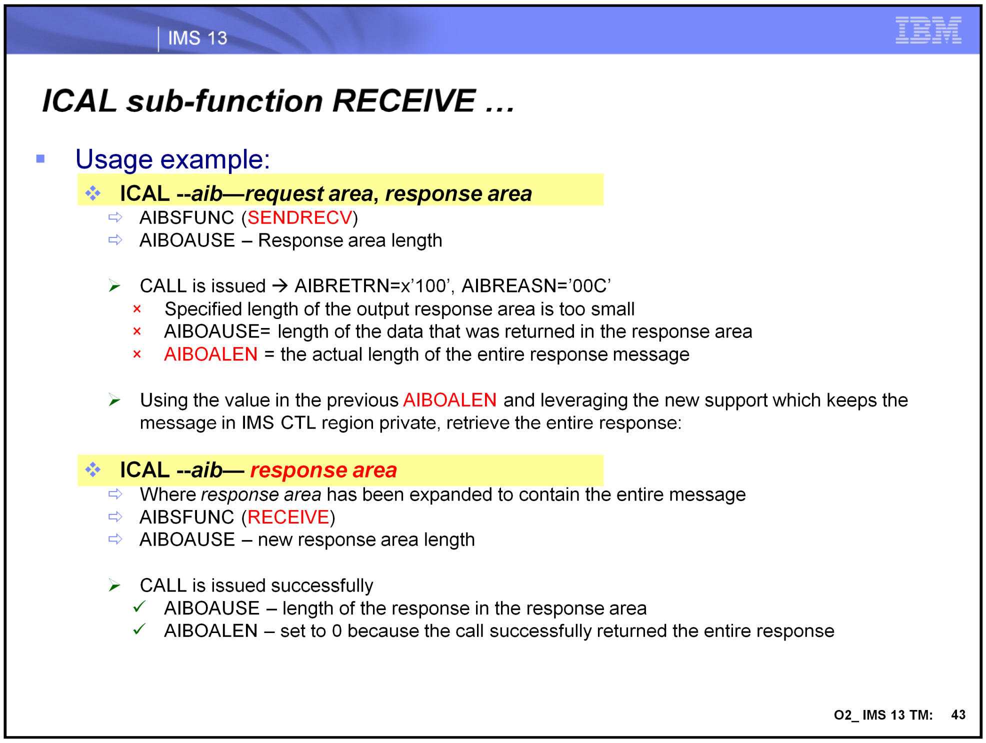
•AIBLEN = AIB length

•AIBOALEN - request area length. This parameter is an input and output parameter for the SENDRECV subfunction. When this parameter is used for the RECEIVE subfunction, it is used as an output parameter. When the complete response data is returned in the response data area, this output field will be set to 0. When partial data is returned (AIB return code X'100', AIB reason code X'00C'), this field contains the actual length of the response message.

•AIBOAUSE - Response area length. This parameter is an input and output parameter. As an input parameter, this 4-byte field contains the length of the output response area that is specified in the call list. As an output parameter, this field is updated by IMS with the length of the response message that is returned in the response area. When partial data is returned because the response area is not large enough, AIBOAUSE contains the length of the data that is returned in the response area, and AIBOALEN contains the actual length of the response message.

•AIBRETRN = AIB Return code

•AIBREASN = AIB Reason code.



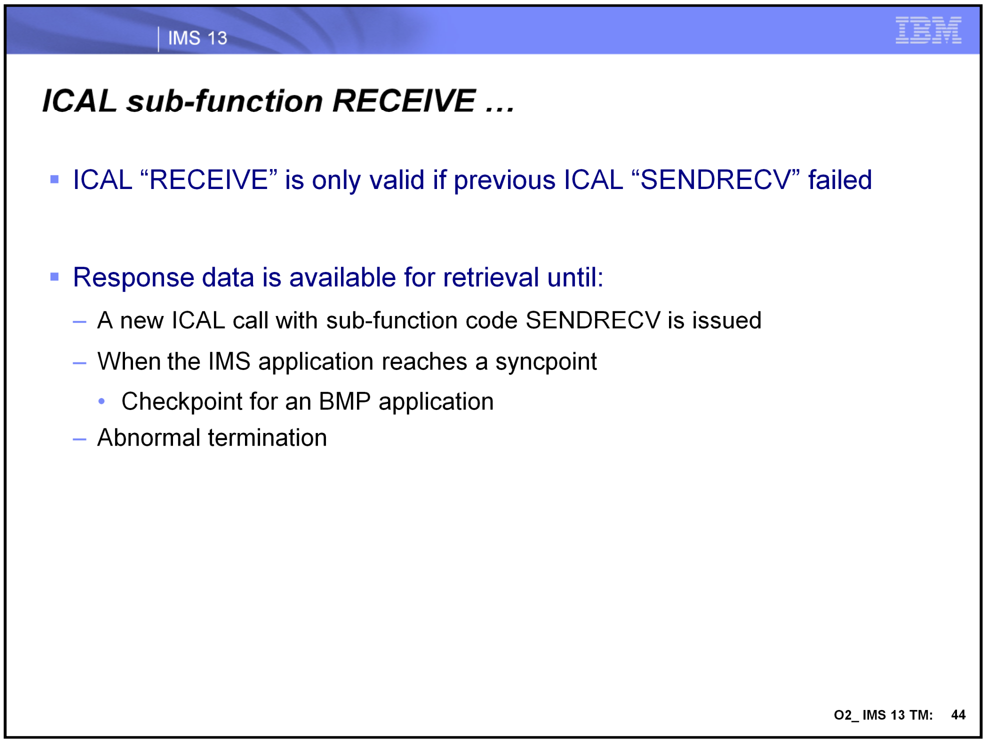
IMS 13

ICAL sub-function RECEIVE …

- Usage example:
  - ICAL --*aib—request area, response area*
    - AIBSFUNC (SENDRECV)
    - AIBOAUSE – Response area length
    - CALL is issued → AIBRETRN=x'100', AIBREASN='00C'
      - × Specified length of the output response area is too small
      - × AIBOAUSE= length of the data that was returned in the response area
      - × AIBOALEN = the actual length of the entire response message
    - Using the value in the previous AIBOALEN and leveraging the new support which keeps the message in IMS CTL region private, retrieve the entire response:
  - ICAL --*aib— response area*
    - Where *response area* has been expanded to contain the entire message
    - AIBSFUNC (RECEIVE)
    - AIBOAUSE – new response area length
    - CALL is issued successfully
      - ✓ AIBOAUSE – length of the response in the response area
      - ✓ AIBOALEN – set to 0 because the call successfully returned the entire response

O2_ IMS 13 TM: 43

If an ICAL call with sub-function code SENDRECV is issued with an insufficient response buffer defined, the ICAL will be returned with AIB return code X'100' and AIB reason code X'00C'. A subsequent ICAL call with sub-function code RECEIVE can still be issued to retrieve the entire response message as long as it is defined with the expanded response buffer.



ICAL sub-function RECEIVE …

- ICAL “RECEIVE” is only valid if previous ICAL “SENDRECV” failed

- Response data is available for retrieval until:
  - A new ICAL call with sub-function code SENDRECV is issued
  - When the IMS application reaches a syncpoint
    - Checkpoint for an BMP application
  - Abnormal termination

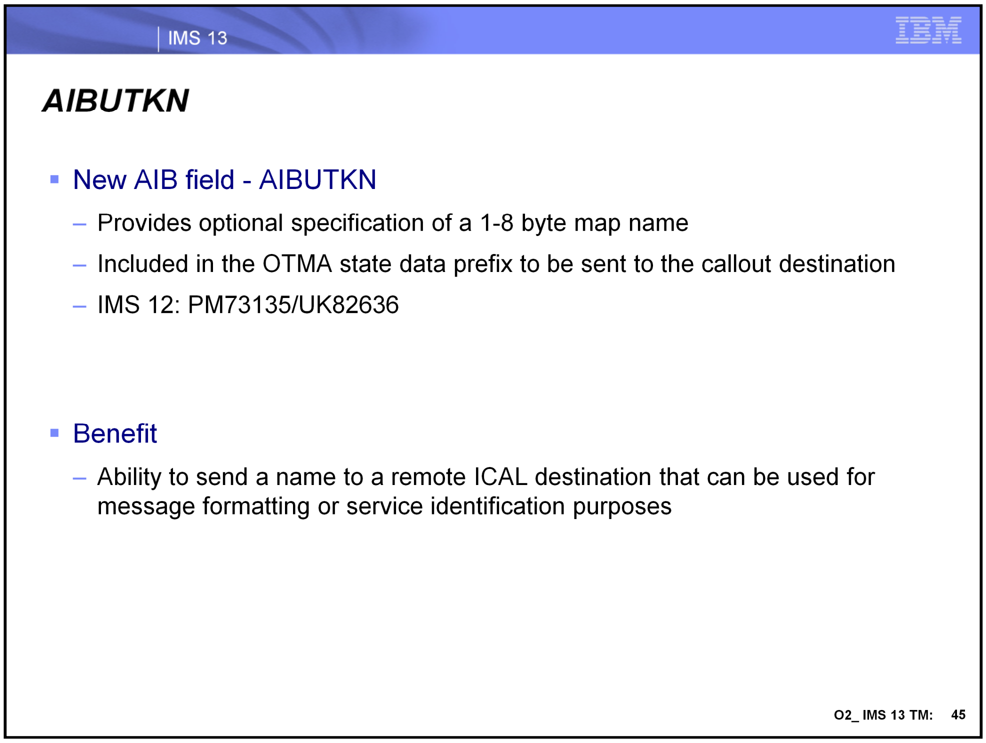
In order to issue an ICAL call with the sub-function code RECEIVE to retrieve the entire response message, the ICAL call with sub-function code SENDRECV must have previously been issued and returned with AIB return code X’100’ and AIB reason code X’00C’.

IMS keeps a copy of the entire response message in the control region private for any potential subsequent ICAL “RECEIVE” calls. Note that if this copy of the response message in IMS is not subsequently retrieved by the ICAL RECEIVE call, it will be discarded and the storage will be released. The response message is kept until:

•A new ICAL call with sub-function code SENDRECV is issued

•The IMS application issues ROLB call or CHECKPOINT

•The IMS application reaches syncpoint or terminates abnormally



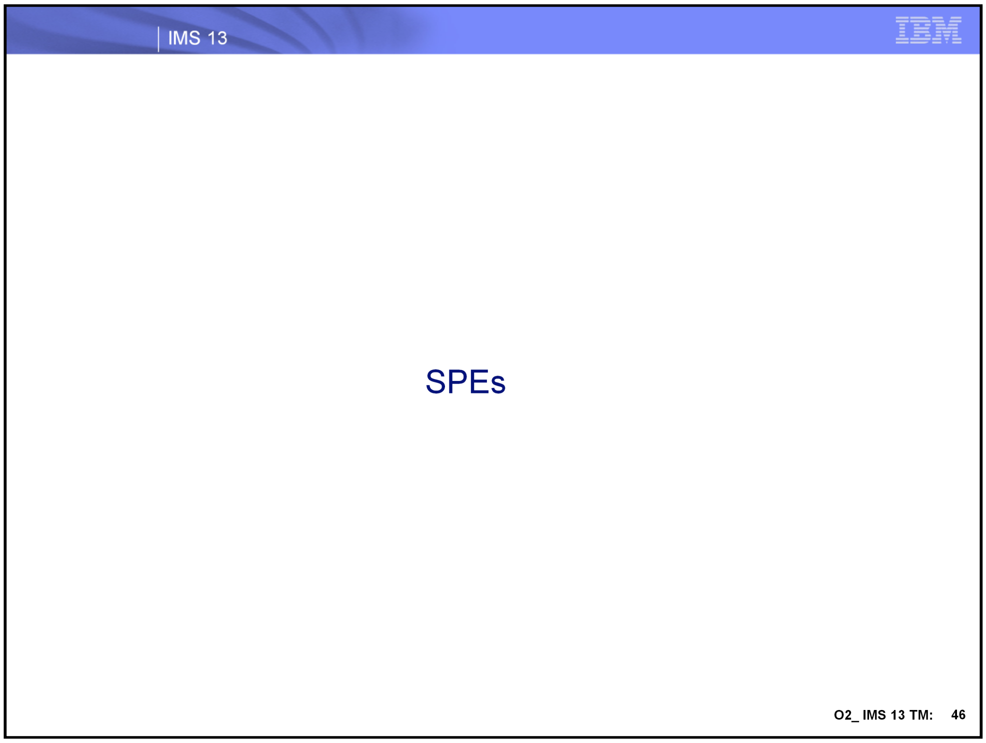
IMS 13

AIBUTKN

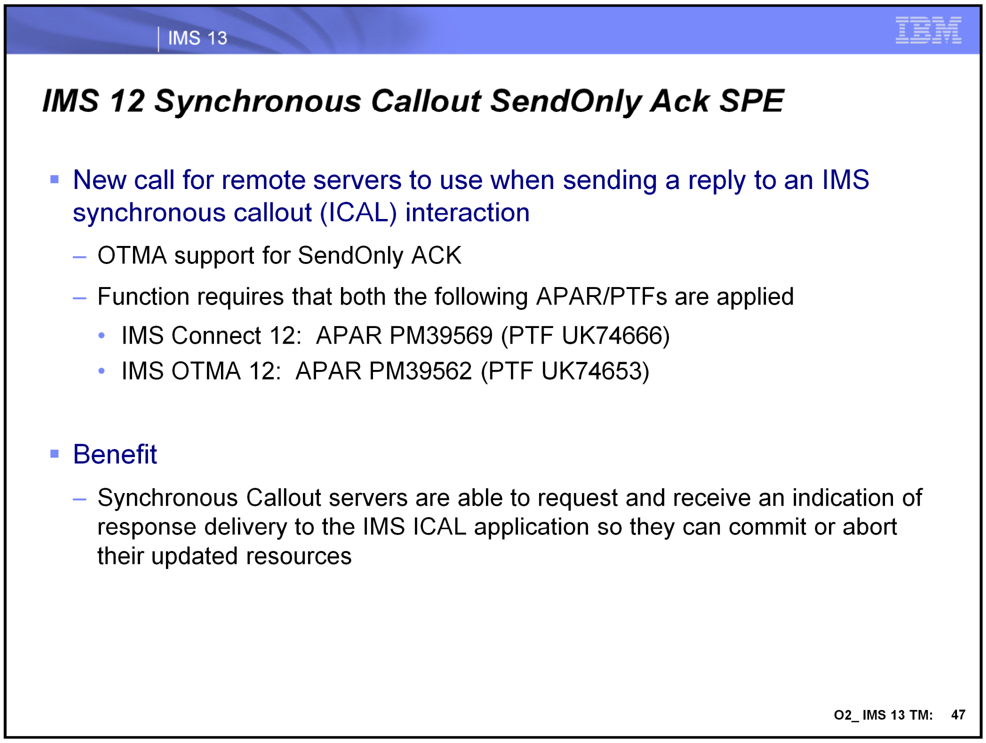
- New AIB field - AIBUTKN
  - Provides optional specification of a 1-8 byte map name
  - Included in the OTMA state data prefix to be sent to the callout destination
  - IMS 12: PM73135/UK82636

- Benefit
  - Ability to send a name to a remote ICAL destination that can be used for message formatting or service identification purposes

The DL/I ICAL call for synchronous callout can optionally use the first 8 bytes of a new AIBUTKN parameter to specify a map name in the application interface block (AIB). This value is passed in the OTMA state data prefix is can be uses by the remote destination of message formatting or service identification purposes.



SPEs



IMS 13

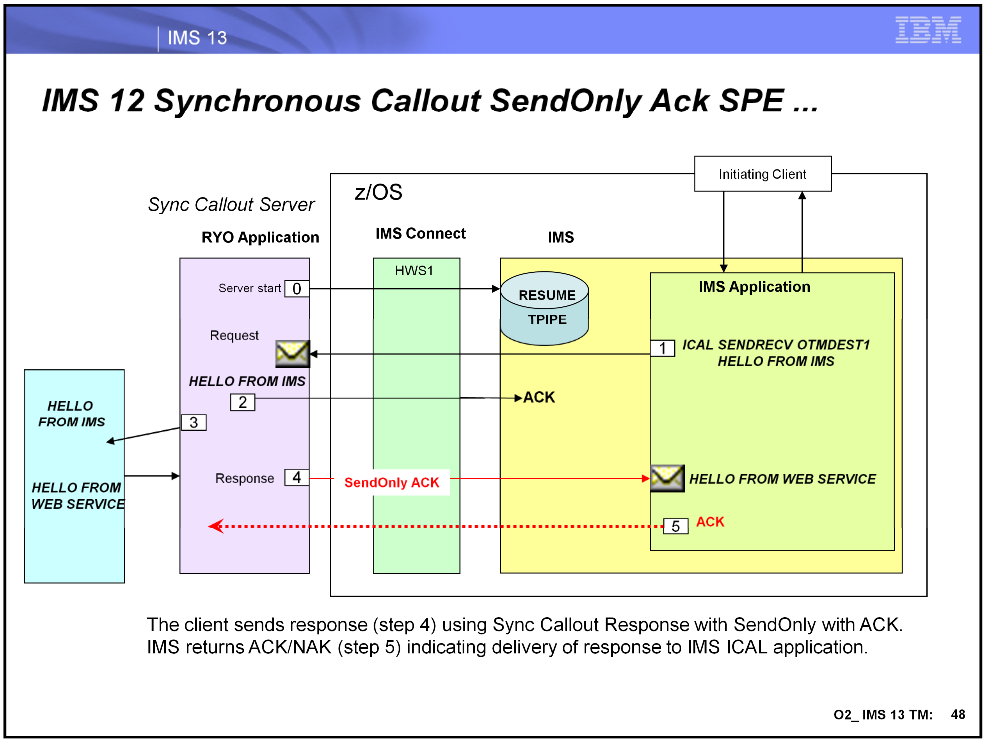
IMS 12 Synchronous Callout SendOnly Ack SPE

- New call for remote servers to use when sending a reply to an IMS synchronous callout (ICAL) interaction
  - OTMA support for SendOnly ACK
  - Function requires that both the following APAR/PTFs are applied
    - IMS Connect 12: APAR PM39569 (PTF UK74666)
    - IMS OTMA 12: APAR PM39562 (PTF UK74653)

- Benefit
  - Synchronous Callout servers are able to request and receive an indication of response delivery to the IMS ICAL application so they can commit or abort their updated resources

The Synchronous Callout SendOnly ACK SPE has been delivered to address IMS 12 environments and is part of base IMS 13. To enable the function both APARs must be applied – one to IMS OTMA and the other to IMS Connect.

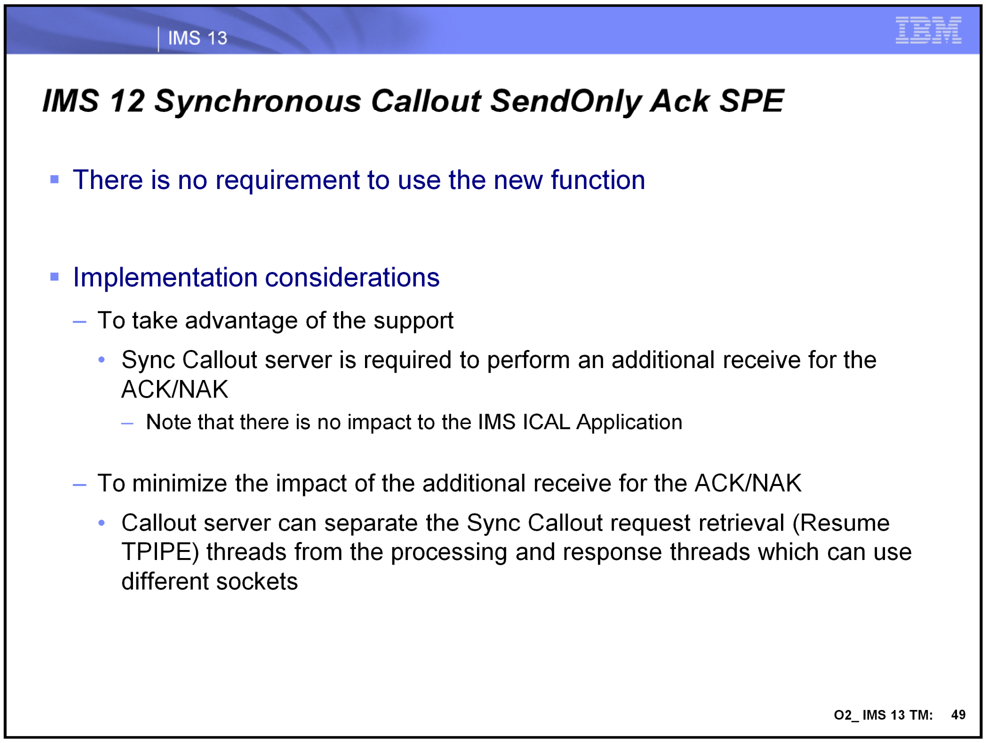
This support allows a synchronous callout server, i.e., the target of IMS's outbound ICAL, to process the request and when sending the response to request that IMS acknowledge receipt of the response.



IMS 12 Synchronous Callout SendOnly Ack SPE …

The client sends response (step 4) using Sync Callout Response with SendOnly with ACK. IMS returns ACK/NAK (step 5) indicating delivery of response to IMS ICAL application.

This visual gives an example of the callout process including a request for SendOnly ACK.



IMS 13

IMS 12 Synchronous Callout SendOnly Ack SPE

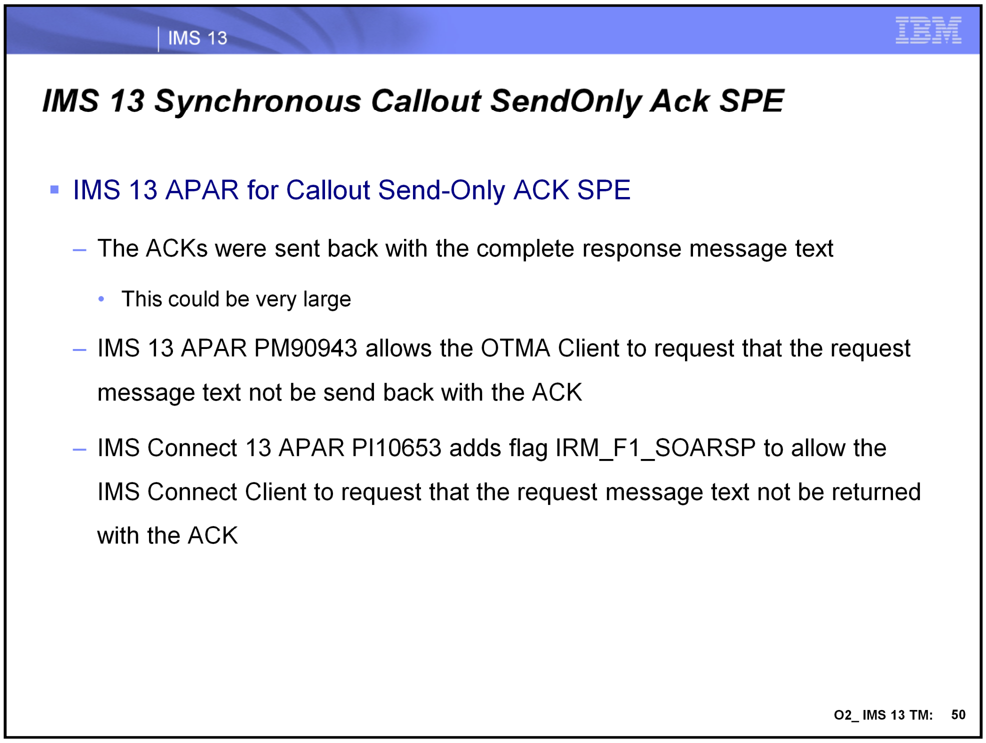
- There is no requirement to use the new function

- Implementation considerations
  - To take advantage of the support
    - Sync Callout server is required to perform an additional receive for the ACK/NAK
      - Note that there is no impact to the IMS ICAL Application
  - To minimize the impact of the additional receive for the ACK/NAK
    - Callout server can separate the Sync Callout request retrieval (Resume TPIPE) threads from the processing and response threads which can use different sockets

O2_ IMS 13 TM: 49

For a synchronous callout server application to take advantage of this new function, an additional receive for the ACK (acknowledgement) or NAK (negative acknowledgement or rejection) must be coded. This implementation does not impact or require any changes to the IMS application that issues the ICAL. The ACK/NAK is sent by OTMA.

To minimize the possible performance impact of coding an additional receive, the server has the option of designing the interaction to separate the request retrieval threads from the processing and response threads. Different threads can use different sockets.

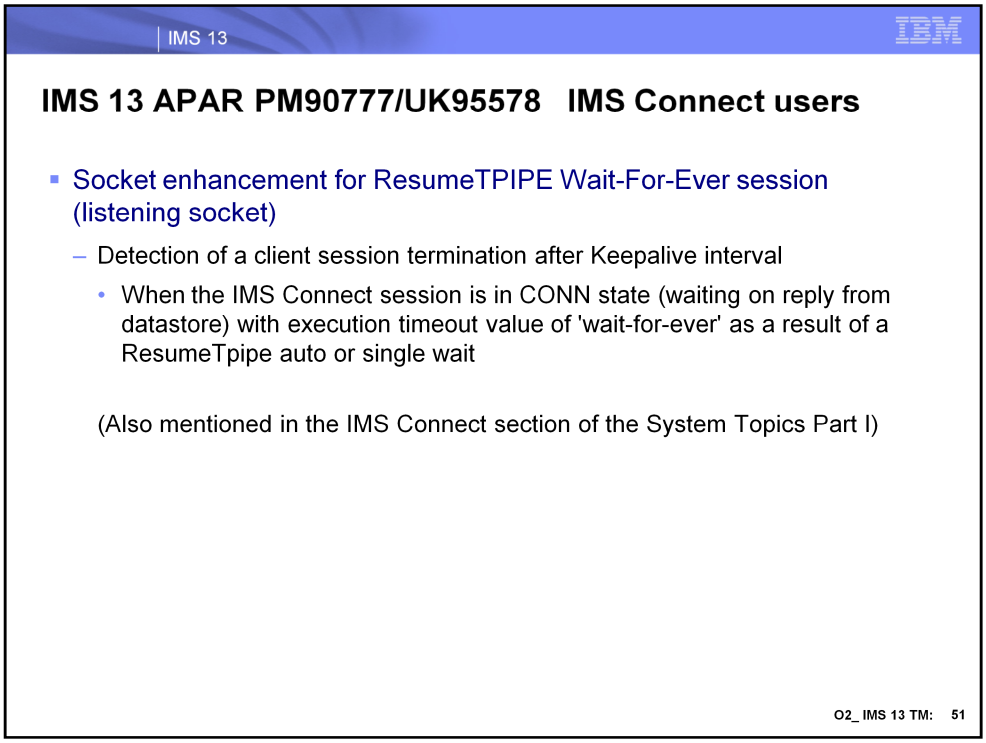


IMS 13 Synchronous Callout SendOnly Ack SPE

- IMS 13 APAR for Callout Send-Only ACK SPE
  - The ACKs were sent back with the complete response message text
    - This could be very large
  - IMS 13 APAR PM90943 allows the OTMA Client to request that the request message text not be send back with the ACK
  - IMS Connect 13 APAR PI10653 adds flag IRM_F1_SOARSP to allow the IMS Connect Client to request that the request message text not be returned with the ACK

For a synchronous callout server application to take advantage of this new function, an additional receive for the ACK (acknowledgement) or NAK (negative acknowledgement or rejection) must be coded. This implementation does not impact or require any changes to the IMS application that issues the ICAL. The ACK/NAK is sent by OTMA.

To minimize the possible performance impact of coding an additional receive, the server has the option of designing the interaction to separate the request retrieval threads from the processing and response threads. Different threads can use different sockets.

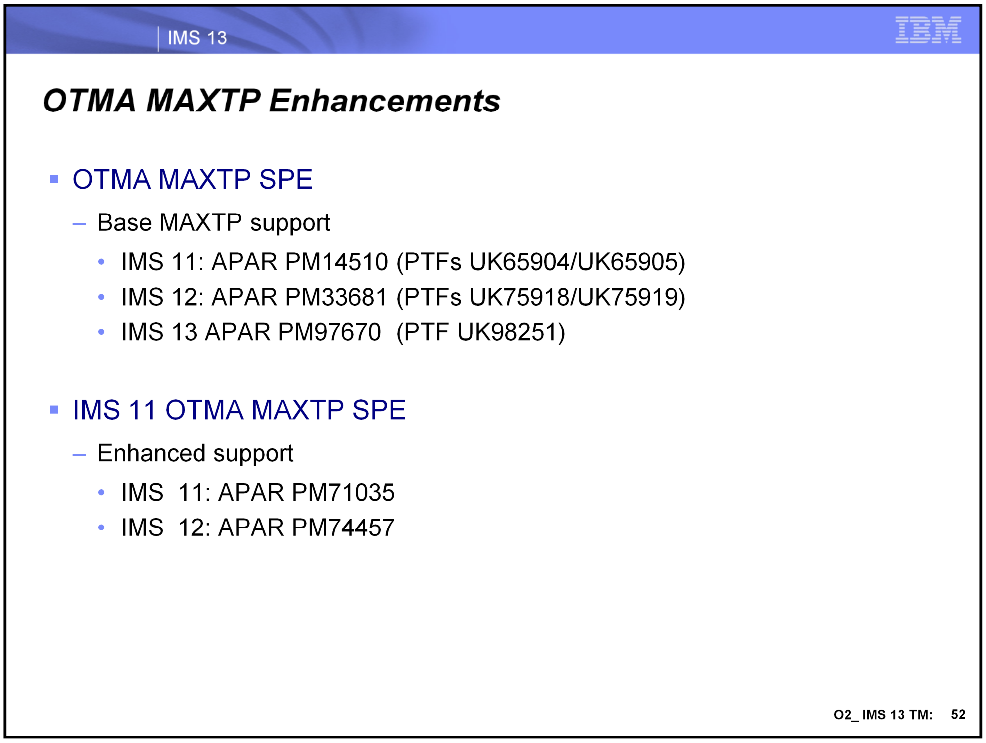


IMS 13 APAR PM90777/UK95578 IMS Connect users

- Socket enhancement for ResumeTPIPE Wait-For-Ever session (listening socket)
  - Detection of a client session termination after Keepalive interval
    - When the IMS Connect session is in CONN state (waiting on reply from datastore) with execution timeout value of 'wait-for-ever' as a result of a ResumeTpipe auto or single wait

(Also mentioned in the IMS Connect section of the System Topics Part I)

With this service the IMS Connect session that is in CONN state (waiting on reply from datastore) with execution timeout value of 'wait-for-ever' as a result of a ResumeTpipe ‘auto’ or ‘single wait’ will immediately detect a session termination of the client application.



IMS 13

OTMA MAXTP Enhancements

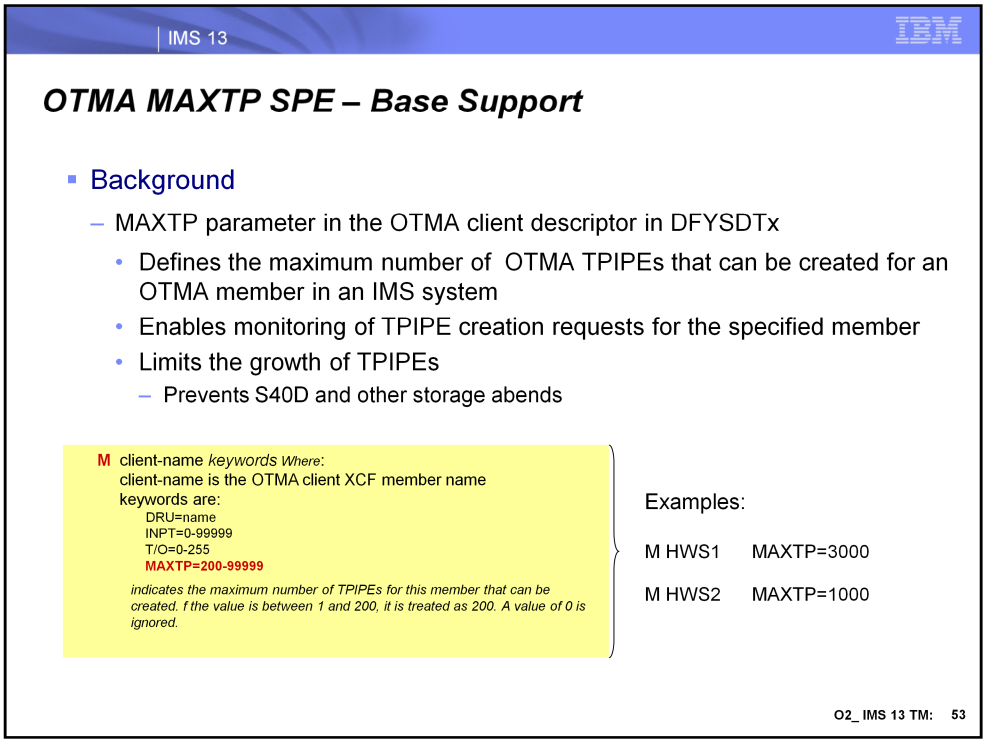
- OTMA MAXTP SPE
  - Base MAXTP support
    - IMS 11: APAR PM14510 (PTFs UK65904/UK65905)
    - IMS 12: APAR PM33681 (PTFs UK75918/UK75919)
    - IMS 13 APAR PM97670 (PTF UK98251)
- IMS 11 OTMA MAXTP SPE
  - Enhanced support
    - IMS 11: APAR PM71035
    - IMS 12: APAR PM74457

A TPIPE is a logical connection between IMS and the OTMA client, through which all input and output for a given client or client ID is routed and queued. OTMA creates TPIPEs dynamically at the request of the OTMA client. Each TPIPE has extended CSA storage and extended private storage associated with it. When there are too many TPIPEs in the IMS system, it can be a storage concern. In order to limit the growth of OTMA TPIPE creation in the system, the optional parameter, MAXTP, in the OTMA client descriptor was introduced to set a TPIPE limit for an OTMA member. Once this parameter is defined for an OTMA member, OTMA begins monitoring requests for TPIPE creation.

The base support for OTMA MAXTP was previously introduced with APARs to IMS 11, and IMS 12.

Enhancements to this base support are provided with the IMS 11 OTMA MAXTP SPE which includes IMS 11 and IMS 12 environments.

All of these capability are part of the IMS 13 base.



OTMA MAXTP SPE – Base Support

- Background
  - MAXTP parameter in the OTMA client descriptor in DFYSDTx
    - Defines the maximum number of OTMA TPIPEs that can be created for an OTMA member in an IMS system
    - Enables monitoring of TPIPE creation requests for the specified member
    - Limits the growth of TPIPEs
      - Prevents S40D and other storage abends

```
M  client-name keywords Where:
   client-name is the OTMA client XCF member name
   keywords are:
        DRU=name
        INPT=0-99999
        T/O=0-255
        MAXTP=200-99999
```

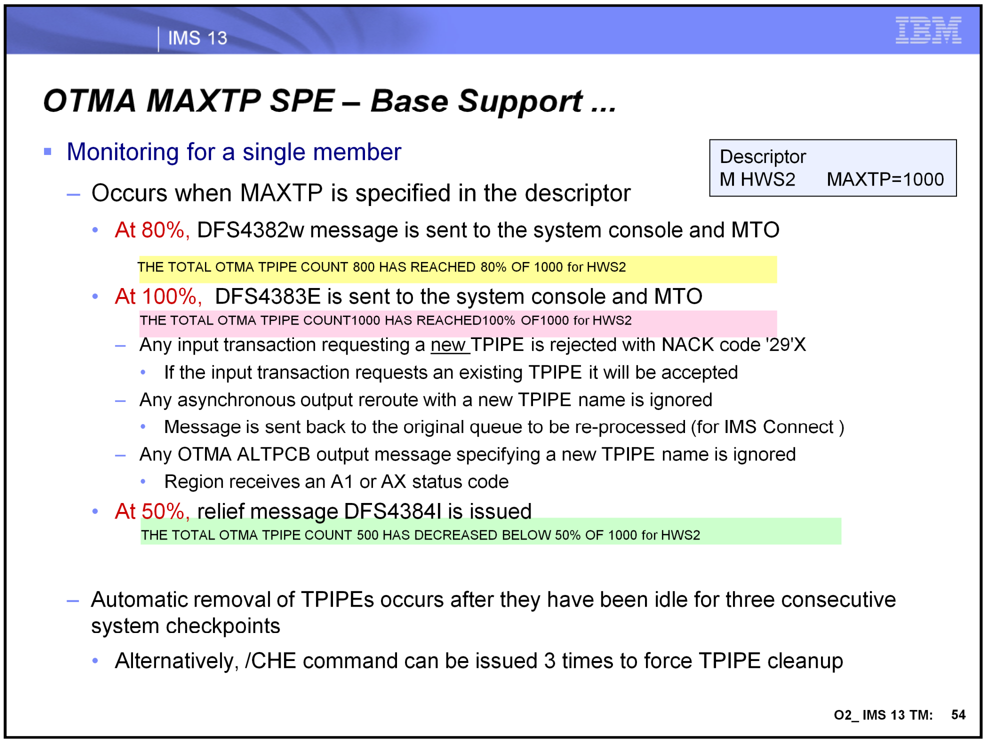
indicates the maximum number of TPIPEs for this member that can be created. f the value is between 1 and 200, it is treated as 200. A value of 0 is ignored.

Examples:

```
M HWS1    MAXTP=3000
M HWS2    MAXTP=1000
```

MAXTP is a parameter in the OTMA client descriptor that defines the maximum number of OTMA TPIPEs that can be created for that member. Once the parameter is defined, OTMA begins monitoring of the TPIPE creation requests. This capability is beneficial for limiting the unrestrained growth of TPIPES which could result in a S40D or other storage abend.

The example on this visual describes the parameters and shows some examples.



IMS 13

OTMA MAXTP SPE – Base Support ...

Descriptor
M HWS2 MAXTP=1000

- Monitoring for a single member
  - Occurs when MAXTP is specified in the descriptor
    - At 80%, DFS4382w message is sent to the system console and MTO

      THE TOTAL OTMA TPIPE COUNT 800 HAS REACHED 80% OF 1000 for HWS2
    - At 100%, DFS4383E is sent to the system console and MTO

      THE TOTAL OTMA TPIPE COUNT1000 HAS REACHED100% OF1000 for HWS2
      - Any input transaction requesting a new TPIPE is rejected with NACK code '29'X
        - If the input transaction requests an existing TPIPE it will be accepted
      - Any asynchronous output reroute with a new TPIPE name is ignored
        - Message is sent back to the original queue to be re-processed (for IMS Connect)
      - Any OTMA ALTPCB output message specifying a new TPIPE name is ignored
        - Region receives an A1 or AX status code
    - At 50%, relief message DFS4384I is issued

      THE TOTAL OTMA TPIPE COUNT 500 HAS DECREASED BELOW 50% OF 1000 for HWS2
  - Automatic removal of TPIPEs occurs after they have been idle for three consecutive system checkpoints
    - Alternatively, /CHE command can be issued 3 times to force TPIPE cleanup

O2_ IMS 13 TM: 54

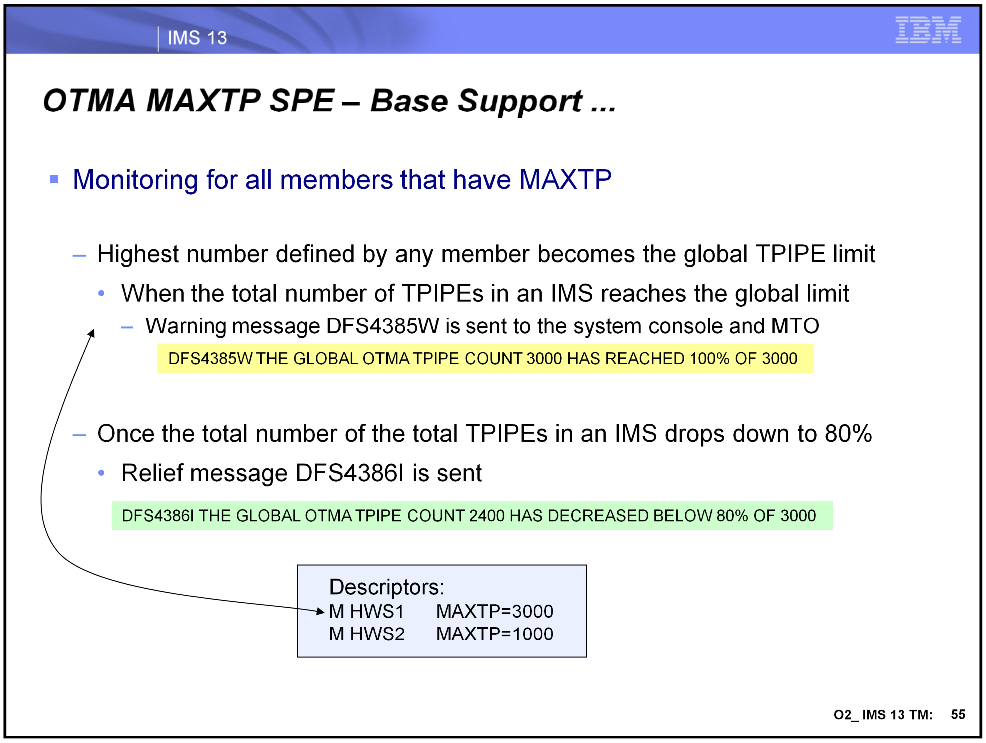
Once the MAXTP parameter is defined for an OTMA member, the request for the TPIPE creation for the OTMA member will be monitored.

When the total number of the TPIPEs reaches 80% of the maximum, IMS generates a warning message DFS4382W to system console and MTO. Note that the DFS4382W message is displayed only once (in the situation, for example, where the percentage bounces from 80% to 75% to 84%, etc.). Only after the number drops down to 50% are the internal flags reset. After this reset, if the TPIPE number once again gets to 80%, then the DFS4382W message will be issued.

When the total number of TPIPEs gets to 100%, an error message DFS4383E is sent to the system console and MTO. Any input transaction requesting a new TPIPE will be rejected with a new NACK code '29'X when the maximum TPIPE limit has been reached. On the other hand, if the input transaction requests an exiting TPIPE, it will be accepted. Any asynchronous output reroute with a new TPIPE name is ignored and the message will be washed back to the original queue to be re-processed. (This is mainly for IMS Connect customers). Additionally, OTMA ALTPCB output messages specifying a new TPIPE name will not be processed. The region will receive an A1 or AX status.

When the number of TPIPEs for the member drops back down to 50% of the value then relief message DFS4384I is issued.

IMS automatically attempts to remove transaction pipes after they have been idle for three consecutive system checkpoints. Or, you can issue /CHE commands 3 times to force the tpipe cleanup.



IMS 13

OTMA MAXTP SPE – Base Support ...

- Monitoring for all members that have MAXTP
  - Highest number defined by any member becomes the global TPIPE limit
    - When the total number of TPIPEs in an IMS reaches the global limit
      - Warning message DFS4385W is sent to the system console and MTO

```
DFS4385W THE GLOBAL OTMA TPIPE COUNT 3000 HAS REACHED 100% OF 3000
```

  - Once the total number of the total TPIPEs in an IMS drops down to 80%
    - Relief message DFS4386I is sent

```
DFS4386I THE GLOBAL OTMA TPIPE COUNT 2400 HAS DECREASED BELOW 80% OF 3000
```

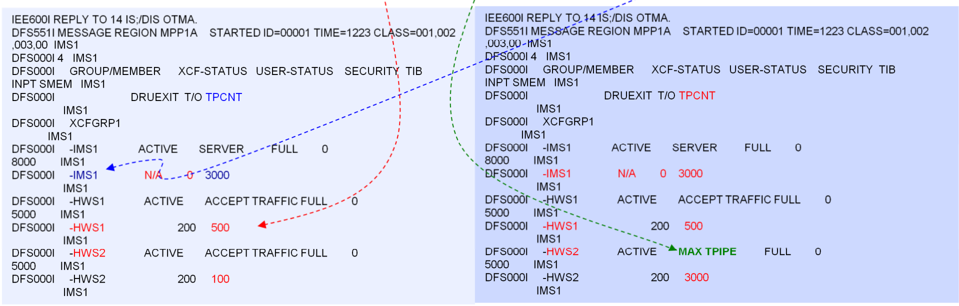
```
Descriptors:
M HWS1     MAXTP=3000
M HWS2     MAXTP=1000
```

O2_ IMS 13 TM: 55

When multiple OTMA member clients specify the MAXTP value in their client descriptors, the highest number defined among the members will be treated as the global TPIPE limit for the members. When the total number of TPIPEs in an IMS reaches the global limit, a warning message DFS4385W will be sent to the system console and MTO. Once the total number of the total TPIPEs in an IMS drops down to 80% of the global limit, a relief message DFS4386I will be sent out.

OTMA MAXTP SPE – Base Support ...

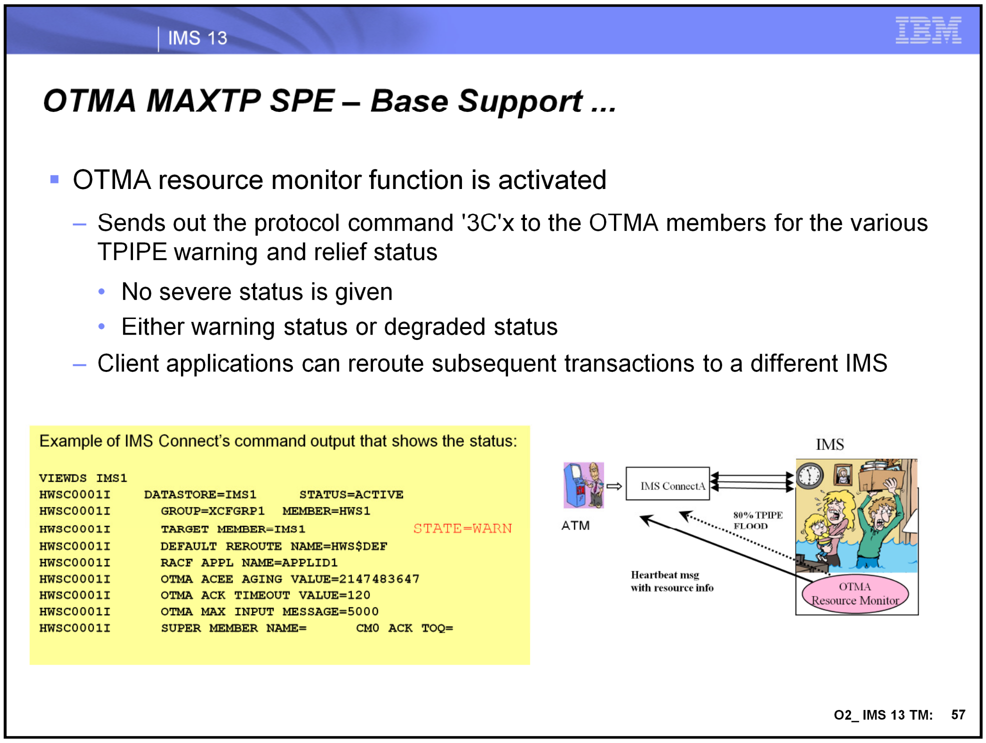
- /DISPLAY OTMA and /DISPLAY TMEMBER command output
  - Shows the current number of TPIPEs for members using the function
  - Once the limit has been reached, MAX TPIPE will be displayed as the USER_STATUS for the member
  - Global TPIPE limit is displayed under the new output field TPCNT for the IMS server



```
IEE600I REPLY TO 14 IS;/DIS OTMA.
DFS551I MESSAGE REGION MPP1A    STARTED ID=00001 TIME=1223 CLASS=001,002
,003,00  IMS1
DFS000I 4   IMS1
DFS000I    GROUP/MEMBER     XCF-STATUS   USER-STATUS    SECURITY  TIB
INPT SMEM   IMS1
DFS000I                    DRUEXIT  T/O TPCNT
            IMS1
DFS000I    XCFGRP1
        IMS1
DFS000I    -IMS1          ACTIVE      SERVER       FULL       0
8000       IMS1
DFS000I    -IMS1           N/A        0    3000
            IMS1
DFS000I    -HWS1           ACTIVE      ACCEPT TRAFFIC FULL       0
5000       IMS1
DFS000I    -HWS1                  200    500
            IMS1
DFS000I    -HWS2           ACTIVE      ACCEPT TRAFFIC FULL       0
5000       IMS1
DFS000I    -HWS2                  200    100
            IMS1
```

```
IEE600I REPLY TO 14 IS;/DIS OTMA.
DFS551I MESSAGE REGION MPP1A    STARTED ID=00001 TIME=1223 CLASS=001,002
,003,00  IMS1
DFS000I 4   IMS1
DFS000I    GROUP/MEMBER     XCF-STATUS   USER-STATUS    SECURITY  TIB
INPT SMEM   IMS1
DFS000I                    DRUEXIT  T/O TPCNT
            IMS1
DFS000I    XCFGRP1
        IMS1
DFS000I    -IMS1          ACTIVE      SERVER       FULL       0
8000       IMS1
DFS000I    -IMS1           N/A        0    3000
            IMS1
DFS000I    -HWS1           ACTIVE      ACCEPT TRAFFIC FULL       0
5000       IMS1
DFS000I    -HWS1                  200    500
            IMS1
DFS000I    -HWS2           ACTIVE      MAX TPIPE       FULL       0
5000       IMS1
DFS000I    -HWS2                  200    3000
            IMS1
```

The /DISPLAY OTMA and /DISPLAY TMEMBER commands have been enhanced to allow the display output to show the current number of TPIPES for those OTMA members that have enabled the MAXTP support. Additionally, the ‘MAX TPIPE’ status is displayed when the limit has been reached.



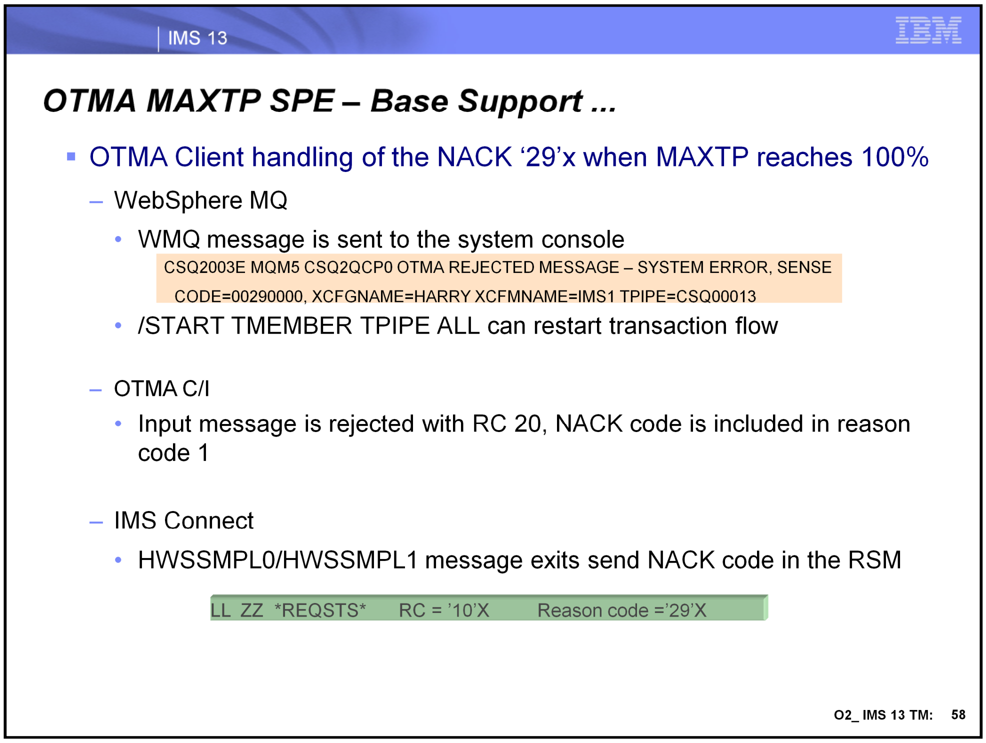
OTMA MAXTP SPE – Base Support ...

- OTMA resource monitor function is activated
  - Sends out the protocol command '3C'x to the OTMA members for the various TPIPE warning and relief status
    - No severe status is given
    - Either warning status or degraded status
  - Client applications can reroute subsequent transactions to a different IMS

Example of IMS Connect's command output that shows the status:

```
VIEWDS IMS1
HWSC0001I    DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I      TARGET MEMBER=IMS1                 STATE=WARN
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
```

The OTMA resource monitor function is also activated with this support and sends out the protocol command '3C'x to the OTMA members for the various TPIPE warning and relief status (no severe status will be given, the most is the warning status or degraded status.) Client applications can then reroute the subsequent transactions to a different IMS if needed.



IMS 13

OTMA MAXTP SPE – Base Support ...

- OTMA Client handling of the NACK ‘29’x when MAXTP reaches 100%
  - WebSphere MQ
    - WMQ message is sent to the system console

```
CSQ2003E MQM5 CSQ2QCP0 OTMA REJECTED MESSAGE – SYSTEM ERROR, SENSE
  CODE=00290000, XCFGNAME=HARRY XCFMNAME=IMS1 TPIPE=CSQ00013
```

-
  -
    - /START TMEMBER TPIPE ALL can restart transaction flow
  - OTMA C/I
    - Input message is rejected with RC 20, NACK code is included in reason code 1
  - IMS Connect
    - HWSSMPL0/HWSSMPL1 message exits send NACK code in the RSM

```
LL  ZZ  *REQSTS*     RC = ’10’X        Reason code =’29’X
```

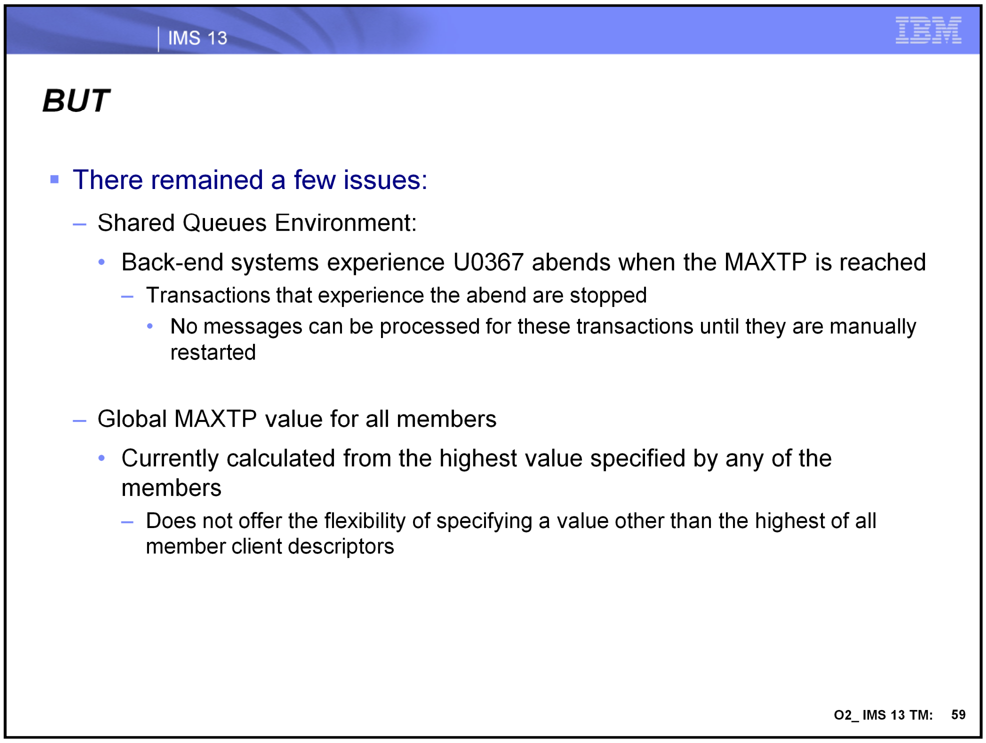
O2_ IMS 13 TM: 58

This visual describes OTMA client handling of the NACK ’29’x when MAXTP reaches 100%.

When IMS rejects a transaction request with ’29’x, WebSphere MQ sends CSQ2003E error message to the system console. The /START TMEMBER TPIPE ALL can be used to allow WMQ to start submitting transactions to IMS

OTMA C/I clients get a rejection of the input message with a RC 20 along with the NACK code in Reason Code 1.

IMS Connect users that take advantage of the HWSSMPL0 or HWSSMPL1 exit routine receive the ’29’x NACK code in the RSM.g

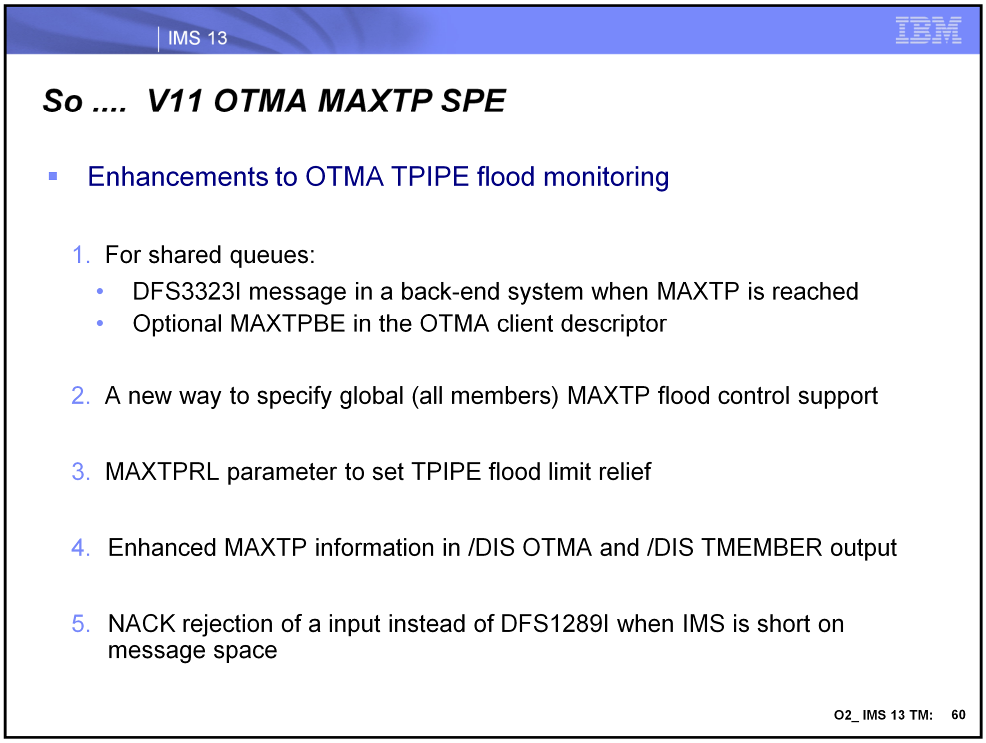


BUT

- There remained a few issues:
  - Shared Queues Environment:
    - Back-end systems experience U0367 abends when the MAXTP is reached
      - Transactions that experience the abend are stopped
        - No messages can be processed for these transactions until they are manually restarted
  - Global MAXTP value for all members
    - Currently calculated from the highest value specified by any of the members
      - Does not offer the flexibility of specifying a value other than the highest of all member client descriptors

After application of the enhancements previously discussed there still remained a few issues. In a Shared Queues environment, U0367 abends occurred in back-end systems when the MAXTP condition was reached. Transactions that receive the abend are stopped and must be manually restarted.

Additionally, calculation for a global MAXTP value is simply based on picking the highest MAXTP value from the OTMA TMEMBERs that specify the value. The method precludes setting a value that accommodates even a partial accumulation of the individual specifications.

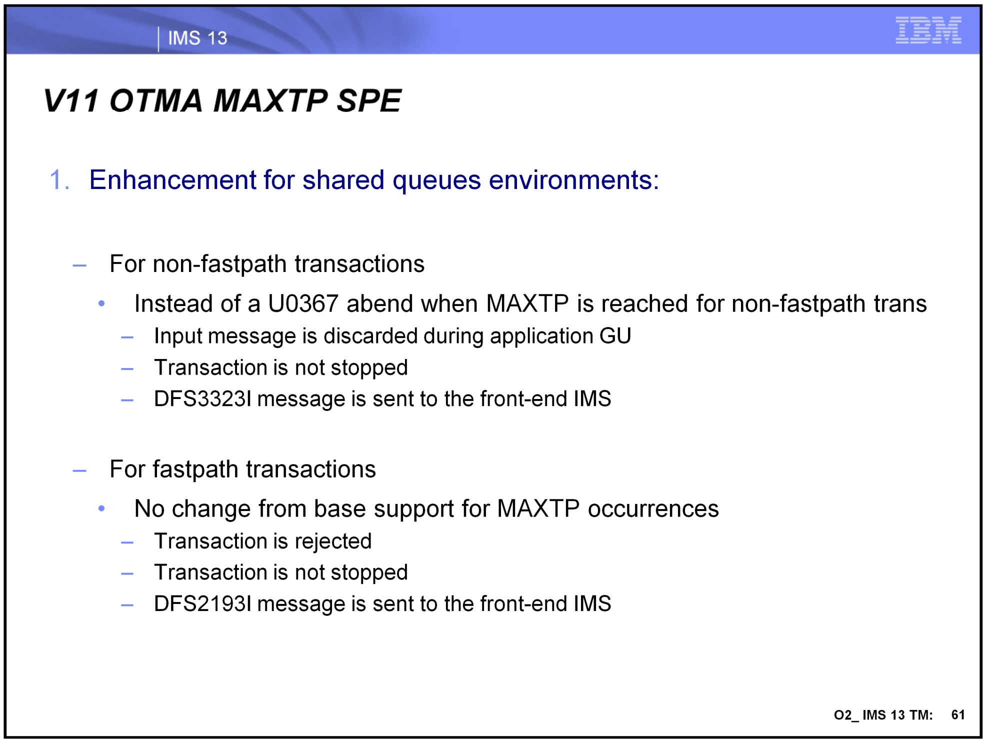


So .... V11 OTMA MAXTP SPE

- Enhancements to OTMA TPIPE flood monitoring

1. For shared queues:
   - DFS3323I message in a back-end system when MAXTP is reached
   - Optional MAXTPBE in the OTMA client descriptor
2. A new way to specify global (all members) MAXTP flood control support
3. MAXTPRL parameter to set TPIPE flood limit relief
4. Enhanced MAXTP information in /DIS OTMA and /DIS TMEMBER output
5. NACK rejection of a input instead of DFS1289I when IMS is short on message space

The V11 OTMA MAXTP SPE answers the issues discussed in the previous visual with 4 enhancements.



IMS 13

V11 OTMA MAXTP SPE

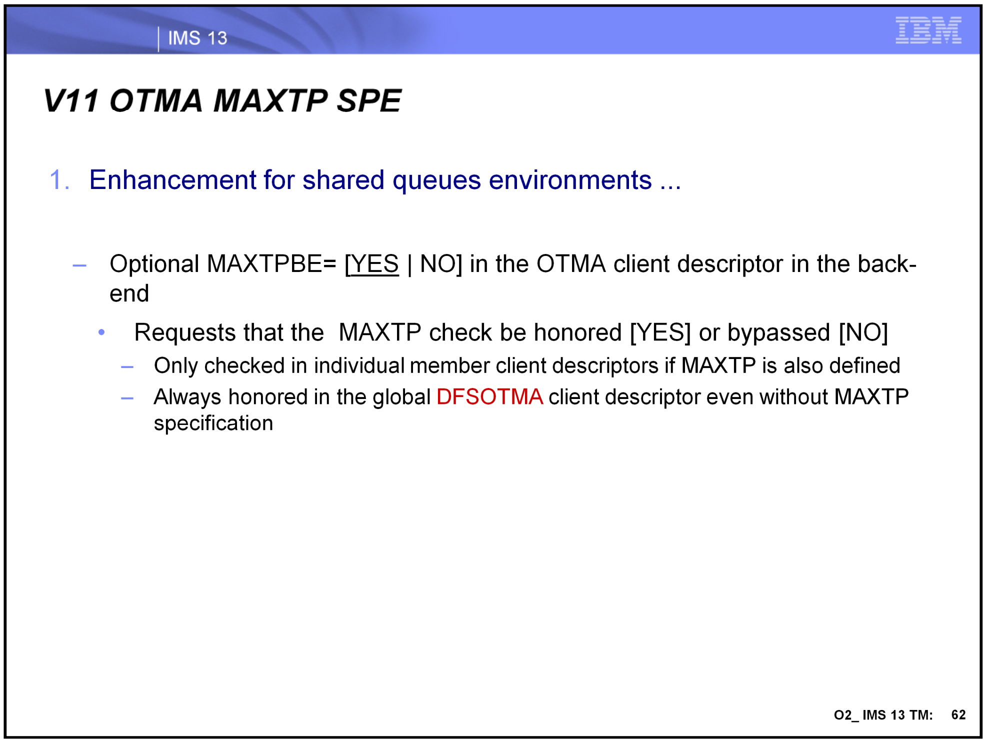
1. Enhancement for shared queues environments:

   - For non-fastpath transactions
     - Instead of a U0367 abend when MAXTP is reached for non-fastpath trans
       - Input message is discarded during application GU
       - Transaction is not stopped
       - DFS3323I message is sent to the front-end IMS
   - For fastpath transactions
     - No change from base support for MAXTP occurrences
       - Transaction is rejected
       - Transaction is not stopped
       - DFS2193I message is sent to the front-end IMS

O2_ IMS 13 TM: 61

The first enhancement address the shared queues back-end system abends. After application of the SPE, when an IMS shared queues back-end system attempts to process a front-end initiated OTMA non-fastpath input transaction at application GU time and MAXTP is reached, abend U0367 does no longer occurs. Rather, the input message is discarded, the transaction is not stopped, and a DFS3323I message is sent to the front-end IMS to inform the OTMA client.

Not there has been to change to fastpath transaction processing in this SPE. When a fathpath transaction is processed at the shared queues back-end IMS and experiences a MAXTP condition, the transaction will be rejected and a DFS2193I message is sent back to the front-end IMS. The transaction in the shared queues back-end is NOT stopped.



IMS 13

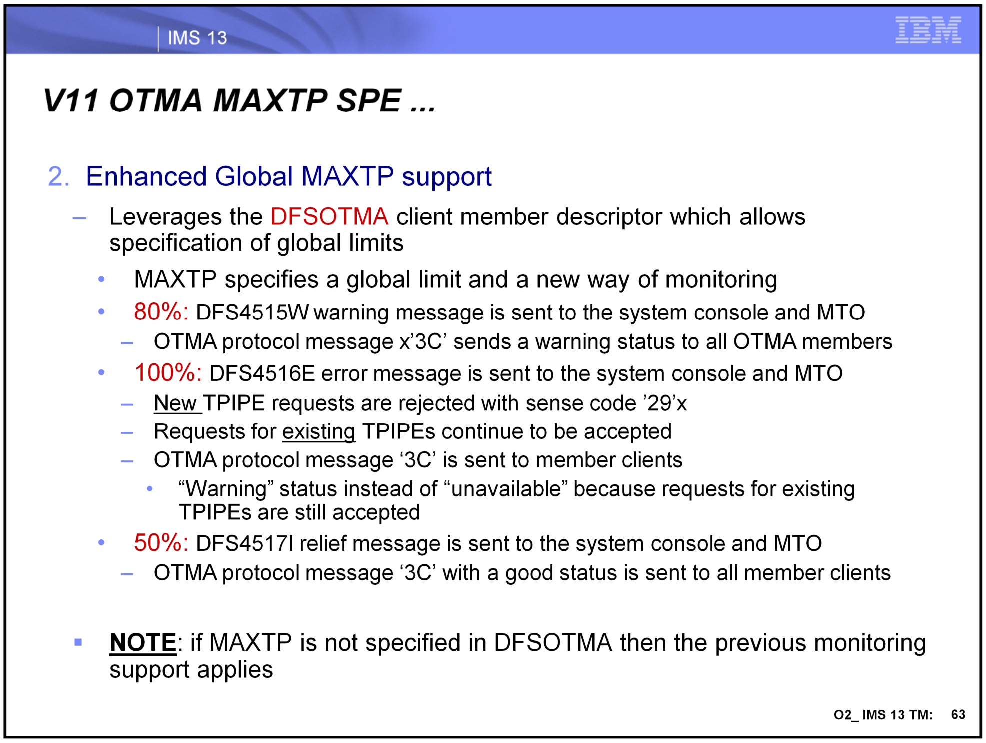
V11 OTMA MAXTP SPE

1. Enhancement for shared queues environments …

   - Optional MAXTPBE= [YES | NO] in the OTMA client descriptor in the back-end
     - Requests that the MAXTP check be honored [YES] or bypassed [NO]
       - Only checked in individual member client descriptors if MAXTP is also defined
       - Always honored in the global DFSOTMA client descriptor even without MAXTP specification

O2_ IMS 13 TM: 62

Optionally, a new parameter MAXTPBE in the OTMA client descriptor of the shared queues back-end can be specified to tell IMS to honor [YES] or ignore [NO] the MAXTP checking at application GU time. Setting this parameter in the back-end IMS allows OTMA front-end initiated transactions to be scheduled without worrying about the number of TPIPE limit defined. This parameter can be specified in the OTMA client descriptor for a member when MAXTP is specified. It will be ignored if the member does not have MAXTP specified. This parameter can also be specified in the special DFSOTMA client member descriptor (discussed earlier in the Global Flood Control Enhancement topic) with or without MAXTP for all the members. It is an optional parameter and defaults to YES.



IMS 13

V11 OTMA MAXTP SPE ...

2. Enhanced Global MAXTP support
   - Leverages the DFSOTMA client member descriptor which allows specification of global limits
     - MAXTP specifies a global limit and a new way of monitoring
     - 80%: DFS4515W warning message is sent to the system console and MTO
       - OTMA protocol message x'3C' sends a warning status to all OTMA members
     - 100%: DFS4516E error message is sent to the system console and MTO
       - New TPIPE requests are rejected with sense code '29'x
       - Requests for existing TPIPEs continue to be accepted
       - OTMA protocol message '3C' is sent to member clients
         - "Warning" status instead of "unavailable" because requests for existing TPIPEs are still accepted
     - 50%: DFS4517I relief message is sent to the system console and MTO
       - OTMA protocol message '3C' with a good status is sent to all member clients

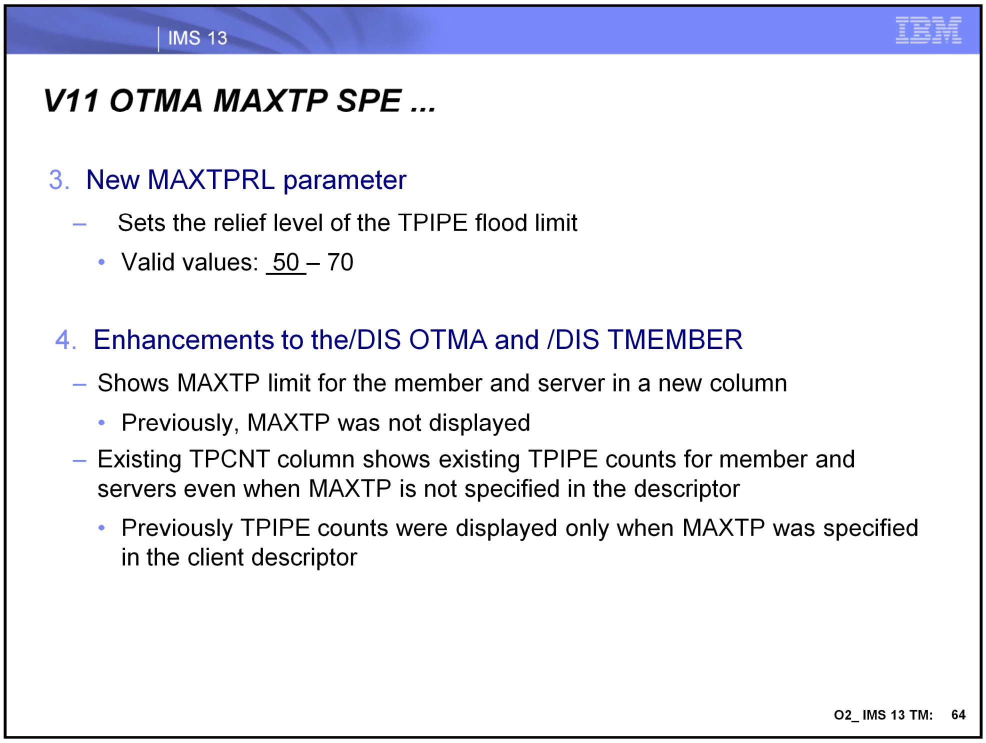
- **NOTE**: if MAXTP is not specified in DFSOTMA then the previous monitoring support applies

O2_ IMS 13 TM: 63

The enhancement for global MAXTP support leverages the DFSOTMA client descriptor and provides a MAXTP global limit for all the members. Once this is specified, IMS no longer uses the highest limit defined by any of the members. The MAXTP limit in DFSOTMA activates a new way of monitoring the TPIPE for all the members:

- When 80% of this global limit is reached, a DFS4515W message is sent to the system console and MTO, and an OTMA protocol message reflecting a warning status is sent to all the OTMA members.
- At 100%, when the global flood limit is reached, OTMA rejects all **new** TPIPE creation requests for all the OTMA members in the IMS. For an OTMA input transaction requesting a new TPIPE, a NAK rejection with OTMA sense code '29'x is sent. Additionally, a DFS4516E error message will be sent to the system console and MTO along with the OTMA protocol message reflecting a **warning** status to all the OTMA members. The reason why the protocol message shows the "warning" status instead of the "unavailable" status is that IMS can still accept transaction requests using existing TPIPEs in the system. A /DISPLAY OTMA command of this degraded system will also show "MAX TPIPE" in the user status of the OTMA server member.
- When the number of TPIPEs for all the members is reduced to 50% of the user specified global limit, a relief message DFS4517I is sent to IMS system console and MTO along. Additionally, an OTMA protocol message reflecting a good status is sent to all the OTMA members.

Note that if MAXTP in the special DFSOTMA client member descriptor is not used, IMS continues to use the highest limit defined among the members for the global MAXTP limit. As defined earlier, new TPIPE creation requests can still be accepted even when the global limit is reached. No rejections occur.



IMS 13

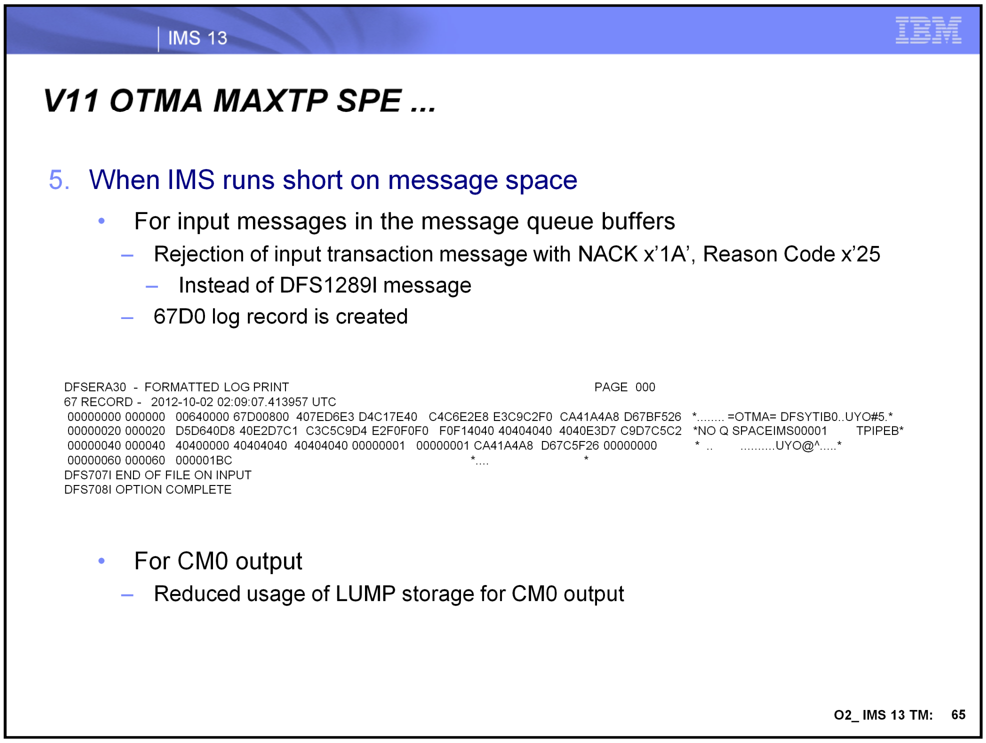
V11 OTMA MAXTP SPE ...

3. New MAXTPRL parameter
   - Sets the relief level of the TPIPE flood limit
     - Valid values: 50 – 70

4. Enhancements to the/DIS OTMA and /DIS TMEMBER
   - Shows MAXTP limit for the member and server in a new column
     - Previously, MAXTP was not displayed
   - Existing TPCNT column shows existing TPIPE counts for member and servers even when MAXTP is not specified in the descriptor
     - Previously TPIPE counts were displayed only when MAXTP was specified in the client descriptor

The new MAXTPRL parameter for the member descriptor sets the relief level of the TPIPE flood limit. The value can be specified between 50 and 70. If not specified, the default level will be 50, which means 50% of the flood limit.

The /DISPLAY OTMA and DISPLAY TMEMBER commands now show the MAXTP limits for the member and server in a new column. Previously, MAXTP for the member was not displayed. Additionally, the TPCNT column has been enhanced to show existing TPIPE counts for members and server, even when MAXTP parameter is not specified in the OTMA client descriptor. (Previously TPCNT displayed TPIPE counts only when MAXTP was specified in the OTMA client descriptor.)



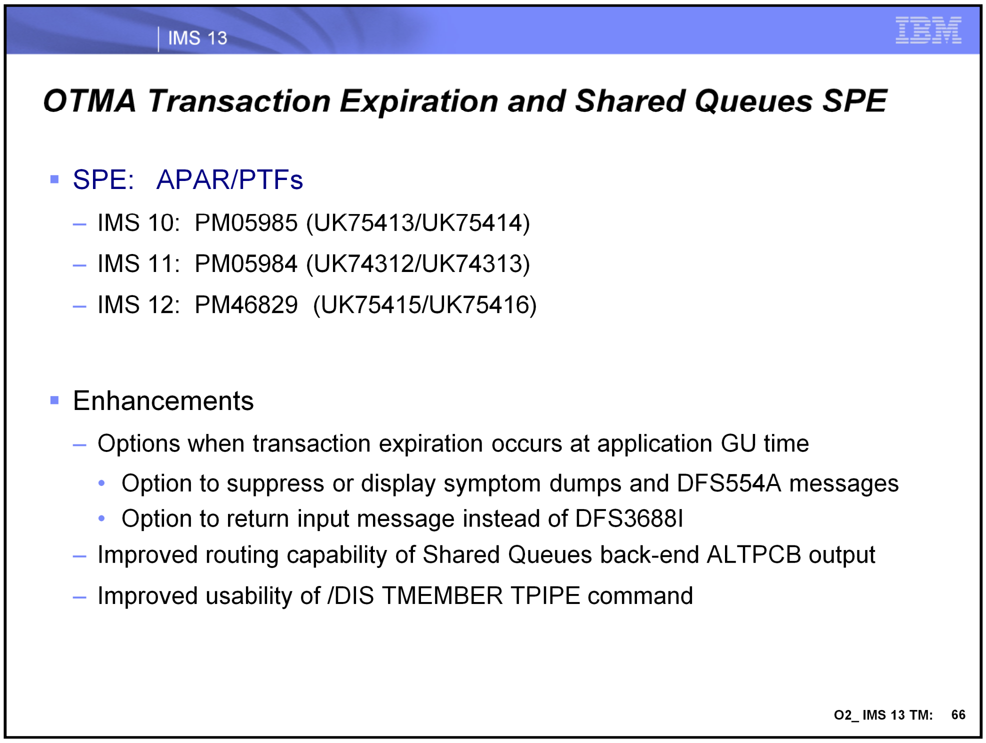
V11 OTMA MAXTP SPE ...

5. When IMS runs short on message space
   - For input messages in the message queue buffers
     - Rejection of input transaction message with NACK x’1A’, Reason Code x’25
       - Instead of DFS1289I message
     - 67D0 log record is created

```
DFSERA30  -  FORMATTED LOG PRINT                                                   PAGE  000
67 RECORD -   2012-10-02 02:09:07.413957 UTC
 00000000 000000   00640000 67D00800  407ED6E3 D4C17E40   C4C6E2E8 E3C9C2F0  CA41A4A8 D67BF526   *........ =OTMA= DFSYTIB0..UYO#5.*
 00000020 000020   D5D640D8 40E2D7C1  C3C5C9D4 E2F0F0F0   F0F14040 40404040  4040E3D7 C9D7C5C2   *NO Q SPACEIMS00001       TPIPEB*
 00000040 000040   40400000 40404040  40404040 00000001   00000001 CA41A4A8  D67C5F26 00000000    *  ..        ..........UYO@^.....*
 00000060 000060   000001BC                                            *....                  *
DFS707I END OF FILE ON INPUT
DFS708I OPTION COMPLETE
```

   - For CM0 output
     - Reduced usage of LUMP storage for CM0 output

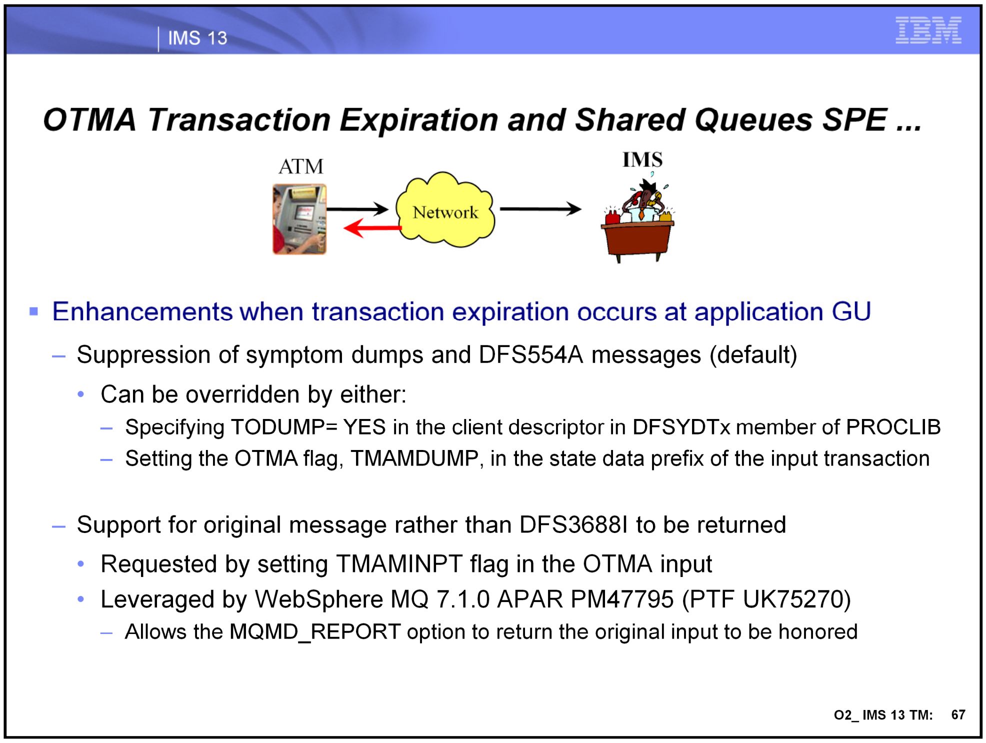
When IMS runs short on message queue space, input transactions are rejected with a NACK x’1A’ reason x’25’ instead of issuing a DFS1289I message. Additionally, a 67D0 log record will be produced.



OTMA Transaction Expiration and Shared Queues SPE

- SPE: APAR/PTFs
  - IMS 10: PM05985 (UK75413/UK75414)
  - IMS 11: PM05984 (UK74312/UK74313)
  - IMS 12: PM46829 (UK75415/UK75416)

- Enhancements
  - Options when transaction expiration occurs at application GU time
    - Option to suppress or display symptom dumps and DFS554A messages
    - Option to return input message instead of DFS3688I
  - Improved routing capability of Shared Queues back-end ALTPCB output
  - Improved usability of /DIS TMEMBER TPIPE command


ATM
Network
IMS

IMS 13

OTMA Transaction Expiration and Shared Queues SPE ...

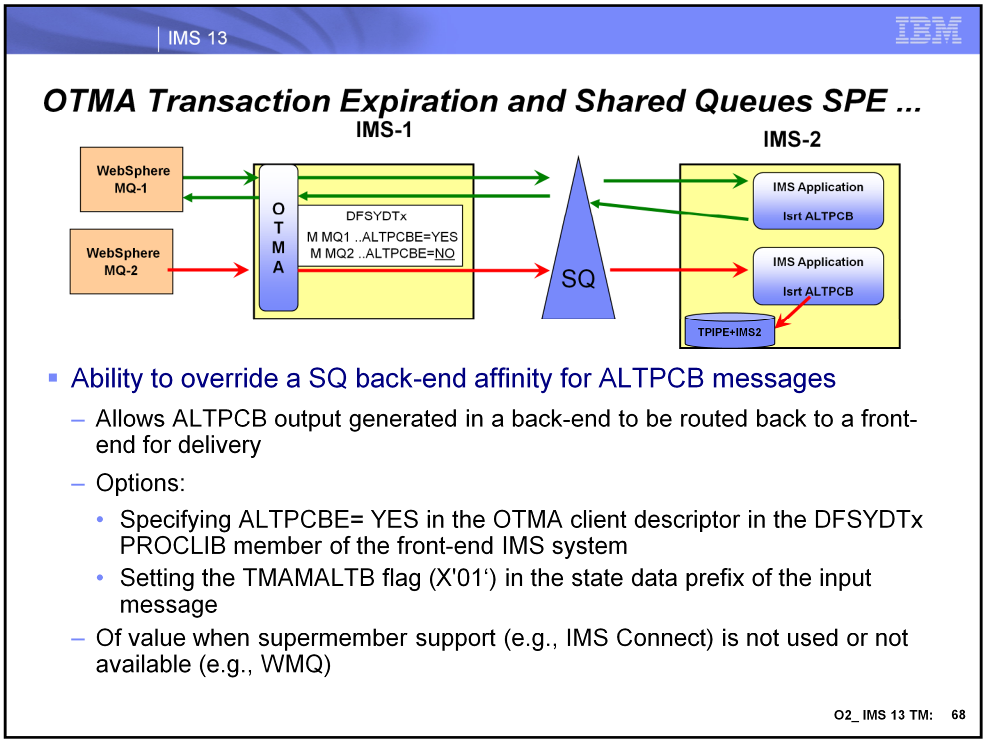
- Enhancements when transaction expiration occurs at application GU
  - Suppression of symptom dumps and DFS554A messages (default)
    - Can be overridden by either:
      - Specifying TODUMP= YES in the client descriptor in DFSYDTx member of PROCLIB
      - Setting the OTMA flag, TMAMDUMP, in the state data prefix of the input transaction
  - Support for original message rather than DFS3688I to be returned
    - Requested by setting TMAMINPT flag in the OTMA input
    - Leveraged by WebSphere MQ 7.1.0 APAR PM47795 (PTF UK75270)
      - Allows the MQMD_REPORT option to return the original input to be honored

O2_ IMS 13 TM: 67

OTMA transaction expiration at GU time has been updated to provide two enhancements:

•By default, the symptom dump of U243 abend and its DFS554A message for an expired transaction are no longer generated to save the CPU cycles. OTMA, however, can be configured to generate the symptom dump and DFS554A message by specifying TODUMP=YES in the DFSYDTx PROCLIB member or by setting the TMAMDUMP flag in the OTMA state data prefix of the input message.

•Also by default, a DFS3688I message instead of DFS555I/DFS2224 is sent to the OTMA client at application GU transaction expiration. OTMA, however, now honors a request to return the original input message instead of the DFS3688I if the TMAMINPT flag is set in the OTMA prefix of the input message. WMQ 7.1.0 with APAR PM47795 supports this capability and sets the flag for requests where the MQMD_REPORT option is to return the original input message.

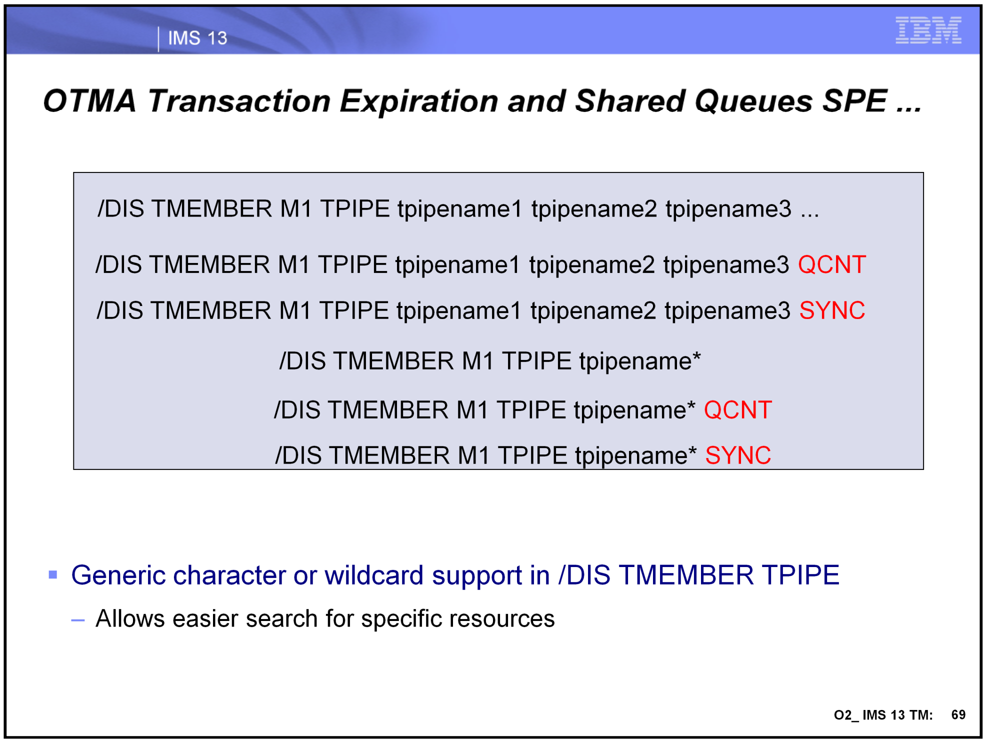


OTMA Transaction Expiration and Shared Queues SPE ...

- Ability to override a SQ back-end affinity for ALTPCB messages
  - Allows ALTPCB output generated in a back-end to be routed back to a front-end for delivery
  - Options:
    - Specifying ALTPCBE= YES in the OTMA client descriptor in the DFSYDTx PROCLIB member of the front-end IMS system
    - Setting the TMAMALTB flag (X'01‘) in the state data prefix of the input message
  - Of value when supermember support (e.g., IMS Connect) is not used or not available (e.g., WMQ)

Shared queues environments can sometimes pose problems for ALTPCB output that is generated at a back-end IMS. By default, the back-end ALTPCB output is queued to the shared queues with an affinity to that system and must be retrieved from there. IMS Connect has resolved this issue with its supermember support. Other OTMA member clients, however, like WebSphere MQ that are connected to the front-end IMS will not easily be able to retrieve the message. Note that IMS Connect will also have this problem if supermember support is not enabled.

With the enhancements introduced in this SPE, routing of transaction ALTPCB responses in an IMS shared queues environment has been improved. Response output issued to an IMS ALTPCB destination by a back-end IMS system can be more easily returned to non-supermember clients attached to the front-end. This optional IMS function is enabled by specifying ALTPCBE= YES in the OTMA DFSYDTx PROCLIB member or by setting the TMAMALTB flag in the state data prefix of the input transaction.

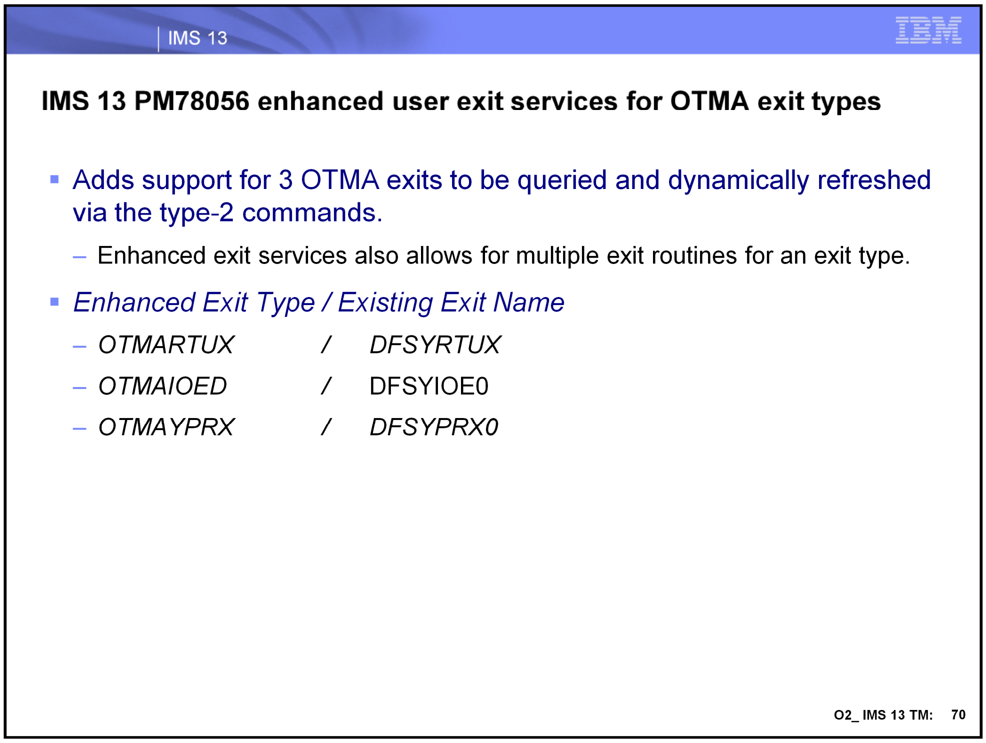


OTMA Transaction Expiration and Shared Queues SPE ...

```
/DIS TMEMBER M1 TPIPE tpipename1 tpipename2 tpipename3 ...
/DIS TMEMBER M1 TPIPE tpipename1 tpipename2 tpipename3 QCNT
/DIS TMEMBER M1 TPIPE tpipename1 tpipename2 tpipename3 SYNC
/DIS TMEMBER M1 TPIPE tpipename*
/DIS TMEMBER M1 TPIPE tpipename* QCNT
/DIS TMEMBER M1 TPIPE tpipename* SYNC
```

- Generic character or wildcard support in /DIS TMEMBER TPIPE
  - Allows easier search for specific resources

IMS improves OTMA usability by adding support for a generic character or wildcard at the end of the TPIPE name in the IMS command /DISPLAY TMEMBER TPIPE. By using the wildcard character, a series of TPIPE queues that are similarly named can be easily displayed. The generic form of the TPIPE name can also be used with the QCNT or SYNC keywords. Note that the TPIPE name with a generic character cannot be mixed with other TPIPE names.

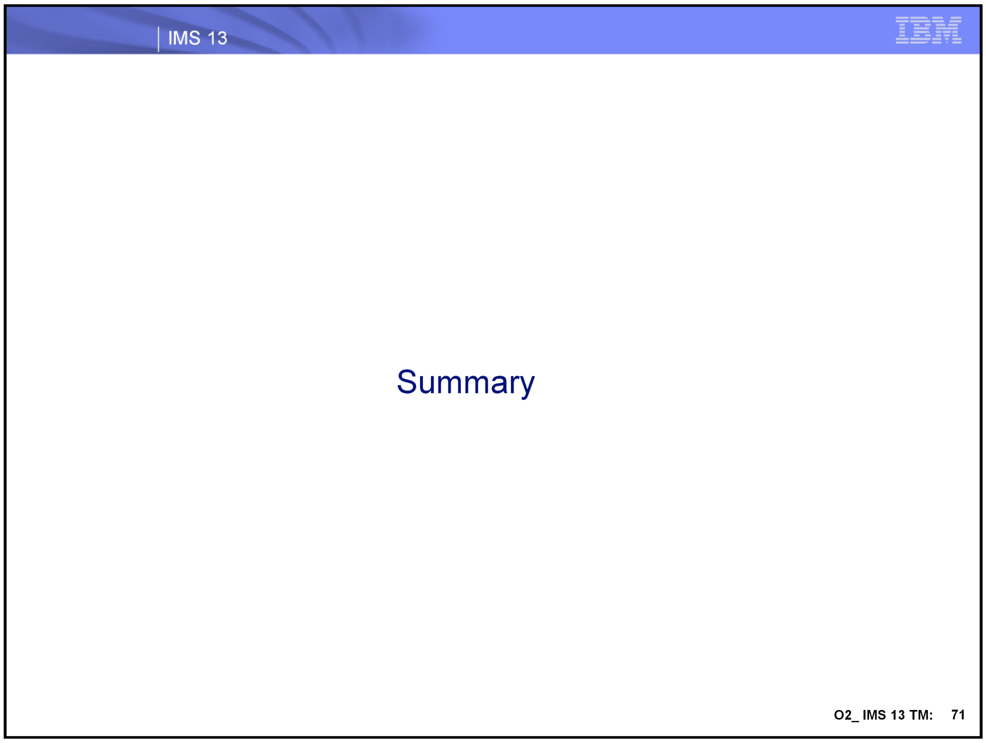


IMS 13 PM78056 enhanced user exit services for OTMA exit types

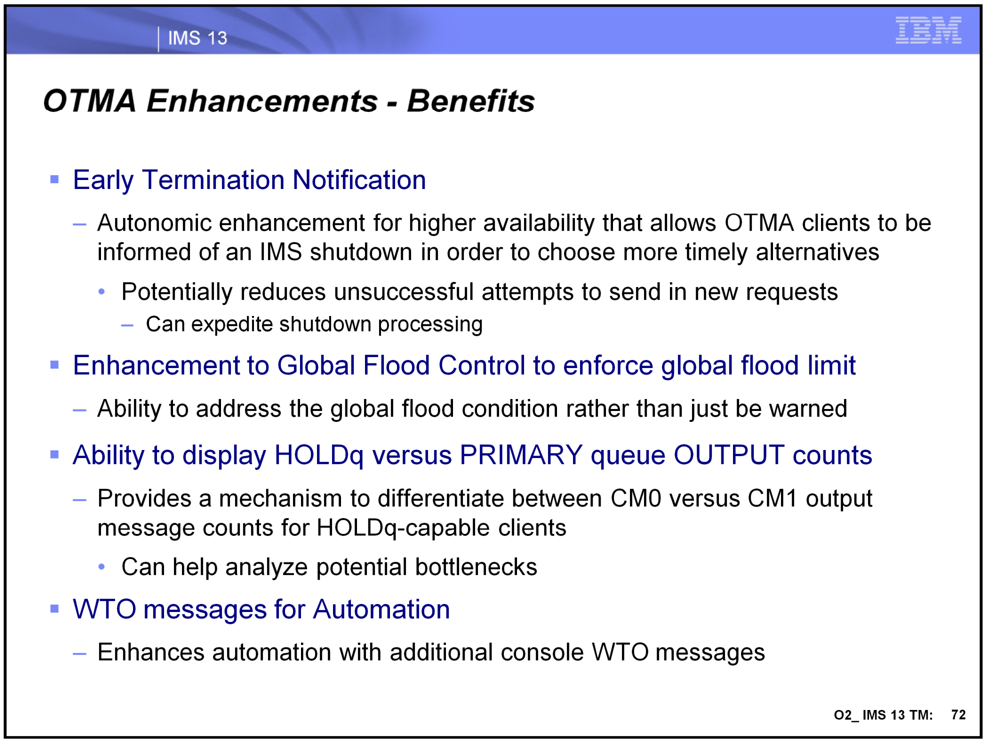
- Adds support for 3 OTMA exits to be queried and dynamically refreshed via the type-2 commands.
  - Enhanced exit services also allows for multiple exit routines for an exit type.
- *Enhanced Exit Type / Existing Exit Name*

OTMARTUX	*/*	*DFSYRTUX*
OTMAIOED	*/*	DFSYIOE0
OTMAYPRX	*/*	*DFSYPRX0*

Enable 3 OTMA exit types for the enhanced user exit services. If an installation wishes to not use the enhanced services, all of the existing exits will continue to work as they currently do.

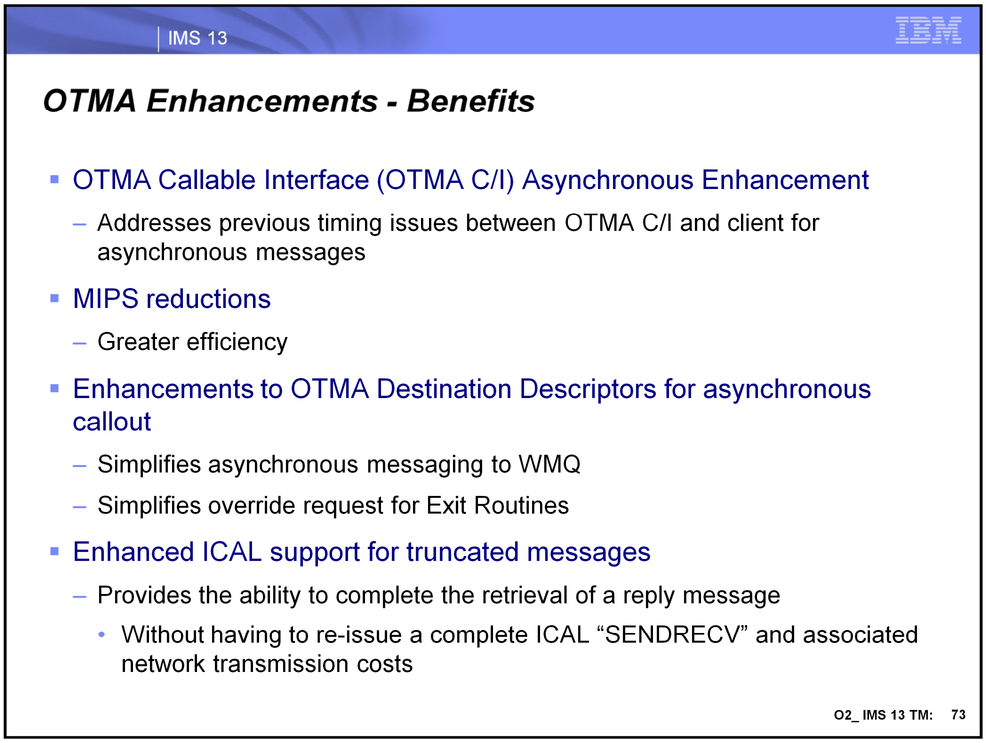


Summary



OTMA Enhancements - Benefits

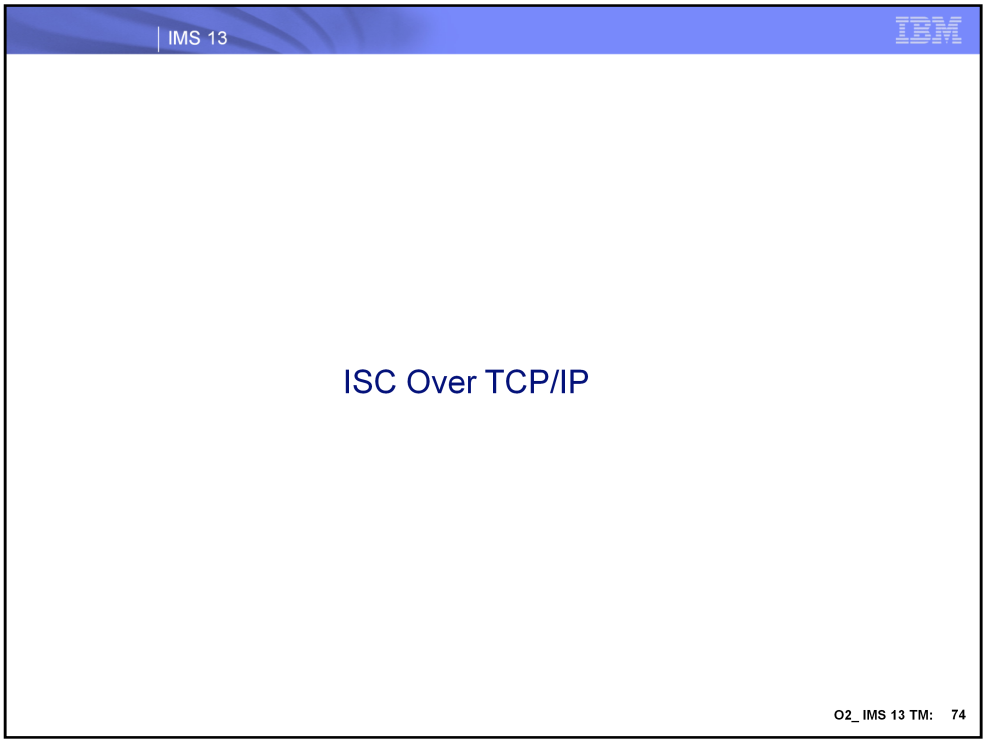
- Early Termination Notification
  - Autonomic enhancement for higher availability that allows OTMA clients to be informed of an IMS shutdown in order to choose more timely alternatives
    - Potentially reduces unsuccessful attempts to send in new requests
      - Can expedite shutdown processing
- Enhancement to Global Flood Control to enforce global flood limit
  - Ability to address the global flood condition rather than just be warned
- Ability to display HOLDq versus PRIMARY queue OUTPUT counts
  - Provides a mechanism to differentiate between CM0 versus CM1 output message counts for HOLDq-capable clients
    - Can help analyze potential bottlenecks
- WTO messages for Automation
  - Enhances automation with additional console WTO messages



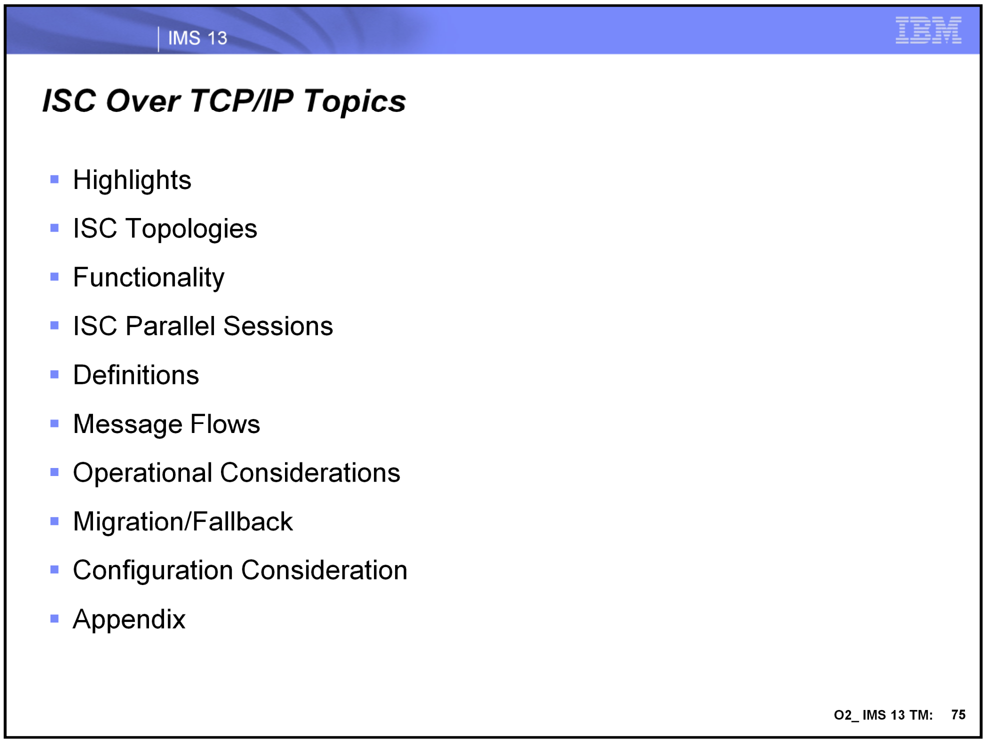
IMS 13

OTMA Enhancements - Benefits

- OTMA Callable Interface (OTMA C/I) Asynchronous Enhancement
  - Addresses previous timing issues between OTMA C/I and client for asynchronous messages
- MIPS reductions
  - Greater efficiency
- Enhancements to OTMA Destination Descriptors for asynchronous callout
  - Simplifies asynchronous messaging to WMQ
  - Simplifies override request for Exit Routines
- Enhanced ICAL support for truncated messages
  - Provides the ability to complete the retrieval of a reply message
    - Without having to re-issue a complete ICAL “SENDRECV” and associated network transmission costs

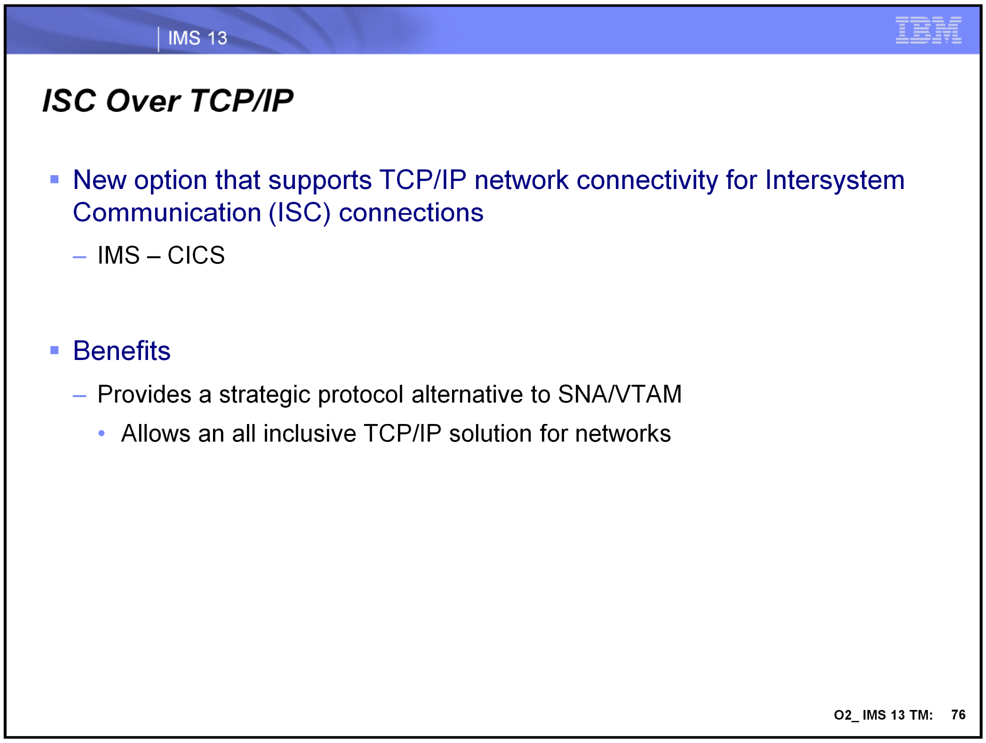


ISC Over TCP/IP



ISC Over TCP/IP Topics

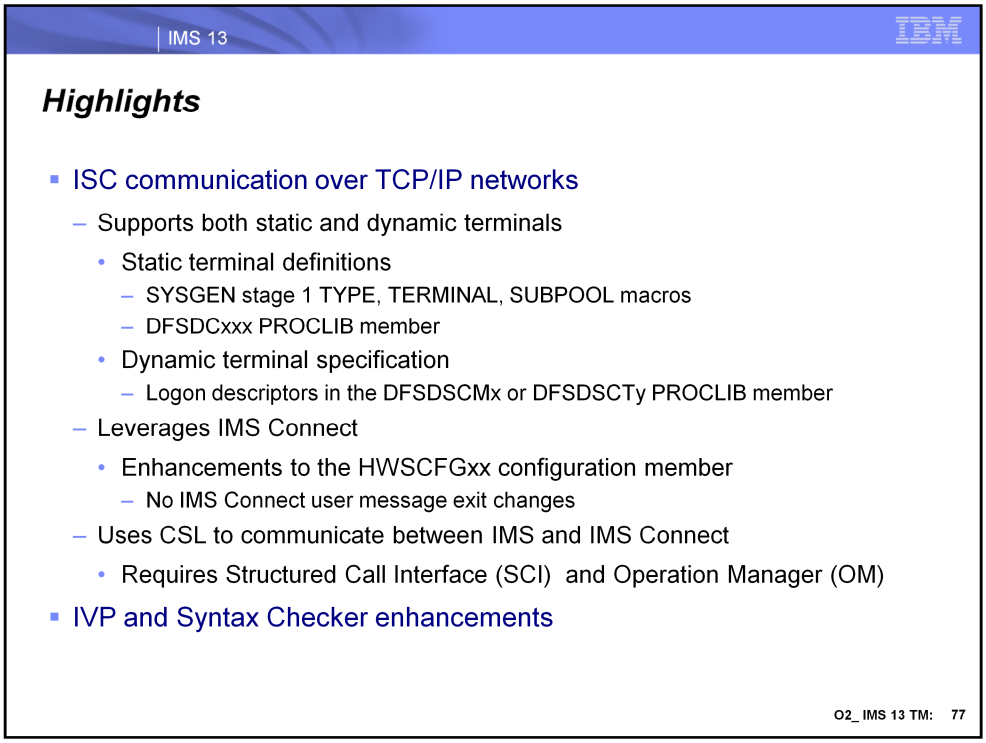
- Highlights
- ISC Topologies
- Functionality
- ISC Parallel Sessions
- Definitions
- Message Flows
- Operational Considerations
- Migration/Fallback
- Configuration Consideration
- Appendix



ISC Over TCP/IP

- New option that supports TCP/IP network connectivity for Intersystem Communication (ISC) connections
  - IMS – CICS
- Benefits
  - Provides a strategic protocol alternative to SNA/VTAM
    - Allows an all inclusive TCP/IP solution for networks

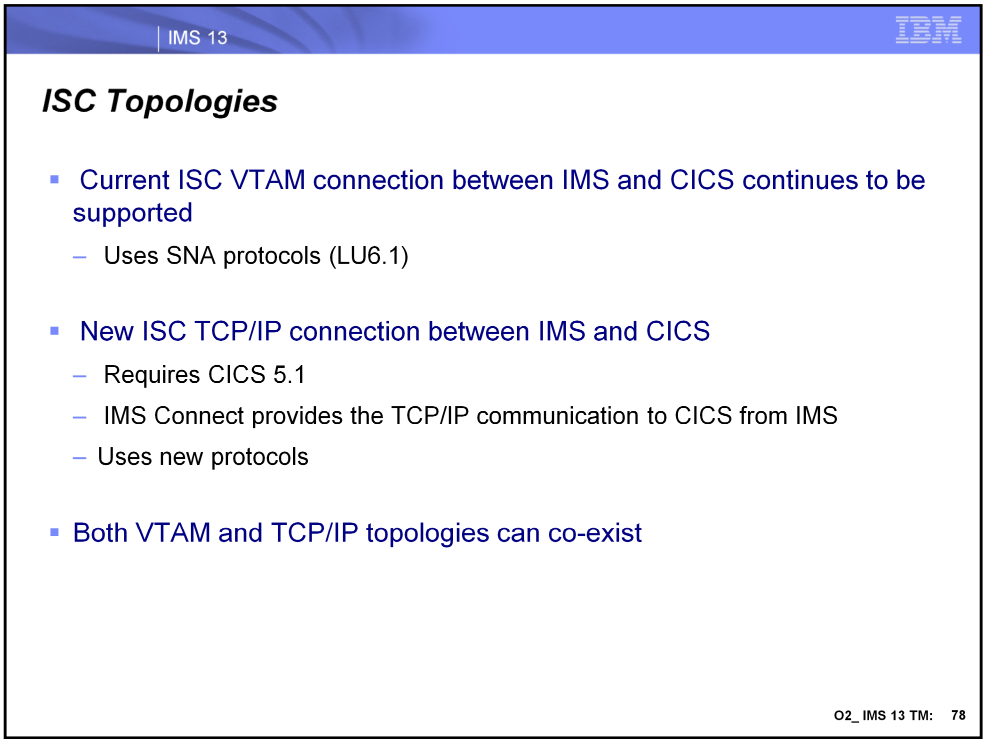
IMS 13 enhances Intersystem Communication (ISC) connectivity to included the TCP/IP protocol. This capability applies to connections between IMS and Customer Information Control System (CICS) systems. This new support provides a solution for environments that want to implement an all inclusive TCP/IP network environment.



Highlights

- ISC communication over TCP/IP networks
  - Supports both static and dynamic terminals
    - Static terminal definitions
      - SYSGEN stage 1 TYPE, TERMINAL, SUBPOOL macros
      - DFSDCxxx PROCLIB member
    - Dynamic terminal specification
      - Logon descriptors in the DFSDSCMx or DFSDSCTy PROCLIB member
  - Leverages IMS Connect
    - Enhancements to the HWSCFGxx configuration member
      - No IMS Connect user message exit changes
  - Uses CSL to communicate between IMS and IMS Connect
    - Requires Structured Call Interface (SCI) and Operation Manager (OM)
- IVP and Syntax Checker enhancements

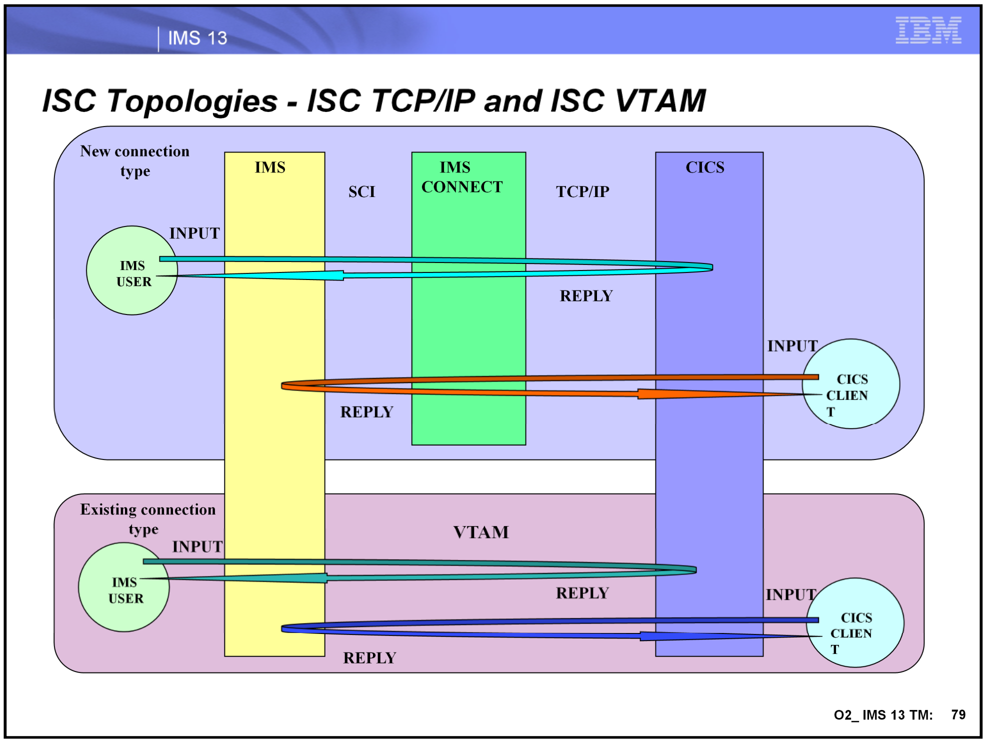
At a high level, the ISC support using TCP/IP protocols can be defined in IMS environments both statically and dynamically (ETO). It uses the SCI (structure call interface) of the CSL (common service layer) between IMS and IMS Connect in order to leverage the existing IMS Connect TCP/IP support.



ISC Topologies

- Current ISC VTAM connection between IMS and CICS continues to be supported
  - Uses SNA protocols (LU6.1)
- New ISC TCP/IP connection between IMS and CICS
  - Requires CICS 5.1
  - IMS Connect provides the TCP/IP communication to CICS from IMS
  - Uses new protocols
- Both VTAM and TCP/IP topologies can co-exist

ISC VTAM (SNA LU 6.1) connections continue to be supported in IMS 13 alongside the new ISC TCP/IP capability. Connectivity to CICS using TCP/IP requires CICS 5.1. Both VTAM and TCP/IP connections can co-exist in both environments.



IMS 13

ISC Topologies - ISC TCP/IP and ISC VTAM

New connection type

IMS

SCI

IMS CONNECT

TCP/IP

CICS

INPUT

IMS USER

REPLY

INPUT

CICS CLIENT

REPLY

Existing connection type

VTAM

INPUT

IMS USER

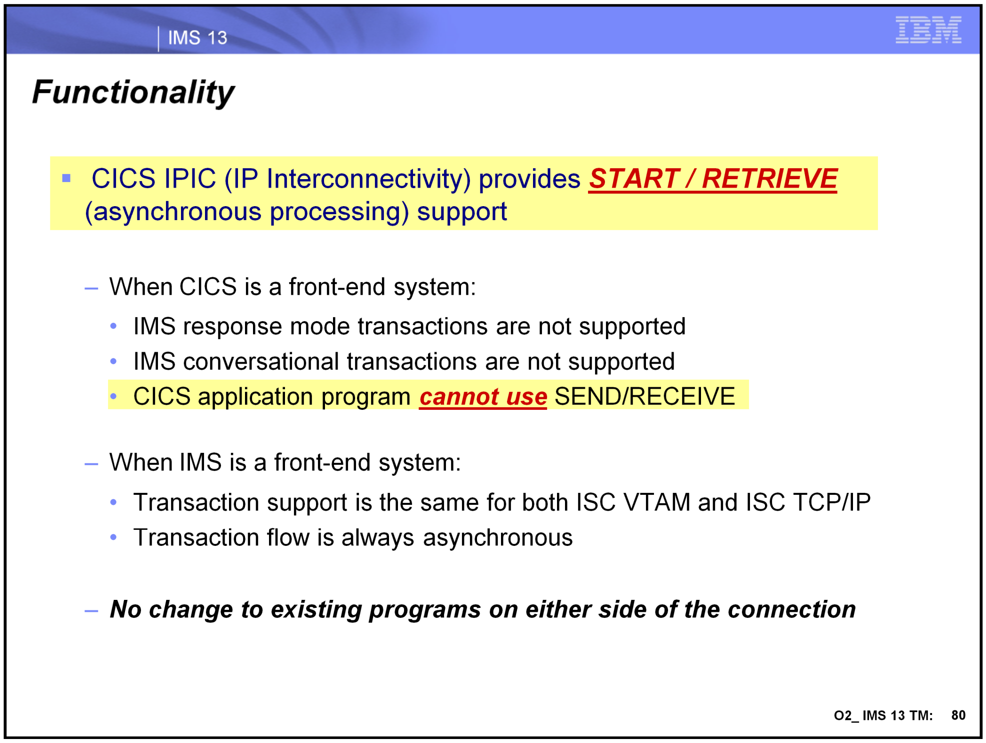
REPLY

INPUT

CICS CLIENT

REPLY

This visual depicts an IMS and CICS system supporting both VTAM and TCP/IP connection. For ISC TCP/IP, IMS communicates with IMS Connect via the Structured Call Interface (SCI). IMS Connect communicates with CICS via TCP/IP network.



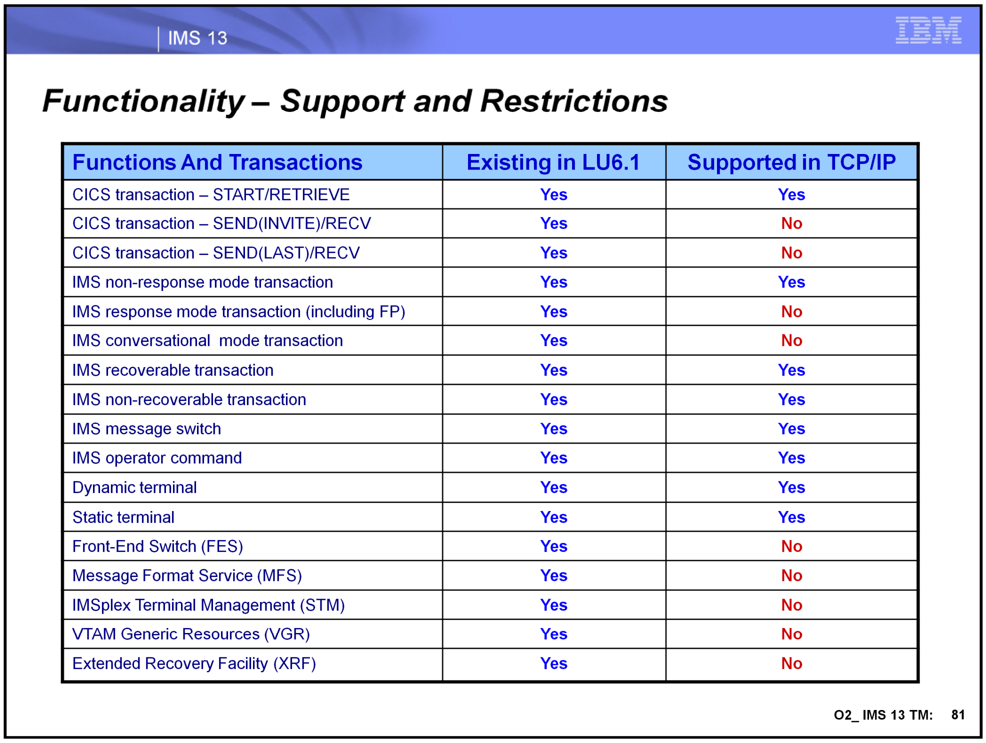
Functionality

- CICS IPIC (IP Interconnectivity) provides ***START / RETRIEVE*** (asynchronous processing) support
  - When CICS is a front-end system:
    - IMS response mode transactions are not supported
    - IMS conversational transactions are not supported
    - CICS application program ***cannot use*** SEND/RECEIVE
  - When IMS is a front-end system:
    - Transaction support is the same for both ISC VTAM and ISC TCP/IP
    - Transaction flow is always asynchronous
  - ***No change to existing programs on either side of the connection***

Connectivity between CICS and IMS only supports the START/RETRIEVE or asynchronous support in IMS 13. This means that when CICS is the front-end system, neither IMS response mode or conversational transactions can be invoked. Additionally, CICS application programs are not able to issue a SEND/RECEIVE (synchronous) type of request.

When IMS is the front-end system, transaction flow is always asynchronous. This interaction has always been true for the ISC VTAM support and is now the same for ISC TCP/IP.

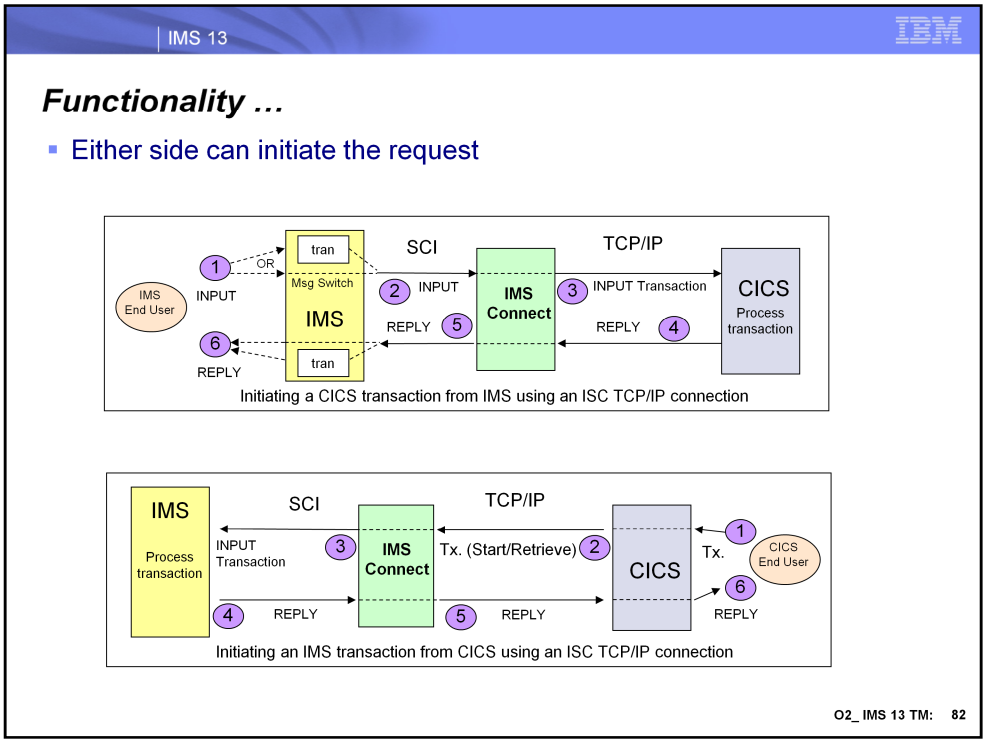
For IMS programs and CICS START/RETRIEVE programs, no changes on either side of the connection is required.



Functionality – Support and Restrictions

Functions And Transactions	Existing in LU6.1	Supported in TCP/IP
CICS transaction – START/RETRIEVE	Yes	Yes
CICS transaction – SEND(INVITE)/RECV	Yes	No
CICS transaction – SEND(LAST)/RECV	Yes	No
IMS non-response mode transaction	Yes	Yes
IMS response mode transaction (including FP)	Yes	No
IMS conversational mode transaction	Yes	No
IMS recoverable transaction	Yes	Yes
IMS non-recoverable transaction	Yes	Yes
IMS message switch	Yes	Yes
IMS operator command	Yes	Yes
Dynamic terminal	Yes	Yes
Static terminal	Yes	Yes
Front-End Switch (FES)	Yes	No
Message Format Service (MFS)	Yes	No
IMSplex Terminal Management (STM)	Yes	No
VTAM Generic Resources (VGR)	Yes	No
Extended Recovery Facility (XRF)	Yes	No

This visual documents the supported/restricted functionality of the ISC TCP/IP support.



IMS 13

Functionality …

- Either side can initiate the request

Initiating a CICS transaction from IMS using an ISC TCP/IP connection

Initiating an IMS transaction from CICS using an ISC TCP/IP connection

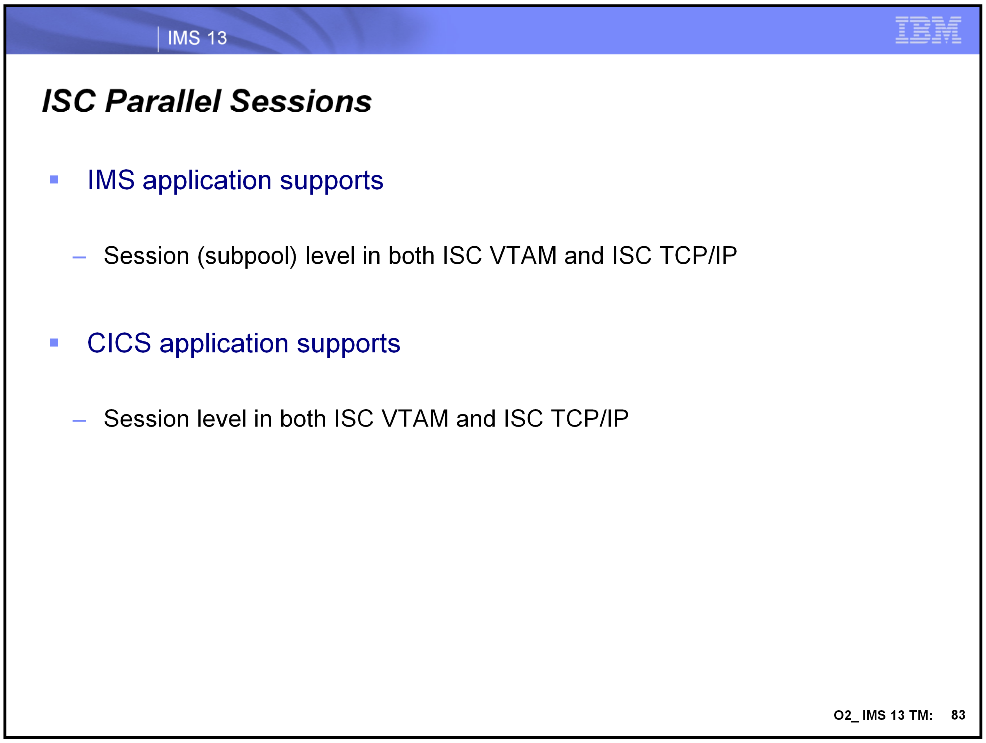
O2_ IMS 13 TM: 82

Initiating a CICS transaction from IMS using an ISC TCP/IP connection:

1. An IMS end user sends in a message-switch that is to be sent to CICS or an input message to an IMS transaction that issues and ISRT, ALTPCB for CICS.
2. IMS uses an active ISC TCP/IP connection to pass the message to IMS Connect using SCI.
3. IMS Connect uses an active socket connection to send the message to CICS.
4. CICS receives the message as a transaction from TCP/IP network, processes it and uses an active ISC TCP/IP connection to send the transaction reply back to IMS.
5. IMS Connect receives the transaction reply from TCP/IP network, passes it to IMS over an active ISC TCP/IP connection using SCI.
6. The reply can invoke an IMS transaction or be a reply that is targeted for an IMS end-user.

Initiating an IMS transaction from CICS using an ISC TCP/IP connection:

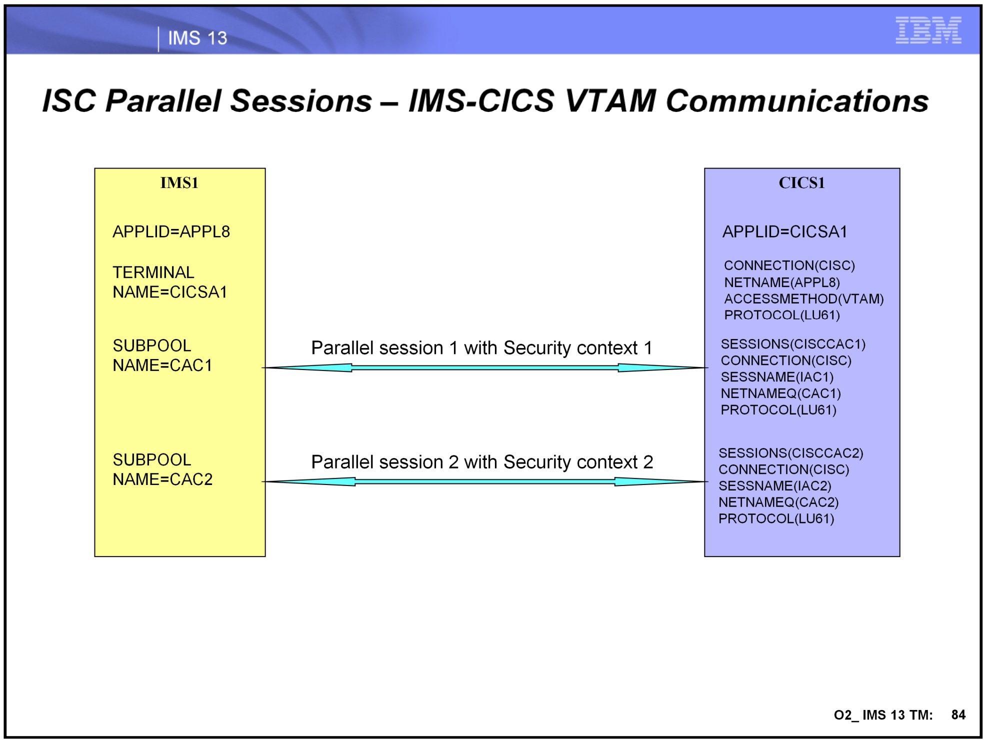
1. CICS client enters a transaction request.
2. CICS uses an active ISC TCP/IP connection to send the transaction to IMS.
3. IMS Connect receives the transaction from TCP/IP network and uses an active ISC TCP/IP connection to pass the message to IMS using SCI.
4. IMS processes the transaction, uses an active ISC TCP/IP connection using SCI to pass the transaction reply to IMS Connect.
5. IMS Connect uses an active socket connection to send transaction reply to CICS.
6. CICS sends the reply back to the client.



ISC Parallel Sessions

- IMS application supports
  - Session (subpool) level in both ISC VTAM and ISC TCP/IP
- CICS application supports
  - Session level in both ISC VTAM and ISC TCP/IP

Parallel sessions are supported in ISC TCP/IP connections as they are in VTAM.



IMS 13

ISC Parallel Sessions – IMS-CICS VTAM Communications

IMS1

APPLID=APPL8

TERMINAL
NAME=CICSA1

SUBPOOL
NAME=CAC1

SUBPOOL
NAME=CAC2

Parallel session 1 with Security context 1

Parallel session 2 with Security context 2

CICS1

APPLID=CICSA1

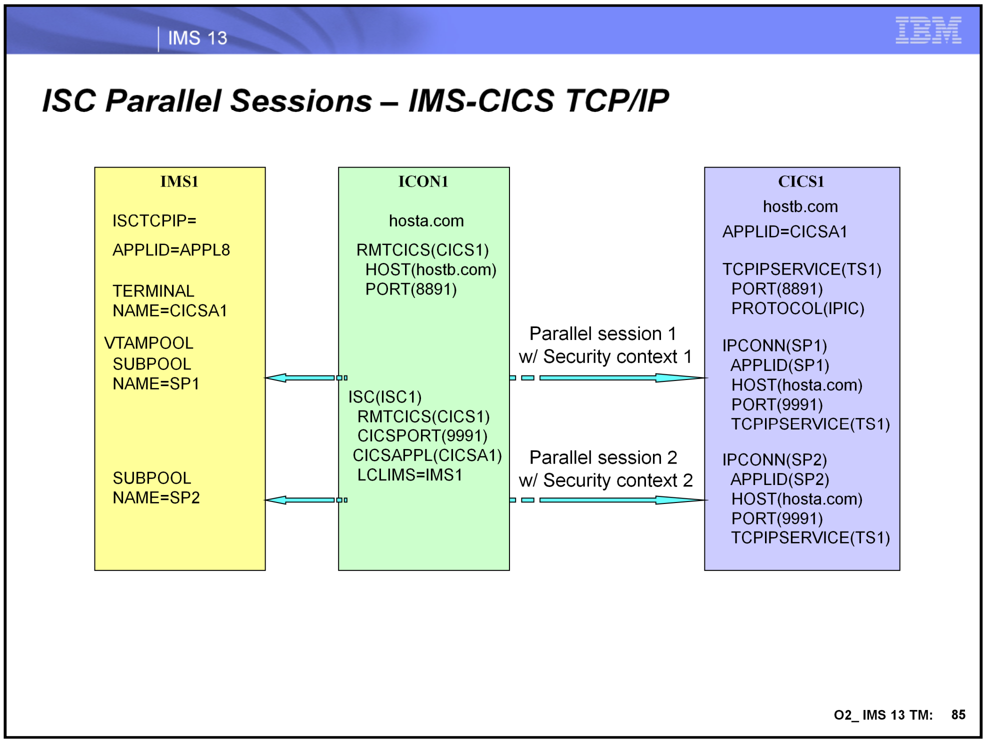
CONNECTION(CISC)
NETNAME(APPL8)
ACCESSMETHOD(VTAM)
PROTOCOL(LU61)

SESSIONS(CISCCAC1)
CONNECTION(CISC)
SESSNAME(IAC1)
NETNAMEQ(CAC1)
PROTOCOL(LU61)

SESSIONS(CISCCAC2)
CONNECTION(CISC)
SESSNAME(IAC2)
NETNAMEQ(CAC2)
PROTOCOL(LU61)

O2_ IMS 13 TM: 84

This visual gives an example of an existing set of ISC VTAM sessions between IMS and CICS.



ISC Parallel Sessions – IMS-CICS TCP/IP

IMS1

ISCTCPIP=
APPLID=APPL8

TERMINAL
NAME=CICSA1

VTAMPOOL
SUBPOOL
NAME=SP1

SUBPOOL
NAME=SP2

ICON1

hosta.com

RMTCICS(CICS1)
HOST(hostb.com)
PORT(8891)

ISC(ISC1)
RMTCICS(CICS1)
CICSPORT(9991)
CICSAPPL(CICSA1)
LCLIMS=IMS1

Parallel session 1
w/ Security context 1

Parallel session 2
w/ Security context 2

CICS1

hostb.com

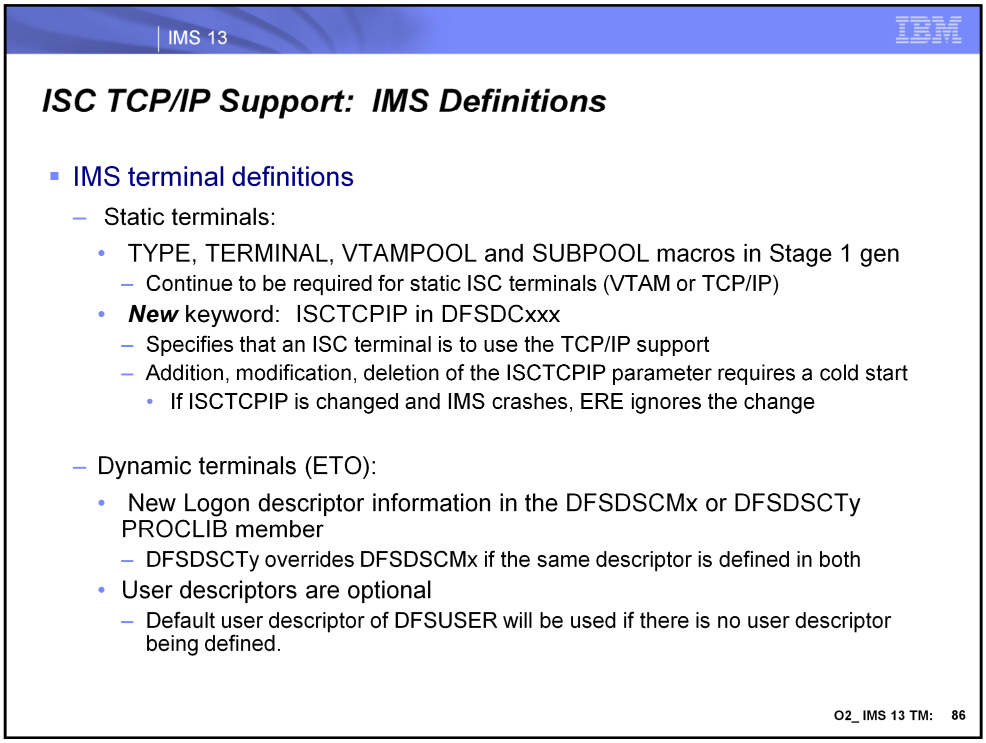
APPLID=CICSA1

TCPIPSERVICE(TS1)
PORT(8891)
PROTOCOL(IPIC)

IPCONN(SP1)
APPLID(SP1)
HOST(hosta.com)
PORT(9991)
TCPIPSERVICE(TS1)

IPCONN(SP2)
APPLID(SP2)
HOST(hosta.com)
PORT(9991)
TCPIPSERVICE(TS1)

This visual gives an example of an existing set of ISC TCP/IP sessions between IMS and CICS. The next step is look at the definition requirements in IMS, IMS Connect and CICS.

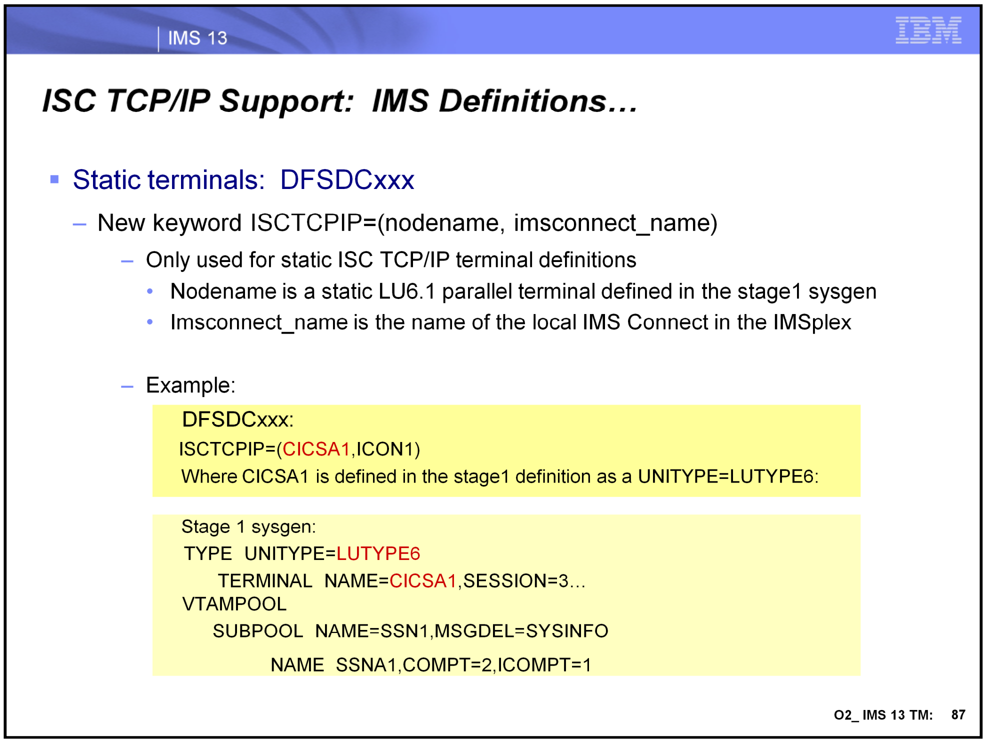


ISC TCP/IP Support: IMS Definitions

- IMS terminal definitions
  - Static terminals:
    - TYPE, TERMINAL, VTAMPOOL and SUBPOOL macros in Stage 1 gen
      - Continue to be required for static ISC terminals (VTAM or TCP/IP)
    - ***New*** keyword: ISCTCPIP in DFSDCxxx
      - Specifies that an ISC terminal is to use the TCP/IP support
      - Addition, modification, deletion of the ISCTCPIP parameter requires a cold start
        - If ISCTCPIP is changed and IMS crashes, ERE ignores the change
  - Dynamic terminals (ETO):
    - New Logon descriptor information in the DFSDSCMx or DFSDSCTy PROCLIB member
      - DFSDSCTy overrides DFSDSCMx if the same descriptor is defined in both
    - User descriptors are optional
      - Default user descriptor of DFSUSER will be used if there is no user descriptor being defined.

In IMS 13, both dynamic (ETO) and static ISC terminals can use the new TCP/IP support. Static terminal support requires a new parameter, ISCTCPIP, that is specified in the DFSDCxxx member of the IMS PROCLIB dataset. Dynamic/ETO (Extended Terminal Option) ISC terminals use logon descriptors that can be specified in either the DFSDSCMx or the DFSDSCTy members of the IMS.PROCLIB data set. The use of static versus dynamic ISC TCP/IP terminals provides equivalency in many respects, e.g., connections as well as status can be initiated or terminated via IMS commands and both input and output messages can be transported using either terminal type. Where they differ is that dynamic terminals do not require a system definition or a cold start of IMS thereby providing much greater flexibility than static ISC terminals.

For dynamic/ETO terminals new logon descriptor can be added to either the IMS DFSDSCMx or DFSDSCTy PROCLIB member. At least one logon descriptor must exist for the dynamic ISC TCP/IP support and a unique logon descriptor is required for each ISC TCP/IP connection. A default logon descriptor name is DFSLU6T. As a reminder, IMS optionally produces ETO descriptors during system definition placing them in DFSDCMx by specifying the ETOFEAT keyword in the IMSCTRL macro. Subsequent system definitions of the same stage 1 input deck override the DFSDSCMx members. If TSO or a z/OS utility is used to create descriptors, these are placed in member DFSDSCTy which avoids loss when member DFSDSCMx is replaced. TSO or a z/OS utility can also update descriptors in DFSDSCMx that were created during system definition.



IMS 13

ISC TCP/IP Support: IMS Definitions…

- Static terminals: DFSDCxxx
  - New keyword ISCTCPIP=(nodename, imsconnect_name)
    - Only used for static ISC TCP/IP terminal definitions
      - Nodename is a static LU6.1 parallel terminal defined in the stage1 sysgen
      - Imsconnect_name is the name of the local IMS Connect in the IMSplex
    - Example:

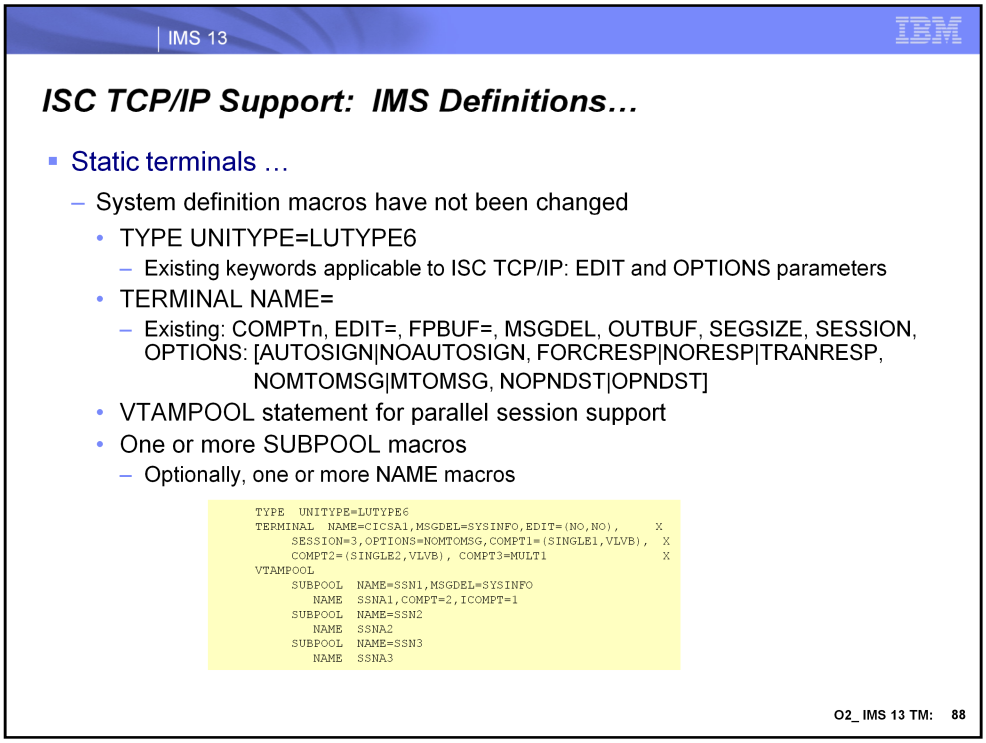
```
DFSDCxxx:
ISCTCPIP=(CICSA1,ICON1)
Where CICSA1 is defined in the stage1 definition as a UNITYPE=LUTYPE6:
```

```
Stage 1 sysgen:
TYPE  UNITYPE=LUTYPE6
     TERMINAL  NAME=CICSA1,SESSION=3…
VTAMPOOL
     SUBPOOL  NAME=SSN1,MSGDEL=SYSINFO
              NAME  SSNA1,COMPT=2,ICOMPT=1
```

O2_ IMS 13 TM: 87

IMS enhances the DFSDCxxx PROCLIB member to provide static ISC TCP/IP terminal support for parallel sessions. Static ISC TCP/IP terminals do not support single ISC sessions. Because ISC terminals use VTAM support by default, the ISCTCPIP parameter in the DFSDCxxx PROCLIB member must be defined to enable TCP/IP for static LU 6.1 ISC terminals.

ISCTCPIP Sets a static LU 6.1 ISC terminal to use TCP/IP communication protocols. The *nodename* value is the name of an LU 6.1 ISC terminal, as specified on the NAME parameter of the TERMINAL macro that is part of the terminal system definition. The *imsconnect_name* value is the IMSplex name of a local IMS Connect instance that manages TCP/IP communications for this IMS. The value must match the name specified on the MEMBER parameter of either the ISC statement or the IMSplex statement in the IMS Connect configuration PROCLIB member (HWSCFGxx).



ISC TCP/IP Support: IMS Definitions…

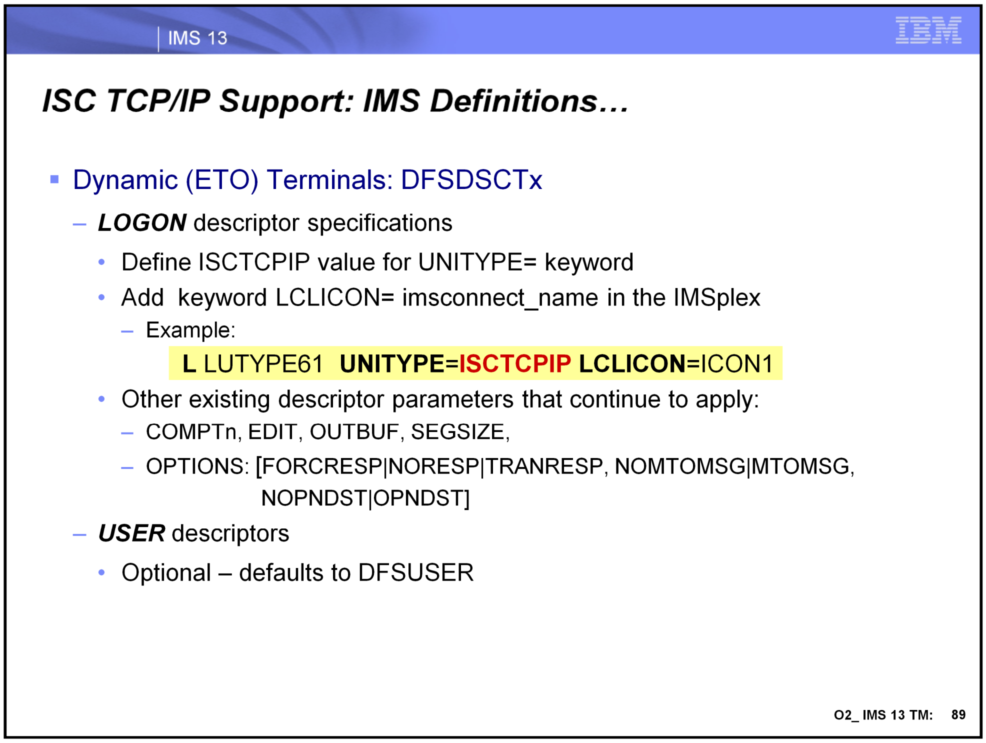
- Static terminals …
  - System definition macros have not been changed
    - TYPE UNITYPE=LUTYPE6
      - Existing keywords applicable to ISC TCP/IP: EDIT and OPTIONS parameters
    - TERMINAL NAME=
      - Existing: COMPTn, EDIT=, FPBUF=, MSGDEL, OUTBUF, SEGSIZE, SESSION, OPTIONS: [AUTOSIGN|NOAUTOSIGN, FORCRESP|NORESP|TRANRESP, NOMTOMSG|MTOMSG, NOPNDST|OPNDST]
    - VTAMPOOL statement for parallel session support
    - One or more SUBPOOL macros
      - Optionally, one or more NAME macros

```
TYPE  UNITYPE=LUTYPE6
TERMINAL  NAME=CICSA1,MSGDEL=SYSINFO,EDIT=(NO,NO),     X
     SESSION=3,OPTIONS=NOMTOMSG,COMPT1=(SINGLE1,VLVB),  X
     COMPT2=(SINGLE2,VLVB), COMPT3=MULT1                X
VTAMPOOL
     SUBPOOL  NAME=SSN1,MSGDEL=SYSINFO
        NAME  SSNA1,COMPT=2,ICOMPT=1
     SUBPOOL  NAME=SSN2
        NAME  SSNA2
     SUBPOOL  NAME=SSN3
        NAME  SSNA3
```

When switching an existing terminal from VTAM to TCP/IP, ensure that UNITTYPE=LUTYPE6 is specified on the TYPE macro that defines the ISC TCP/IP terminal. If UNITTYPE=LUTYPE6 is not specified, an error message will be issued when the ISCTCPIP parameter (discussed in the previous visual) is processed in the DFSDCxxx member and TCP/IP will not be enabled for the terminal. Other existing parameters in the TYPE macros continue to apply to the new support. The EDIT parameter can continue to be used to define any user-provided physical terminal output edit routine and/or physical terminal input edit routine. For OPTIONS: AUTOSIGN indicates that IMS automatically signs on to static terminals. For ISC terminals the SUBPOOL name is the user ID. NOAUTSGN indicates that IMS does not automatically sign on to static terminals.. This is the default. The AUTOSIGN and NOAUTSGN definitions on the TERMINAL macro override those on the TYPE.

For the TERMINAL macro, existing keywords that apply to the new TCP/IP capability include: COMPTn, EDIT=, FPBUF=, MSGDEL=, OUTBUF=, SEGSIZE=, SESSION= and OPTIONS. The following OPTIONS are applicable to this support: AUTOSIGN|NOAUTOSIGN, FORCRESP|NORESP|TRANRESP, NOMTOMSG|MTOMSG, NOPNDST|OPNDST.

Although the ISC TCP/IP support does not use VTAMPOOL there must be at least 1 VTAMPOOL statement defined for parallel session support and one or more subsequent SUBPOOL macros, each of which can optionally be followed by one or more NAME macros (if at least one SUBPOOL statement has an associated NAME statement).



IMS 13

ISC TCP/IP Support: IMS Definitions…

- Dynamic (ETO) Terminals: DFSDSCTx
  - ***LOGON*** descriptor specifications
    - Define ISCTCPIP value for UNITYPE= keyword
    - Add keyword LCLICON= imsconnect_name in the IMSplex
      - Example:

        L LUTYPE61 **UNITYPE**=ISCTCPIP **LCLICON**=ICON1
    - Other existing descriptor parameters that continue to apply:
      - COMPTn, EDIT, OUTBUF, SEGSIZE,
      - OPTIONS: [FORCRESP|NORESP|TRANRESP, NOMTOMSG|MTOMSG, NOPNDST|OPNDST]
  - ***USER*** descriptors
    - Optional – defaults to DFSUSER

O2_ IMS 13 TM: 89

As mentioned earlier, with the ETOFEAT enabled, stage 1 system definitions for ISC terminals can generate logon descriptors which are placed in the DFSDSCMx member of PROCLIB. On the other hand, you can code logon descriptors directly in the DFSDSCTx member. The descriptors that you can specify in the DFSDSCTy member, as well as the syntax and parameters of those descriptors, are exactly the same as the descriptors that can exist in DFSDSCMx. If a descriptor is specified on both members, however, the specifications coded in the DFSDSCTy member for the descriptor override the specifications coded in the DFSDSCMx member.

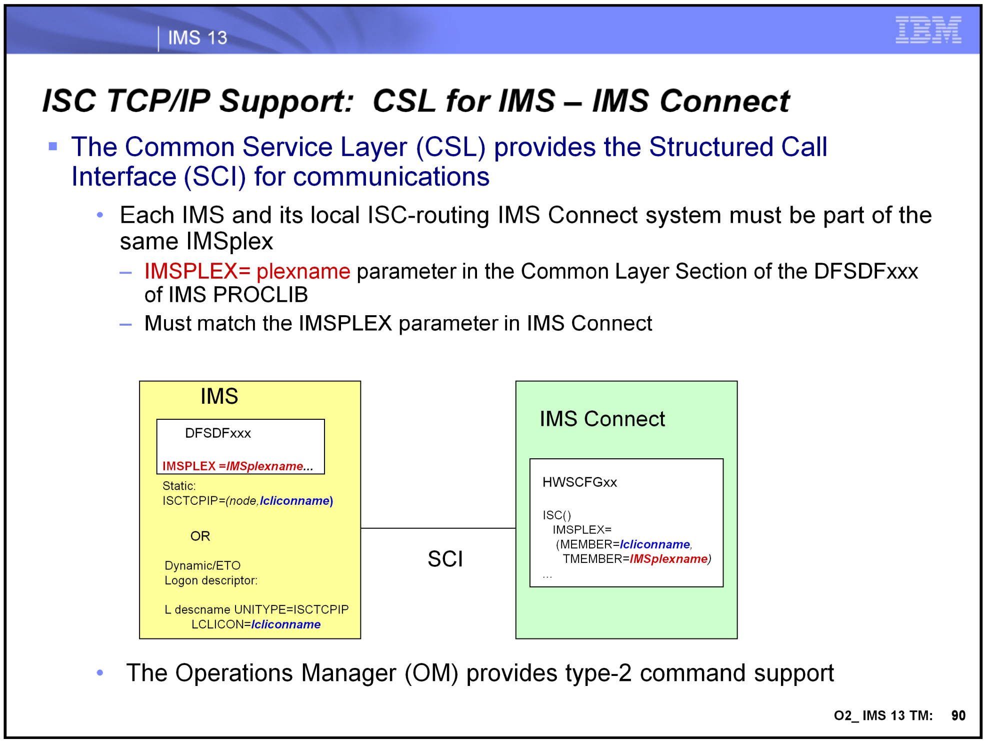
The logon descriptor for ISC TCP/IP terminal support includes:

•Specification of UNITYPE=ISCTCPIP

• The LCLICON keyword which is only applicable when UNITYPE=ISCTCPIP is specified. For ISC TCP/IP, this keyword identifies the local IMS Connect (IMSCON) member within an IMSplex that this LUTYPE6T terminal connects to via SCI. *Lcl_imsconnect_name* is a 1 to 8 alphanumeric character name.

As with static definitions, other existing parameters for ISC logon descriptors apply. These include: COMPTn, EDIT=, OUTBUF=, SEGSIZE=, SESSION= and OPTIONS. The following OPTIONS are applicable to this support: FORCRESP|NORESP|TRANRESP, NOMTOMSG|MTOMSG, NOPNDST|OPNDST.

As in previous releases, USER descriptors are not required and default to the use of DFSUSER. If specified, the following keywords are applicable to the ISC TCP/IP support: LTERM, OPTIONS [see logon descriptor], RCVYCONV, RCVYFP, RCVYRESP, RCVYSTSN, and SRMDEF.



ISC TCP/IP Support: CSL for IMS – IMS Connect

- The Common Service Layer (CSL) provides the Structured Call Interface (SCI) for communications
  - Each IMS and its local ISC-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
    - Must match the IMSPLEX parameter in IMS Connect

IMS
SCI
IMS Connect

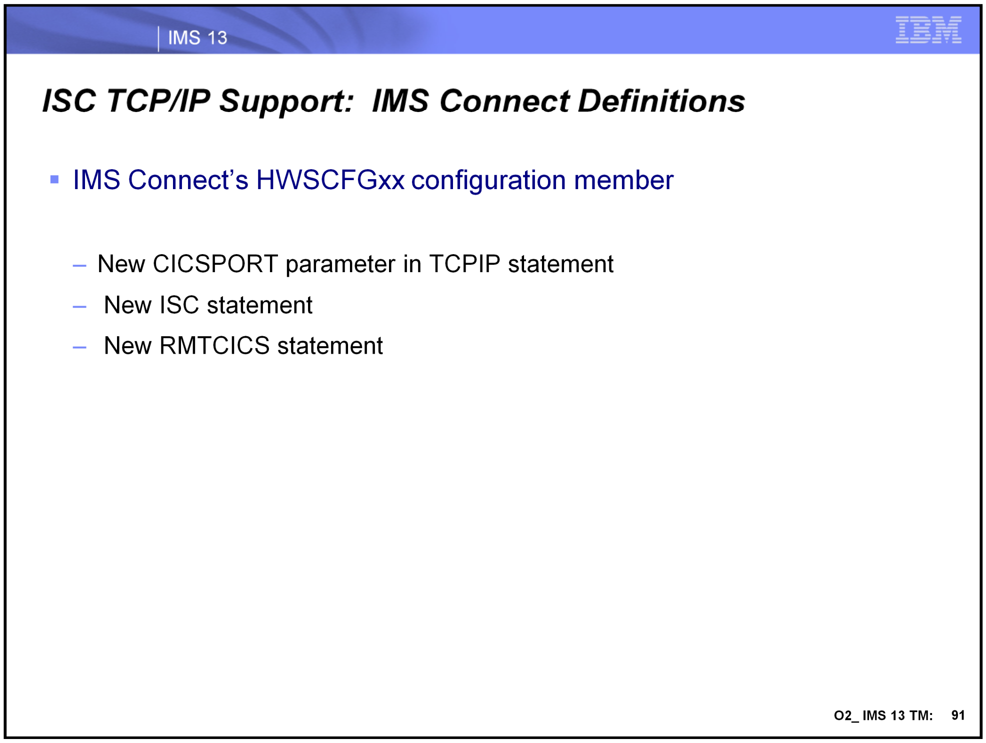
```
DFSDFxxx

IMSPLEX =IMSplexname...
```

```
Static:
ISCTCPIP=(node,lcliconname)

    OR

Dynamic/ETO
Logon descriptor:

L descname UNITYPE=ISCTCPIP
     LCLICON=lcliconname
```

```
HWSCFGxx

ISC()
  IMSPLEX=
   (MEMBER=lcliconname,
     TMEMBER=IMSplexname)
...
```

  - The Operations Manager (OM) provides type-2 command support

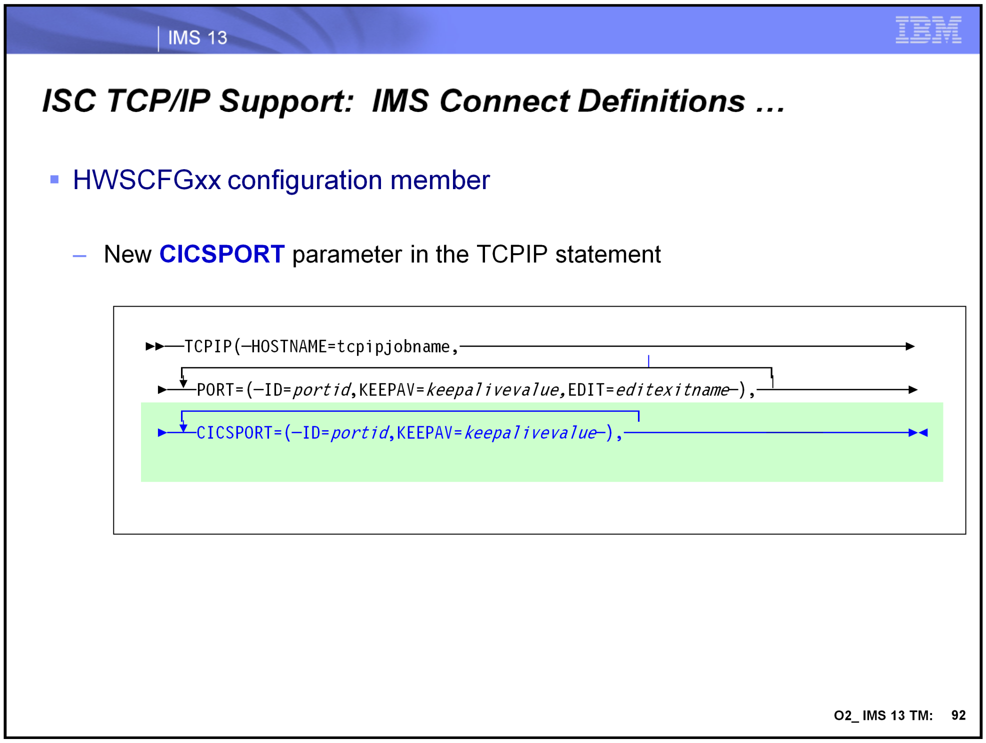
A basic Common Service Layer (CSL) configuration is required to support ISC TCP/IP. This minimum configuration includes the Structure Call Interface (SCI) and SCI address space for communication between IMS components such as IMS to IMS Connect. The Operations Manager (OM) provides type-2 command support for the environment. The new ISC TCP/IP 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 IMS Connect setup definitions for ISC.



ISC TCP/IP Support: IMS Connect Definitions

- IMS Connect’s HWSCFGxx configuration member
  - New CICSPORT parameter in TCPIP statement
  - New ISC statement
  - New RMTCICS statement

TCP/IP support for ISC is provided by IMS Connect and is defined by coding the appropriate configuration statements in the IMS Connect PROCLIB member. These include a new CICSPORT parameter in the existing TCPIP configuration statement, and new ISC as well as RMTCICS statements.



IMS 13

ISC TCP/IP Support: IMS Connect Definitions …

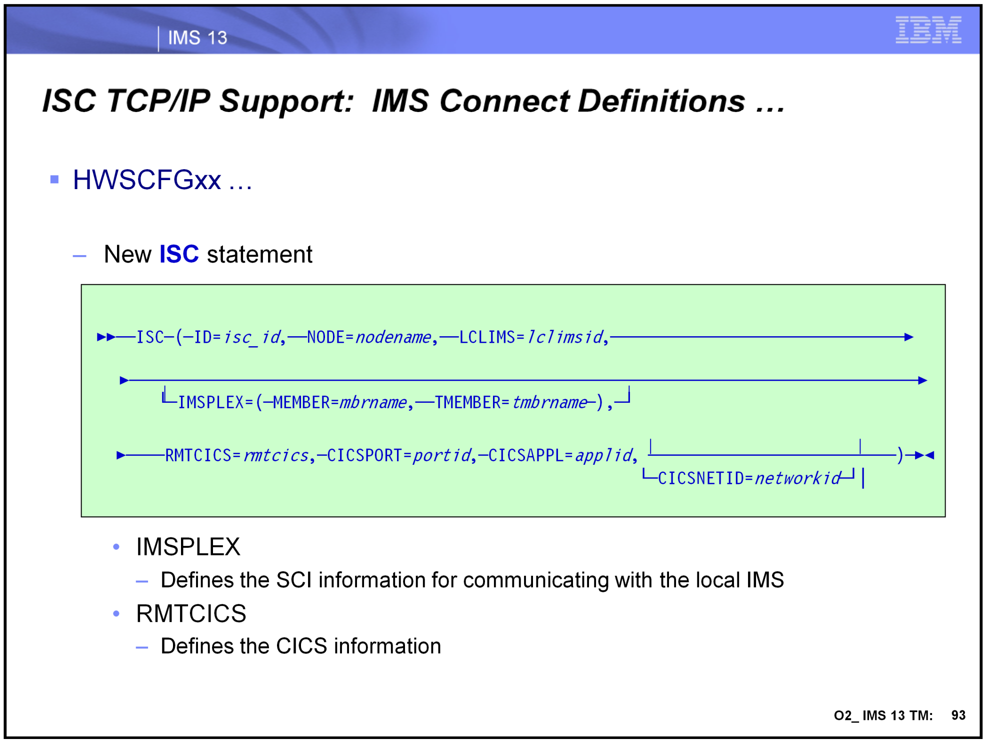
- HWSCFGxx configuration member
  - New **CICSPORT** parameter in the TCPIP statement

```
►►──TCPIP(─HOSTNAME=tcpipjobname,──────────────────────────────────────►

  ┌───────────────────────────────────────────────────────┐
►─▼─PORT=(─ID=portid,KEEPAV=keepalivevalue,EDIT=editexitname─),─┴──────►

  ┌────────────────────────────────────────────┐
►─▼─CICSPORT=(─ID=portid,KEEPAV=keepalivevalue─),─┴────────────────────►◄
```

O2_ IMS 13 TM: 92

The HWSCFGxx TCPIP statement has been enhanced to add the CICSPORT parameter. This new parameter identifies specific CICS port(s) to use for the connection. Remember that you can define up to 50 TCP/IP ports for an instance of IMS connect. The combined total for the CICS port(s) plus all others that may be defined in this IMS Connect instance cannot exceed 50.

- ID= defines the port(s) to use and must be unique for a specific instance of IMS Connect without conflicting with other ports already selected.
- KEEPAV= specifies the number of seconds for the TCP/IP KeepAlive interval for sockets on this port. TCP/IP accepts a range from 1- 2147460 seconds. If you specify zero, the KeepAlive interval value is bypassed and the setting for the TCP/IP stack is used (this is the default). TCP/IP KeepAlive enables you to detect error situations on inactive sockets.



IMS 13

ISC TCP/IP Support: IMS Connect Definitions …

- HWSCFGxx …
  - New **ISC** statement

```
►►──ISC─(─ID=isc_id,──NODE=nodename,──LCLIMS=lclimsid,──────────────────►

 ►──┬──────────────────────────────────────────────────┬──►
    └─IMSPLEX=(─MEMBER=mbrname,──TMEMBER=tmbrname─),─┘

 ►───RMTCICS=rmtcics,─CICSPORT=portid,─CICSAPPL=applid,─┬─────────────────────┬──)─►◄
                                                        └─CICSNETID=networkid─┘
```

- IMSPLEX
  - Defines the SCI information for communicating with the local IMS
- RMTCICS
  - Defines the CICS information

O2_ IMS 13 TM: 93

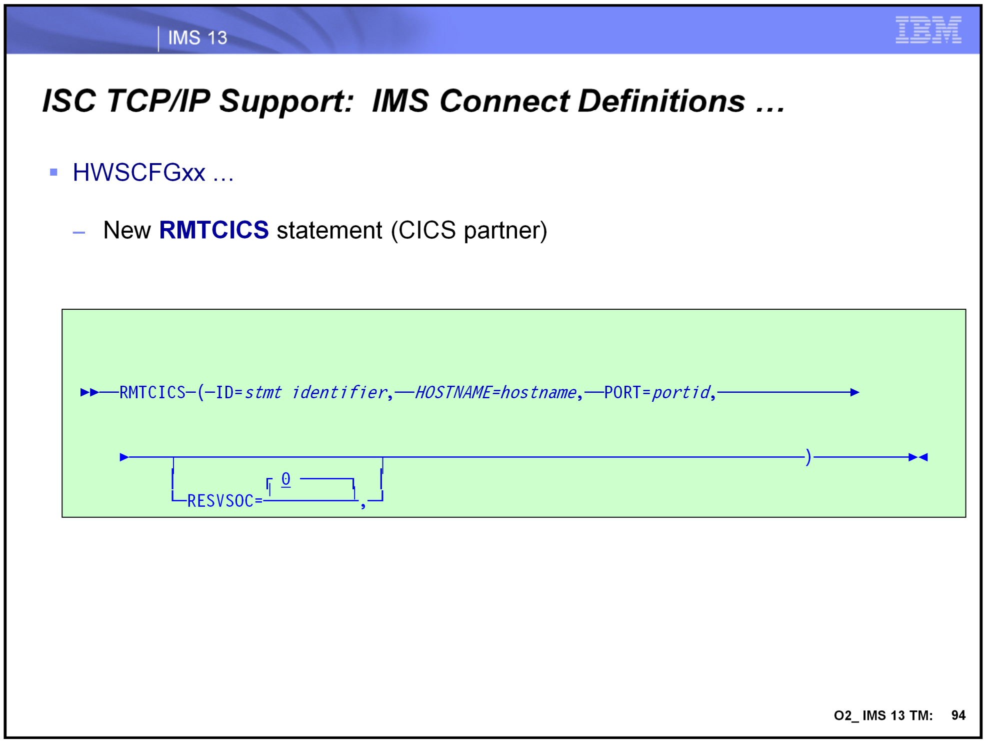
The ISC statement defines the send path for an ISC link from the local IMS system to a remote CICS system. This ISC statement in conjunction with a RMTCICS statement establishes a one way send socket path. On the other side of the link, CICS definitions establish the corresponding receive socket for the connection. Additionally the CICS's IPCONN definition establishes a send socket on the CICS side which is matched with a receive socket in IMS. Each ISC parallel session, therefore, establishes two socket connections in IMS Connect. Note that the value specified in the LCLIMS parameter and the value specified in the NODE parameter must be unique for each ISC statement.

The RMTCICS value specifies the remote CICS connection and is the same as the ID= parameter of one of the RMTCICS statements (described in the next visual). ISC links that use separately-defined TCP/IP connections to the same CICS subsystem must specify different IDs on the RMTCICS parameter; however, the different RMTCICS statements that each ISC statement references must specify the same host name or IP address and port number.

– CICSAPPL= equates to the specified APPLID value of the DFHSIT macro definition in the remote CICS. This parameter must be specified if RMTCICS parameter is also specified, otherwise, IMS Connect will issue an error message and an abend on startup.

– CICSNETID= is either the VTAM NETID or the UOWNETQL= parameter of the DFHSIT macro definition in the remote CICS subsystem.

– CICSPORT= specifies a CICSPORT *portid* as defined in the TCPIP statement. Multiple ISC statements can specify the same CICSPORT *portid* if and only if they specify the same LCLIMS. However, if these ISC statements specify the same CICSAPPL value, they must have unique CICSPORT values. Otherwise, IMS Connect will issue an error message and an abend on startup.



IMS 13

ISC TCP/IP Support: IMS Connect Definitions …

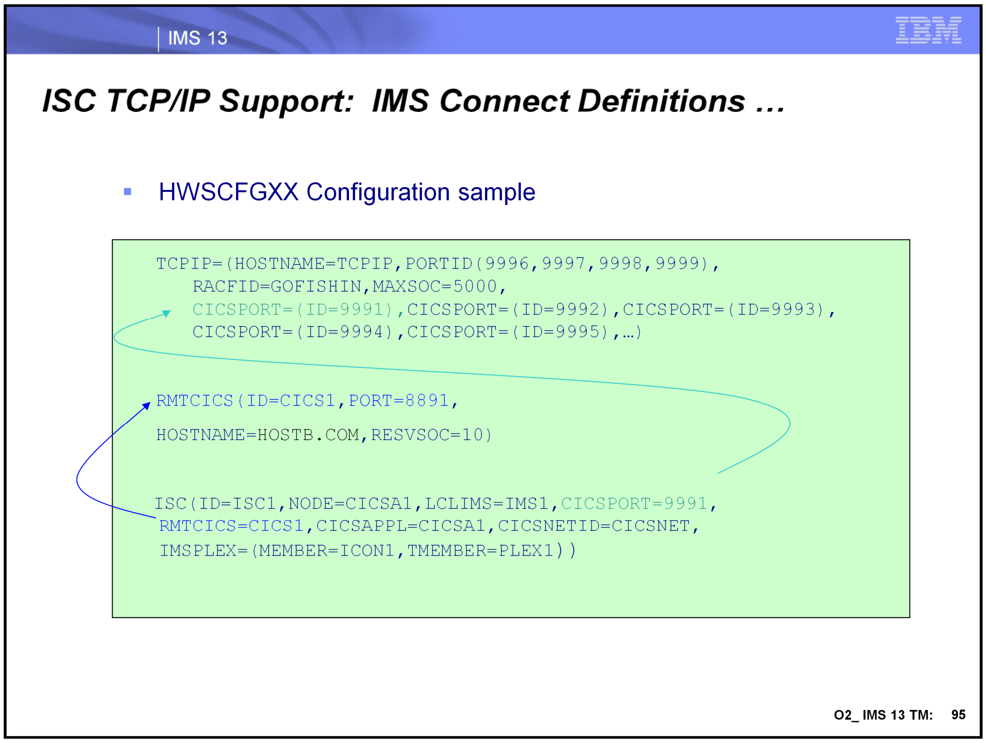
- HWSCFGxx …
  - New **RMTCICS** statement (CICS partner)

```
►►──RMTCICS─(─ID=stmt identifier,──HOSTNAME=hostname,──PORT=portid,──────────►

   ►──┬──────────────────────────┬───────────────────────────────)──────►◄
      │             ┌─0───────┐   │
      └─RESVSOC=────┴─────────┴─,─┘
```

O2_ IMS 13 TM: 94

The RMTCICS() statement defines each remote CICS connection that can communicate with this instance of IMS Connect.

- ID= identifies this statement. It matches one or more RMTCICS specification(s) on the ISC() statement (previous visual).
- HOSTNAME= is the hostname of the remote CICS, e.g., CICSA.ibm.com. This parameter can accept up to 60 alphanumeric characters.
- PORT= is the remote CICS port number and must match a definition in the remote CICS.
- RESVSOC= defines the number of sockets to reserve for use by this connection. This value should be equivalent to the number of parallel sessions. IMS Connect reserves 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 ISC above this RESVSOC count, IMS Connect will get them, if possible, will count the additional sockets towards the max sockets count. The sum of all the RESVSOC= values specified in all the RMTCICS 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.



IMS 13

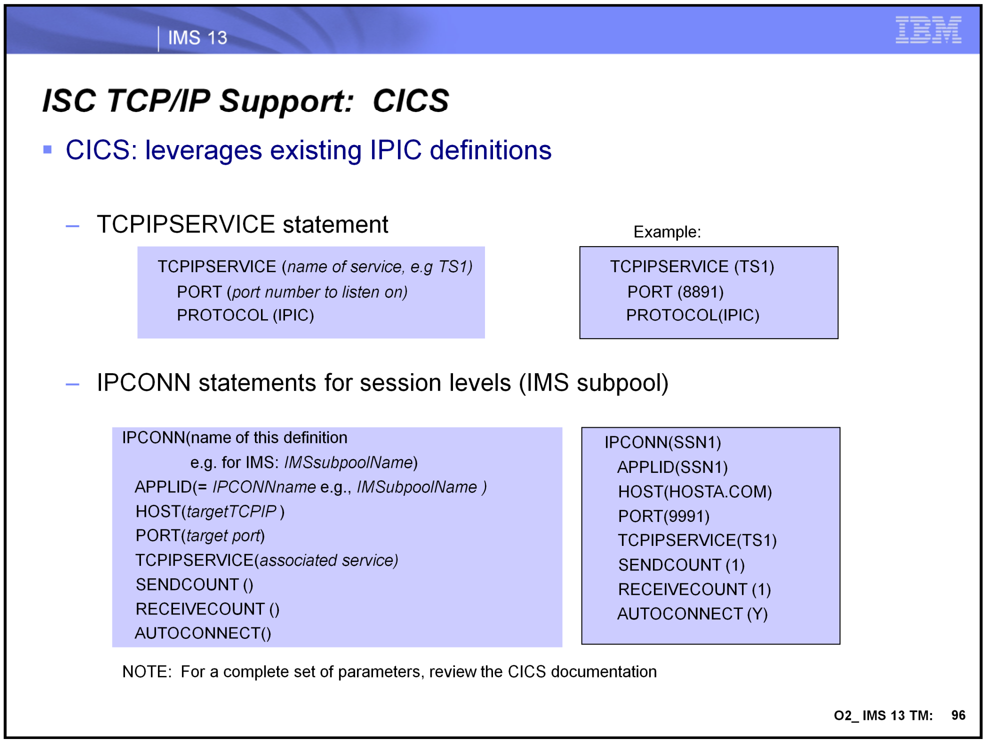
ISC TCP/IP Support: IMS Connect Definitions …

- HWSCFGXX Configuration sample

```
TCPIP=(HOSTNAME=TCPIP,PORTID(9996,9997,9998,9999),
   RACFID=GOFISHIN,MAXSOC=5000,
   CICSPORT=(ID=9991),CICSPORT=(ID=9992),CICSPORT=(ID=9993),
   CICSPORT=(ID=9994),CICSPORT=(ID=9995),…)

RMTCICS(ID=CICS1,PORT=8891,

HOSTNAME=HOSTB.COM,RESVSOC=10)


ISC(ID=ISC1,NODE=CICSA1,LCLIMS=IMS1,CICSPORT=9991,
 RMTCICS=CICS1,CICSAPPL=CICSA1,CICSNETID=CICSNET,
 IMSPLEX=(MEMBER=ICON1,TMEMBER=PLEX1))
```

In this configuration example, note the RMTCICS definition. This shows that the local IMS Connect instance has been defined to use port 8891 for outbound communications with CICS. This is for establishing the SEND sockets from IMS Connect.

The ISC statement shows that IMS Connect has specified that CICSPORT 9991 is for inbound communication from CICS. This is for establishing the RECV sockets for messages from CICS. Also note that the CICSPORT number matches one that has been defined in the CICSPORT parameter of the TCP/IP statement.



IMS 13

ISC TCP/IP Support: CICS

- CICS: leverages existing IPIC definitions
  - TCPIPSERVICE statement

```
TCPIPSERVICE (name of service, e.g TS1)
   PORT (port number to listen on)
   PROTOCOL (IPIC)
```

Example:

```
TCPIPSERVICE (TS1)
   PORT (8891)
   PROTOCOL(IPIC)
```

  - IPCONN statements for session levels (IMS subpool)

```
IPCONN(name of this definition
           e.g. for IMS: IMSsubpoolName)
  APPLID(= IPCONNname e.g., IMSubpoolName )
  HOST(targetTCPIP )
  PORT(target port)
  TCPIPSERVICE(associated service)
  SENDCOUNT ()
  RECEIVECOUNT ()
  AUTOCONNECT()
```

```
IPCONN(SSN1)
  APPLID(SSN1)
  HOST(HOSTA.COM)
  PORT(9991)
  TCPIPSERVICE(TS1)
  SENDCOUNT (1)
  RECEIVECOUNT (1)
  AUTOCONNECT (Y)
```

NOTE: For a complete set of parameters, review the CICS documentation

O2_ IMS 13 TM: 96

IPIC or IP interconnectivity requires resources to be defined using the IPCONN and TCPIPSERVICE statements. A TCPIPSERVICE is the CICS resource that listens for TCP/IP communications from a partner region.

TCPIPSERVICE () is the name of the local TCPIPSERVICE resource where inbound requests are received:

•PORT is the port number that CICS will listen on for this service

•PROTOCOL for this type of connection is always IPIC

IPCONN () name of the definition – this matches the IMS subpool name:

•APPLID for the IMS environment this must match the IPCONN name, e.g., IMS subpool name. Since the APPLID value is equal to the IMS subpool name then the connection can be established either by IMS or CICS.

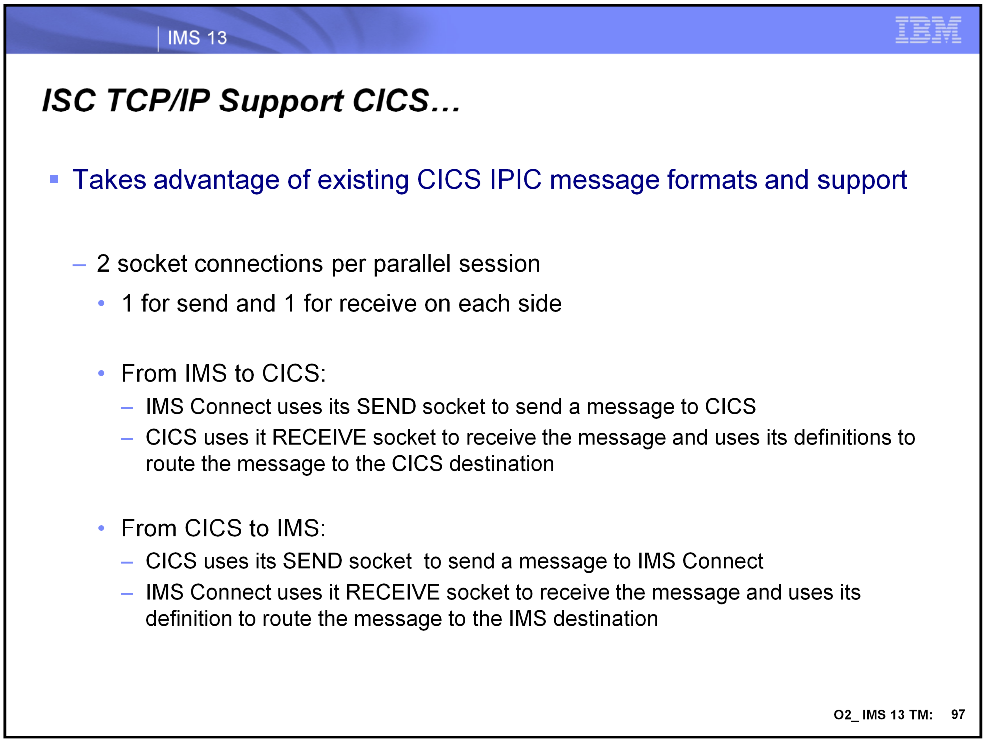
•AUTOCONNECT specifies whether Y | N sessions are to be established when the IPCONN definition is installed .

•HOST is the name of the TCP/IP Stack on the target system MVS image

•PORT is the port named on the TCPIPSERVICE resource of the remote system

•RECEIVECOUNT(1-999) specifies the number of sessions that receive incoming requests. The actual number of receive sessions that are used depends also on the number of send sessions defined in the remote system, e.g., IMS. When the connection is established, these values are exchanged and the lower value is used.

•SENDCOUNT(0-999) specifies the number of sessions that send outgoing requests. The actual number of send sessions that are used depends also on the number of receive sessions defined in the remote system, e.g. IMS. When the connection is established, these values are exchanged and the lower value is used. If 0 is specified, then this IPCONN can only process incoming work, cannot send requests to the connected system, and cannot be acquired. A SENDCOUNT value greater than zero requires PORT to have a numeric value.



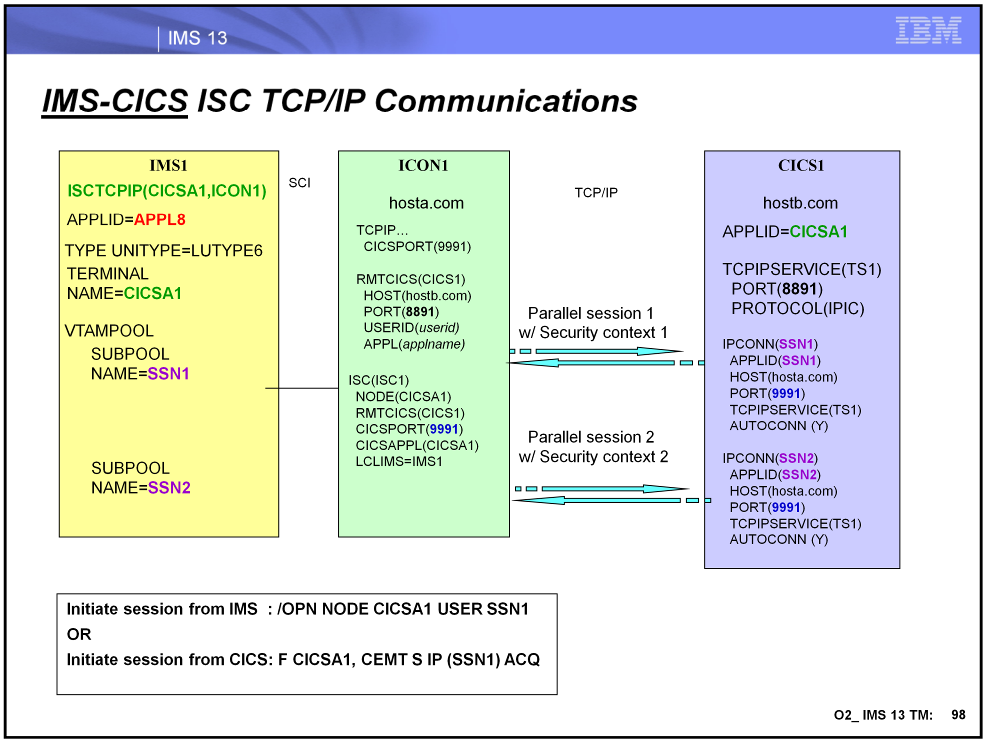
IMS 13

ISC TCP/IP Support CICS…

- Takes advantage of existing CICS IPIC message formats and support
  - 2 socket connections per parallel session
    - 1 for send and 1 for receive on each side
    - From IMS to CICS:
      - IMS Connect uses its SEND socket to send a message to CICS
      - CICS uses it RECEIVE socket to receive the message and uses its definitions to route the message to the CICS destination
    - From CICS to IMS:
      - CICS uses its SEND socket to send a message to IMS Connect
      - IMS Connect uses it RECEIVE socket to receive the message and uses its definition to route the message to the IMS destination

O2_ IMS 13 TM: 97

The CICS support leverages the IPIC connectivity capability that exists in CICS.



IMS 13

IMS-CICS ISC TCP/IP Communications

IMS1

ISCTCPIP(CICSA1,ICON1)

APPLID=APPL8

TYPE UNITYPE=LUTYPE6

TERMINAL
NAME=CICSA1

VTAMPOOL
SUBPOOL
NAME=SSN1

SUBPOOL
NAME=SSN2

SCI

ICON1

hosta.com

```
TCPIP...
  CICSPORT(9991)

RMTCICS(CICS1)
  HOST(hostb.com)
  PORT(8891)
  USERID(userid)
  APPL(applname)

ISC(ISC1)
 NODE(CICSA1)
 RMTCICS(CICS1)
 CICSPORT(9991)
 CICSAPPL(CICSA1)
 LCLIMS=IMS1
```

TCP/IP

Parallel session 1
w/ Security context 1

Parallel session 2
w/ Security context 2

CICS1

hostb.com

APPLID=CICSA1

```
TCPIPSERVICE(TS1)
 PORT(8891)
 PROTOCOL(IPIC)

IPCONN(SSN1)
  APPLID(SSN1)
  HOST(hosta.com)
  PORT(9991)
  TCPIPSERVICE(TS1)
  AUTOCONN (Y)

IPCONN(SSN2)
  APPLID(SSN2)
  HOST(hosta.com)
  PORT(9991)
  TCPIPSERVICE(TS1)
  AUTOCONN (Y)
```

Initiate session from IMS : /OPN NODE CICSA1 USER SSN1

OR

Initiate session from CICS: F CICSA1, CEMT S IP (SSN1) ACQ

O2_ IMS 13 TM: 98

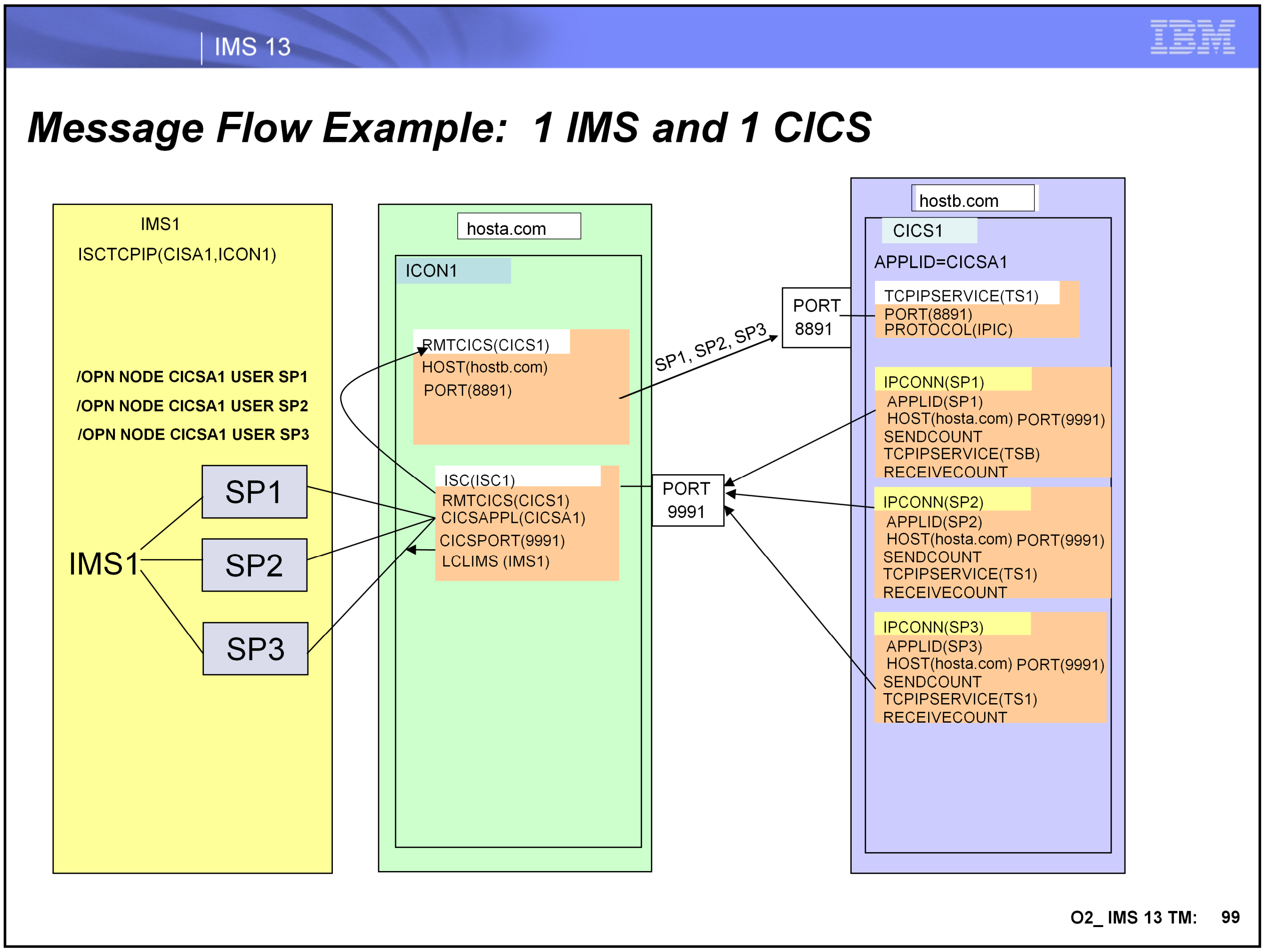
Now that the definitions have been described, this same visual that was shown earlier should make more sense. On the IMS side, this configuration example shows two statically defined parallel sessions and subpools SSN1 and SSN2. Corresponding definitions are provided in CICS with IPCONN statements matching the subpools.

IMS initiates the session using the /OPNDST command:

•Example: /OPN NODE CICSA1 USER SSN1

Or, alternatively, CICS can initiate the session using the CEMT transaction:

•Example: F CICSA1,CEMT S IP(SSN1) ACQ



Message Flow Example: 1 IMS and 1 CICS

IMS1

ISCTCPIP(CISA1,ICON1)

/OPN NODE CICSA1 USER SP1

/OPN NODE CICSA1 USER SP2

/OPN NODE CICSA1 USER SP3

IMS1

SP1

SP2

SP3

hosta.com

ICON1

RMTCICS(CICS1)
HOST(hostb.com)
PORT(8891)

ISC(ISC1)
RMTCICS(CICS1)
CICSAPPL(CICSA1)
CICSPORT(9991)
LCLIMS (IMS1)

SP1, SP2, SP3

PORT
8891

PORT
9991

hostb.com

CICS1

APPLID=CICSA1

TCPIPSERVICE(TS1)
PORT(8891)
PROTOCOL(IPIC)

IPCONN(SP1)
APPLID(SP1)
HOST(hosta.com) PORT(9991)
SENDCOUNT
TCPIPSERVICE(TSB)
RECEIVECOUNT

IPCONN(SP2)
APPLID(SP2)
HOST(hosta.com) PORT(9991)
SENDCOUNT
TCPIPSERVICE(TS1)
RECEIVECOUNT

IPCONN(SP3)
APPLID(SP3)
HOST(hosta.com) PORT(9991)
SENDCOUNT
TCPIPSERVICE(TS1)
RECEIVECOUNT

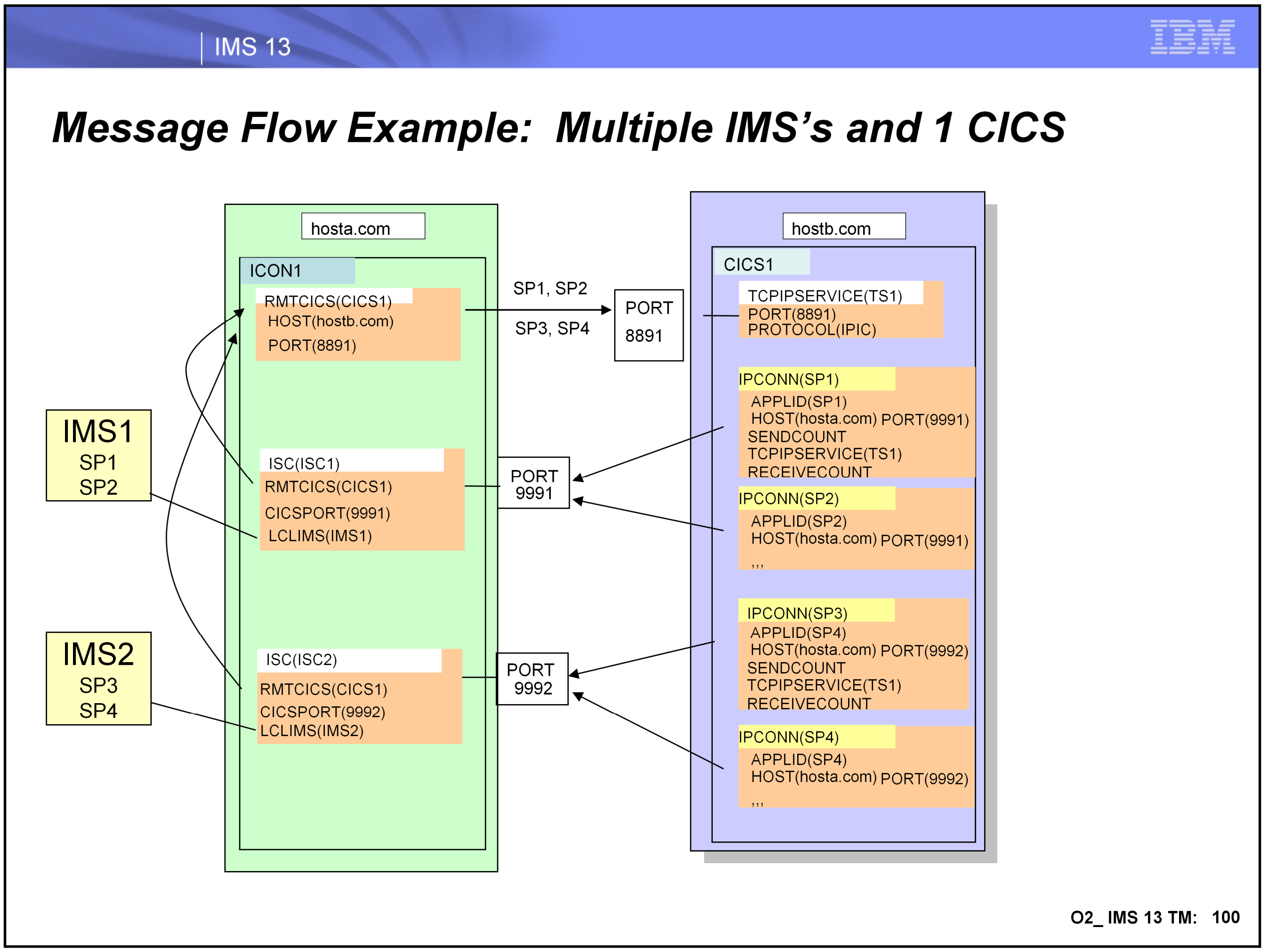
When an /OPN NODE CICSA1 USER SP1 command is entered from IMS1, the session connection request is sent to IMS Connect ICON1. ICON1 uses the ISC1 specification to determine that it will use CICS1 RMTCICS specification to communicate with CICS. ICON1 then uses the hostname of hostb.com and port 8891 specified in the RMTCICS statement to send the session connection request to CICS1.

At the other end of the link, CICS1 uses the TCPIPSERVICE specification listening in on Port 8891 to receive the session connection request. CICS1 selects the IPCONN statement whose name is SSN1 to send the reply back to IMS, using the hostname defined in the statement which in this case is hosta.com and port 9991.

ICON1 receives the reply message from CICS1 coming into port 9991. ICON1 then uses the ISC1 definition to forward the reply message back to the LCLIMS which is IMS1.

Note the following:

1. There are no duplicate IPCONN statements with the same NETID and APPLID.
2. The IPCONN name that equates to IMS subpools, e.g. SP1, are defined in IMS.



Message Flow Example: Multiple IMS's and 1 CICS

hosta.com

ICON1

RMTCICS(CICS1)
HOST(hostb.com)
PORT(8891)

ISC(ISC1)
RMTCICS(CICS1)
CICSPORT(9991)
LCLIMS(IMS1)

ISC(ISC2)
RMTCICS(CICS1)
CICSPORT(9992)
LCLIMS(IMS2)

IMS1
SP1
SP2

IMS2
SP3
SP4

SP1, SP2

SP3, SP4

PORT
8891

PORT
9991

PORT
9992

hostb.com

CICS1

TCPIPSERVICE(TS1)
PORT(8891)
PROTOCOL(IPIC)

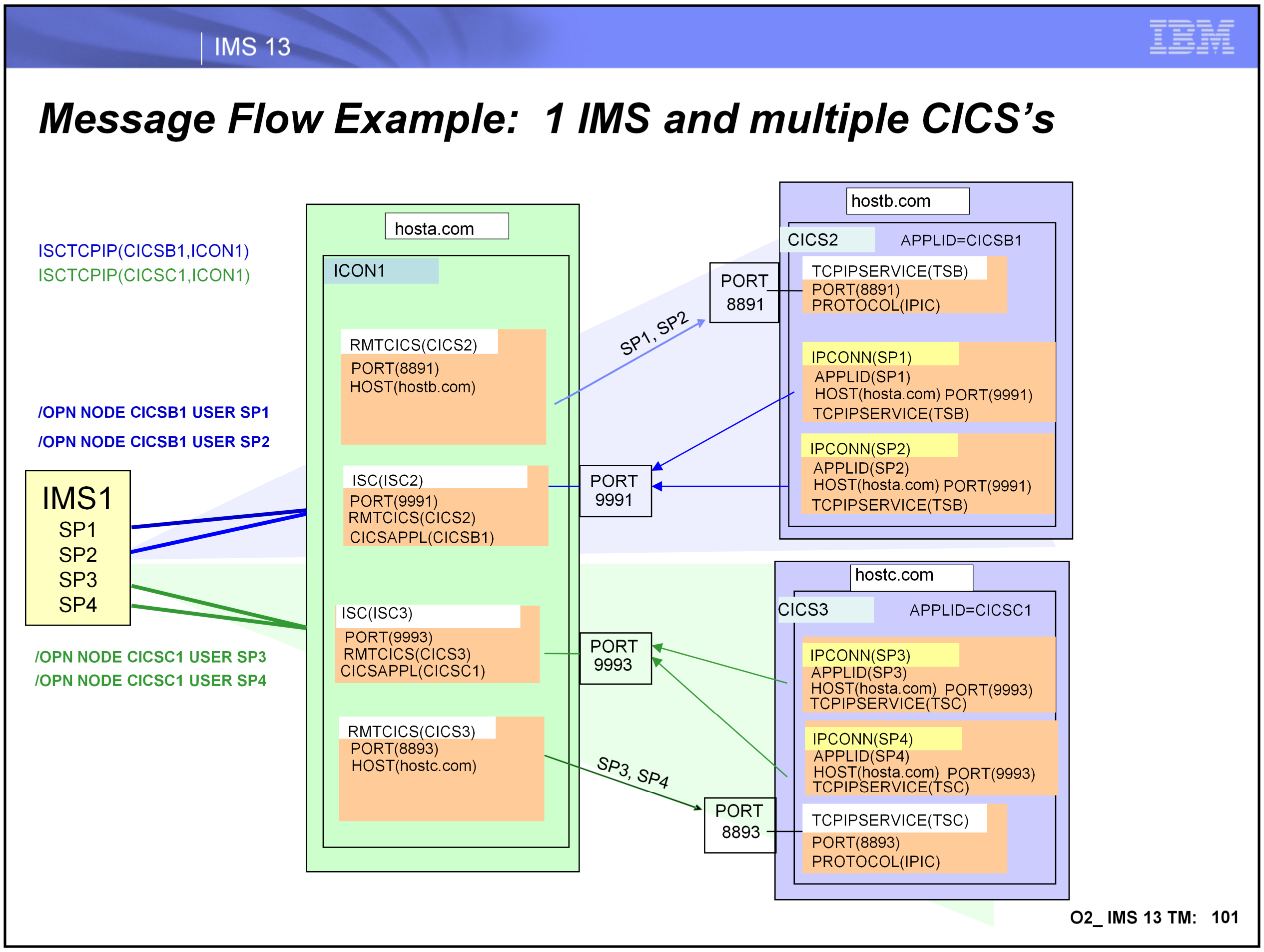
IPCONN(SP1)
APPLID(SP1)
HOST(hosta.com) PORT(9991)
SENDCOUNT
TCPIPSERVICE(TS1)
RECEIVECOUNT

IPCONN(SP2)
APPLID(SP2)
HOST(hosta.com) PORT(9991)
,,,

IPCONN(SP3)
APPLID(SP4)
HOST(hosta.com) PORT(9992)
SENDCOUNT
TCPIPSERVICE(TS1)
RECEIVECOUNT

IPCONN(SP4)
APPLID(SP4)
HOST(hosta.com) PORT(9992)
,,,

Multiple IMS systems can use one IMS Connect to route traffic to a single CICS partner.



Message Flow Example: 1 IMS and multiple CICS's

ISCTCPIP(CICSB1,ICON1)
ISCTCPIP(CICSC1,ICON1)

/OPN NODE CICSB1 USER SP1
/OPN NODE CICSB1 USER SP2

IMS1
SP1
SP2
SP3
SP4

/OPN NODE CICSC1 USER SP3
/OPN NODE CICSC1 USER SP4

hosta.com

ICON1

RMTCICS(CICS2)
PORT(8891)
HOST(hostb.com)

ISC(ISC2)
PORT(9991)
RMTCICS(CICS2)
CICSAPPL(CICSB1)

ISC(ISC3)
PORT(9993)
RMTCICS(CICS3)
CICSAPPL(CICSC1)

RMTCICS(CICS3)
PORT(8893)
HOST(hostc.com)

hostb.com

CICS2 APPLID=CICSB1

TCPIPSERVICE(TSB)
PORT(8891)
PROTOCOL(IPIC)

IPCONN(SP1)
APPLID(SP1)
HOST(hosta.com) PORT(9991)
TCPIPSERVICE(TSB)

IPCONN(SP2)
APPLID(SP2)
HOST(hosta.com) PORT(9991)
TCPIPSERVICE(TSB)

hostc.com

CICS3 APPLID=CICSC1

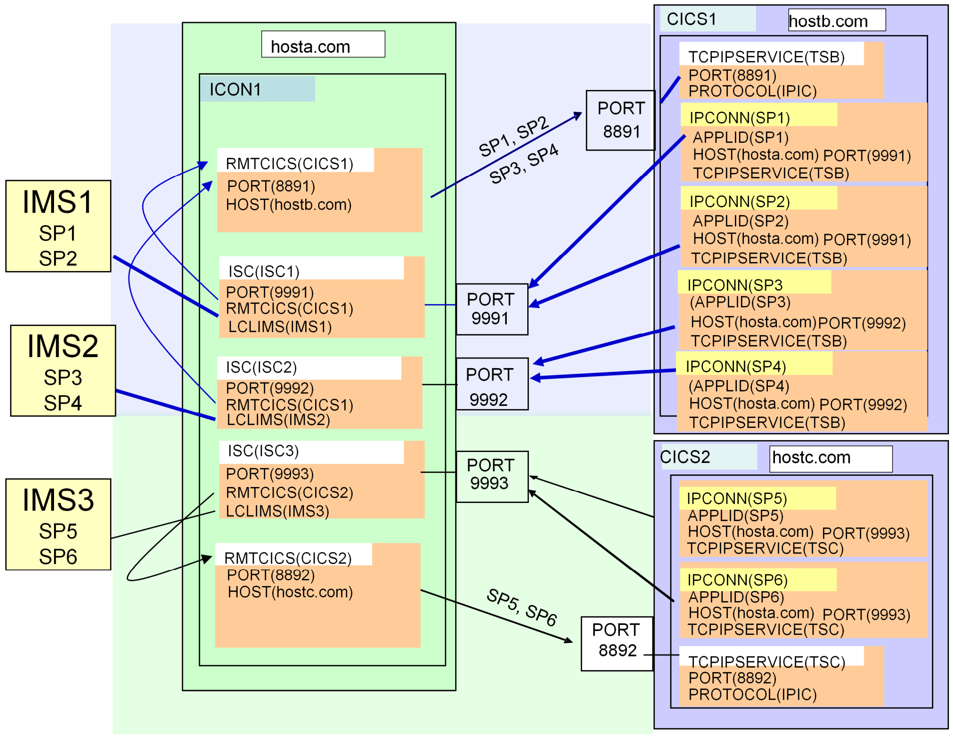
IPCONN(SP3)
APPLID(SP3)
HOST(hosta.com) PORT(9993)
TCPIPSERVICE(TSC)

IPCONN(SP4)
APPLID(SP4)
HOST(hosta.com) PORT(9993)
TCPIPSERVICE(TSC)

TCPIPSERVICE(TSC)
PORT(8893)
PROTOCOL(IPIC)

One IMS system can communicate with multiple CICS's using one IMS Connect.

Message Flow Example: Multiple IMS and Multiple CICS



IMS1
SP1
SP2

IMS2
SP3
SP4

IMS3
SP5
SP6

hosta.com

ICON1

RMTCICS(CICS1)
PORT(8891)
HOST(hostb.com)

ISC(ISC1)
PORT(9991)
RMTCICS(CICS1)
LCLIMS(IMS1)

ISC(ISC2)
PORT(9992)
RMTCICS(CICS1)
LCLIMS(IMS2)

ISC(ISC3)
PORT(9993)
RMTCICS(CICS2)
LCLIMS(IMS3)

RMTCICS(CICS2)
PORT(8892)
HOST(hostc.com)

SP1, SP2
SP3, SP4

PORT 8891

PORT 9991

PORT 9992

PORT 9993

SP5, SP6

PORT 8892

CICS1
hostb.com

TCPIPSERVICE(TSB)
PORT(8891)
PROTOCOL(IPIC)

IPCONN(SP1)
APPLID(SP1)
HOST(hosta.com) PORT(9991)
TCPIPSERVICE(TSB)

IPCONN(SP2)
APPLID(SP2)
HOST(hosta.com) PORT(9991)
TCPIPSERVICE(TSB)

IPCONN(SP3
(APPLID(SP3)
HOST(hosta.com)PORT(9992)
TCPIPSERVICE(TSB)

IPCONN(SP4)
(APPLID(SP4)
HOST(hosta.com) PORT(9992)
TCPIPSERVICE(TSB)

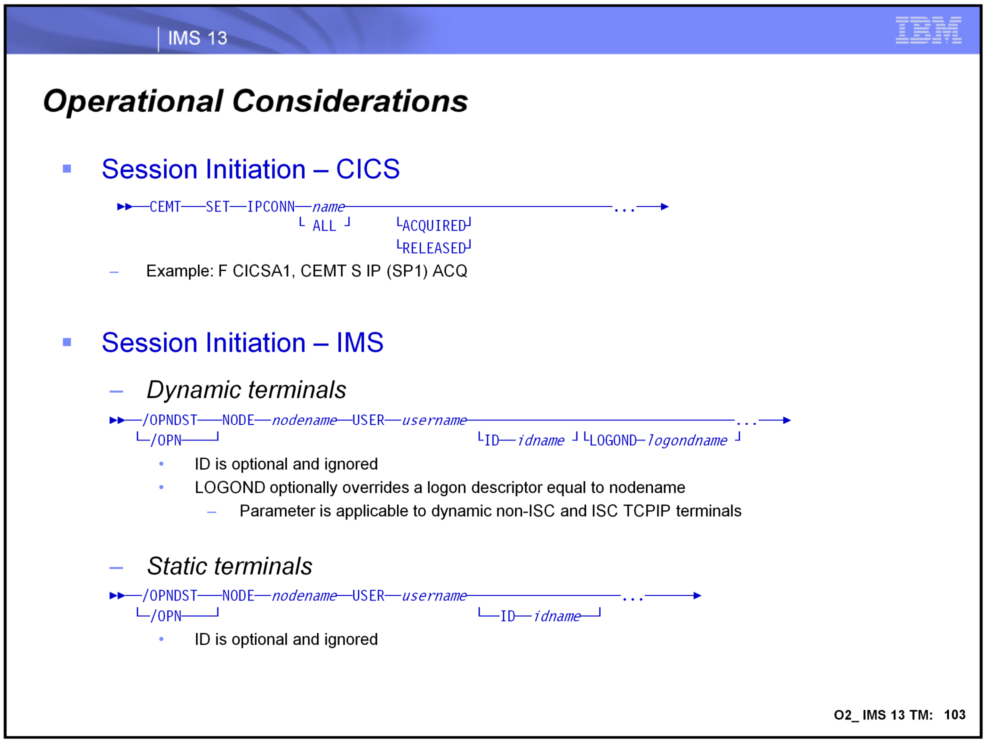
CICS2
hostc.com

IPCONN(SP5)
APPLID(SP5)
HOST(hosta.com) PORT(9993)
TCPIPSERVICE(TSC)

IPCONN(SP6)
APPLID(SP6)
HOST(hosta.com) PORT(9993)
TCPIPSERVICE(TSC)

TCPIPSERVICE(TSC)
PORT(8892)
PROTOCOL(IPIC)

Configurations can even support complex scenarios.



IMS 13

Operational Considerations

- Session Initiation – CICS

```
►►─CEMT──SET─IPCONN─name──────────────────────────...──►
                   └ ALL ┘      └ACQUIRED┘
                                └RELEASED┘
```

  – Example: F CICSA1, CEMT S IP (SP1) ACQ

- Session Initiation – IMS

  – *Dynamic terminals*

```
►►─/OPNDST──NODE─nodename─USER─username─────────────────────────────...──►
  └─/OPN──┘                             └ID─idname ┘└LOGOND─logondname ┘
```

  - ID is optional and ignored
  - LOGOND optionally overrides a logon descriptor equal to nodename
    – Parameter is applicable to dynamic non-ISC and ISC TCPIP terminals

  – *Static terminals*

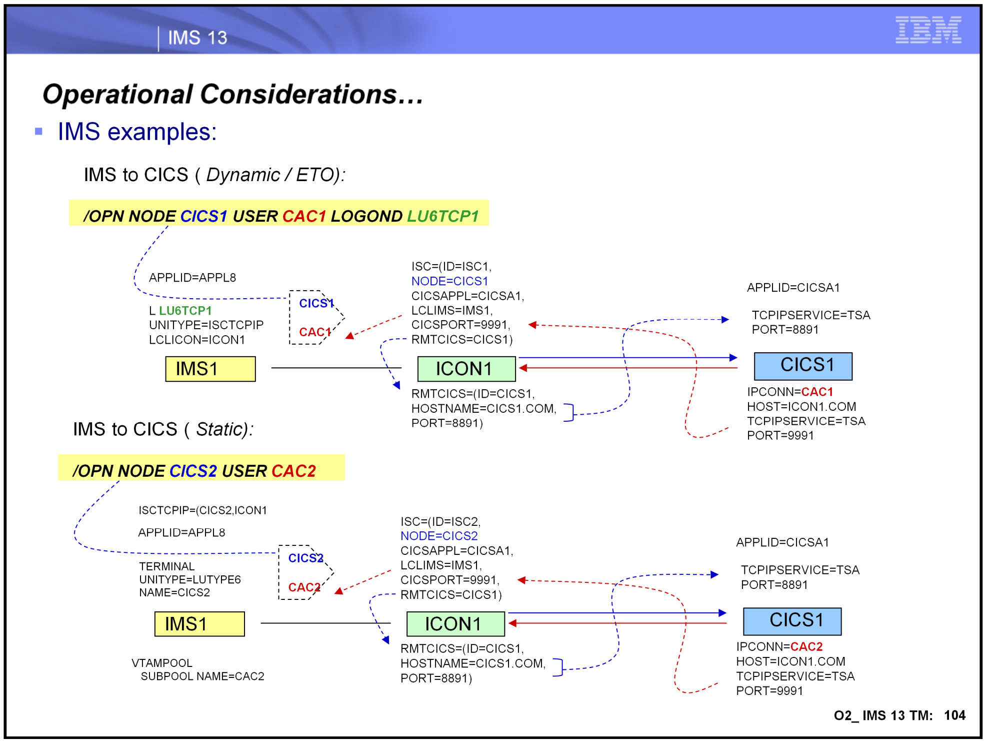
```
►►─/OPNDST──NODE─nodename─USER─username────────────────...────►
  └─/OPN──┘                             └─ID─idname─┘
```

  - ID is optional and ignored

O2_ IMS 13 TM: 103

The existing /OPNDST command in IMS can be used to initiate a session from the IMS side. If specified, ID *idname* which is optional is ignored (originally for the VTAM half-session qualifier). The LOGOND *logndname* parameter for dynamic terminals is also optional but if specified must reference an ISC TCP/IP logon descriptor in the DFSDSCMx / DFSDSCTx PROCLIB member. The logon descriptor specified as the LOGOND value for ISC must be one that was defined with UNITYPE=ISCTCPIP. Note that the LOGOND parameter which is not available for dynamic ISC VTAM terminals is supported for ISC TCPIP.

From the CICS side, sessions are initiated using the master terminal CEMT transaction. CEMT SET IPCONN (CEMT S IP) followed by one or more connection identifiers or ALL, followed in turn by one or more attribute settings that you want to change. For example, CEMT S IP ALL resets the values for all connections to make them available for use (inservice).



Operational Considerations…

- IMS examples:

IMS to CICS (*Dynamic / ETO):*

/OPN NODE CICS1 USER CAC1 LOGOND LU6TCP1

APPLID=APPL8

L LU6TCP1
UNITYPE=ISCTCPIP
LCLICON=ICON1

CICS1
CAC1

ISC=(ID=ISC1,
NODE=CICS1
CICSAPPL=CICSA1,
LCLIMS=IMS1,
CICSPORT=9991,
RMTCICS=CICS1)

APPLID=CICSA1

TCPIPSERVICE=TSA
PORT=8891

IMS1

ICON1

CICS1

RMTCICS=(ID=CICS1,
HOSTNAME=CICS1.COM,
PORT=8891)

IPCONN=CAC1
HOST=ICON1.COM
TCPIPSERVICE=TSA
PORT=9991

IMS to CICS (*Static):*

/OPN NODE CICS2 USER CAC2

ISCTCPIP=(CICS2,ICON1

APPLID=APPL8

TERMINAL
UNITYPE=LUTYPE6
NAME=CICS2

CICS2
CAC2

ISC=(ID=ISC2,
NODE=CICS2
CICSAPPL=CICSA1,
LCLIMS=IMS1,
CICSPORT=9991,
RMTCICS=CICS1)

APPLID=CICSA1

TCPIPSERVICE=TSA
PORT=8891

IMS1

ICON1

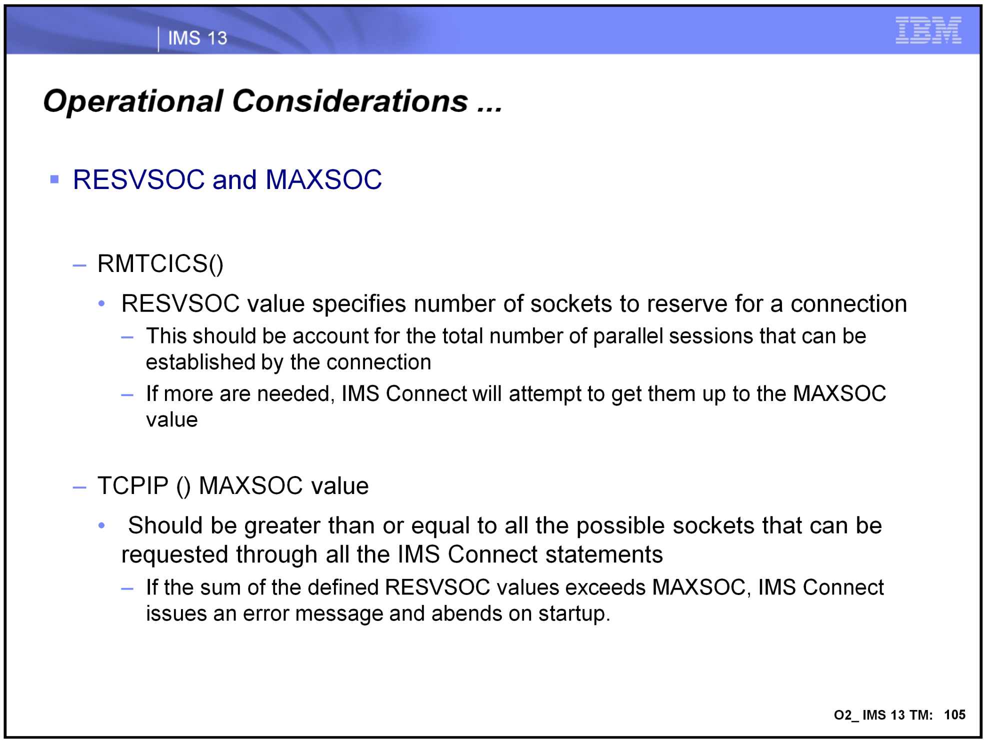
CICS1

VTAMPOOL
SUBPOOL NAME=CAC2

RMTCICS=(ID=CICS1,
HOSTNAME=CICS1.COM,
PORT=8891)

IPCONN=CAC2
HOST=ICON1.COM
TCPIPSERVICE=TSA
PORT=9991

These are examples of what happens when the /OPN command is issued in IMS.



IMS 13

Operational Considerations ...

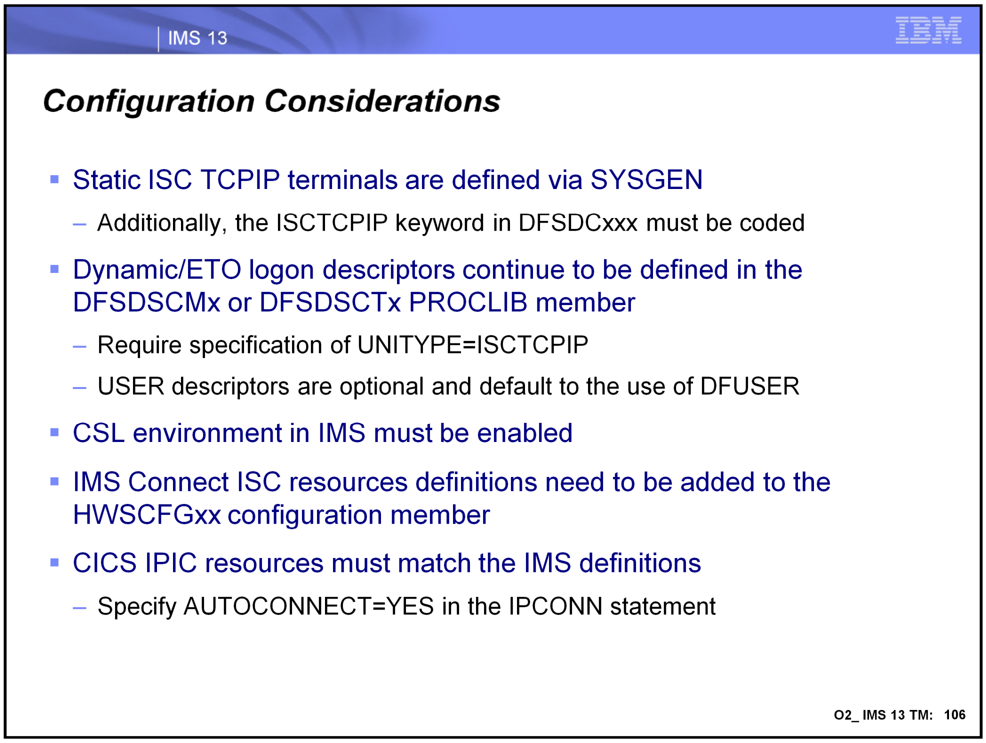
- RESVSOC and MAXSOC
  - RMTCICS()
    - RESVSOC value specifies number of sockets to reserve for a connection
      - This should be account for the total number of parallel sessions that can be established by the connection
      - If more are needed, IMS Connect will attempt to get them up to the MAXSOC value
  - TCPIP () MAXSOC value
    - Should be greater than or equal to all the possible sockets that can be requested through all the IMS Connect statements
      - If the sum of the defined RESVSOC values exceeds MAXSOC, IMS Connect issues an error message and abends on startup.

O2_ IMS 13 TM: 105

The RESVSOC value which is defined in the RMTCICS() statement is used to specify the number of sockets to reserve for use by a connection. For ISC TCP/IP, this value should be equivalent to the number of parallel sessions that can be established. 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 ISC above the RESVSOC count, IMS Connect will get them if possible and they will count towards the max sockets count.

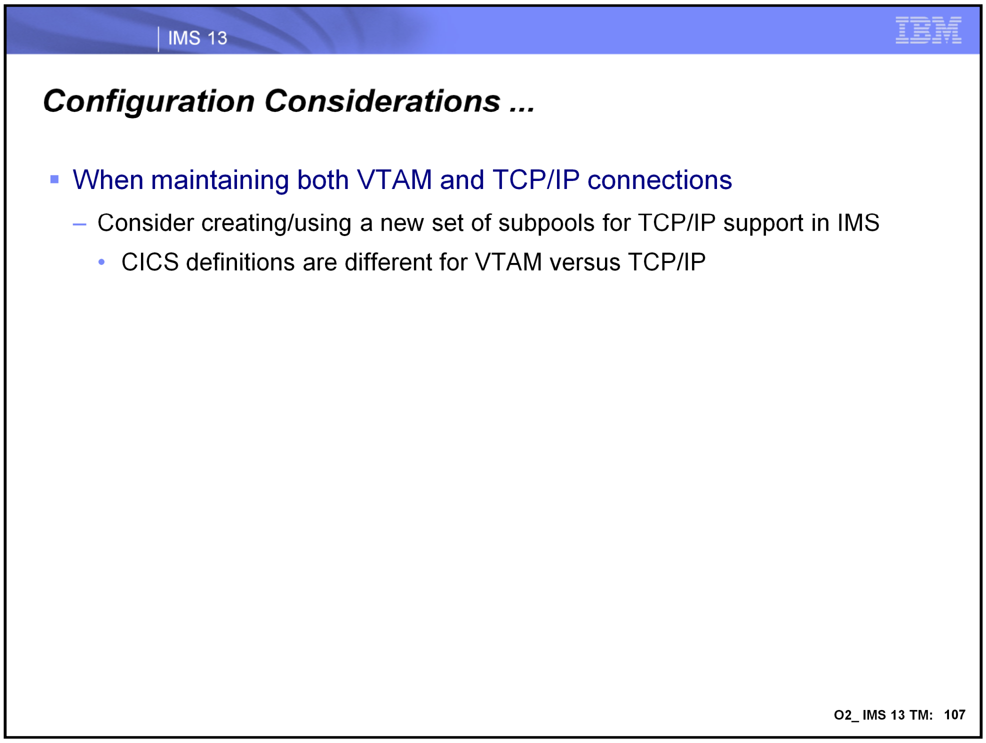
The sum of all the RESVSOC= values in the RMTCICS and RMTIMSCON statements must not exceed the value specified in the TCPIP MAXSOC= parameter. If the sum exceeds the MAXSOC value, then IMS Connect will issue an error message and abend on startup.

This is an optional parameter. The default value is 0.



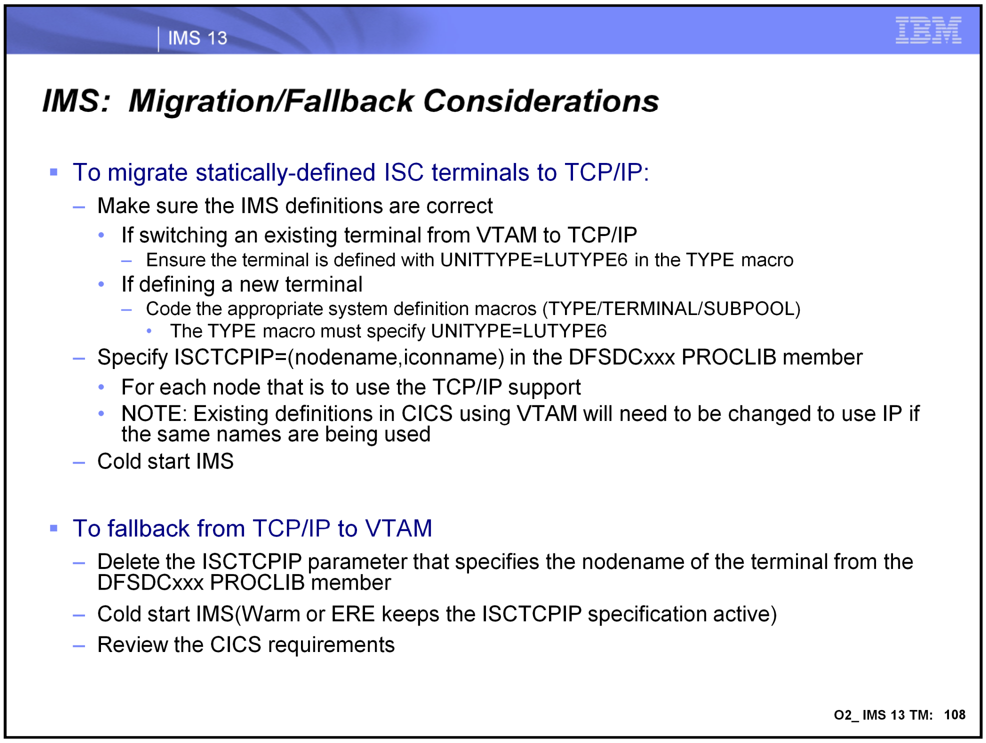
Configuration Considerations

- Static ISC TCPIP terminals are defined via SYSGEN
  - Additionally, the ISCTCPIP keyword in DFSDCxxx must be coded
- Dynamic/ETO logon descriptors continue to be defined in the DFSDSCMx or DFSDSCTx PROCLIB member
  - Require specification of UNITYPE=ISCTCPIP
  - USER descriptors are optional and default to the use of DFUSER
- CSL environment in IMS must be enabled
- IMS Connect ISC resources definitions need to be added to the HWSCFGxx configuration member
- CICS IPIC resources must match the IMS definitions
  - Specify AUTOCONNECT=YES in the IPCONN statement



Configuration Considerations ...

- When maintaining both VTAM and TCP/IP connections
  - Consider creating/using a new set of subpools for TCP/IP support in IMS
    - CICS definitions are different for VTAM versus TCP/IP

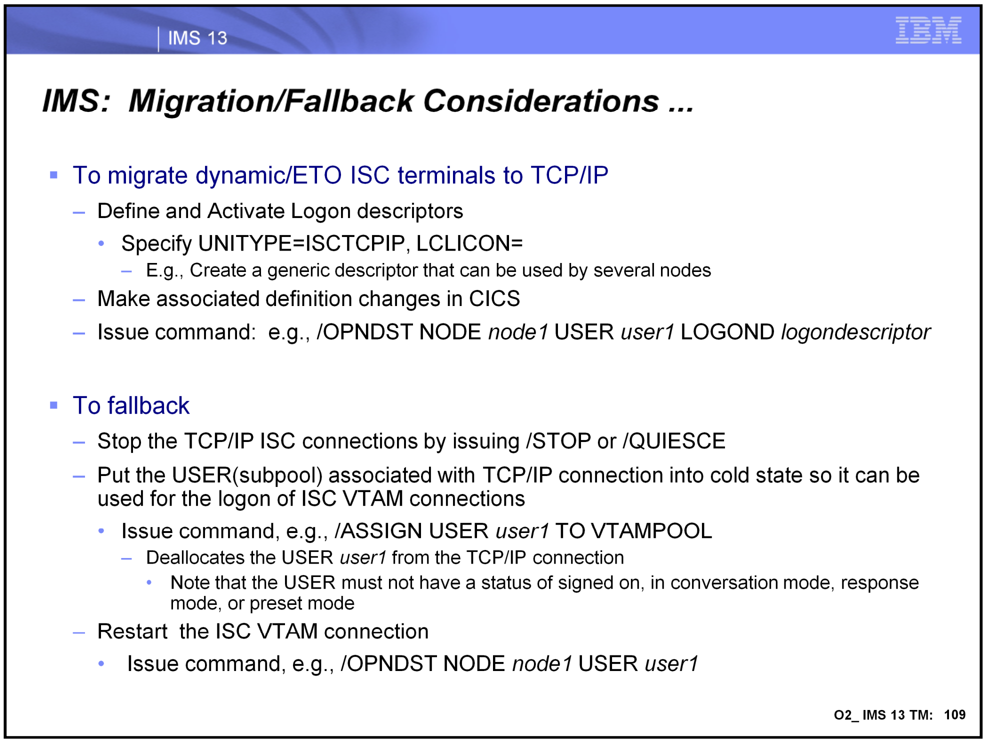


IMS 13

IMS: Migration/Fallback Considerations

- To migrate statically-defined ISC terminals to TCP/IP:
  - Make sure the IMS definitions are correct
    - If switching an existing terminal from VTAM to TCP/IP
      - Ensure the terminal is defined with UNITTYPE=LUTYPE6 in the TYPE macro
    - If defining a new terminal
      - Code the appropriate system definition macros (TYPE/TERMINAL/SUBPOOL)
        - The TYPE macro must specify UNITYPE=LUTYPE6
  - Specify ISCTCPIP=(nodename,iconname) in the DFSDCxxx PROCLIB member
    - For each node that is to use the TCP/IP support
    - NOTE: Existing definitions in CICS using VTAM will need to be changed to use IP if the same names are being used
  - Cold start IMS

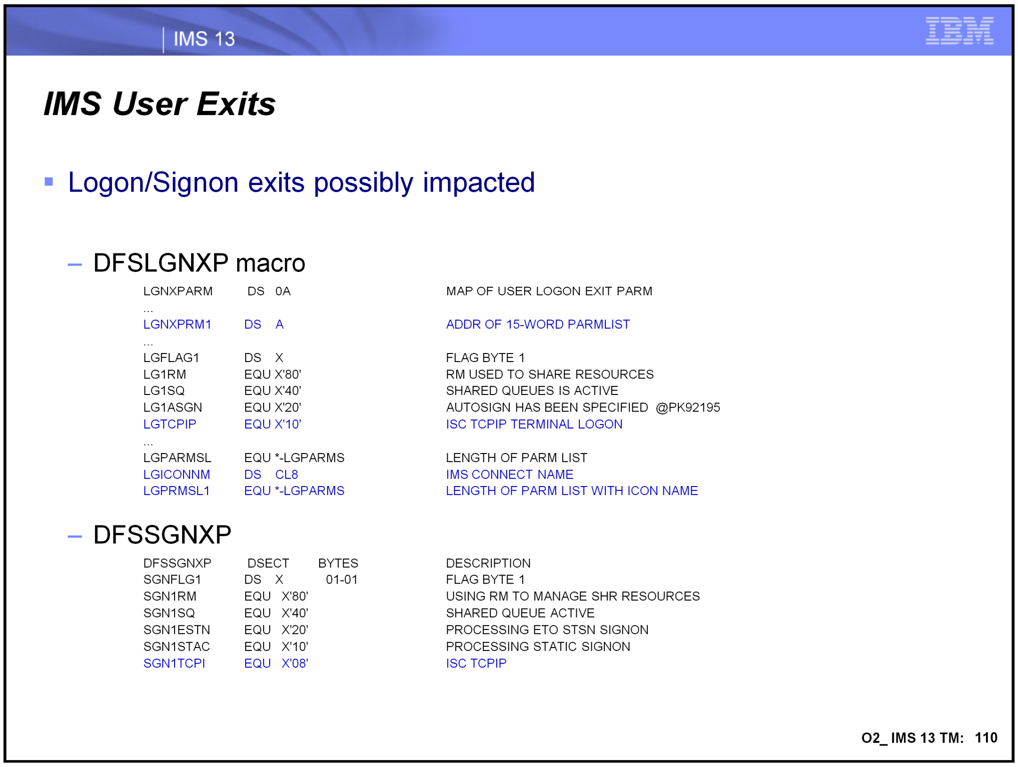
- To fallback from TCP/IP to VTAM
  - Delete the ISCTCPIP parameter that specifies the nodename of the terminal from the DFSDCxxx PROCLIB member
  - Cold start IMS(Warm or ERE keeps the ISCTCPIP specification active)
  - Review the CICS requirements



IMS: Migration/Fallback Considerations ...

- To migrate dynamic/ETO ISC terminals to TCP/IP
  - Define and Activate Logon descriptors
    - Specify UNITYPE=ISCTCPIP, LCLICON=
      - E.g., Create a generic descriptor that can be used by several nodes
  - Make associated definition changes in CICS
  - Issue command: e.g., /OPNDST NODE *node1* USER *user1* LOGOND *logondescriptor*

- To fallback
  - Stop the TCP/IP ISC connections by issuing /STOP or /QUIESCE
  - Put the USER(subpool) associated with TCP/IP connection into cold state so it can be used for the logon of ISC VTAM connections
    - Issue command, e.g., /ASSIGN USER *user1* TO VTAMPOOL
      - Deallocates the USER *user1* from the TCP/IP connection
        - Note that the USER must not have a status of signed on, in conversation mode, response mode, or preset mode
  - Restart the ISC VTAM connection
    - Issue command, e.g., /OPNDST NODE *node1* USER *user1*



IMS User Exits

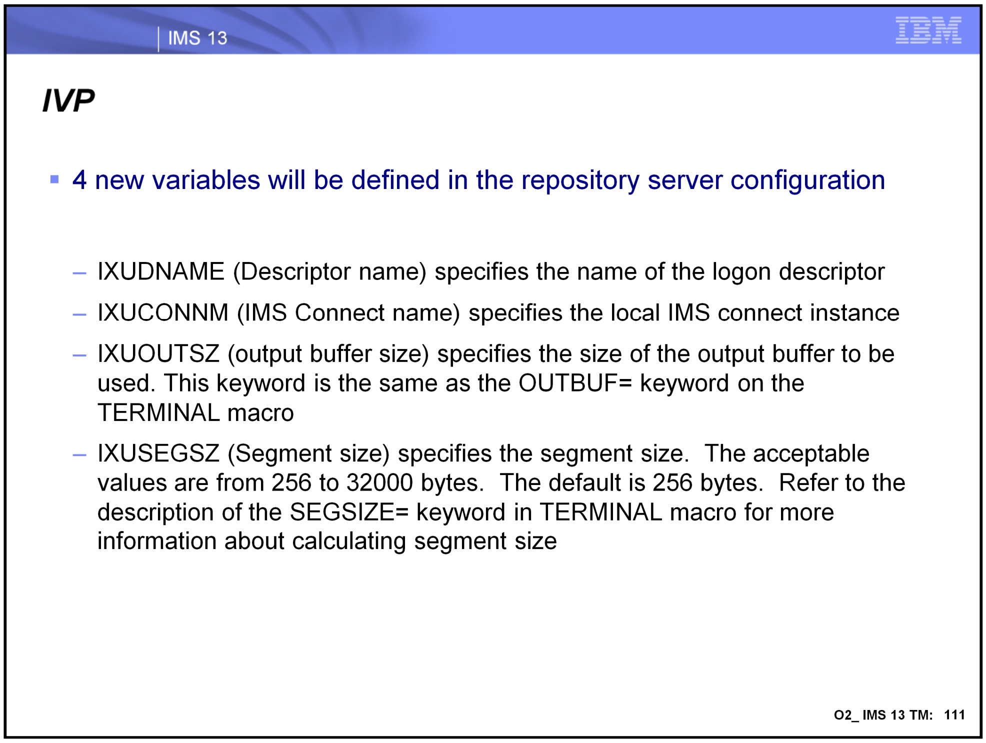
- Logon/Signon exits possibly impacted

– DFSLGNXP macro

```
LGNXPARM     DS   0A                    MAP OF USER LOGON EXIT PARM
...
LGNXPRM1     DS   A                     ADDR OF 15-WORD PARMLIST
...
LGFLAG1      DS   X                     FLAG BYTE 1
LG1RM        EQU X'80'                  RM USED TO SHARE RESOURCES
LG1SQ        EQU X'40'                  SHARED QUEUES IS ACTIVE
LG1ASGN      EQU X'20'                  AUTOSIGN HAS BEEN SPECIFIED  @PK92195
LGTCPIP      EQU X'10'                  ISC TCPIP TERMINAL LOGON
...
LGPARMSL     EQU *-LGPARMS              LENGTH OF PARM LIST
LGICONNM     DS   CL8                   IMS CONNECT NAME
LGPRMSL1     EQU *-LGPARMS              LENGTH OF PARM LIST WITH ICON NAME
```

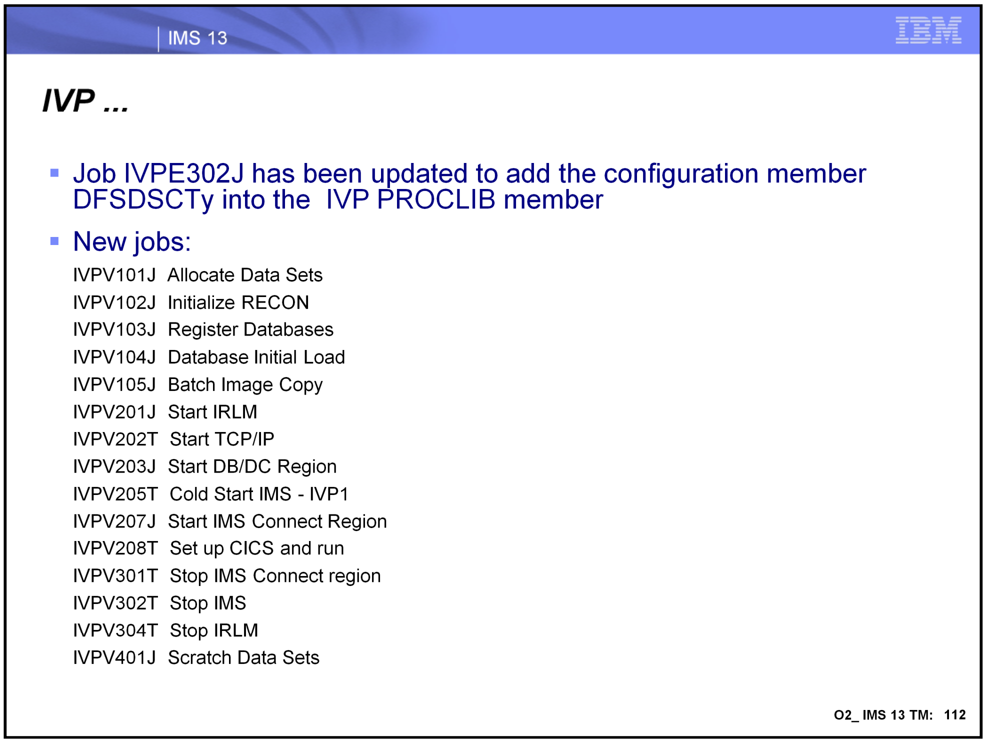
– DFSSGNXP

```
DFSSGNXP     DSECT     BYTES            DESCRIPTION
SGNFLG1      DS   X     01-01           FLAG BYTE 1
SGN1RM       EQU  X'80'                 USING RM TO MANAGE SHR RESOURCES
SGN1SQ       EQU  X'40'                 SHARED QUEUE ACTIVE
SGN1ESTN     EQU  X'20'                 PROCESSING ETO STSN SIGNON
SGN1STAC     EQU  X'10'                 PROCESSING STATIC SIGNON
SGN1TCPI     EQU  X'08'                 ISC TCPIP
```



IVP

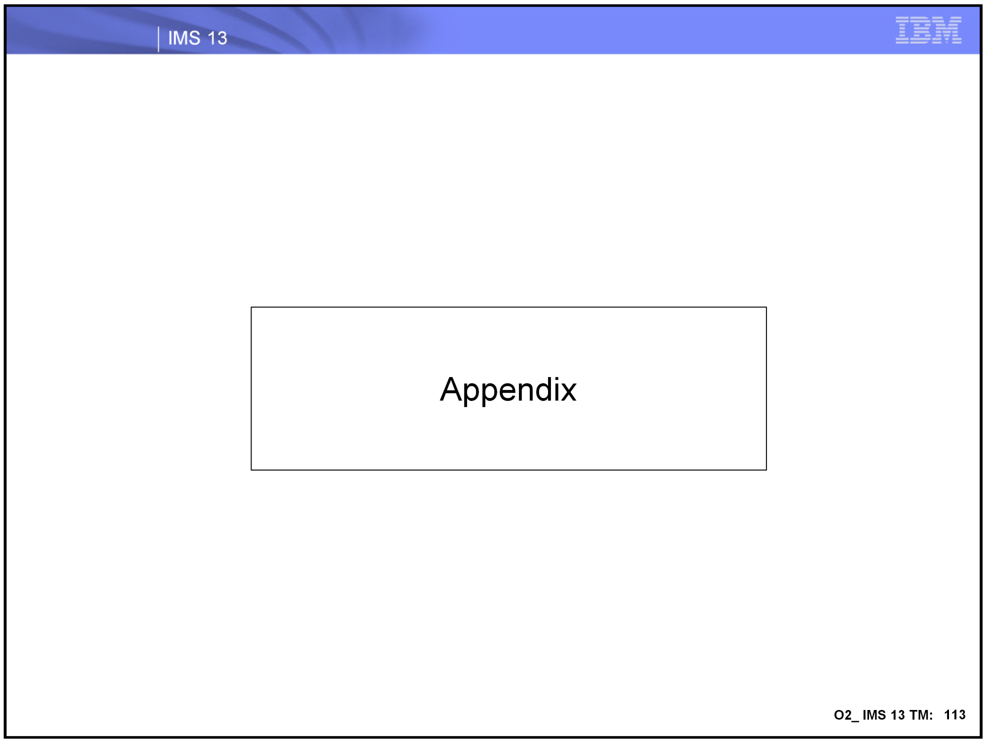
- 4 new variables will be defined in the repository server configuration
  - IXUDNAME (Descriptor name) specifies the name of the logon descriptor
  - IXUCONNM (IMS Connect name) specifies the local IMS connect instance
  - IXUOUTSZ (output buffer size) specifies the size of the output buffer to be used. This keyword is the same as the OUTBUF= keyword on the TERMINAL macro
  - IXUSEGSZ (Segment size) specifies the segment size. The acceptable values are from 256 to 32000 bytes. The default is 256 bytes. Refer to the description of the SEGSIZE= keyword in TERMINAL macro for more information about calculating segment size



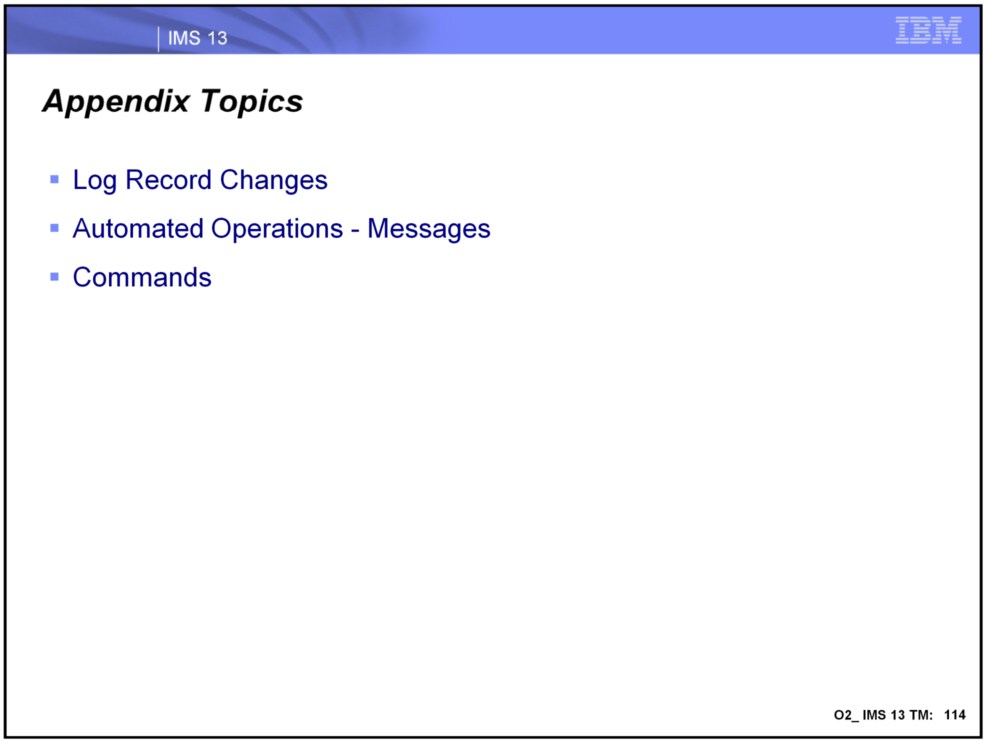
IVP ...

- Job IVPE302J has been updated to add the configuration member DFSDSCTy into the IVP PROCLIB member
- New jobs:

IVPV101J	Allocate Data Sets
IVPV102J	Initialize RECON
IVPV103J	Register Databases
IVPV104J	Database Initial Load
IVPV105J	Batch Image Copy
IVPV201J	Start IRLM
IVPV202T	Start TCP/IP
IVPV203J	Start DB/DC Region
IVPV205T	Cold Start IMS - IVP1
IVPV207J	Start IMS Connect Region
IVPV208T	Set up CICS and run
IVPV301T	Stop IMS Connect region
IVPV302T	Stop IMS
IVPV304T	Stop IRLM
IVPV401J	Scratch Data Sets

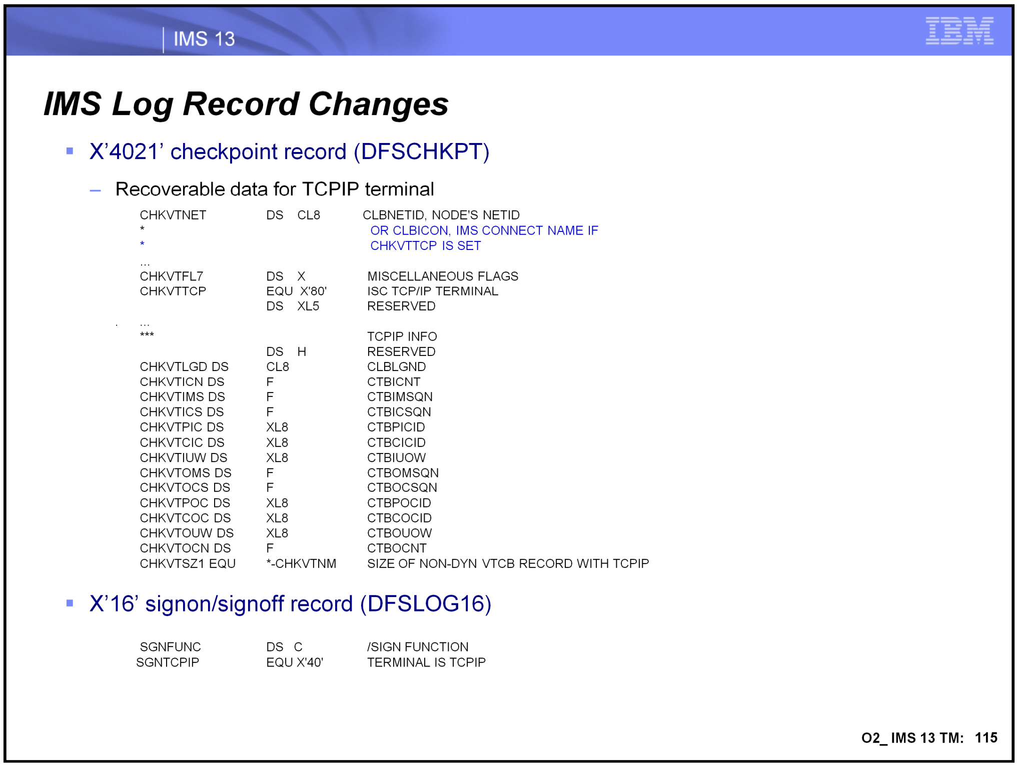


Appendix



Appendix Topics

- Log Record Changes
- Automated Operations - Messages
- Commands



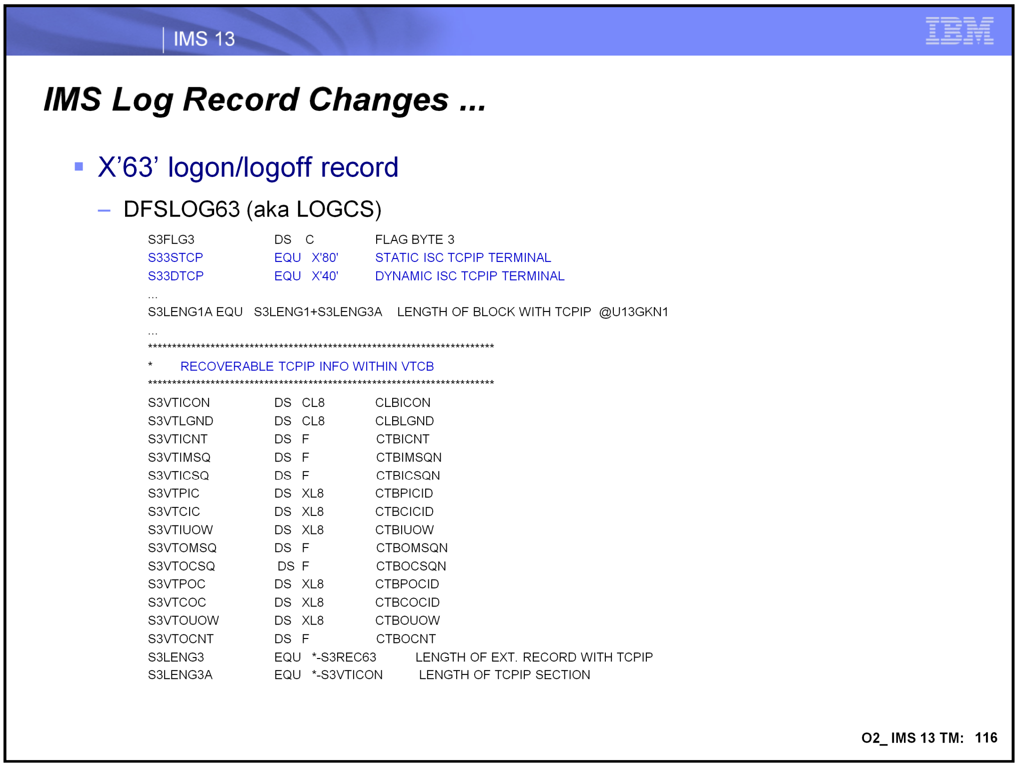
IMS Log Record Changes

- X’4021’ checkpoint record (DFSCHKPT)
  - Recoverable data for TCPIP terminal

```
CHKVTNET          DS   CL8        CLBNETID, NODE'S NETID
*                                  OR CLBICON, IMS CONNECT NAME IF
*                                  CHKVTTCP  IS SET
...
CHKVTFL7          DS   X           MISCELLANEOUS FLAGS
CHKVTTCP          EQU  X'80'       ISC TCP/IP TERMINAL
                  DS   XL5         RESERVED
.   ...
***                                TCPIP INFO
                  DS   H           RESERVED
CHKVTLGD DS       CL8              CLBLGND
CHKVTICN DS       F                CTBICNT
CHKVTIMS DS       F                CTBIMSQN
CHKVTICS DS       F                CTBICSQN
CHKVTPIC DS       XL8              CTBPICID
CHKVTCIC DS       XL8              CTBCICID
CHKVTIUW DS       XL8              CTBIUOW
CHKVTOMS DS       F                CTBOMSQN
CHKVTOCS DS       F                CTBOCSQN
CHKVTPOC DS       XL8              CTBPOCID
CHKVTCOC DS       XL8              CTBCOCID
CHKVTOUW DS       XL8              CTBOUOW
CHKVTOCN DS       F                CTBOCNT
CHKVTSZ1 EQU      *-CHKVTNM        SIZE OF NON-DYN  VTCB RECORD WITH TCPIP
```

- X’16’ signon/signoff record (DFSLOG16)

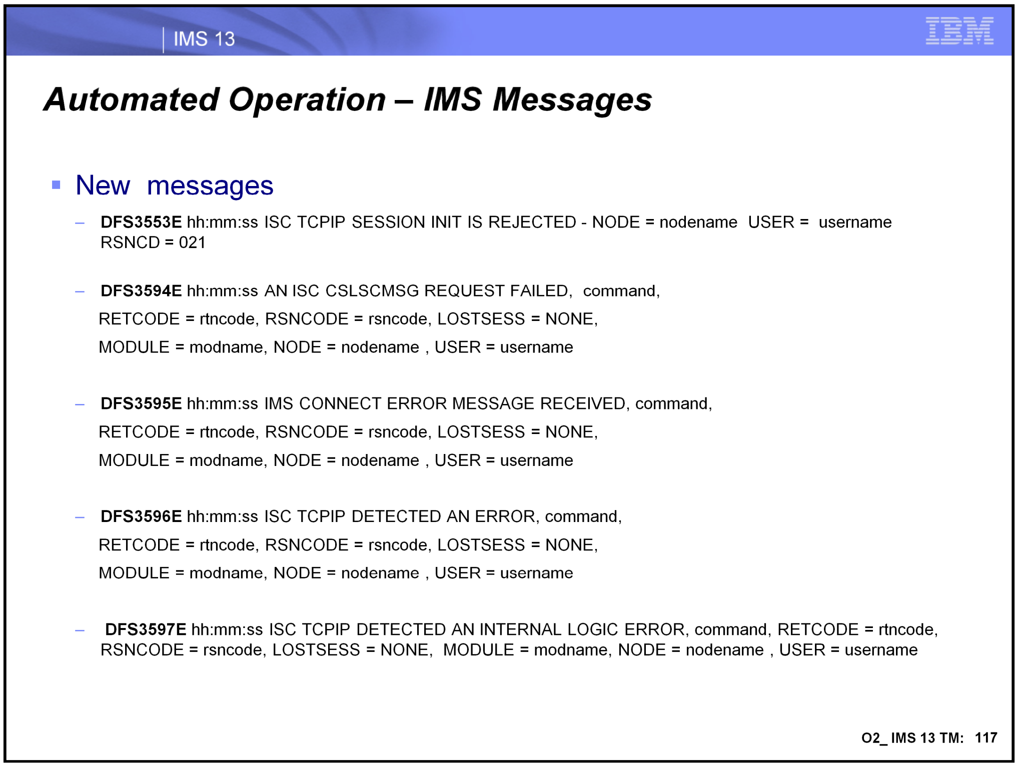
```
 SGNFUNC          DS   C           /SIGN FUNCTION
SGNTCPIP          EQU X'40'        TERMINAL IS TCPIP
```



IMS Log Record Changes …

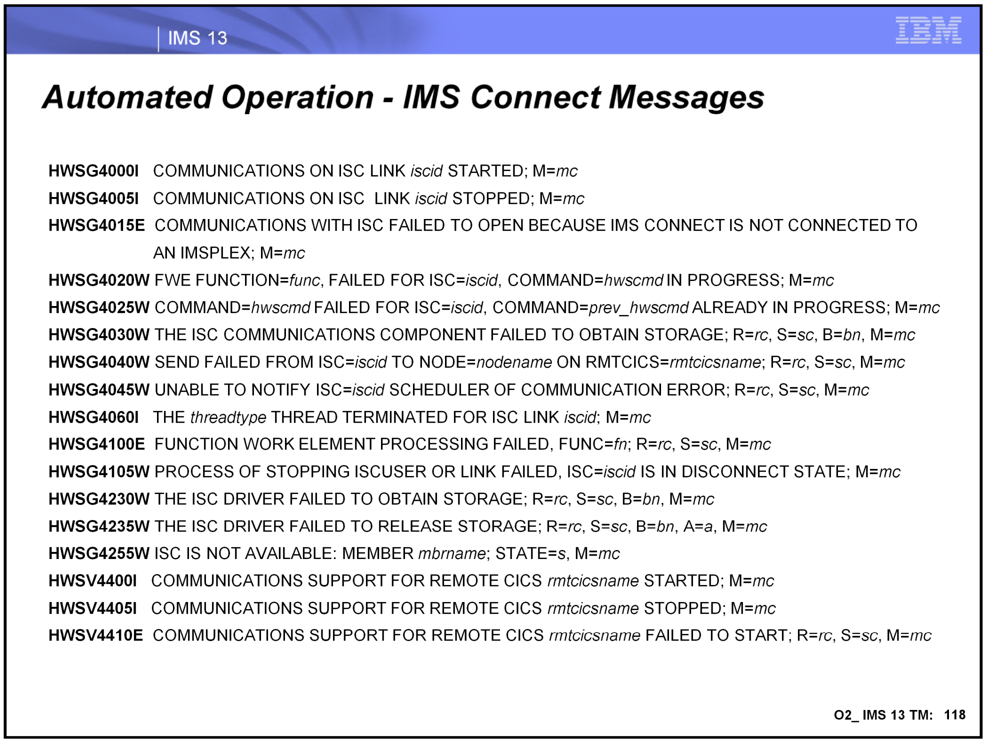
- X’63’ logon/logoff record
  - DFSLOG63 (aka LOGCS)

```
S3FLG3          DS   C          FLAG BYTE 3
S33STCP         EQU  X'80'      STATIC ISC TCPIP TERMINAL
S33DTCP         EQU  X'40'      DYNAMIC ISC TCPIP TERMINAL
...
S3LENG1A EQU  S3LENG1+S3LENG3A   LENGTH OF BLOCK WITH TCPIP  @U13GKN1
...
***********************************************************************
*    RECOVERABLE TCPIP INFO WITHIN VTCB
***********************************************************************
S3VTICON        DS  CL8         CLBICON
S3VTLGND        DS  CL8         CLBLGND
S3VTICNT        DS  F           CTBICNT
S3VTIMSQ        DS  F           CTBIMSQN
S3VTICSQ        DS  F           CTBICSQN
S3VTPIC         DS  XL8         CTBPICID
S3VTCIC         DS  XL8         CTBCICID
S3VTIUOW        DS  XL8         CTBIUOW
S3VTOMSQ        DS  F           CTBOMSQN
S3VTOCSQ         DS F           CTBOCSQN
S3VTPOC         DS  XL8         CTBPOCID
S3VTCOC         DS  XL8         CTBCOCID
S3VTOUOW        DS  XL8         CTBOUOW
S3VTOCNT        DS  F           CTBOCNT
S3LENG3         EQU  *-S3REC63      LENGTH OF EXT. RECORD WITH TCPIP
S3LENG3A        EQU  *-S3VTICON      LENGTH OF TCPIP SECTION
```



Automated Operation – IMS Messages

- New messages
  - **DFS3553E** hh:mm:ss ISC TCPIP SESSION INIT IS REJECTED - NODE = nodename USER = username RSNCD = 021
  - **DFS3594E** hh:mm:ss AN ISC CSLSCMSG REQUEST FAILED, command,
    RETCODE = rtncode, RSNCODE = rsncode, LOSTSESS = NONE,
    MODULE = modname, NODE = nodename , USER = username
  - **DFS3595E** hh:mm:ss IMS CONNECT ERROR MESSAGE RECEIVED, command,
    RETCODE = rtncode, RSNCODE = rsncode, LOSTSESS = NONE,
    MODULE = modname, NODE = nodename , USER = username
  - **DFS3596E** hh:mm:ss ISC TCPIP DETECTED AN ERROR, command,
    RETCODE = rtncode, RSNCODE = rsncode, LOSTSESS = NONE,
    MODULE = modname, NODE = nodename , USER = username
  - **DFS3597E** hh:mm:ss ISC TCPIP DETECTED AN INTERNAL LOGIC ERROR, command, RETCODE = rtncode, RSNCODE = rsncode, LOSTSESS = NONE, MODULE = modname, NODE = nodename , USER = username



Automated Operation - IMS Connect Messages

HWSG4000I COMMUNICATIONS ON ISC LINK *iscid* STARTED; M=*mc*

HWSG4005I COMMUNICATIONS ON ISC LINK *iscid* STOPPED; M=*mc*

HWSG4015E COMMUNICATIONS WITH ISC FAILED TO OPEN BECAUSE IMS CONNECT IS NOT CONNECTED TO AN IMSPLEX; M=*mc*

HWSG4020W FWE FUNCTION=*func*, FAILED FOR ISC=*iscid*, COMMAND=*hwscmd* IN PROGRESS; M=*mc*

HWSG4025W COMMAND=*hwscmd* FAILED FOR ISC=*iscid*, COMMAND=*prev_hwscmd* ALREADY IN PROGRESS; M=*mc*

HWSG4030W THE ISC COMMUNICATIONS COMPONENT FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

HWSG4040W SEND FAILED FROM ISC=*iscid* TO NODE=*nodename* ON RMTCICS=*rmtcicsname*; R=*rc*, S=*sc*, M=*mc*

HWSG4045W UNABLE TO NOTIFY ISC=*iscid* SCHEDULER OF COMMUNICATION ERROR; R=*rc*, S=*sc*, M=*mc*

HWSG4060I THE *threadtype* THREAD TERMINATED FOR ISC LINK *iscid*; M=*mc*

HWSG4100E FUNCTION WORK ELEMENT PROCESSING FAILED, FUNC=*fn*; R=*rc*, S=*sc*, M=*mc*

HWSG4105W PROCESS OF STOPPING ISCUSER OR LINK FAILED, ISC=*iscid* IS IN DISCONNECT STATE; M=*mc*

HWSG4230W THE ISC DRIVER FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

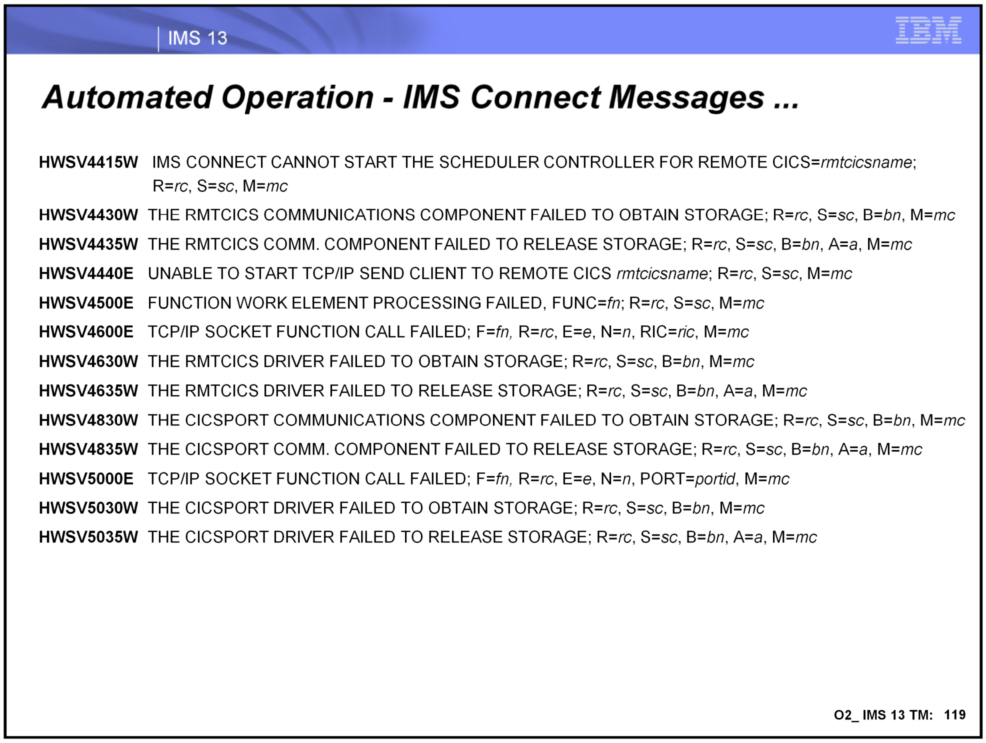
HWSG4235W THE ISC DRIVER FAILED TO RELEASE STORAGE; R=*rc*, S=*sc*, B=*bn*, A=*a*, M=*mc*

HWSG4255W ISC IS NOT AVAILABLE: MEMBER *mbrname*; STATE=*s*, M=*mc*

HWSV4400I COMMUNICATIONS SUPPORT FOR REMOTE CICS *rmtcicsname* STARTED; M=*mc*

HWSV4405I COMMUNICATIONS SUPPORT FOR REMOTE CICS *rmtcicsname* STOPPED; M=*mc*

HWSV4410E COMMUNICATIONS SUPPORT FOR REMOTE CICS *rmtcicsname* FAILED TO START; R=*rc*, S=*sc*, M=*mc*



Automated Operation - IMS Connect Messages ...

HWSV4415W IMS CONNECT CANNOT START THE SCHEDULER CONTROLLER FOR REMOTE CICS=*rmtcicsname*; R=*rc*, S=*sc*, M=*mc*

HWSV4430W THE RMTCICS COMMUNICATIONS COMPONENT FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

HWSV4435W THE RMTCICS COMM. COMPONENT FAILED TO RELEASE STORAGE; R=*rc*, S=*sc*, B=*bn*, A=*a*, M=*mc*

HWSV4440E UNABLE TO START TCP/IP SEND CLIENT TO REMOTE CICS *rmtcicsname*; R=*rc*, S=*sc*, M=*mc*

HWSV4500E FUNCTION WORK ELEMENT PROCESSING FAILED, FUNC=*fn*; R=*rc*, S=*sc*, M=*mc*

HWSV4600E TCP/IP SOCKET FUNCTION CALL FAILED; F=*fn,* R=*rc*, E=*e*, N=*n*, RIC=*ric*, M=*mc*

HWSV4630W THE RMTCICS DRIVER FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

HWSV4635W THE RMTCICS DRIVER FAILED TO RELEASE STORAGE; R=*rc*, S=*sc*, B=*bn*, A=*a*, M=*mc*

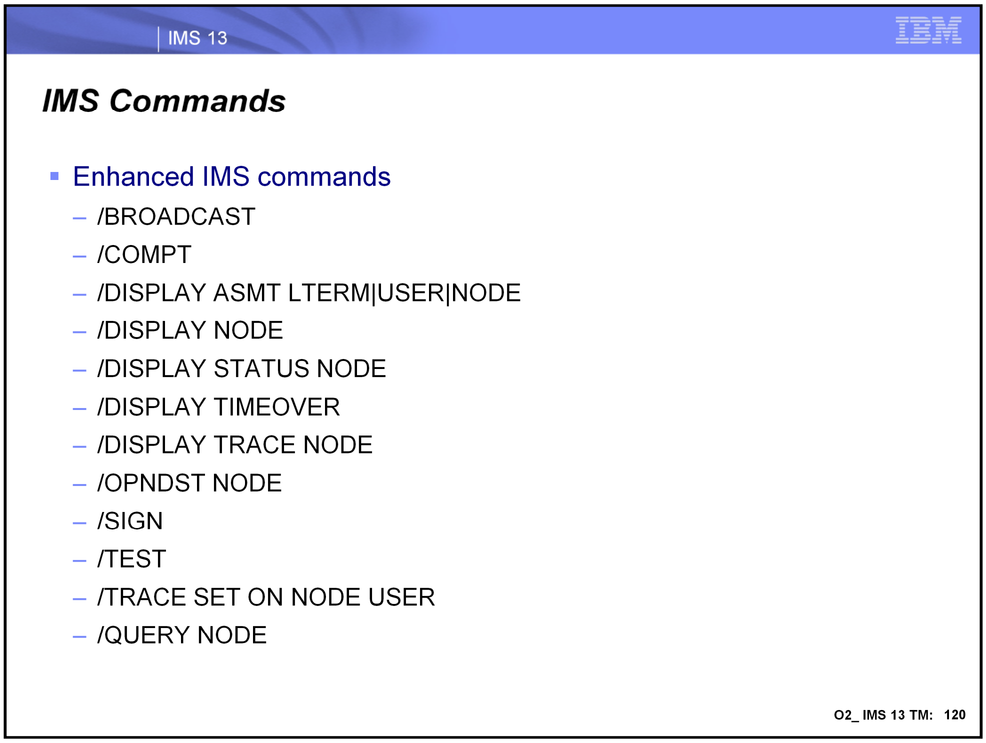
HWSV4830W THE CICSPORT COMMUNICATIONS COMPONENT FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

HWSV4835W THE CICSPORT COMM. COMPONENT FAILED TO RELEASE STORAGE; R=*rc*, S=*sc*, B=*bn*, A=*a*, M=*mc*

HWSV5000E TCP/IP SOCKET FUNCTION CALL FAILED; F=*fn,* R=*rc*, E=*e*, N=*n*, PORT=*portid*, M=*mc*

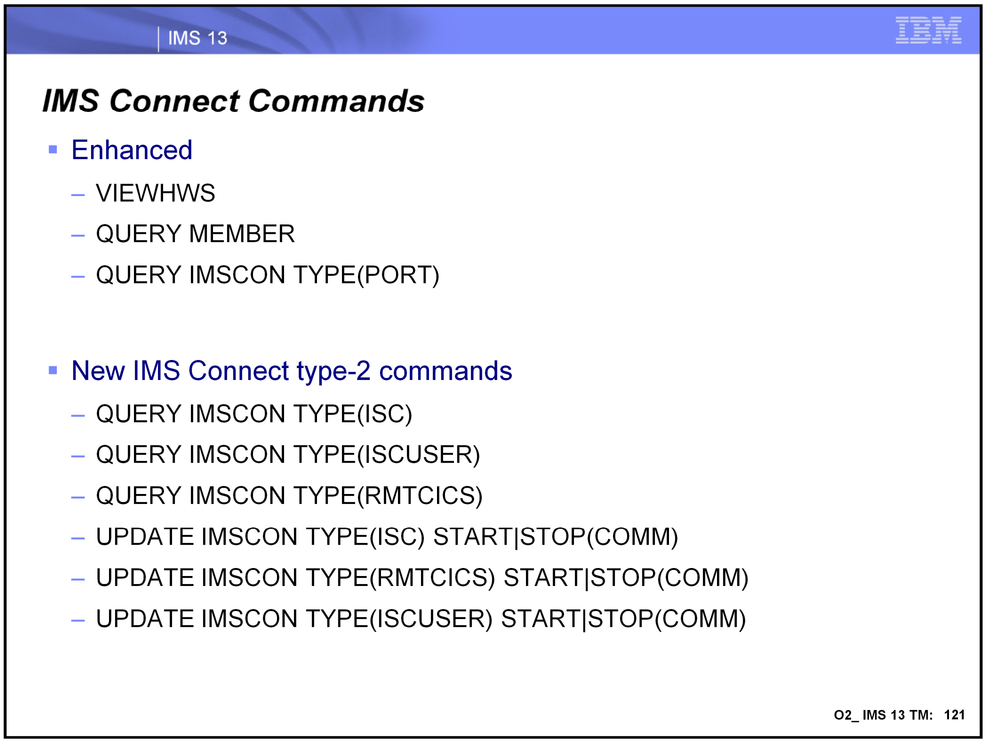
HWSV5030W THE CICSPORT DRIVER FAILED TO OBTAIN STORAGE; R=*rc*, S=*sc*, B=*bn*, M=*mc*

HWSV5035W THE CICSPORT DRIVER FAILED TO RELEASE STORAGE; R=*rc*, S=*sc*, B=*bn*, A=*a*, M=*mc*



IMS Commands

- Enhanced IMS commands
  - /BROADCAST
  - /COMPT
  - /DISPLAY ASMT LTERM|USER|NODE
  - /DISPLAY NODE
  - /DISPLAY STATUS NODE
  - /DISPLAY TIMEOVER
  - /DISPLAY TRACE NODE
  - /OPNDST NODE
  - /SIGN
  - /TEST
  - /TRACE SET ON NODE USER
  - /QUERY NODE



IMS 13

IMS Connect Commands

- Enhanced
  - VIEWHWS
  - QUERY MEMBER
  - QUERY IMSCON TYPE(PORT)

- New IMS Connect type-2 commands
  - QUERY IMSCON TYPE(ISC)
  - QUERY IMSCON TYPE(ISCUSER)
  - QUERY IMSCON TYPE(RMTCICS)
  - UPDATE IMSCON TYPE(ISC) START|STOP(COMM)
  - UPDATE IMSCON TYPE(RMTCICS) START|STOP(COMM)
  - UPDATE IMSCON TYPE(ISCUSER) START|STOP(COMM)

The following are still to be enhanced:

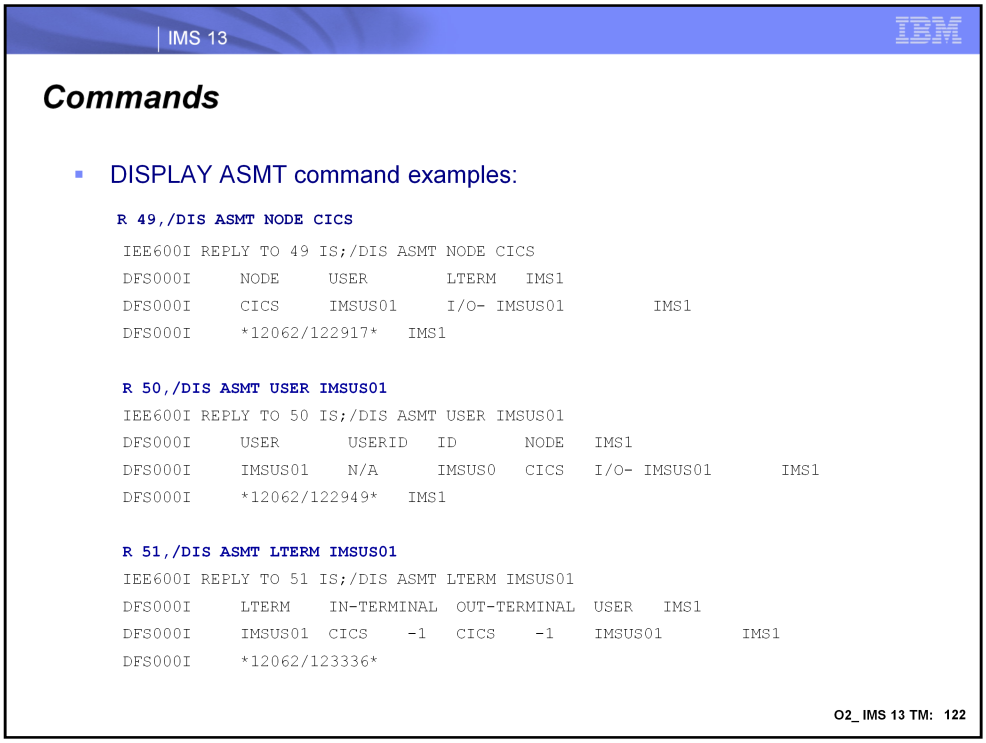
•STOPCLNT

•STOPSCLN

•QUERY IMSCON TYPE(SENDCLNT)

•UPDATE IMSCON TYPE(SENDCLNT) STOP(COMM)

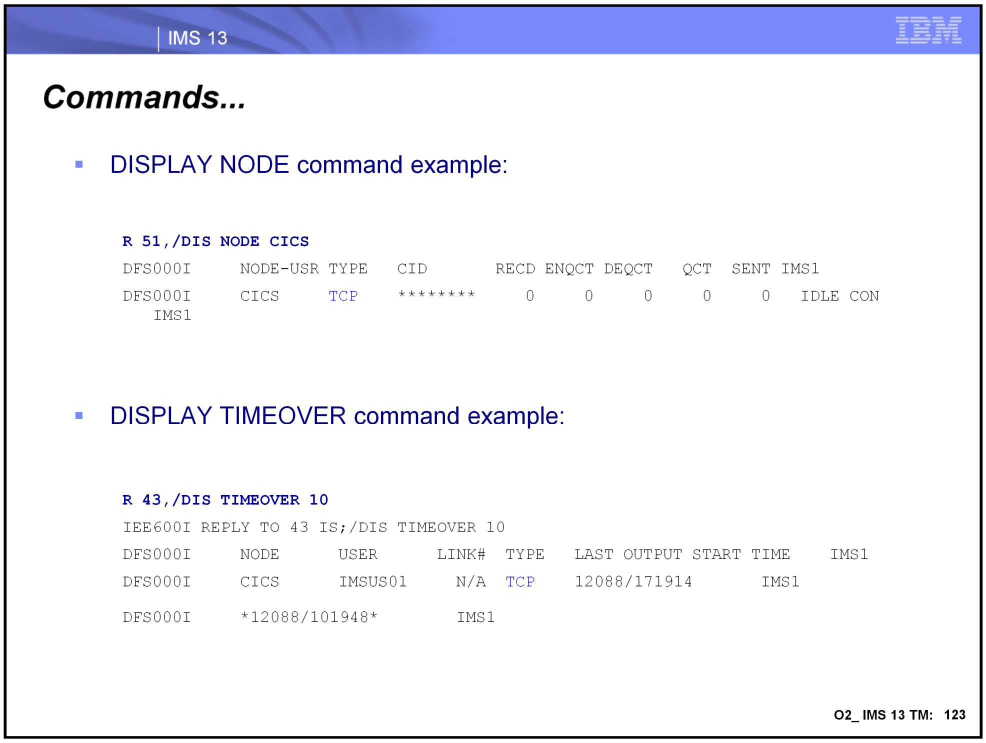
•UPDATE IMSCON TYPE(ISCUSER) START|STOP(COMM)



Commands

- DISPLAY ASMT command examples:

```
R 49,/DIS ASMT NODE CICS
IEE600I REPLY TO 49 IS;/DIS ASMT NODE CICS
DFS000I     NODE     USER        LTERM   IMS1
DFS000I     CICS     IMSUS01     I/O- IMSUS01        IMS1
DFS000I     *12062/122917*   IMS1

R 50,/DIS ASMT USER IMSUS01
IEE600I REPLY TO 50 IS;/DIS ASMT USER IMSUS01
DFS000I     USER       USERID   ID       NODE   IMS1
DFS000I     IMSUS01    N/A      IMSUS0   CICS   I/O- IMSUS01       IMS1
DFS000I     *12062/122949*   IMS1

R 51,/DIS ASMT LTERM IMSUS01
IEE600I REPLY TO 51 IS;/DIS ASMT LTERM IMSUS01
DFS000I     LTERM    IN-TERMINAL  OUT-TERMINAL  USER   IMS1
DFS000I     IMSUS01  CICS    -1   CICS    -1    IMSUS01        IMS1
DFS000I     *12062/123336*
```



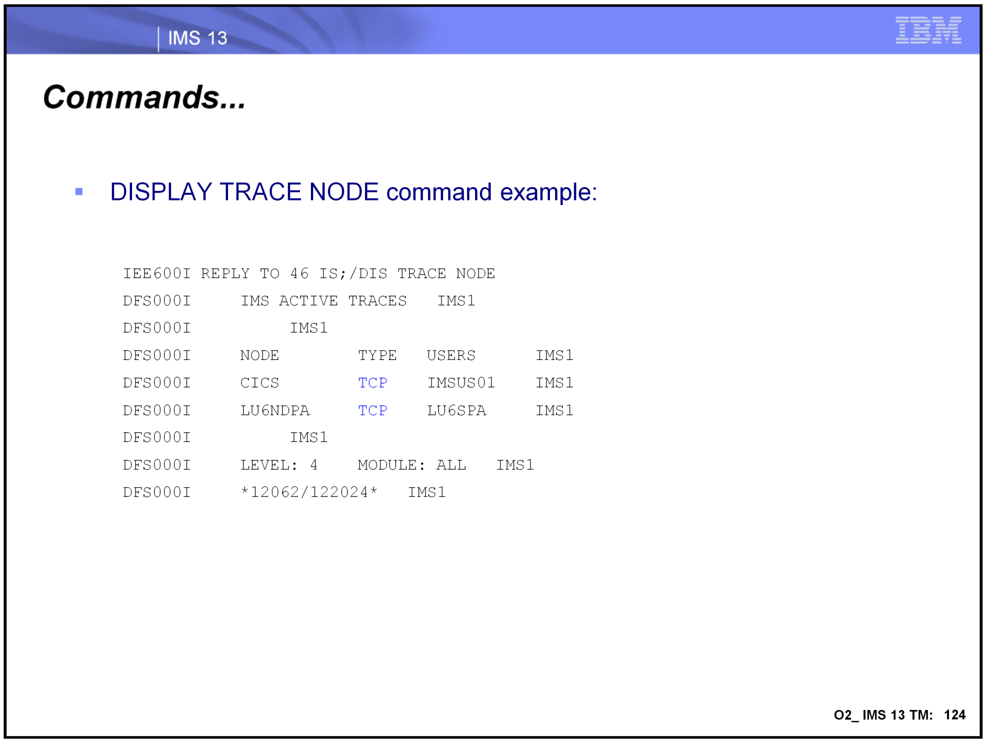
Commands...

- DISPLAY NODE command example:

```
R 51,/DIS NODE CICS
DFS000I     NODE-USR TYPE   CID        RECD ENQCT DEQCT   QCT  SENT IMS1
DFS000I     CICS     TCP    ********      0     0     0     0     0   IDLE CON
   IMS1
```

- DISPLAY TIMEOVER command example:

```
R 43,/DIS TIMEOVER 10
IEE600I REPLY TO 43 IS;/DIS TIMEOVER 10
DFS000I     NODE      USER      LINK#  TYPE   LAST OUTPUT START TIME    IMS1
DFS000I     CICS      IMSUS01     N/A  TCP    12088/171914       IMS1

DFS000I     *12088/101948*        IMS1
```



Commands...

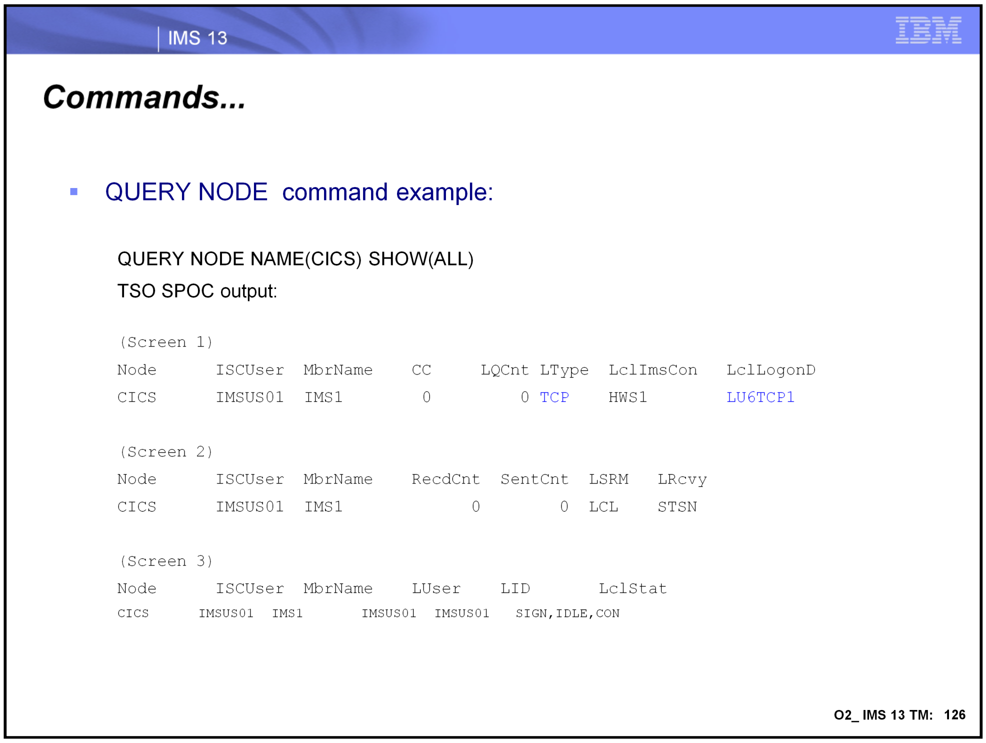
- DISPLAY TRACE NODE command example:

```
IEE600I REPLY TO 46 IS;/DIS TRACE NODE
DFS000I     IMS ACTIVE TRACES   IMS1
DFS000I          IMS1
DFS000I     NODE         TYPE   USERS      IMS1
DFS000I     CICS         TCP    IMSUS01    IMS1
DFS000I     LU6NDPA      TCP    LU6SPA     IMS1
DFS000I          IMS1
DFS000I     LEVEL: 4     MODULE: ALL   IMS1
DFS000I     *12062/122024*   IMS1
```

Commands...

- QUERY NODE command syntax:
  - Addition of IMSCON and LOGOND parameters to the SHOW() keyword to show IMS Connect name and the logon descriptor name used for dynamic terminal

```
          ┌,──────────┐
►──────  ──ALL──────────────────────────────────────────────────►
         ├─AFFIN────┤
         ├─CID──────┤
         ├─CONV─────┤
         ├─COUNT────┤
         ├─EMHQ─────┤
         ├─ID───────┤
         ├─IMSCON───┤
         ├─LOGOND───┤
         ├─LTERM────┤
         ├─MODETBL──┤
         ├─OWNER────┤
         ├─PRESET───┤
         ├─QCNT─────┤
         ├─RECOVERY─┤
         ├─STATUS───┤
         ├─TYPE─────┤
         ├─USER─────┤
         ├─USERID───┤
         └─VERSION──┘
```



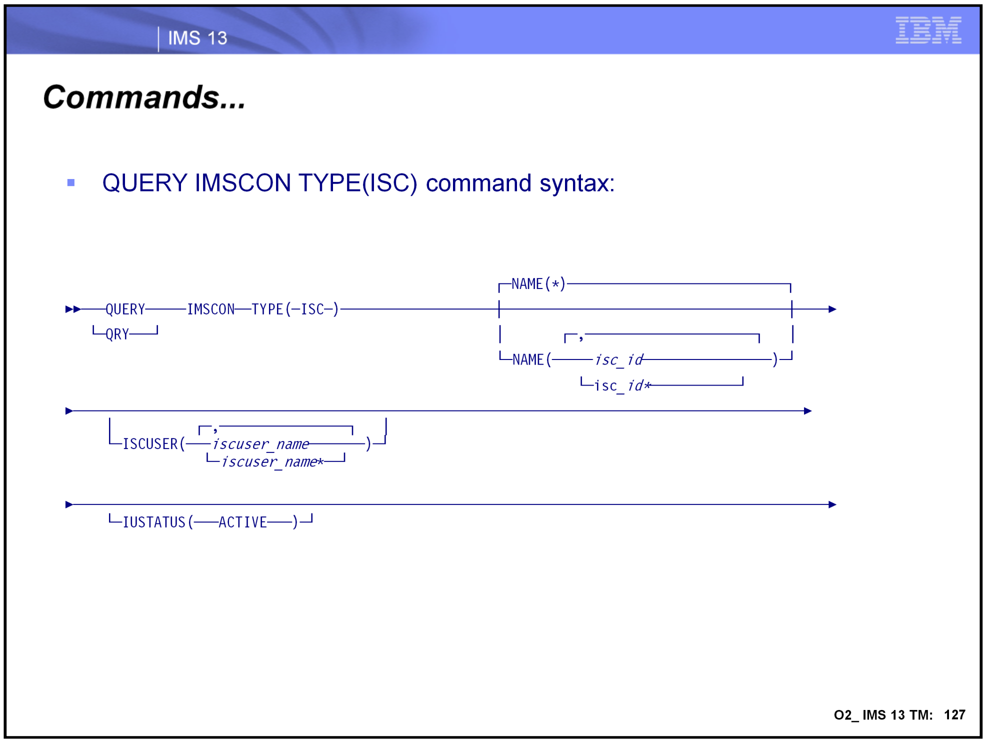
Commands...

- QUERY NODE command example:

QUERY NODE NAME(CICS) SHOW(ALL)

TSO SPOC output:

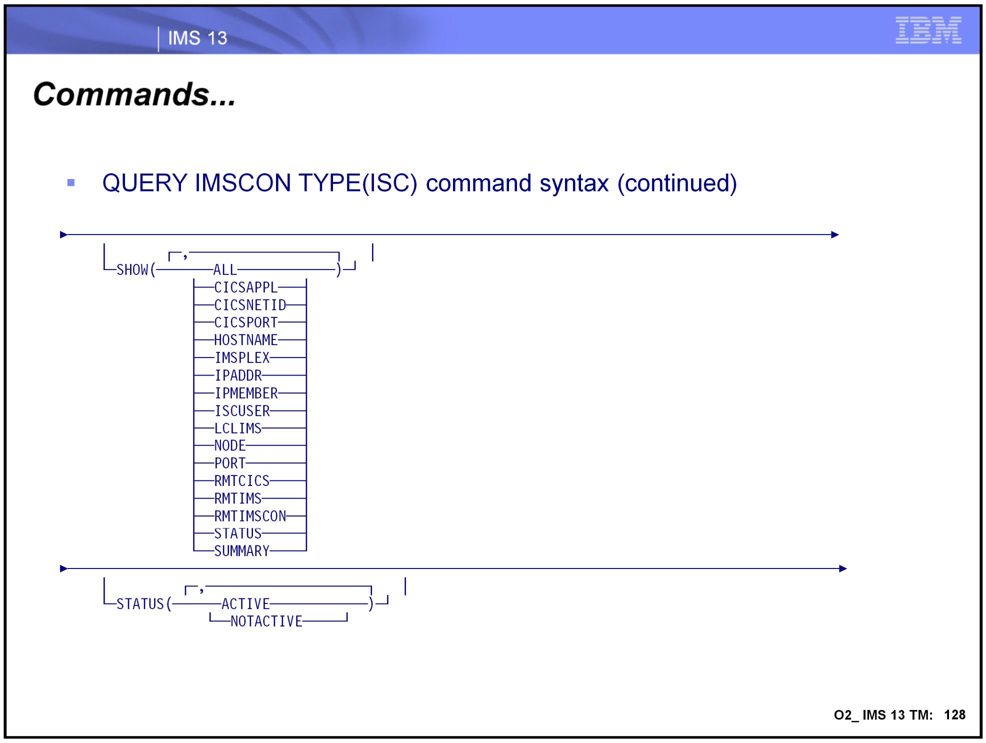
```
(Screen 1)
Node      ISCUser  MbrName    CC      LQCnt LType  LclImsCon    LclLogonD
CICS      IMSUS01  IMS1        0          0 TCP    HWS1         LU6TCP1

(Screen 2)
Node      ISCUser  MbrName    RecdCnt  SentCnt  LSRM   LRcvy
CICS      IMSUS01  IMS1             0        0  LCL    STSN

(Screen 3)
Node      ISCUser  MbrName    LUser    LID       LclStat
CICS      IMSUS01  IMS1       IMSUS01  IMSUS01   SIGN,IDLE,CON
```



Commands...

- QUERY IMSCON TYPE(ISC) command syntax:

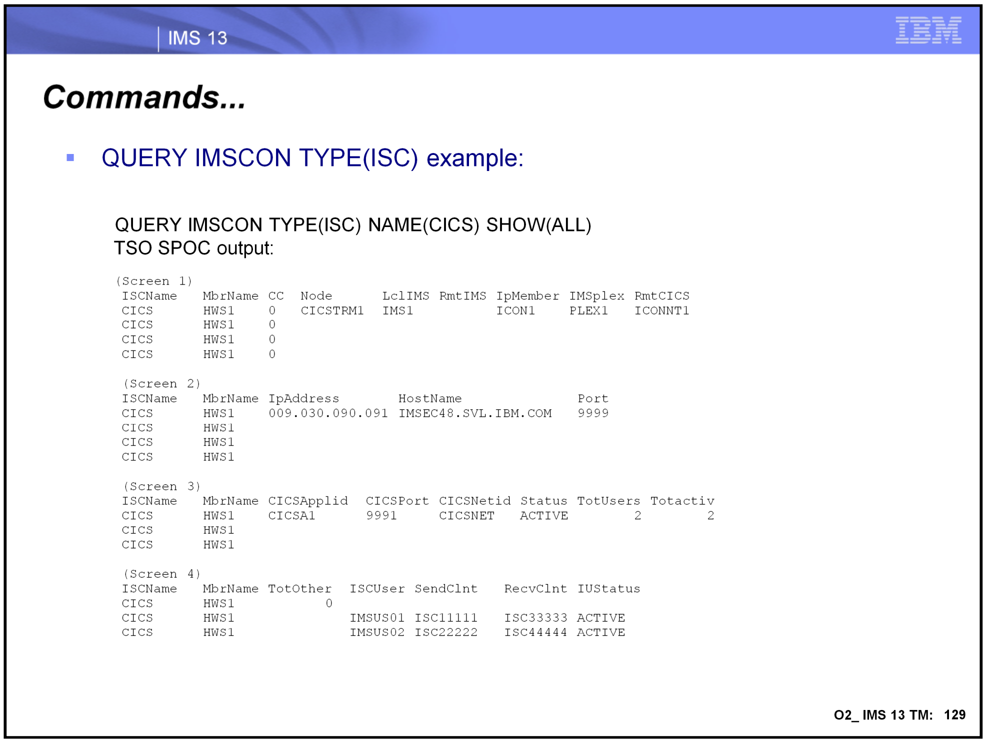
```
                                                      ┌─NAME(*)──────────────────────┐
▶▶──┬─QUERY─┬──IMSCON──TYPE(─ISC─)────────────────────┼──────────────────────────────┼──▶
    └─QRY───┘                                         │        ┌─,──────────────┐    │
                                                      └─NAME(──┬─isc_id──────┬──┴─)──┘
                                                               └─isc_id*─────┘

▶──┬──────────────────────────────────────────┬──────────────────────────────────▶
   └─ISCUSER(──┬─,──────────────┐──)──────────┘
               ├─iscuser_name───┤
               └─iscuser_name*──┘

▶──┬──────────────────────────┬────────────────────────────────────────────────────▶
   └─IUSTATUS(──ACTIVE──)─────┘
```



Commands...

- QUERY IMSCON TYPE(ISC) command syntax (continued)

```
►────────────────────────────────────────────────────────────────────►
  │    ┌─,──────────────────┐   │
  └─SHOW(───────ALL──────────────┴─)─┘
              ├─CICSAPPL───┤
              ├─CICSNETID──┤
              ├─CICSPORT───┤
              ├─HOSTNAME───┤
              ├─IMSPLEX────┤
              ├─IPADDR─────┤
              ├─IPMEMBER───┤
              ├─ISCUSER────┤
              ├─LCLIMS─────┤
              ├─NODE───────┤
              ├─PORT───────┤
              ├─RMTCICS────┤
              ├─RMTIMS─────┤
              ├─RMTIMSCON──┤
              ├─STATUS─────┤
              └─SUMMARY────┘
►────────────────────────────────────────────────────────────────────►
  │       ┌─,──────────────────┐   │
  └─STATUS(──────ACTIVE─────────┴─)─┘
               └─NOTACTIVE─────┘
```

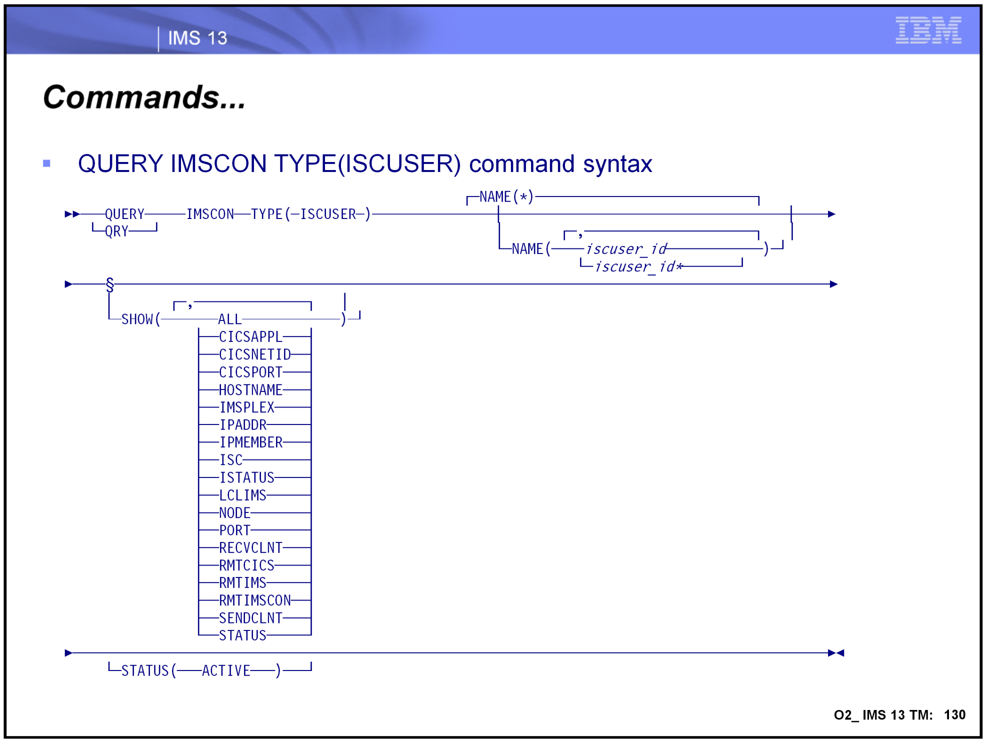


Commands...

- QUERY IMSCON TYPE(ISC) example:

QUERY IMSCON TYPE(ISC) NAME(CICS) SHOW(ALL)
TSO SPOC output:

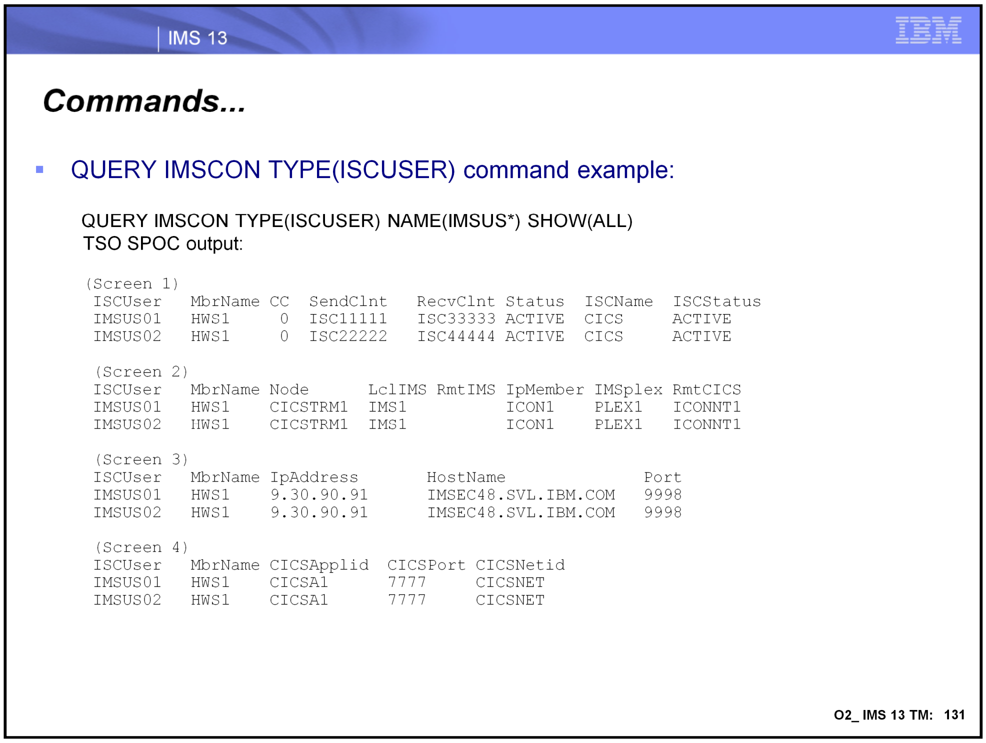
```
(Screen 1)
 ISCName   MbrName CC  Node      LclIMS RmtIMS IpMember IMSplex RmtCICS
 CICS      HWS1    0   CICSTRM1  IMS1          ICON1    PLEX1   ICONNT1
 CICS      HWS1    0
 CICS      HWS1    0
 CICS      HWS1    0

 (Screen 2)
 ISCName   MbrName IpAddress       HostName              Port
 CICS      HWS1    009.030.090.091 IMSEC48.SVL.IBM.COM   9999
 CICS      HWS1
 CICS      HWS1
 CICS      HWS1

 (Screen 3)
 ISCName   MbrName CICSApplid  CICSPort CICSNetid Status TotUsers Totactiv
 CICS      HWS1    CICSA1      9991     CICSNET   ACTIVE        2        2
 CICS      HWS1
 CICS      HWS1

 (Screen 4)
 ISCName   MbrName TotOther  ISCUser SendClnt   RecvClnt IUStatus
 CICS      HWS1           0
 CICS      HWS1              IMSUS01 ISC11111   ISC33333 ACTIVE
 CICS      HWS1              IMSUS02 ISC22222   ISC44444 ACTIVE
```



Commands...

- QUERY IMSCON TYPE(ISCUSER) command syntax

```
                                                  ┌─NAME(*)──────────────────────────┐
►►──┬─QUERY─┬──IMSCON──TYPE(─ISCUSER─)────────────┼──────────────────────────────────┼──►
    └─QRY───┘                                     │       ┌─,────────────────────┐   │
                                                  └─NAME(─┬─iscuser_id───────────┬┴─)─┘
                                                          └─iscuser_id*──────────┘

►──┬─§──────────────────────────────────────────────────────────────────────────────┬──►
   │        ┌─,─────────────┐        │
   └─SHOW(──┬─ALL───────────────────┬─)─┘
            ├─CICSAPPL──────┤
            ├─CICSNETID─────┤
            ├─CICSPORT──────┤
            ├─HOSTNAME──────┤
            ├─IMSPLEX───────┤
            ├─IPADDR────────┤
            ├─IPMEMBER──────┤
            ├─ISC───────────┤
            ├─ISTATUS───────┤
            ├─LCLIMS────────┤
            ├─NODE──────────┤
            ├─PORT──────────┤
            ├─RECVCLNT──────┤
            ├─RMTCICS───────┤
            ├─RMTIMS────────┤
            ├─RMTIMSCON─────┤
            ├─SENDCLNT──────┤
            └─STATUS────────┘

►──┬──────────────────────────┬────────────────────────────────────────────────────────►◄
   └─STATUS(──ACTIVE──)───────┘
```

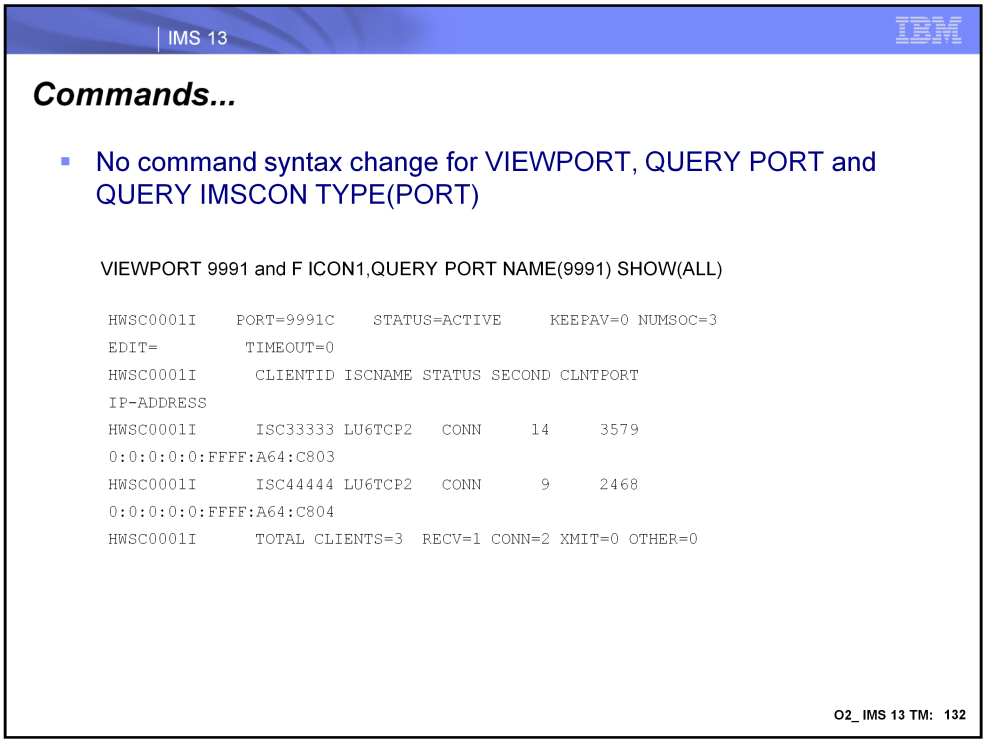


Commands…

- QUERY IMSCON TYPE(ISCUSER) command example:

QUERY IMSCON TYPE(ISCUSER) NAME(IMSUS*) SHOW(ALL)
TSO SPOC output:

```
(Screen 1)
 ISCUser   MbrName CC  SendClnt   RecvClnt Status  ISCName  ISCStatus
 IMSUS01   HWS1     0  ISC11111   ISC33333 ACTIVE  CICS     ACTIVE
 IMSUS02   HWS1     0  ISC22222   ISC44444 ACTIVE  CICS     ACTIVE

 (Screen 2)
 ISCUser   MbrName Node      LclIMS RmtIMS IpMember IMSplex RmtCICS
 IMSUS01   HWS1    CICSTRM1  IMS1          ICON1    PLEX1   ICONNT1
 IMSUS02   HWS1    CICSTRM1  IMS1          ICON1    PLEX1   ICONNT1

 (Screen 3)
 ISCUser   MbrName IpAddress       HostName              Port
 IMSUS01   HWS1    9.30.90.91      IMSEC48.SVL.IBM.COM   9998
 IMSUS02   HWS1    9.30.90.91      IMSEC48.SVL.IBM.COM   9998

 (Screen 4)
 ISCUser   MbrName CICSApplid  CICSPort CICSNetid
 IMSUS01   HWS1    CICSA1      7777     CICSNET
 IMSUS02   HWS1    CICSA1      7777     CICSNET
```



IMS 13

Commands...

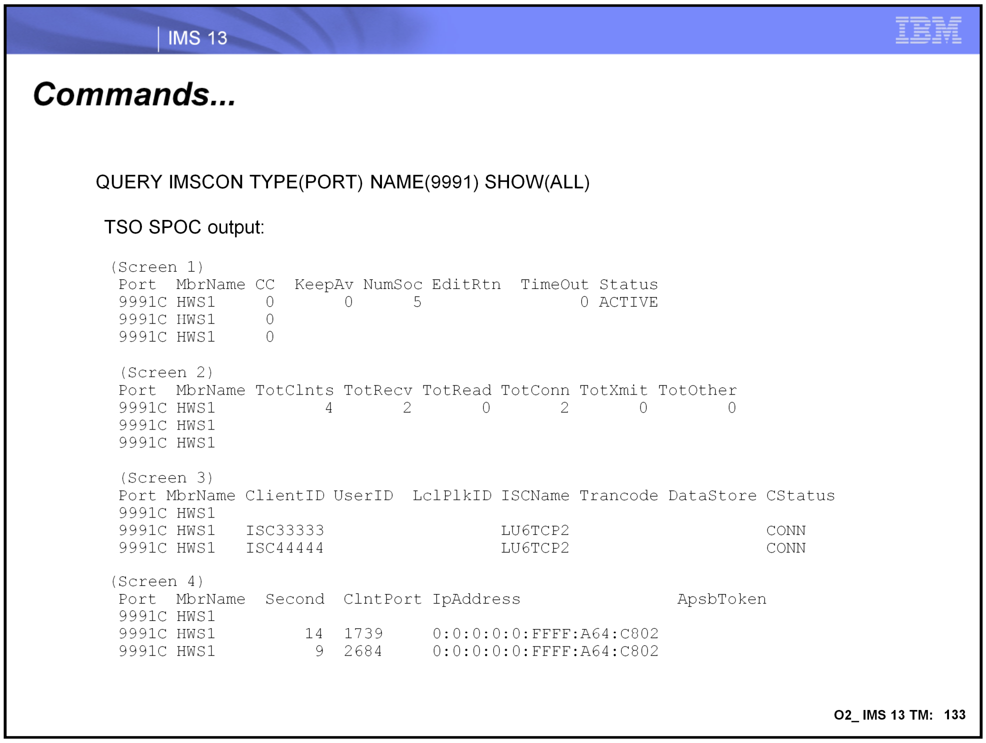
- No command syntax change for VIEWPORT, QUERY PORT and QUERY IMSCON TYPE(PORT)

VIEWPORT 9991 and F ICON1,QUERY PORT NAME(9991) SHOW(ALL)

```
HWSC0001I    PORT=9991C    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3
EDIT=         TIMEOUT=0
HWSC0001I      CLIENTID ISCNAME STATUS SECOND CLNTPORT
IP-ADDRESS
HWSC0001I      ISC33333 LU6TCP2   CONN     14     3579
0:0:0:0:0:FFFF:A64:C803
HWSC0001I      ISC44444 LU6TCP2   CONN      9     2468
0:0:0:0:0:FFFF:A64:C804
HWSC0001I      TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
```

O2_ IMS 13 TM: 132

VIEWPORT, QUERY PORT, and QUERY IMSCON TYPE(PORT) do not have syntax change



Commands...

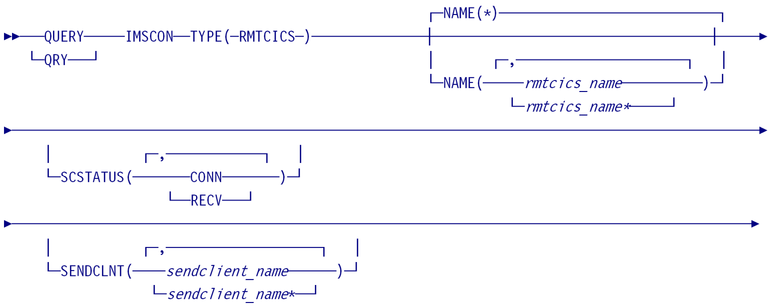
QUERY IMSCON TYPE(PORT) NAME(9991) SHOW(ALL)

TSO SPOC output:

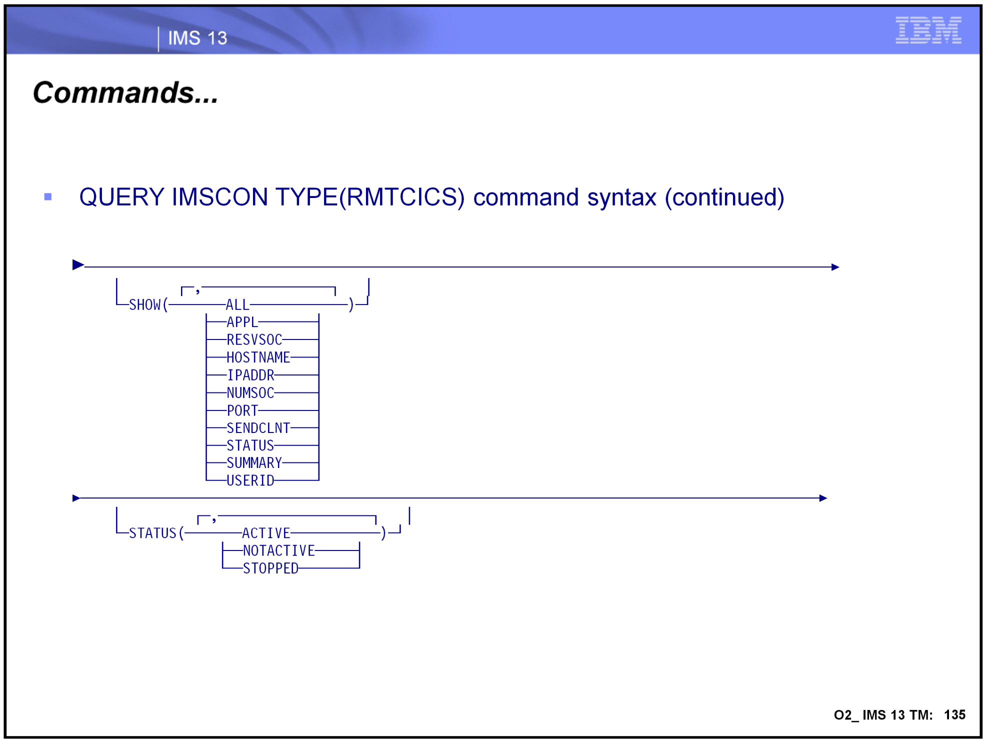
```
(Screen 1)
 Port  MbrName CC  KeepAv NumSoc EditRtn  TimeOut Status
 9991C HWS1     0       0      5                0 ACTIVE
 9991C HWS1     0
 9991C HWS1     0

 (Screen 2)
 Port  MbrName TotClnts TotRecv TotRead TotConn TotXmit TotOther
 9991C HWS1           4       2       0       2       0        0
 9991C HWS1
 9991C HWS1

 (Screen 3)
 Port MbrName ClientID UserID  LclPlkID ISCName Trancode DataStore CStatus
 9991C HWS1
 9991C HWS1   ISC33333                  LU6TCP2                    CONN
 9991C HWS1   ISC44444                  LU6TCP2                    CONN

(Screen 4)
 Port  MbrName  Second  ClntPort IpAddress                ApsbToken
 9991C HWS1
 9991C HWS1         14  1739     0:0:0:0:0:FFFF:A64:C802
 9991C HWS1          9  2684     0:0:0:0:0:FFFF:A64:C802
```

Commands...

- QUERY IMSCON TYPE(RMTCICS) command syntax



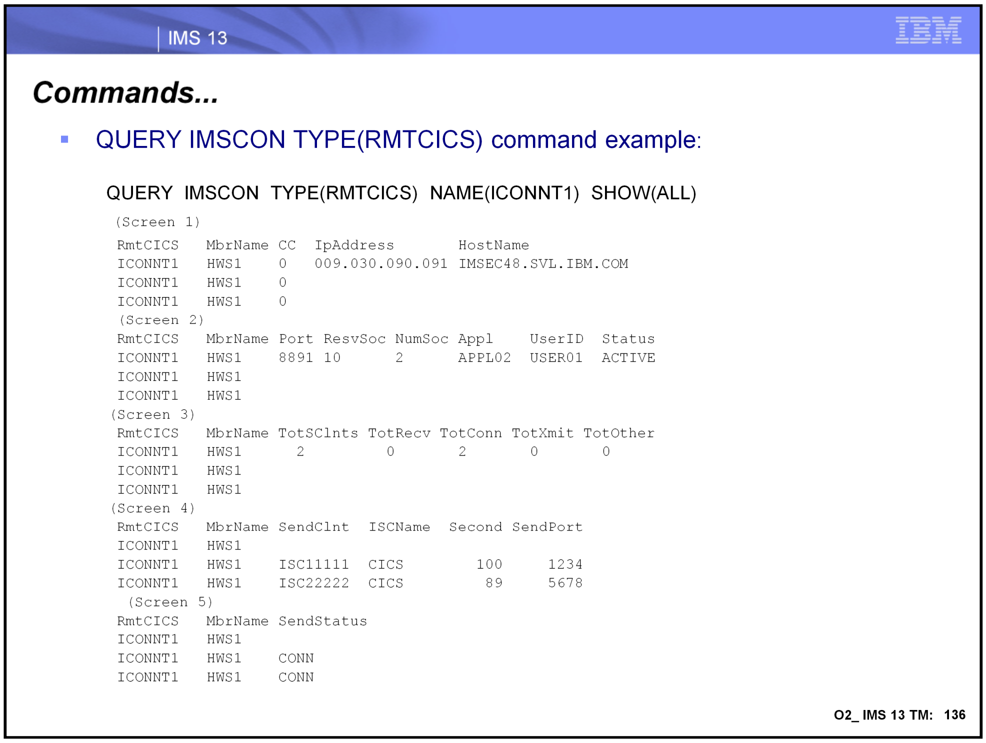
```
                                                         ┌─NAME(*)──────────────────────────┐
▶▶──QUERY──────IMSCON──TYPE(─RMTCICS─)───────────────────┼──────────────────────────────────┼──▶
    └─QRY───┘                                            │       ┌─,──────────────────────┐ │
                                                         └─NAME(─────rmtcics_name─────────────)─┘
                                                                   └─rmtcics_name*──────┘
▶──────────────────────────────────────────────────────────────────────────────────────────────▶
     │            ┌─,─────────────┐    │
     └─SCSTATUS(──────────CONN──────────)─┘
                     └─RECV───┘
▶──────────────────────────────────────────────────────────────────────────────────────────────▶
     │            ┌─,─────────────────────┐    │
     └─SENDCLNT(────sendclient_name──────────)─┘
                   └─sendclient_name*──┘
```



Commands...

- QUERY IMSCON TYPE(RMTCICS) command syntax (continued)

```
►──────────────────────────────────────────────────────────────►
   │   ┌─,──────────────┐  │
   └SHOW(─┬─ALL───────┬─┴─)─┘
          ├─APPL──────┤
          ├─RESVSOC───┤
          ├─HOSTNAME──┤
          ├─IPADDR────┤
          ├─NUMSOC────┤
          ├─PORT──────┤
          ├─SENDCLNT──┤
          ├─STATUS────┤
          ├─SUMMARY───┤
          └─USERID────┘
►─────────────────────────────────────────────────────────────►
   │       ┌─,────────────────┐   │
   └STATUS(─┬─ACTIVE─────┬─┴─)─┘
            ├─NOTACTIVE──┤
            └─STOPPED────┘
```



Commands...

- QUERY IMSCON TYPE(RMTCICS) command example:

```
QUERY  IMSCON  TYPE(RMTCICS)  NAME(ICONNT1)  SHOW(ALL)
```

```
(Screen 1)
RmtCICS   MbrName CC  IpAddress       HostName
ICONNT1   HWS1    0   009.030.090.091 IMSEC48.SVL.IBM.COM
ICONNT1   HWS1    0
ICONNT1   HWS1    0
(Screen 2)
RmtCICS   MbrName Port ResvSoc NumSoc Appl    UserID  Status
ICONNT1   HWS1    8891 10      2       APPL02  USER01  ACTIVE
ICONNT1   HWS1
ICONNT1   HWS1
(Screen 3)
RmtCICS   MbrName TotSClnts TotRecv TotConn TotXmit TotOther
ICONNT1   HWS1      2         0       2       0       0
ICONNT1   HWS1
ICONNT1   HWS1
(Screen 4)
RmtCICS   MbrName SendClnt  ISCName  Second SendPort
ICONNT1   HWS1
ICONNT1   HWS1    ISC11111  CICS        100     1234
ICONNT1   HWS1    ISC22222  CICS         89     5678
 (Screen 5)
RmtCICS   MbrName SendStatus
ICONNT1   HWS1
ICONNT1   HWS1    CONN
ICONNT1   HWS1    CONN
```