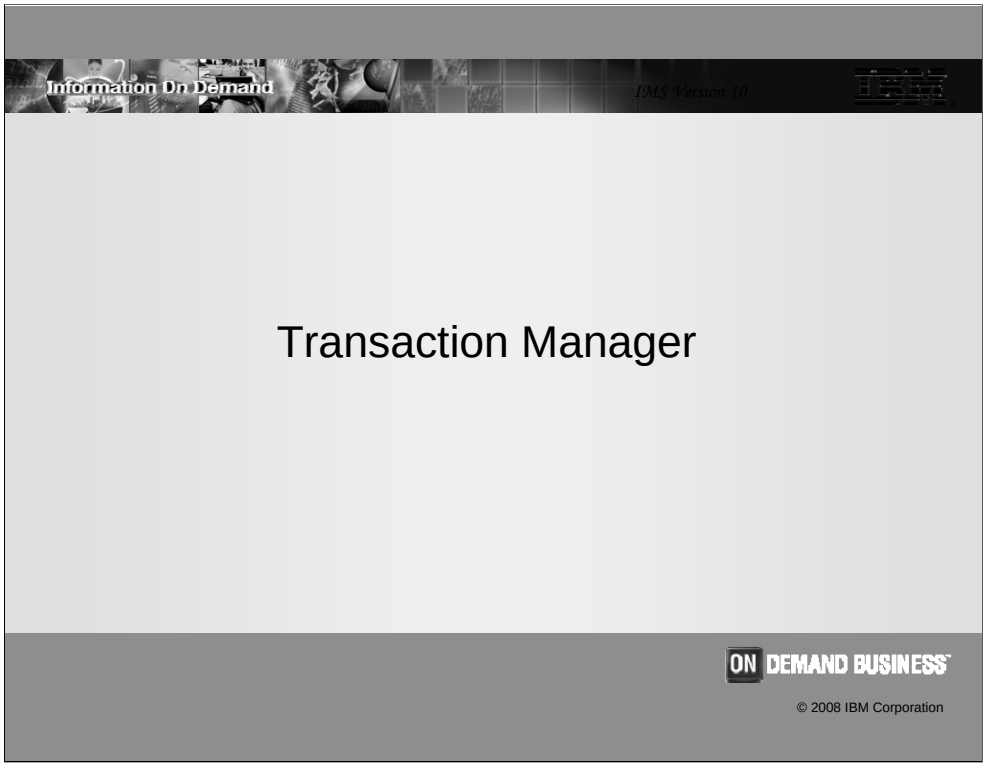

Information On Demand
IMS Version 10
Transaction Manager
ON DEMAND BUSINESS
© 2008 IBM Corporation

Transaction Manager

- MSC
- APPC
- Removal of BTAM Support

MSC

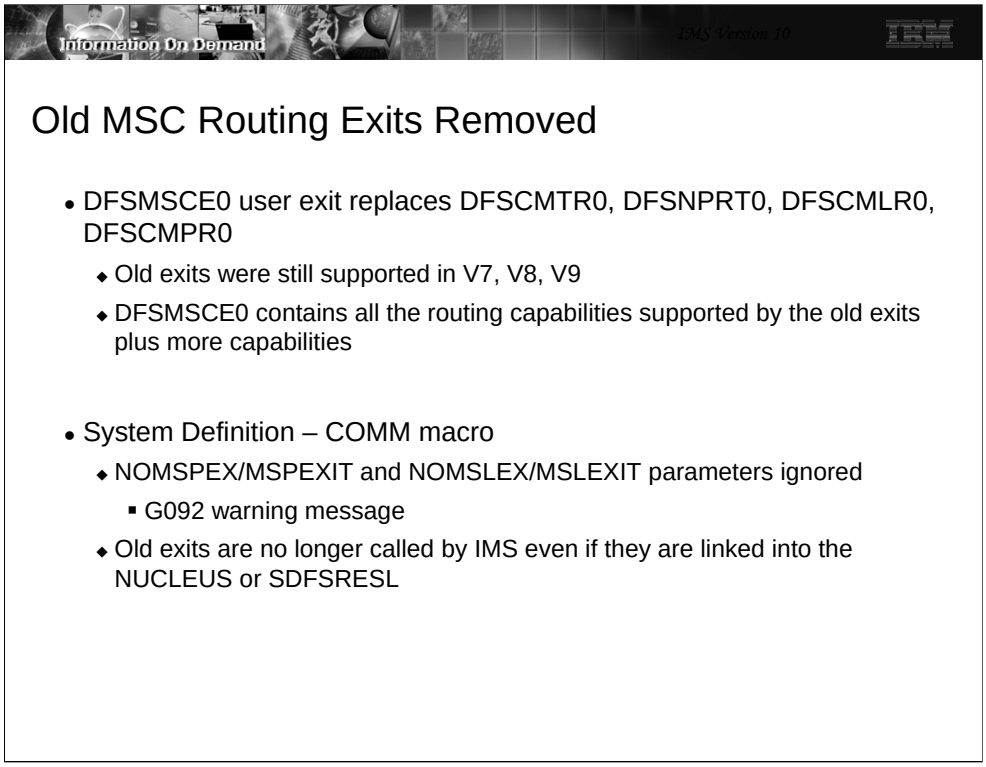
Highlights

- Removal of old routing exit routines DFSCMTR0, DFSNPRT0, DFSCMLR0, DFSCMPR0
  - Replaced by DFSMSCE0 which was introduced in IMS V7
    - Several IMS releases provided dual support and the ability to migrate
- Increased bandwidth to improve MSC link performance
  - Capability to allow multiple messages to be sent in the same buffer
    - Bandwidth mode
  - Reduction of logger I/Os by reducing number of CHECK WRITE calls
  - Expansion of maximum link buffer size to 64K
    - Capability to dynamically increase or decrease size by command
  - Enhanced link QUERY Statistics command
- MSC VGR support

IMS V10 removes the routing exit routines that were replaced by the TM and MSC Message Routing and Control User Exit (DFSMSCE0) beginning in IMS V7. DFSMSCE0 consolidated and replaced the following exits: MSC Terminal Routing Exit (DFSCMTR0), MSC Input Message Routing Exit (DFSNPRT0), MSC Link Receive Routing Exit (DFSCMLR0), and the MSC Program Routing Exit (DFSCMPR0). Although the older exit routines were supported concurrently with DFSMSCE0 in V7, V8, and V9, they are now being removed in V10 and will no longer be supported or called. DFSMSCE0 must be used. Note: DFSMSCE0 can be used in both MSC and non-MSC environments, although not all routing options will apply to non-MSC systems.

To improve the performance and bandwidth requirements of high-volume MSC systems, many customers today use a large number of parallel MSC links between pairs of IMS systems. This scheme can complicate the operations of the IMS systems and adds to the complexity of the system configurations as well as load balancing schemes. IMS V10 introduces several enhancements to increase MSC bandwidth. These include:

- Blocking multiple messages and responses into a single buffer when sending messages across the MSC links. In previous releases, IMS only sent one message or response at a time.
- Reducing the logger I/O operations by reducing the number of CHECK WRITE calls.
- Expanding the maximum link buffer size to 64K (was 32K) so that more messages and responses can fit into a buffer. Additionally, the initial link buffer sizes set during the IMS system definition process can be increased or decreased and displayed dynamically with IMS commands.
- Providing a command to provide statistics on a link. A secondary issue associated with MSC bandwidth is the difficulty that IMS operators and system programmers can encounter when MSC link performance is inadequate due to a message backup or poor performance. Inadequate performance data in previous IMS releases often resulted in days or weeks of analyzing log records and traces to resolve the issue. In IMS V10, this issue is resolved with an enhanced link Query Statistics command.



IMS V10 only supports the TM and MSC Message Routing and Control User Exit Routine (DFSMSCEO) for routing. There is no longer any support for the older routing exit routines DFSCMTR0, DFSNPRT0, DFSCMLR0, DFSCMPR0. DFSMSCE0 was introduced in IMS V7 to provide an opportunity to migrate over several releases.

The COMM macro specification during system definition, therefore, ignores the NOMSPEX/MSPEXIT and NOMSLEX/MSLEXIT parameters. Previous releases of IMS allowed specification of the parameters NOMSPEX/MSPEXIT to exclude/include the MSC Program routing exit (DFSCMPR0) and NOMSLEX/MSLEXIT to exclude/include the MSC Link Receive routing exit (DFSCMLR0). IMS no longer recognizes these routines even if they are included in the IMS resource library.

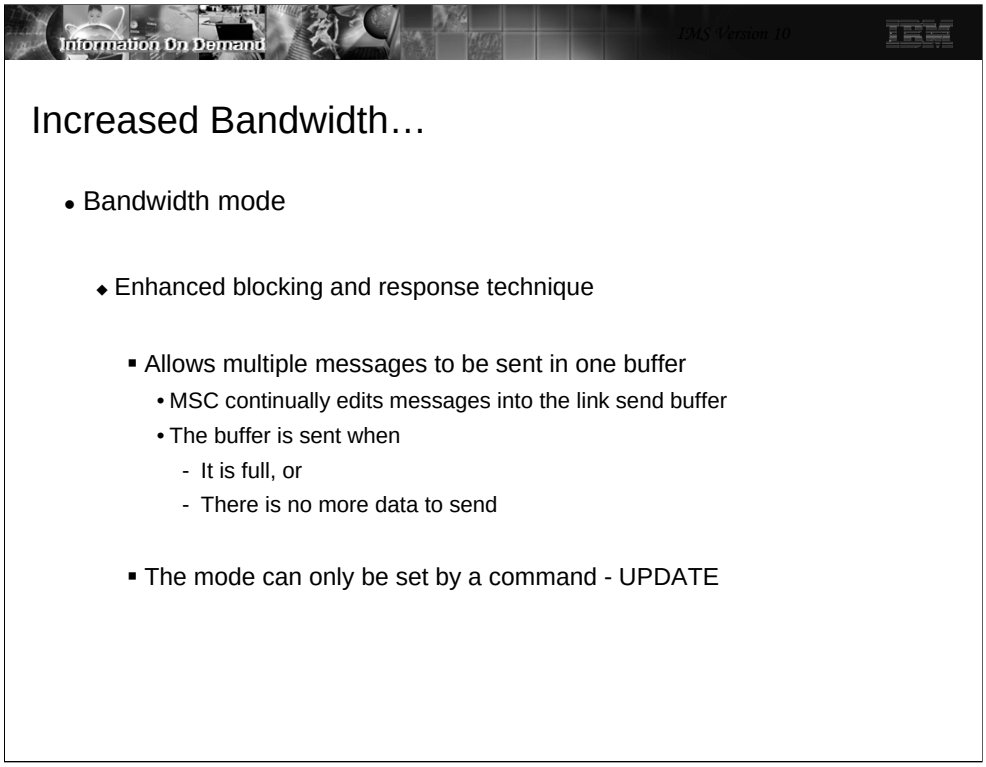
If the older exit routines still exist in the IMS release from which the migration is being done, then migration to DFSMSCE0 becomes a required action for the IMS V10 migration. The IMS Customization Guide documents a sample DFSMSCE0 exit along with the user edit parameter list macro (DFSMSCEP) which provides information on using and customizing the exit. To include the DFSMSCE0 replacement exit into the IMS system, it must be bound (linkedited) into IMS SDFSRESL (or a concatenated library).

Increased Bandwidth - Background

- Prior Releases
  - Input and output buffer to send messages on a physical link
    - Defined via BUFSIZE on MSPLINK macro at system definition
      - Applicable for the duration that the link is active
- Issues
  - Buffer size is fixed and may not account for increased traffic
  - Only one message or response can be sent per buffer
    - Even if the buffer is large enough to hold multiple messages/responses
  - The next message is not sent until the partner IMS responds that it has
    - Received, queued, and logged the message

To understand the changes for MSC bandwidth, a review of the issue is provided on this visual.

The definition of MSC physical links provides a specification for a BUFSIZE on the MSPLINK system definition macro. The buffer sizes specified are fixed at system definition. When the link is started, an input and output buffer is acquired and held for the duration of the link restart (i.e. link active), at the specified size to send and receive data (messages) across the link. Since each side (partner) of the link has a send and receive buffer, a message or response may be simultaneously sent each way. However, only one message or response is sent per buffer, even if the buffer is large enough to hold multiple messages/responses. Another message is not sent until the partner IMS responds that it has received, queued, and logged the message. For high-volume systems, the wait associated for a freed buffer can be unacceptable. To get around this issue, high-volume systems oftentimes are defined with a large number of parallel links to support the concurrent traffic from one IMS to another.



The MSC bandwidth mode in IMS V10 is a mechanism to determine whether or not to send multiple messages in one buffer. In non-bandwidth mode, MSC sends a maximum of one message or response per I/O operation (i.e. Send or write). In bandwidth mode, IMS attempts to maximize the capacity of a link by sending as many messages that are queued and ready to go, and responses that are owed for messages received, in the same buffer. By increasing the link buffer size, more and more messages and responses may be sent simultaneously.

Note that BANDWIDTH mode is not a system definition option. It can only be set ON with a command. The default is non-Bandwidth mode which allows MSC to function as in previous releases.

Increased Bandwidth…

- Bandwidth mode …
  - Considerations
    - Bandwidth mode requires both partner IMS systems to at IMS 10 level
      - Established by command on a link by link basis
        - Support for IMS 10 to IMS 10 links
        - No support for IMS 10 to IMS version 8 or 9 links
    - Non-bandwidth mode is compatible with V9/V8
      - No migration fixes required
        - Compatibility is handled on the IMS 10 side
    - Both sides of the link must specify the same bandwidth mode
      - Otherwise, link restart is rejected with a DFS3218 message

MSC bandwidth mode is only available on IMS 10 to IMS 10 connections.

The bandwidth mode may be set ON/OFF by the IMS UPDATE command on a link by link basis for all the CTC, MTM, or VTAM links.

The specified mode must be the same on both sides of the link or the link restart will be rejected with a DFS3218 message. IMS V10 MSC links that are connected to a down level release of IMS (i.e. V8, V9) will not be able to set bandwidth mode on the V8/V9 side of the link so the link, in effect, will not be able to operate in bandwidth mode.

Increased Bandwidth…

- Reduced Logger I/O
  - Prior Releases
    - IMS Check Writes (CHKW) to the WADS
      - Recoverable messages
        - On the send side when the last part of a message is sent
        - On the receive side when the message is enqueued
  - IMS 10
    - Bandwidth mode
      - One CHKW per send buffer regardless of the number of messages in the buffer
        - E.g., 5 messages in one buffer, results in 1 CHKW on the send side
      - 1 CHKW per message on the receive side
    - Non-bandwidth mode
      - One CHKW per message on the send side (same as pre-V10 systems)
        - E.g., 5 messages are sent in 5 buffers and result in 5 CHKWs on the send side
      - 1 CHKW per message on the receive side

Turning bandwidth mode ON can also impact logger I/O. As a general rule, there are 2 CHKWs in the path of a recoverable message that is sent across an MSC link, one on the send side when the last part of a message is sent, and one on the receive side when the received message is enqueued. There are no CHKWs issued for non-recoverable messages. Note, however that even though a message that is sent from a front-end IMS to a back-end system is non-recoverable (therefore no CHKWs), its response is always considered recoverable and will incur CHKWs. Therefore, recoverable remote transactions incur 4 CHKWs, two for the message going to the back-end and two for the response. Non-recoverable remote transactions incur 2 CHKWs, none for the non-recoverable message that is sent to the back-end, and two for the reply which IMS always sets to recoverable.

The choice of bandwidth mode, however, does impact the number of CHKWs that can be issued in an IMS V10 environment. Bandwidth mode potentially reduces the number of CHKWs issued because only one CHKW is written per send buffer regardless of how many messages are contained in the buffer. For example, if there are five messages in the buffer then only one CHKW is written rather than five. Note that on the receive side, the back-end IMS continues to issue one CHKW per message. When the response messages are ready, the back-end IMS will attempt to buffer as many messages as can fit in a buffer and issue one CHKW for the buffer. The front-end IMS which is the receiving system for the responses will issue one CHKW per message.

Non-bandwidth mode provides the same processing as pre-IMS V10 systems. For example, if the front-end IMS has five messages to send, these messages are all sent in separate buffers and five CHKWs will occur, one for each message. On the receive side, IMS will again issue one CHKW per message. The same processing and number of CHKWs occur for the responses from the back-end IMS to the front-end system.

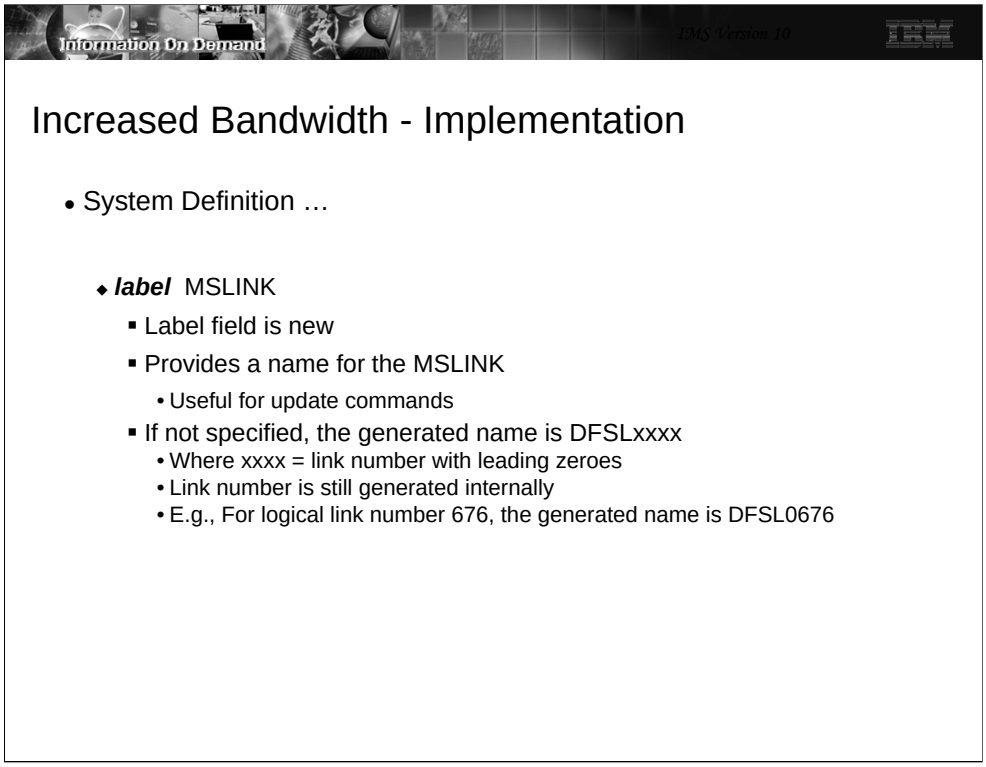
Increased Bandwidth

- Increase in link buffer size range
  - System Definition
    - MSPLINK BUFSIZE=size *where size is **1024** (from 160) **to 65536** (from 32K)*
      - Applicable to all link types: CTC, MTM and VTAM
    - If using Bandwidth mode, 4096 should be the minimum
    - IMS V10 initializes the buffers to the size specified
      - If link partner is IMS 10, no overhead is added
      - If link partner is verion 9 or earlier, overhead is added for compatibility
        - CTC and MTM: 78 at definition + 28 at initialization
        - VTAM: 288 + 28 (restricted to values in the VTAM table)
        - Reference the VTAM table in the MSPLINK macro in the manual

IMS 10 increases the range of buffer size specifications. The new MSPLINK BUFSIZE range is from 1024 to 65536 and specifies the input and output buffer sizes for each logical link defined for a physical link. The range is standardized and applicable to CTC, MTM and VTAM links. Prior to version 10, the size for CTC and MTM links ranged from 160 to 32689 and from 208 to 30720 for VTAM links. Note that when bandwidth mode is used, a BUFSIZE of 1024 may be too small to take advantage of the ability to send multiple messages with one buffer especially if the messages include some of the extended headers that capabilities such as OTMA impose. A value of 4096 is more realistic.

Additionally, in prior releases IMS system definition added 78 bytes to the MSPLINK BUFSIZE for CTC and MTM links, and 288 bytes to VTAM links. IMS Initialization added another 28 bytes. This overhead has been removed in V10. IMS initializes the MSC buffers to the actual BUFSIZE specified. For V10 to be compatible with V9 and below, however, IMS will continue to add the overhead (i.e. 78+28 for CTC and MTM links, 288+28 for VTAM links), at link restart if the partner IMS is below a V10 level. This action insures that the buffer sizes are compatible. If the partner is at V10 or higher, the overhead is not added.

Also note that in prior IMS releases, VTAM required buffer sizes to reflect a formula of X times 2 to the power of Y, where X had to be a value of 8 through 15, and Y has to be a value from 3 to 13. This restricted the MSC VTAM buffer sizes to certain values that were documented in a VTAM table under the MSPLINK macro in the Installation Guide, Volume 2 manual. For V10, these restrictions are removed, but remain for compatibility when communicating to a V9 or earlier release of IMS**.** Any BUFSIZE from 1024 to 65536 is acceptable. The table in the manual has been updated to reflect the applicable sizes.



A label field has been added to the MSLINK macro to allow a name to be assigned to the logical link. The name, if provided, can be used for the new command capabilities that are provided to affect specific links. If not specified, IMS will generate the name DFSLxxxx where xxx is the link number. Link numbers, as in previous releases, continue to be assigned automatically by IMS based on the relative position of the MSLINK macro in the SYSGEN.

Increased Bandwidth - Commands

- New command capability to change the link characteristics
  - Recoverable across a warm or emergency restart
  - Type 1 command

    /UPD MSLINK NAME (*name)* **SET**((BANDWIDTH(*ON* | *OFF)* BUFSIZE(*xxxxx))*
  - Type 2 command

    UPDATE MSLINK NAME (*name)* **SET**((BANDWIDTH(*ON* | *OFF)* BUFSIZE(*xxxxx))*
  - Note: Dynamic buffer size changes only apply to IMS V10
    - V9/V8 still requires a SYSGEN to change link buffer sizes

To support the increased bandwidth enhancements, IMS V10 introduces several new command capabilities.

The update command has been enhanced to provide a mechanism to change the characteristics of the link associated with the name provided. The name is either the label value on the MSLINK definition or the default DFSLKxxx value. Possible actions that control the bandwidth capacity of the link include increasing and decreasing link buffer sizes, setting a higher limit for buffer size, and setting the bandwidth mode off and on. The command can be entered as either a Type2 (UPDATE) command which uses the Operations Manager (OM) or a Type1 (/UPD) which does not require OM. Any accepted changes are kept across a warm start.

If the characteristics of the link need to be changed, the link must first be stopped and idle. The UPDATE command, when issued, must also be issued on both IMS systems on either side of the link to reflect identical specifications for bandwidth mode and buffer sizes.

Increased Bandwidth - Commands …

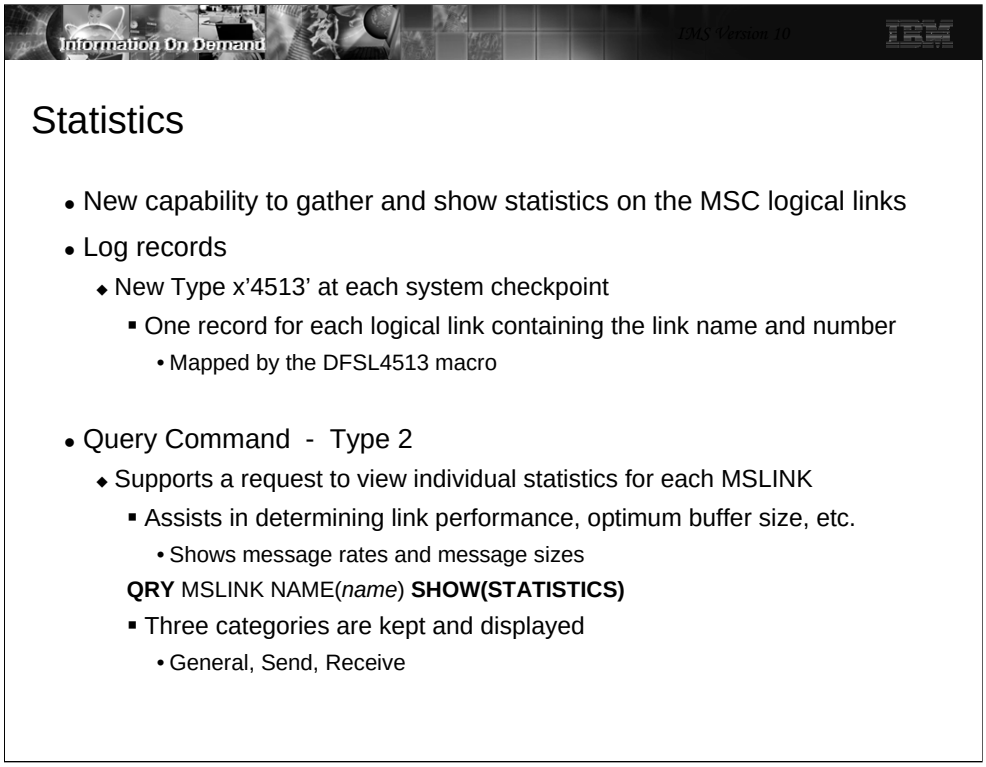
- Commands to show status and properties of the link
  - Display Command - Type 1

    /DISPLAY LINK *number*
    - Existing command that displays link values and BANDWIDTH status if bandwidth mode is set on

    /DISPLAY LINK *number* **OPTION BUFSIZE**
    - New format of the command that displays the link parameters such as link#, linkname, bandwidth, bufsize, and status
  - Query Command - Type 2

    QUERY MSLINK NAME*(name)* **SHOW(ALL | BUFSIZE |BANDWIDTH))**

The /DISPLAY and QUERY commands show the links with their current buffer size and whether or not bandwidth Mode is on or off.



IMS V10 also provides a new capability to gather and show statistics on the MSC logical links.

A new log type x'4513' is available to provide information on each logical link.

The QUERY command provides options to request statistics on the MSC links. This allows for quick and easy access to link performance. The information can be used to determine the efficiency of the link and assist in deciding on an optimum buffer size. Individual statistics for each MSLINK are collected in the DFSMSCWA workarea.

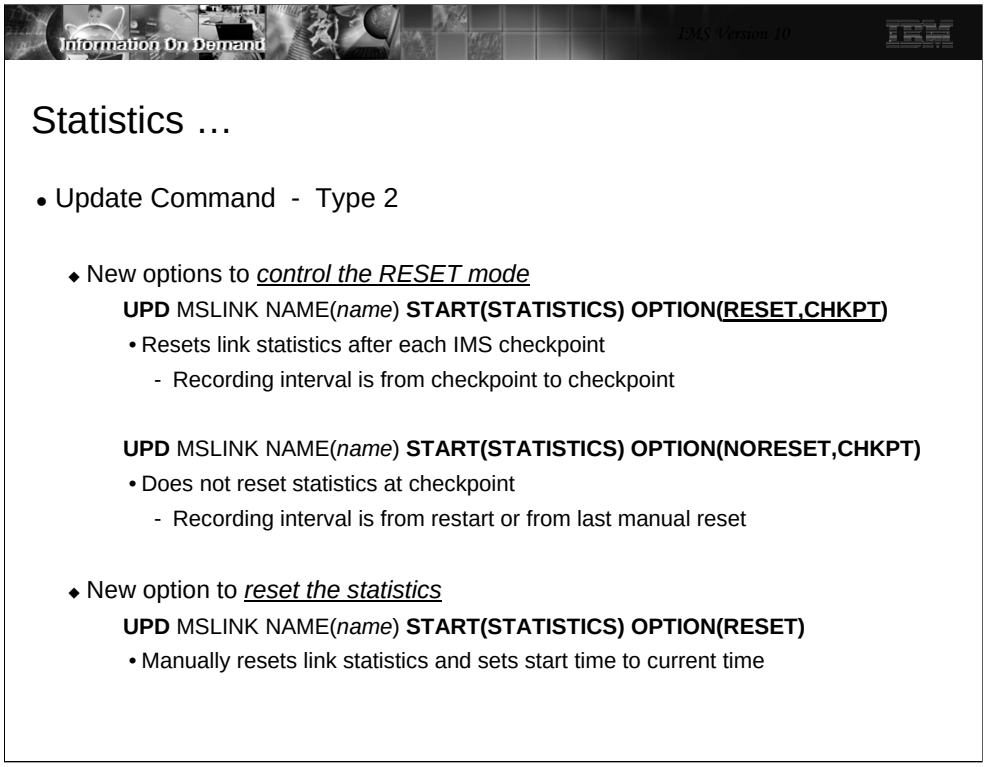
Note that link statistics are not displayed with the SHOW(ALL) keyword. SHOW(STATISTICS) must be used. This keyword also displays the statistics reset mode, RESET,CHKPT or NORESET,CHKPT.

There are three categories of statistics:

•General - such as, statistics start time, ITASK dispatch counts, ITASK processing times, and the rate and number of logger check writes

•Send - such as, messages sent, byte count sent, send message sizes, queue manager get counts and times, and send I/O times

•Receive - such as messages received, byte count received, receive message sizes, QMGR insert counts and times, and receive I/O times



The UPDATE MSLINK command can be invoked to reset the statistics counters and control how the resetting is to be done.

The first two forms of the command control the RESET mode:

•RESET,CHKPT mode causes the link statistics to be reset at each IMS checkpoint, after the statistics are logged. The recording interval, therefore, is from IMS checkpoint to checkpoint. This is the default mode and provides a reasonable interval for the statistics to be gathered. A longer interval might not be as useful in determining problems.

•NORESET,CHKPT mode does not reset the statistics at IMS checkpoint. To be reset, the operator must do so manually. The recording interval, therefore, is from IMS restart or when the last manual reset was issued until the time the command is issued. This mode is useful when running a benchmark or gathering statistics for a longer interval than between IMS checkpoints.

The third form of the command manually resets the link statistics, sets the start time to the current time, and begins a new recording interval.

Statistics …

- Statistics information

General Statistics

Start_Time	Start time for statistics
Tot_DISP_CT	Number of times link ITASK was dispatched by IMS
Tot_Proc_Time	Total link ITASK processing time for all dispatches
Hi_Proc_Time	Highest link processing time per dispatch
Low_Proc_Time	Lowest link ITASK processing time per dispatch
Avg_Proc_Time	Average link ITASK processing time per all dispatches
chkw_CT	Number of logger check writes
chkwio_CT	Number of logger check writes that caused an I/O
chkw_Rate	Logger check write rate calculated by calculating chkw_ct divided by the statistics recording time

The next two pages provide an overview of the statistics provided in the QUERY command. This visual shows the General Statistics.

Statistics …

- Statistics information …

Send Statistics		Receive Statistics	
Tot_Send_CT	Total # of send I/O	Tot_Rec_CT	Total # of receive I/O
Tot_MSG_Send_CT	Total # of msgs sent	Tot_Msg_Rec_CT	Total # of messages received
Tot_MsgByte_Send_CT	Total # of msg bytes sent	Tot_MsgByte_Rec_CT	Total # of msg bytes received
Tot_Byte_Send_CT	Total bytes of data sent	Tot_Byte_Rec_CT	Total bytes of data received
Hi_MSG_Send_SZ	Largest msg size sent	Hi_Msg_Rec_SZ	Largest msg size received
Low_MSG_Send_SZ	Smallest msg size sent	Low_Msg_Rec_SZ	Smallest msg size received
Avg_MSG_Send_SZ	Average msg size sent	Avg_Msg_Rec_SZ	Average msg size received
Send_Msg_Time	Interval bet. 1st & last msg	Rec_Msg_Time	Interval bet. 1st & last msg
Tot_Qget_CT	Total # of QMGR gets	Tot_Qput_CT	Total # of QMGR put calls
Tot_Qget_TM	Total tm to get msg off q	Tot_Qput_TM	Total proc tm for QMGR puts
Hi_Qget_Time	Longest QMGR tm to send	Hi_Qput_Time	Longest QMGR tm to rcv
Low_Qget_Time	Shortest QMGR tm to send	Low_Qput_Time	Shortest QMGR tm to rcv
Avg_Qget_Time	Average QMGR tm to send	Avg_Qput_Time	Average QMGR tm to rcv
Tot_SendIO_Time	Total I/O time to send msgs	Tot_RecIO_Time	Total I/O tm to rcv msgs
Hi_SendIO_Time	Longest send I/O time	Hi_RecIO_Time	Longest I/O tm to rcv msgs
Low_SendIO_Time	Shortest send I/O time	Low_RecIO_Time	Shortest tm to rcv msgs
SendIO_Req_Rate	Send I/O req. per second	RecIO_Req_Rate	Rcv I/O requests/sec
SendIO_Byte_Rate	Rate in bytes/sec sent	RecIO_Byte_Rate	Rate in bytes/sec received

This visual shows the type of information that can be analyzed by reviewing the Send and Receive statistics.

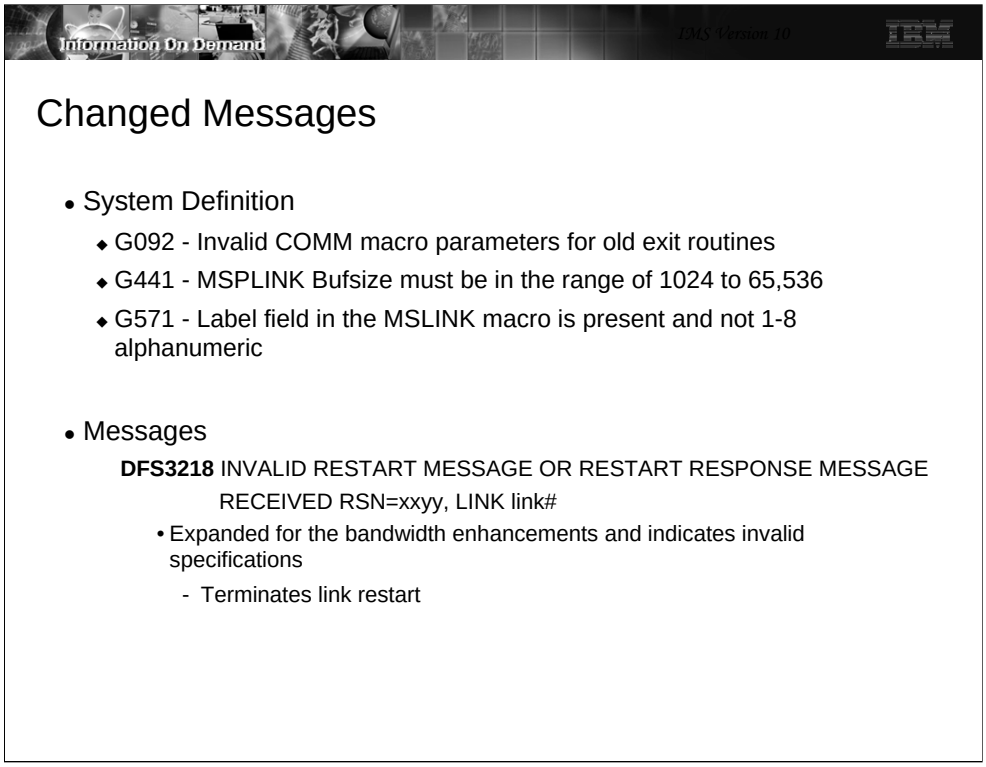
The Send statistics reference QGET calls. These calls are used to get messages off the queue for send processing.

The Receive statistics reference QPUT calls. These calls are used to put messages on the queue as part of receive processing.

Statistics - Report example

```
IMS1 side
 QUERY MSLINK NAME(LNK12V02) SHOW(STATISTICS)

 MSLink    MSLink#  MbrName   CC    Option            Start_Time
 LNK12V02  10       IMS1      0     NORESET,CHKPT     2006.261 19:03:58.77

 Tot_Disp_CT  Tot_Proc_Time  Hi_Proc_Time  Low_Proc_Time  Avg_Proc_Time  Chkw_CT
 98           0.176661       0.059530      0.000003       0.000853       33

 ChkwIO_CT  Chkw_Rate  Tot_Send_CT  Tot_Msg_Send_CT  Tot_MsgByte_Send_CT
 33         0.165016   26           20               10,461

 Tot_Byte_Send_CT  Hi_Msg_Send_SZ  Low_Msg_Send_SZ  Avg_Msg_Send_SZ  Send_Msg_Time
 22,403            578             502              523              2.408548

 Tot_Qget_CT  Tot_Qget_Time  Hi_Qget_Time Low_Qget_Time  Avg_Qget_Time  Tot_SendIO_Time
 51           0.002054       0.001070     0.000004       0.000040       0.505401

 Hi_SendIO_Time  Low_SendIO_Time  SendIO_Req_Rate  SendIO_Byte_Rate  Tot_Rec_CT
 0.193435        0.000708        51.444298        44,327            26

 Tot_Msg_Rec_CT  Tot_MsgByte_Rec_CT  Tot_Byte_Rec_CT  Hi_Msg_Rec_SZ  Low_Msg_Rec_SZ
 20              10,503              22,445           584            502

 Avg_Msg_Rec_SZ  Rec_Msg_Time  Tot_Qput_CT  Tot_Qput_Time  Hi_Qput_Time  Low_Qput_Time
 525             2.320062      40           0.037326       0.015957      0.000008

 Avg_Qput_Time  Tot_RecIO_Time  Hi_RecIO_Time
 0.000933       0.070947        0.025376

 Low_RecIO_Time  RecIO_Req_Rate  RecIO_Byte_Rate
 0.000155        316,363         148,040.09
```

When the QUERY MSLINK NAME(*name)* SHOW(STATISTICS) command is issued, all the statistics information on the link is displayed.



Several system definition messages have been changed to reflect the IMS V10 enhancements:

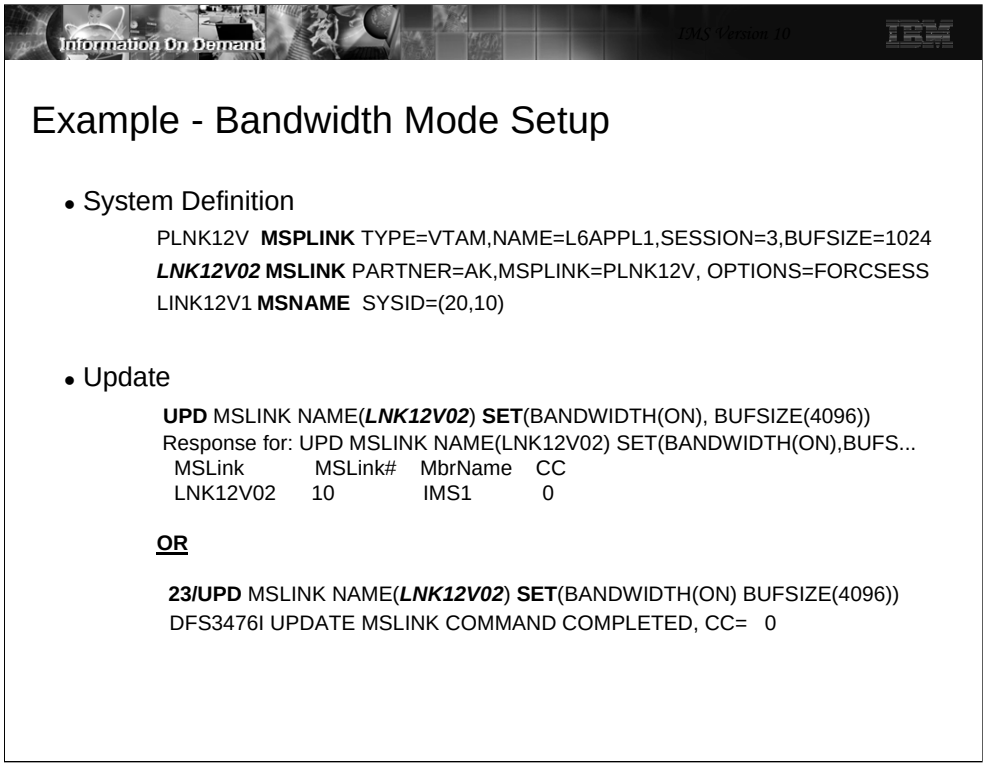
•G092 reflects the removal of the MSC exits on the COMM macro.

•G441 reflects the new BUFSIZE limits on the MSPLINK macro for V10 which must be in the range of 1024 to 65536 for all link types - CTC, MTM, and VTAM.

•G571 is issued if the specified MSLINK macro label field is not blank and is not a one-to-eight character alphanumeric name.

Additionally, Message DFS3218 has been expanded to support the bandwidth enhancements. When an MSC link is restarted, restart messages (referred to as a restart message and restart response message) are exchanged between the partner IMS systems. If the messages are invalid, unexpected, or contain incompatible information, a DFS3218 message will be issued on the IMS that detected the error, and the link restart will be aborted. Bandwidth support added more error conditions (e.g., Buffer sizes not equal, Bandwidth Mode not compatible on both sides). To more easily identify the error, a unique reason code was added for each error condition (for existing errors and new errors). The reason code identifies the module that detected the error and the error condition.

DFS3218 INVALID RESTART MESSAGE OR RESTART RESPONSE MESSAGE RECEIVED RSN=xxyy, LINK link#



These next two visuals provide an example of MSC setup.

During system definition, three macros provide the mechanism to define the connection between 2 IMS systems. The MSPLINK macro has not changed but the range in values that can be specified for BUFSIZE reflect the new minimum and maximum limits. The MSLINK has changed to provide a label. In this example, the name is *LNK12V02* and is important because it will be used in the UPDATE command.

Once the system definition process is complete and IMS is initialized, the parameters on the MSLINK can be modified. Note that in this example, the UPD command identifies the specific MSLINK named *LNK12V02.* Two changes are made. The first is the setting of BANDWIDTH Mode to ON and the second is the increase of the BUFSIZE to 4096. Alternatively, the command can be issued as a Type 1 command using the /UPD form.

Example - Bandwidth Mode Setup …

- Query

```
 QRY MSLINK NAME(LNK12V02) SHOW(ALL)
Response for: QRY MSLINK NAME(LNK12V02) SHOW(ALL)            More:

MSLink     MSLink# MbrName   CC MSPLink    CID         PID  RecdCnt  SentCnt
LNK12V02     10    IMS1        0  PLNK12V  00000000  AK    0            0

Bufsize Bandwidth LclStat
4096    ON          STOCOMM,IDLE,COLDSESS,FORCSESS,TRACE
```

- Display

```
87/DIS LINK 10
DFS000I    LINK PARTNER   RECD  ENQCT  DEQCT    QCT SENT
DFS000I    10    AK          0      0      0      0    0
DFS000I     PSTOPPED IDLE COLD FORCE TRA

88/DIS LINK 10 OPTION BUFSIZE
DFS000I    LINK LINKNAME     BUFSIZE   STATUS
DFS000I    10   LNK12V02     4096        PSTOPPED IDLE COLD TRACE
DFS000I     FORCE BANDWIDTH
```

Once the changes to the link have been made, either the QRY command or the /DIS command can be used to display the link and ensure that the changes have actually been made.

Performance results

Time in seconds to send 1000 queued messages,
Data length = 100 bytes + PFX

CTC Shared Queues

non-bandwidth mode 4k Time = 8.074355 sec
- bandwidth mode 4k Time = 5.279757 sec = 34% less
- bandwidth mode 8k Time = 5.036957 sec = 37% less
- bandwidth mode 16k Time = 4.588605 sec = 43% less
- bandwidth mode 32k Time = 4.47057 sec = 44% less

VTAM Shared Queues

non-bandwidth mode 4k Time = 7.78207 sec
- bandwidth mode 4k Time = 6.97356 sec = 10% less
- bandwidth mode 8k Time = 5.252516 sec = 32% less
- bandwidth mode 16k Time = 4.829623 sec = 37% less
- bandwidth mode 32k Time = 4.446261 sec = 42% less

NOTE:
These results are from a defined and controlled environment and are intended as an example and NOT to provide exact percentages

☆ Results vary based on:
- Buffer sizes, and
- Traffic (number of messages that are queued and ready to be sent)

This visual documents the results of a test comparing the use of Bandwidth Mode against a variety of buffer sizes. Note that the test results vary based on buffer size with a controlled message size and queue depth.

Migration Considerations

- DFSMSCE0
  - Ensure migration from the older routing exit routines DFSCMTR0, DFSNPRT0, DFSCMLR0, DFSCMPR0
- Bandwidth
  - Migrate using the default of non-bandwidth mode in IMS V10
    - Compatible with previous releases
  - After migration, bandwidth mode can be enabled using the UPD command
    - Requires both partners of the link to be IMS V10 systems
    - Adjust buffer sizes
      - Minimum of 4096
      - To be effective, the buffers should be large enough to accommodate multiple messages

When migrating to IMS V10, ensure that the older routing exit routines, if any, have been converted to DFSMSCE0. This migration can be accomplished in any of the supported previous releases (V7, V8, V9) prior to the actual IMS V10 migration. If the exit routines still exist while the V10 migration is in progress then the V10 migration tasks need to include the upgrade of the previous releases to DFSMSCE0.

A like-for-like migration from a previous release to IMS V10 allows MSC to be initialized in non-bandwidth mode. This mode is compatible with MSC operations in previous releases.

The ability to turn on bandwidth mode is provided via the UPD command and requires both partners of the link to be at an IMS V10 level. In this mode, the minimum size of the buffers should be 4096. To be more effective, the buffer sizes should be defined to accommodate multiple messages.

Migration Considerations …

- Buffer size
  - MSPLINK buffer size ranges have changed
    - Minimum is1024, maximum is 65,536
      - Increase over IMS V8/V9 definitions
      - VTAM sizes for IMS V10 can be any size and no longer follow an algorithm
    - Validate that partner IMS systems have matching buffer specifications
      - If the IMSV8/V9 systems have link buffer sizes less than 1024, they must be modified to a minimum of 1024

Note that the buffer size minimum and maximum ranges have changed in IMS V10. If the link is a VTAM link, the buffer size can now be any size defined in the range and no longer requires a size that fits into the previous formula of X times 2 to the power of Y. The MSPLINK macro description in the System Definition Guide still documents a table but only for compatibility purposes.

If the partner of the MSC link is a V8 or V9 IMS system then those definitions must be at a minimum of 1024.

Benefits

- DFSMSCE0
  - Provides a standard exit routine and consistent set of routing capabilities
- Bandwidth enhancements - Bandwidth Mode and buffer size
  - Provide greater control over the bandwidth capacity of MSC links
    - Allows for reduction in MSC parallel links to support throughput
    - Supports a dynamic display and change of link buffer size
  - Improve the efficiency and performance of the link protocols
  - Reduce logger CHKWs
- New QUERY Statistics
  - Provide the ability to fine tune and analyze MSC link performance
    - Quick and easy access to link statistics
      - Information that can assist in defining optimum buffer sizes

The use of the TM and MSC Message Routing and Control User Exit (DFSMSCE0) routine consolidates the older message routing exits and provides a variety of enhanced options. By reducing the number of exit routines, a more consistent set of routing capabilities for all types of messages is provided along with a possible simplification of the coding and maintenance requirements.

Bandwidth is the rate at which data or messages can be sent and received across the MSC links. It is dependent on many factors such as: line speeds, processor speeds and other current activity going on in the system, I/O activity for logging and writing messages to the message queues, as well as the choice of protocols and buffer sizes. The bandwidth enhancements in IMS V10 address these many factors by improving the efficiency of MSC processing across the three link protocols - CTC, MTM and VTAM. Additionally, the new capability to view link statistics in the QUERY command can help determine how well the link is performing. The information can be used to optimize the buffer sizes based on actual traffic as messages flow through the system. This combination of being able to view the statistics and then dynamically change the link buffer sizes allows for more responsive control of the bandwidth capacity of MSC links.

MSC VGR Support

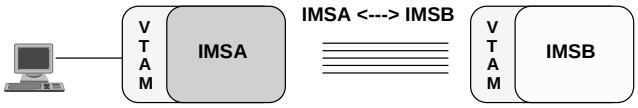
- Single MSC system image for the IMS instances in an IMSplex
  - Enhancement to the existing IMS VGR functionality
    - Access through MSC links using VTAM Generic Resources (VGR)
  - Initial implementation for MSC VGR
    - Local mode with VTAM-managed affinities
      - Local mode - No MSC data in CF, no support for the Resource Manager
    - Future - Global mode and RM

IMS already supports VGR for other terminal types including ISC. This function adds the MSC environment as a local mode VGR with VTAM-managed affinities.

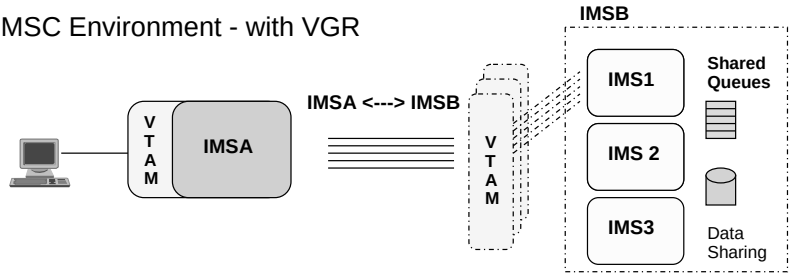
An affinity is a mapping of a node, and in the case of MSC all the parallel sessions associated with a link, to an IMS system.

MSC VGR Support …

- MSC Environment - without VGR



- MSC Environment - with VGR



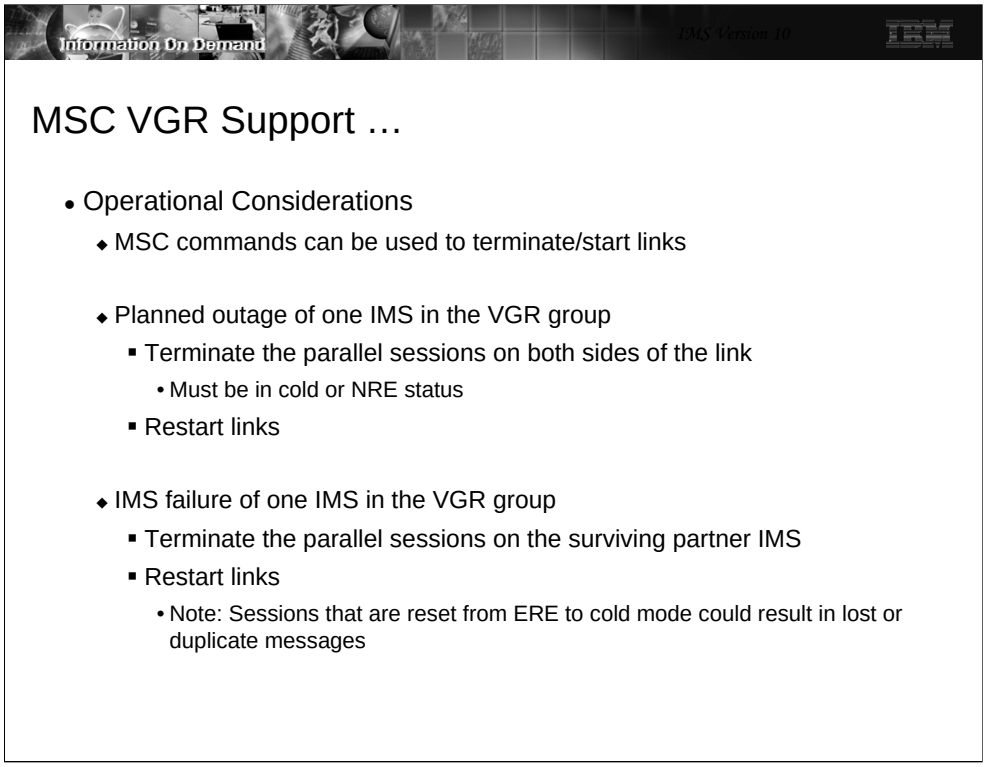
MSC VGR support provides the ability to grow the capacity and capabilities of the IMS MSC environment.

MSC VGR Support …

- Definitions
  - Remote IMS can communicate using a single MSC link
    - To any of the IMS systems in the VGR group
  - IMS systems that are part of the local VGR group
    - Can clone MSC link definitions for communicating with a single remote IMS
- Implementation
  - All parallel MSC sessions are routed to the same IMS

The use of MSC VGR eases the requirements on the definitions that are needed. A remote IMS can use a single set of MSC link definitions to access any of the IMS systems in the VGR group. Likewise, IMS systems that are part of the local VGR group can clone definitions for MSC access to the single remote IMS.

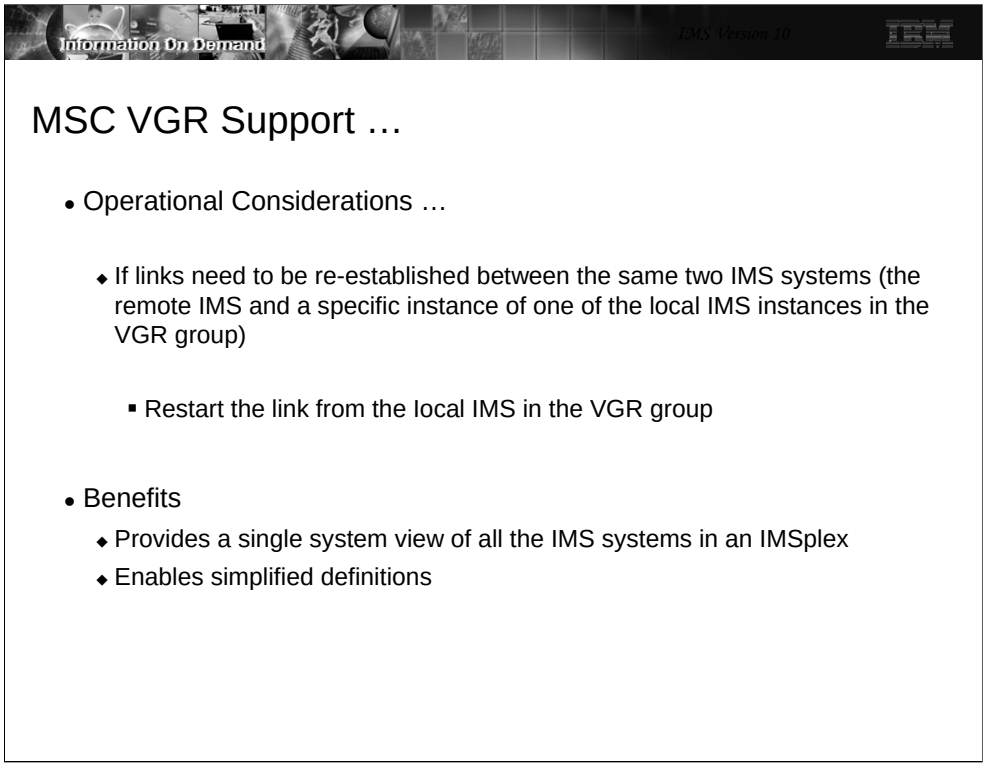
Note that when VGR is used, all parallel sessions associated with the link are routed to the same IMS.



MSC commands are available to terminate and restart the MSC links.

The information that MSC keeps about its sessions includes the link status, active or stopped, and the sequence numbers. If a planned outage is to occur, the parallel sessions on both sides of the link must be stopped and then restarted using the new link partner. To allow the information about sequence numbers to be cleared, the sessions need to be in cold or NRE status on both sides of the link.

On the other hand, if an IMS failure occurs where one IMS system terminates, the surviving IMS must also stop the parallel sessions. Note that if session status is changed from ERE to cold mode, messages could be duplicated or lost.



If the links need to be re-established between the same two instances of IMS, then a way to accomplish this would be to first start the links on the instance of the IMS in the VGR group where the connection is desired. This action establishes affinity for all parallel sessions which remains in effect through subsequent restarts and until the last parallel session is terminated. Once the affinities are established, the links can also be started on the remote IMS system.

APPC

Highlights

- Enhanced timeout granularity
- Support for /LOCK and /UNLOCK
  - From both APPC and OTMA clients
- Local LU Support

32

APPC enhancements include the following three areas:

•Greater level of timeout granularity with support for APPC/MVS timeout in seconds

•Ability for APPC and OTMA clients to issue the /LOCK and /UNLOCK commands

•Local LU Support

Enhanced Timeout Granularity

- Support for APPC/MVS timeout in seconds
  - Enhancement in z/OS V1R7
    - Prior releases provided the timeout capability in minutes
  - IMS enhancements to:
    - APPCIOT keyword in DFSDCxx member of IMS.PROCLIB
    - /CHANGE APPC TIMEOUT command

The APPC/MVS capability for timeout has been supported by IMS since V6. In z/OS V1.R7, APPC/MVS enhanced its timeout support to specify a lower level of granularity. Prior z/OS releases supported timeout in minutes with the shortest value being one minute. Even one minute, however, can be too long a time when resources are held. With IMS V10 and z/OS V1R7, the timeout capability as provided in the first parameter of the APPCIOT keyword and in the /CHANGE APPC TIMEOUT command allow a value that can be specified in seconds.

Providing the ability to set timeout in seconds is particularly helpful when slow downs in the network occur or when APPC clients that are unable to respond in a timely manner cause IMS dependent regions to hang. Another benefit is one that could affect command processing when a slow or non-responding client impacts the IMS command task DFSCMTI0. For example, when an APPC node sends input for a transction that is stopped, IMS replies to this condition by sending a DFS065 message. This error message is sent under the IMS command task which waits for a response from the APPC node. When this node fails to respond, the task hangs until the wait is broken by the APPC/MVS timeout facility. This task, however, is quite vital in the IMS environment and plays a major role in handling most commands. When the condition just described happens, a slowdown in IMS throughput could result. An APPC timeout value of 1 minute, therefore, could be too long when suspending the command task in a production environment.

Enhanced Timeout Granularity - Implementation

- APPCIOT
  - Previous specification:
    - APPCIOT=(mmmmA, mmmmB)

      where mmmmA = timeout value for APPC/MVS in minutes (0-1440)

      mmmmB = timeout value for APPC/IMS in minutes (0-1440)
  - New specification:
    - APPCIOT=(mmmmA:ssA,mmmmB)

      where mmmmA is 0-1440 and ssA= 0-59

      mmmmB is 0-1440

- /CHANGE APPC TIMEOUT
  - Previous specification: /CHANGE APPC TIMEOUT mmmm
  - New specification: /CHANGE APPC TIMEOUT mmmm:ss

The APPCIOT keyword is an existing keyword in the DFSDCxxx member of IMS.PROCLIB. The first parameter of APPCIOT specifies the APPC/MVS time-out value expressed in minutes. This refers to an IMS wait for a requested APPC/MVS service to complete. IMS takes advantage of an APPC/MVS service to enforce the timeout with valid values between 0 and 1440 minutes. The second parameter which was introduced in IMS V9 provides an APPC/IMS timeout. This refers to an APPC/MVS wait for an IMS process to complete. For example, the timer begins when IMS receives a message from APPC/MVS and is reset when another APPC verb is issued, e.g., send of a reply, error message or deallocate. This second value is similar to putting a timeout on a queue to queue response. Like the first parameter, valid values are between 0 and 1440 minutes.

IMS V10 provides an enhancement for the first parameter which affects the APPC/MVS timeout specification. A value in this parameter can now be specified as (# of minutes : # of seconds). Note that the second parameter remains unchanged.

Similarly, the /CHANGE APPC TIMEOUT command has also been enhanced to provide a specification in minutes and seconds. Additionally, since the command can be issued during IMS execution, the value can be dynamically change as needed.

Support for /LOCK and /UNLOCK

- New capability for APPC and OTMA clients to send in the /LOCK and /UNLOCK commands
  - Supports keywords DATABASE, PROGRAM and TRANSACTION
    - No support for keywords LTERM, PTERM and NODE
  - The /LOCK and /UNLOCK command functionality remains unchanged

The restriction to sending in the /LOCK and /UNLOCK commands from APPC and OTMA clients has been lifted. The support allows the DATABASE, PROGRAM, and TRANSACTION keywords but restricts the use of LTERM, PTERM and NODE. Note that the commands themselves are not changed.

Support for /LOCK and /UNLOCK …

- Implementation
  - APPC Client sends
    - ALLOCATE… TPN=*command*
      SEND *remainder of command*
      RECEIVE…
      - Where *command* can now include /LOCK or /UNLOCK
  - IMS responds with
    - DFS058 Command completed
  - Example

```
ALLOCATE LUNAME=IMSLU, TPN=/LOCK. …
SEND_DATA   DB database1
RCV_AND_WAIT                  <------DFS058 LOCK COMMAND COMPLETED
DEALLOCATE
```

This visual gives an example of an APPC client that specifies the /LOCK command as a TPNAME on the ALLOCATE request and the remainder of the command in the subsequent SEND_DATA verb. IMS responds with a DFS058 LOCK COMMAND COMPLETED which simply means that the command was accepted and processed by IMS. Note that the capability to send IMS commands from an APPC client is not new.

Support for /LOCK and /UNLOCK …

- Implementation …
  - OTMA environment supports remote clients
    - IMS Connect: LLLL llzzIRM llzz*command* EOM
      - Where the IRM header provides a command instead of a trancode
      - Where *command* can now include /LOCK or /UNLOCK
  - IMS responds with
    - DFS058 Command completed
  - Example

```
CONNECT
WRITE   LLLL llzz IRM (specify /LOCK in IRM_TRNCOD)  llzz /LOCK DB database1
EOM
READ                                  <------DFS058 LOCK COMMAND COMPLETED
DEALLOCATE
```

Similarly, OTMA clients such as IMS Connect and MQ can also send in the command requests. Each OTMA client provides its own interface for remote applications. The example on this visual shows a request coming in from a remote application through IMS Connect.

Local LU

- Enhancement that allows greater control when specifying which LU name is to be used for asynchronous outbound conversations
  - Allows IMS application to request any LOCAL LU name through a new descriptor keyword for ALTPCB requests
  - Allows incoming LU name (if not the BASE) to be used for outbound asynchronous responses to the IOPCB
  - Terminology
    - Base LU - primary and default LU name associated with APPC/IMS
      - Defined as such in the APPCPMxx member of SYS1.PROCLIB
    - Local LU - name of an alternate LU that can also be associated with an APPC/IMS system

The APPC/IMS support which was introduced in IMS V4 provided the ability to define both a primary VTAM APPC LU name to be associated with IMS as well as multiple secondary LU names. The primary name is defined as the BASE LU and the alternate names as LOCAL LUs.

A Base LU is specified as such in the APPCPMxx member of SYS1.PROCLIB which contains the APPC/MVS definitions. Local LUs must also be specified in the same member as being associated with APPC/IMS. Note that an LU must be unique to an IMS system in an IMSPlex.

IMS V10 enables IMS applications that use the alternate TP PCB (ALTPCB) for APPC asynchronous outbound requests to specify which LU name to use on the outbound ALLOCATE request through a new keyword in the DFS62DTx descriptor. Additionally, a new startup parameter APPCLLU provides the ability to specify whether or not the incoming LU name (if different from the Base LU) is to be used for any associated asynchronous IOPCB outbound conversations.

As a reminder, asynchronous APPC conversations are those that are sent without waiting for a reply. The verb set used is Allocate, Send_data, Deallocate. Synchronous conversations, on the other hand, are those that send a message and wait for a reply using the verb set Allocate, Send_data, Receive_and_wait, Deallocate. The Local LU support in this section applies only to asynchronous conversation requests.

Local LU …

- New OUTBND keyword in the LU62 descriptor DFS62DTX
  - Affects asynchronous outbound ALTPCB messages
    - Specifies the APPC LU to be used

```
• U L62A LUNAME=L62RMT1 TPNAME=MYTRAN MODE=L62MODE1
• U L62A SYNCLEVEL=C OUTBND=IMS62A
```

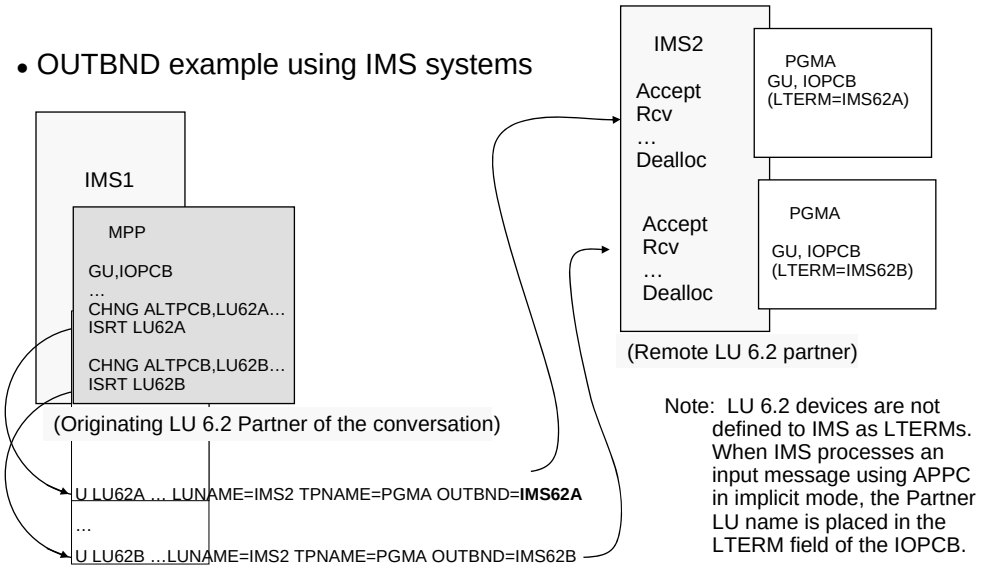
    - ALTPCB destination name = descriptor name
    - If OUTBND is not provided, the default Base LU name is used
- LU 6.2 Edit Exit Routine (DFSLUEE0)
  - Can change the Local LU name
    - Word 13 points to the name

A new OUTBND keyword has been added to the DFS62DTx IMS.PROCLIB member. IMS applications that produce ALTPCB messages control where and how the message is sent through specifications in an LU62 descriptor that is identified with the same name as the ALTPCB destination. IMS uses the information to create the appropriate APPC requests. If the OUTBND keyword is present and the Local LU specified is valid for this IMS then the outbound ALLOCATE is sent using the Local LU name. The partner APPC application sees the same name when it ACCEPTs the conversation. If the OUTBND keyword is not available, then the default Base LU name is used. Note: The DLI API has not been changed to add the OUTBND keyword to the CHNG call's LU 6.2 options.

The LU 6.2 Edit Exit Routine (DFSLUEE0), if it exists, is always called for inbound and outbound conversations managed by IMS. Word 13 in the exit interface points to the Local LU name and can be examined as well as modified by the exit code.

Local LU …

- OUTBND example using IMS systems



The example in this visual shows how specifying different Local LUs in the OUTBND parameter of two descriptors results in the remote partner seeing the different names. If the remote partner is IMS then the target application PGM1, in this example, sees a different LTERM name for each message.

Note that an APPC/IMS implicit environment is one where IMS functions as the LU 6.2 partner on behalf of the IMS application program. In this environment, IMS places the Partner LU name in the LTERM field of the IOPCB. A possible benefit of using the OUTBND capability is that the receiving IMS application can branch to different logic based on the LTERM name.

The remote partner can be any LU 6.2 partner.

Local LU …

- New APPCLLU startup parameter in DFSDCxxx
  - Affects asynchronous outbound IOPCB messages

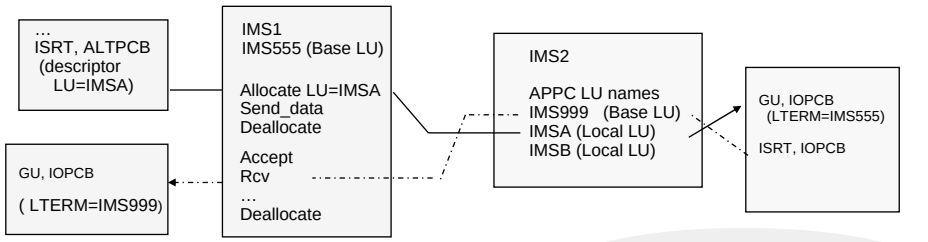
APPCLLU= Y | N

  - Y requests that if a Local LU was used for the inbound message that the same Local LU be used for outbound processing
  - N requests that the Base LU always be used

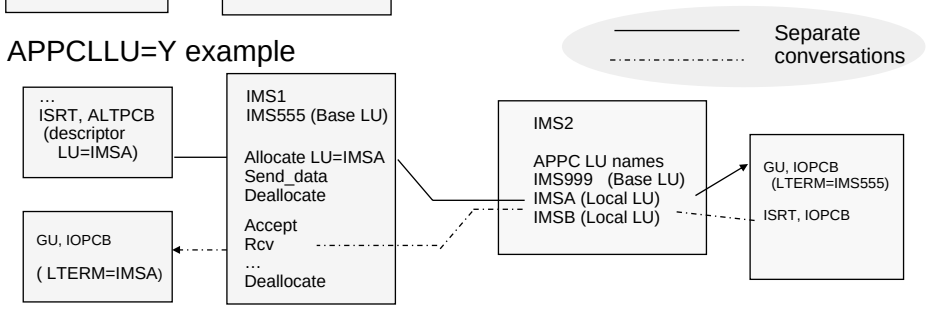
IMS V10 also provides a new startup parameter in the DFSDCxxx member of IMS.PROCLIB. APPCLLU affects asynchronous outbound messages that are inserted to the IOPCB. For this situation to occur, the message that the IMS application processed with a GU IOPCB had to originate in an APPC partner as an asynchronous inbound message.

Local LU …

- APPCLLU=N example
  - default, same as releases prior to IMS V10



- APPCLLU=Y example



The two examples on this visual illustrate the difference between the APPCLLU specifications of N and Y.

In the first example, APPCLLU=N, the IOPCB asynchronous output reply is sent using the Base LU name associated with IMS2 regardless of which LU name was used for the inbound asynchronous conversation.

In the second example, APPCLLU=Y, IMS2 uses the Local LU name of IMSA which is the same name used on the inbound asynchronous request. This capability is of value when the remote application is designed to expect a response from a specific partner LU name.

Local LU …

- Commands that have been enhanced to support Local LU
  - /CHANGE DESC *descriptor* OUTBND *locallu*
    - Modifies the descriptor to use the specified *locallu* name for any new outbound asynchronous conversations associated with the *descriptor*
  - /DISPLAY DESC *descriptor* | ALL
    - Output includes Local LU names

```
DESC    LUNAME      MODE     SIDE      SYNCLEVEL      TYPE    OUTBNDLU
L62A     IMS2      L62MDE02             CONFIRM      MAPPED    IMS62A
    TPNAME : PGMA
L62B     IMS2      L62MDE02             CONFIRM      MAPPED    IMS62B
    TPNAME : PGMA
L62TM    L62RMT1   L62MDE09             CONFIRM      MAPPED
    TPNAME : RMTPGM
```

Will use the Base LU

Some other enhancements provide support for the Local LU capability.

The /CHANGE DESC *descriptor* OUTBND *locallu* command modifies the descriptor to use the specified *locallu* name for any new outbound asynchronous conversations associated with the specified descriptor. Note that this impacts messages that use the descriptor after the command has been issued. Messages that are already on the queue but have not yet been sent are not affected by the change.

The /DISPLAY DESC command has also been enhanced to support the Local LU. If the OUTBND keyword exists in a descriptor then its value is displayed under the OUTBNDLU column. If the value for a descriptor displays as blanks then the Base LU will be used.

Local LU …

- Considerations
  - Remote partner receives the LU name used by IMS
    - Either the Base or Local LU
  - APPC LU names used by IMS must be unique within the IMSplex
  - Descriptors modified by the /CHA command affect new messages created after the change
  - Shared queues group
    - IMS V9 systems use the Base LU

Some areas to consider include the following:

Remote partners that receive an asynchronous conversation request from IMS have the ability to determine the LU name that IMS uses when sending a request. The Local LU capability provides the opportunity for different names to be provided.

Note that LU names are specific to IMS systems and cannot be shared within IMSplex members.

Any modifications that occur as a result of a /CHA command affect new messages that are created after the change.

Local LU Security

- RACROUTE VERIFY
  - IMS V10 Passes the applicable LU (Local or Base) as the invoking application to the security product
  - Previous releases
    - Always passed the Base LU name even if the request came in through a Local LU name
      - Some transactions could pass APPC/MVS security and then fail in IMS with a security violation
        - Userid security access can differ based on LU names

When an inbound conversation in a secured environment is allocated with APPC/MVS, a security check is done using the target LU name for IMS (Base or Local). If authorization is granted, APPC/MVS passes the RACF object to IMS. The RACF object, similar to an ACEE, is used by the IMS security call to check if the userid is authorized to the transaction before scheduling the request. In the situation where the message was sent to IMS using a Local LU, the Local LU name is used. Once this is done, the RACF object is deleted. If RACF=FULL has been specified for APPC/IMS, the ACEE must again be built for the dependent region. Prior to IMS V10, the ACEE is always built using the Base LU. If a mismatch occurs because a secured message was sent in using a Local LU but the dependent region ACEE, using the Base LU, does not authorize the user, then the queued transaction will fail authorization. IMS V10 addresses this situation by using the applicable LU (Base or Local) that was used for the inbound message when building the dependent region ACEE.

Benefits

- Enhanced timeout capability
  - Releases resources more responsively
    - In less than one minute when needed
- Support for /LOCK and /UNLOCK
  - Enhances command capability for APPC clients such as the IMS Command Control Facility
    - Implemented for OTMA clients also
- Local LU capability
  - Provides greater control of the LU names that IMS uses for asynchronous outbound requests
  - Enhances the security environment

The enhanced timeout granularity that provides values in the seconds range allows resources to be released more quickly when needed. The APPCIOT specification which is set at IMS initialization can be overriden dynamically by the /CHA command.

The support for /LOCK and /UNLOCK commands enhances the ability for tools such as the IMS Command Control Facility (CCF) which uses APPC to send commands to IMS.

The Local LU enhancement provides greater flexibility for the IMS environment to support multiple LU names. This is beneficial for application designs that rely on different partner names to trigger different events.

Removal of BTAM Support

Highlights

- IBM BTAM products were withdrawn from service several years ago
  - IMS continued to support BTAM through IMS V9
- IMS V10 removes BTAM support
  - Ignores all macro statements associated with the unsupported BTAM terminals during IMS system definition
    - Issues warning message

      G411 MACRO STATEMENT ASSOCIATED WITH AN UNSUPPORTED BTAM TERMINAL

      - A severity code of 2 is issued to allow system definition to continue
  - Devices such as Spool, Reader, Printer, Punch, Tape and Disk are not affected.

Although IBM withdrew marketing and service of BTAM products several years ago, IMS continued to support the BTAM macros through IMS V9. IMS V10 removes this support. The following list shows the device types affected:

BTAM Device Type	Comments or Other Specifications
1050	Switched Terminal
2740	Non-Station-Control
2740	Non-switched, model 1
2740	Non-switched, model 2
2740	Switched Terminal, model 1
2741	Non-switched
2741	Switched Terminal
2260	Local
2780	
3270	Remote, Non-switched
3270	Local
3270	Switched Terminal
3275	Switched Terminal
3741	Switched Terminal
SYSTEM/3	
SYSTEM/7	BSC, BSC and Contention
SYSTEM/7	Start/Stop, Start/Stop and Contention

Warning message G411 will be issued if the macro statement operand has an unsupported BTAM terminal specification during the IMS STAGE 1 system definition process. In addition, a severity code of 2 will be issued to allow system definition to continue. This warning message and severity code will be documented in the IMS V10 Messages and Codes manual.

An Example

```
*
*      NONSWITCHED AND POINT-TO-POINT 2740 LINE GROUPS
*
       SPACE 2
       LINEGRP DDNAME=DD2740S,UNITYPE=2740
         LINE ADDR=0B2
        TERMINAL ADDR=C6,OPTIONS=MFS
         NAME CTRL,HOWARD
        TERMINAL ADDR=E2,OPTIONS=MFS
         NAME LARRY
       LINE ADDR=0B3,MODEL=(2,440)
        TERMINAL ADDR=45,BUFSIZE=248,OPTIONS=MFS
         NAME MODEL2
        TERMINAL ADDR=46,OPTIONS=MFS
         NAME MODEL2K
       SPACE
```

```
178        LINEGRP DDNAME=DD2740S,UNITYPE=2740
179+*  GOTO   LEVPLAS; /*
180+        PUSH  PRINT
181+        PRINT OFF
197+        POP   PRINT
198+**/
199+*LEVPLAS: ;
200+*DECLARE DFSCID1 CHAR EXTERNAL  ;
201+*DECLARE DFSLEV  CHAR EXTERNAL  ;
202+*DFSCID1 = '5655J5400' ;
203+*DFSLEV  = '1010' ;
206+*,CHANGEID = 1 FOR PID-1010 IN LINEGRP
208+*,CHANGEID = 2 FOR UP10L1JW IN LINEGRP
175 *       NONSWITCHED AND POINT-TO-POINT 2740 LINE GROUPS
210+*,CHANGEID = 3 FOR UP10L1SF IN LINEGRP
213+    2,G411 MACRO STATEMENT ASSOCIATED WITH AN
214+    2,     UNSUPPORTED BTAM TERMINAL.
215        LINE ADDR=0B2
```

This example shows the warning message that is issued when attempting to run the system definition process specifying an unsupported BTAM device.

Migration Considerations

- Affected macros that previously supported BTAM include:
  - CONFIG
  - CTLUNIT
  - DCLIST
  - IDLIST
  - LINE
  - LINEGRP
  - MSPLINK
  - NAME
  - POOL
  - STATION
  - SUBPOOL
  - TERMINAL
  - TYPE

This page lists the different IMS macros that previously supported BTAM specifications.