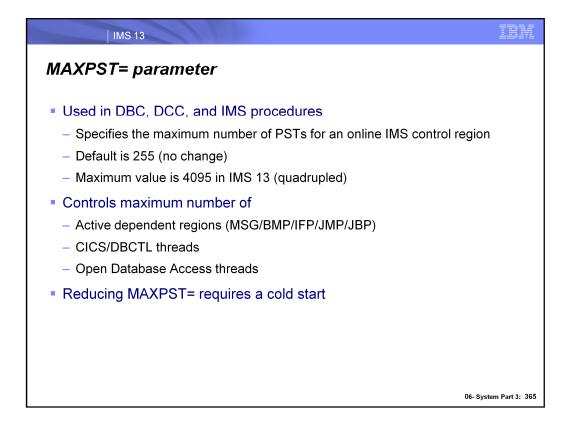
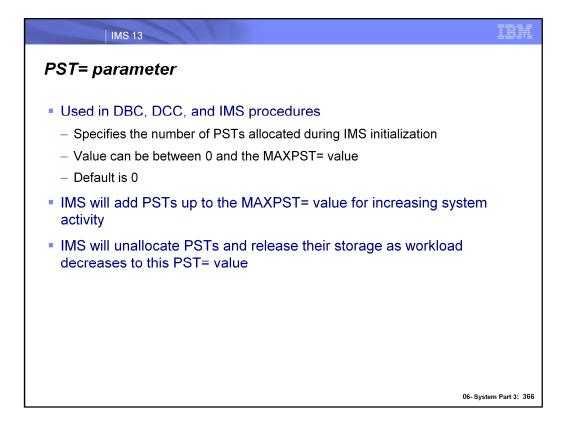
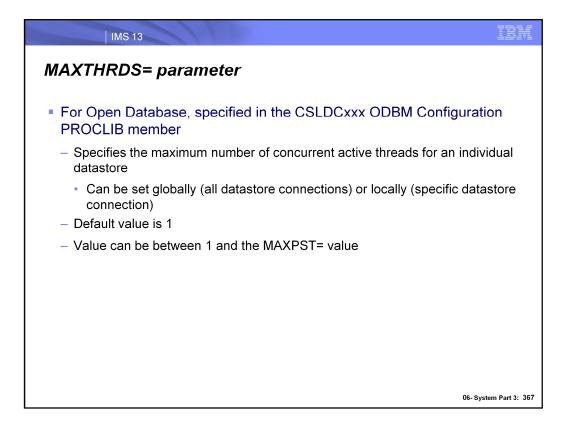


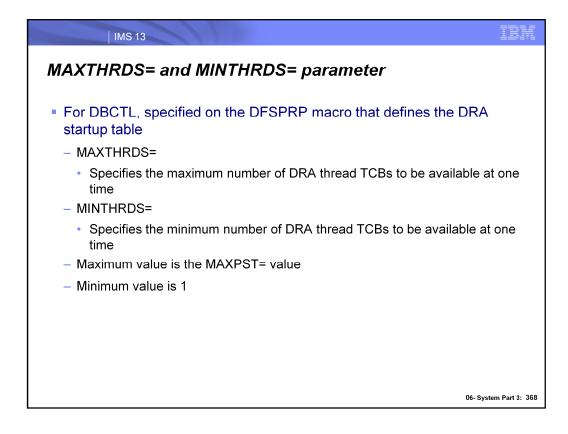
IMS systems continue to grow, needing more dependent regions to process new workloads associated with usage of open database, internet connectivity options, and applications with longer dependent region residency times.

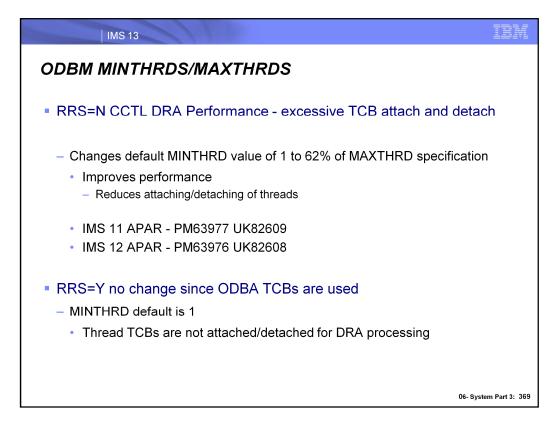
IMS 13 has quadrupled the maximum number of PSTs (Partition Specification Table) that can run concurrently in a particular IMS image.





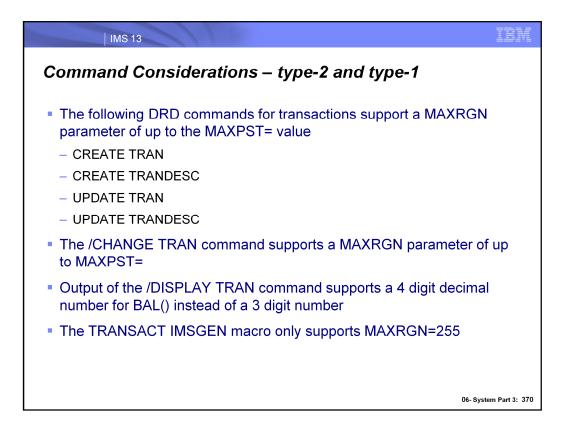


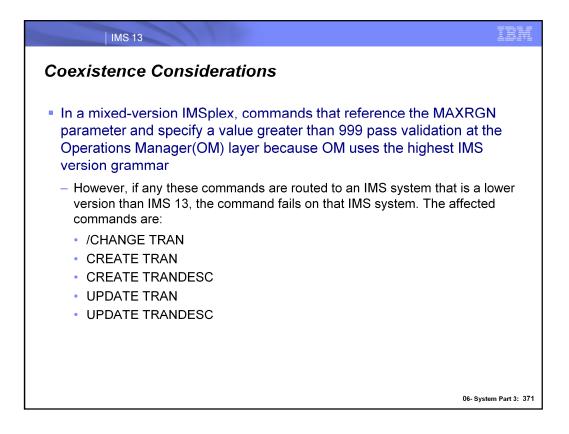


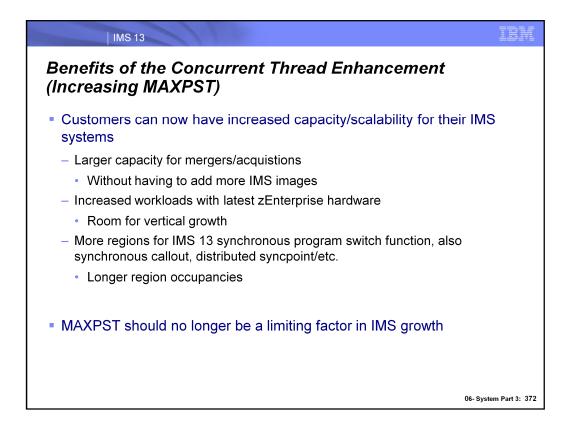


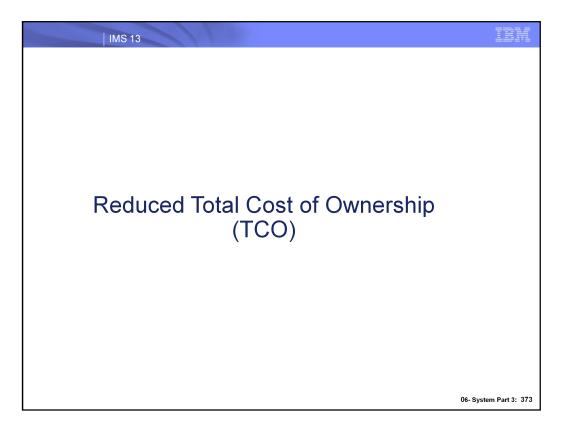
The default MINTHRD value of 1, used in an RRS=N environment, can result in an excessive number of attach and detach processes, which can result in performance degradation. With this enhancement, the MINTHRD value in an RRS=N ODBM environment is set to approximately 62% of the value of MAXTHRD. By establishing a higher minimum number of threads, the performance issues associated with attaching and detaching threads are reduced.

ODBM uses ODBA in the RRS=Y case so thread TCBs are not attached or detached for the dra processing. MINTHRD is still 1 for ODBA.

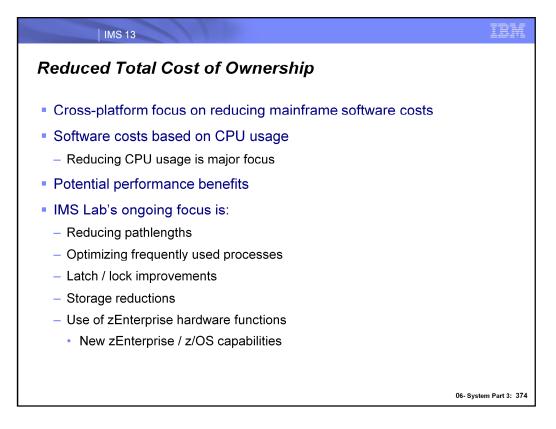




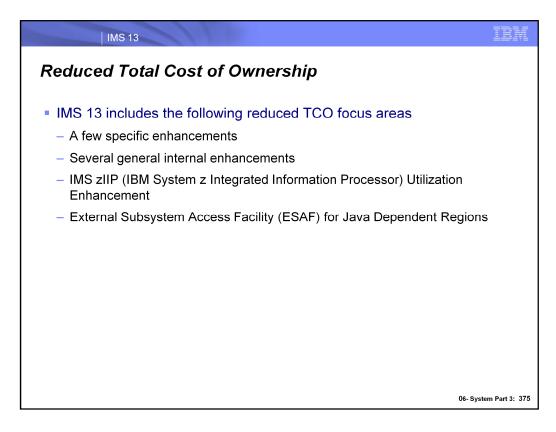




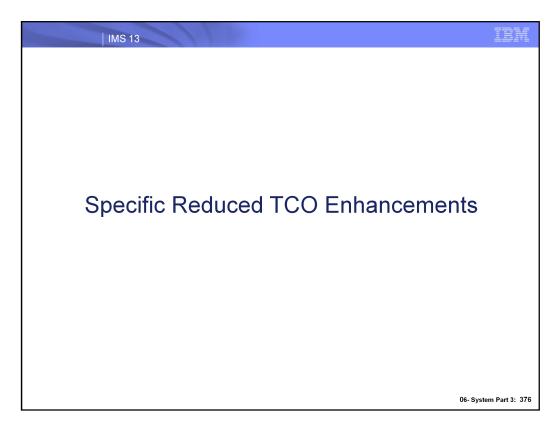
Reducing total cost of ownership for IMS users is a major focus in IMS 13.



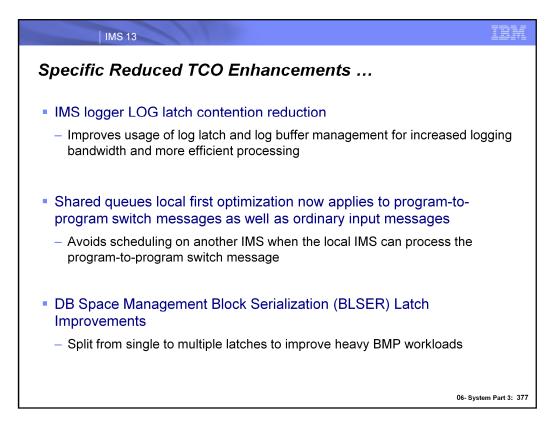
Reducing the costs of running on the mainframe platform is a major, continuing focus for all IBM products on this platform.



Reduced TCO enhancements can be grouped into four categories above.



This section includes a few more important enhancements.



IMS logger LOG latch contention reduction

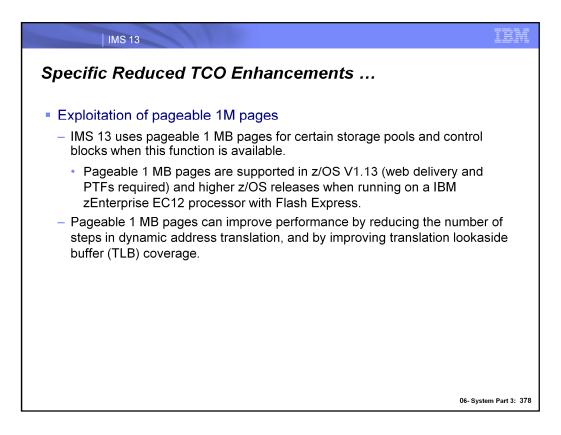
Change the IMS logger so that log data can be moved to a buffer without requiring an exclusive latch, to reduce contention on high activity systems.

Shared queues local first optimization now applies to PTP switch messages as well as ordinary input messages

Transaction messages inserted by an application program now are considered for local first processing, subject to the same rules as transaction messages that originate from the network. This can improve performance by avoiding scheduling for a specific transaction instance on other IMSs in the same shared queues group when the message can be processed by the local IMS.

DB Space Management Block Serialization ("BLSER") Latch Improvements

The DB block serialization latch has been split from a single latch to multiple latches. Normally, this latch is not a problem; however, it can be when there is heavy BMP activity (multiple concurrently-executing BMP jobs). This change reduces contention for the latch and can help reduce BMP elapsed time.



Exploitation of pageable 1M pages

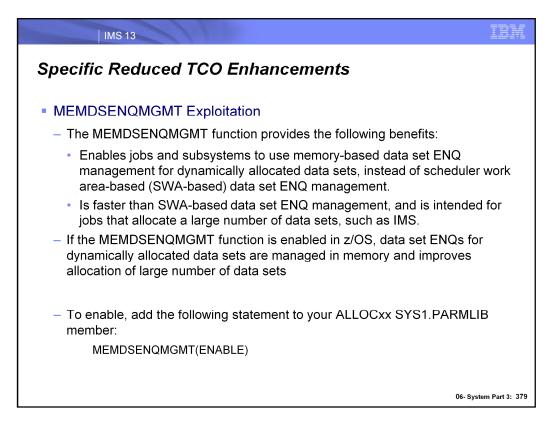
On zEC12 machines with Flash Express, z/OS will support 31-bit storage backed by 1M large real pages. Large page-backed storage improves performance by reducing the number of steps in dynamic address translation, and by improving translation lookaside buffer (TLB) coverage.

IMS 13 requests the following areas to be backed by pageable large pages, when available:

CQS interface buffers CQSPUT data buffers* Several highly-used internal control blocks in the CQS address space* DPSB pool (DLI/SAS PSB pool) ** DLDP pool (DMB pool) ** DBWP pool (DMB work pool) **

* Only when running on z/OS 2.1 or higher.

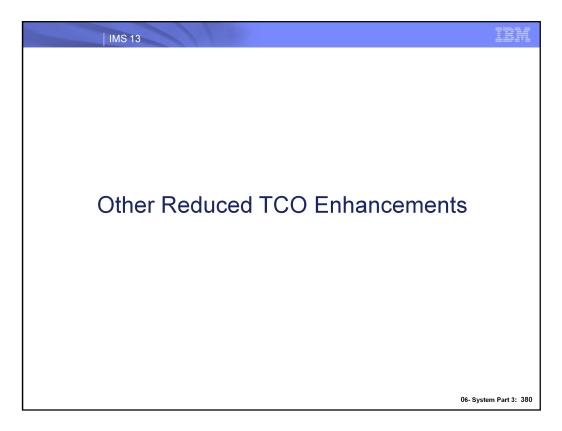
** If you page fix these pools, then they will not be backed by 1M pages unless running on z/OS 2.1 or higher.



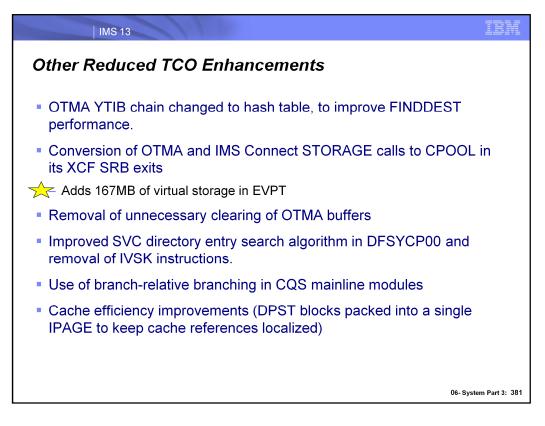
MEMDSENQMGMT Exploitation

IMS V13 issues an IEFDDSRV DSENQMGMT=MEMORY call from both control region and DLI/SAS to request memory-based data set ENQ management when using z/OS 1.12 or later. Memory-based management can be more efficient for jobs that allocate a large number of data sets. To have this change be effective, add the following statement in your ALLOCxx SYS1.PARMLIB member:

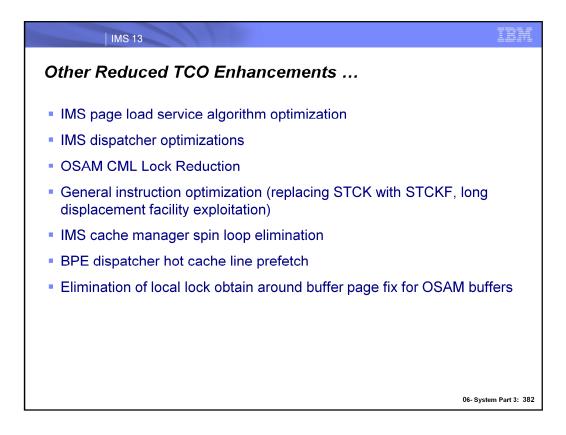
MEMDSENQMGMT(ENABLE)

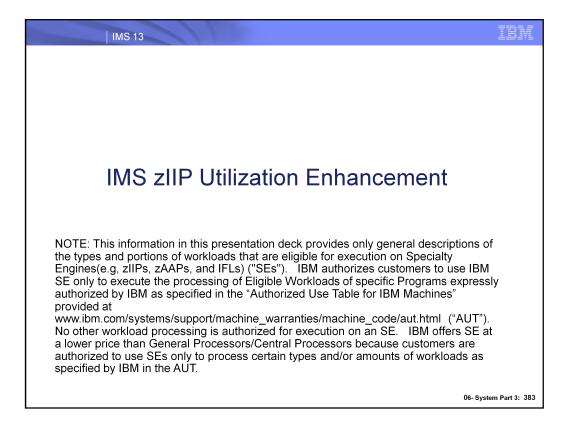


This section includes several smaller IMS internal changes.

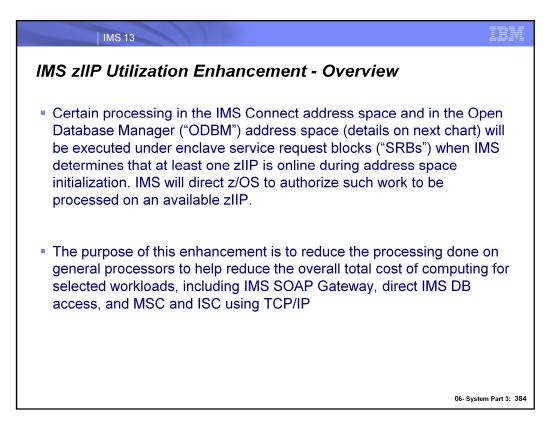


These enhancements are across many components of IMS, showing the reduced TCO focus for the whole IMS product.

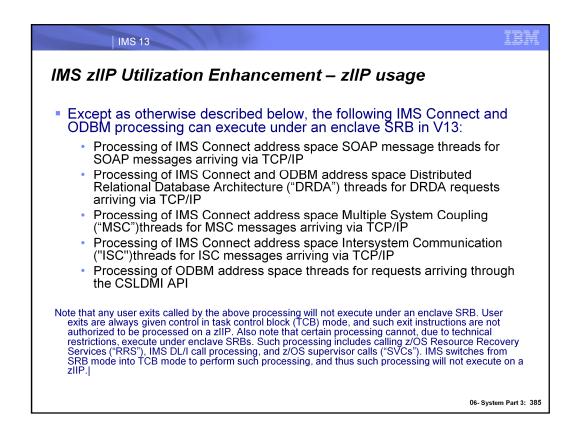




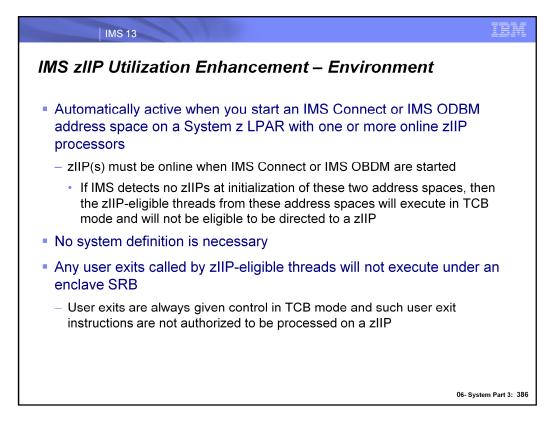
In IMS 13, designated areas of IMS processing will exploit IBM System z Integrated Information Processors (zIIPs).



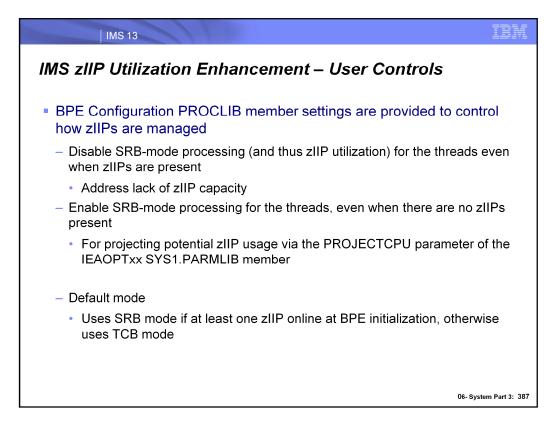
IMS 13 exploits using IBM System z Integrated Information Processors (zIIPs) as part of its overall focus on reducing total cost of ownership (TCO).



Listed above are the selected types of IMS processing that may exploit zIIP processors.



Usage of zIIP for these selected IMS functions is automatic if the environment described above exists.



Additional information on the PROJECTCPU capability is available in the MVS Initialization and Tuning Reference, SA22-7592-23.

IMS 13
IMS zIIP Utilization Enhancement – User Controls (Disable)
Disabling SRB-mode processing and zIIP utilization
 Create a new or update an existing BPE Configuration PROCLIB member for the IMS Connect or the IMS ODBM address space
 For IMS Connect, add the following statement to its BPE Configuration PROCLIB member:
CONDSRB(NEVER, HWS)
 For IMS ODBM, add the following statement to its BPE Configuration PROCLIB member
CONDSRB(NEVER,ODBM)
 Start the IMS Connect or IMS ODBM address space with the BPECFG= parameter in the PARMS= parameter string of the address space startup JCL, specifying the name of the created or updated BPE Configuration PROCLIB member
06- System Part 3: 388

Running on a zIIP in SRB mode can reduce software license charges. However, SRB-mode execution is more restrictive than TCB mode. There can be additional processing overhead involved if a thread running in SRB mode has to switch back to TCB mode to perform some processing that is not supported in SRB mode. Additionally, if you have IIPHONORPRIORITY=NO specified in the z/OS IEAOPTxx SYS1.PARMLIB member, ready zIIP-eligible work will wait for a zIIP processor when one is not immediately available, even if there is a standard processor available that could process the work.

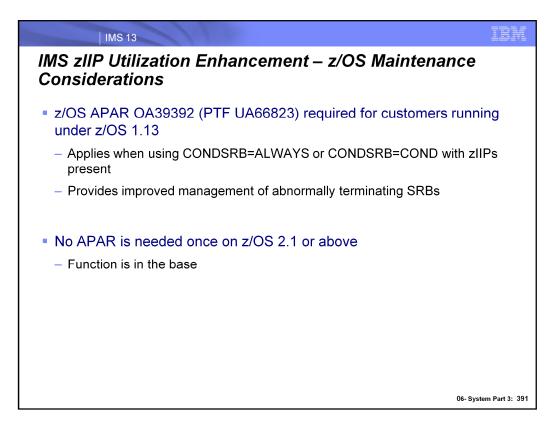
Thus, you can use the NEVER setting of CONDSRB to avoid SRB-TCB mode switching overhead, and to force work that could have been zIIP-eligible to run instead on standard processors. No execution on zIIPs will occur for the specified IMS component (i.e., IMS Connect or ODBM).

For additional information about the IIPHONORPRIORITY parameter, see the z/OS manual MVS Initialization and Tuning Reference, SA22-7592.

IMS 13
IMS zIIP Utilization Enhancement – User Controls (Enable)
Enabling SRB-mode Processing
 Create a new or update an existing BPE Configuration PROCLIB member for the IMS Connect or the IMS ODBM address space
 For IMS Connect, add the following statement to its BPE Configuration PROCLIB member:
CONDSRB(ALWAYS,HWS)
 For IMS ODBM, add the following statement to its BPE Configuration PROCLIB member
CONDSRB(ALWAYS,ODBM)
 Start the IMS Connect or IMS ODBM address space with the BPECFG= parameter in the PARMS= parameter string of the address space startup JCL, specifying the name of the created or updated BPE Configuration PROCLIB member
06- System Part 3: 389

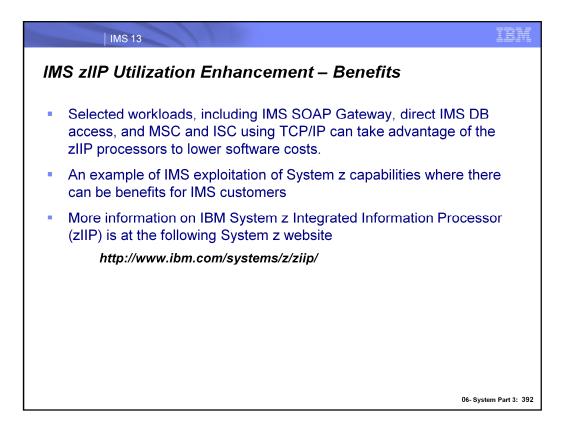
If you do not have a zIIP available, you still might want to run the threads in SRB mode. z/OS provides the ability to project the amount of CPU you could have executed on a zIIP had one been installed. You request this projection via the PROJECTCPU parameter of the IEAOPTxx SYS1.PARMLIB member (see MVS Initialization and Tuning Reference, SA22-7592-23). For this projection to be effective, you must also tell BPE to run the threads in SRB mode always, even when there is no zIIP online.

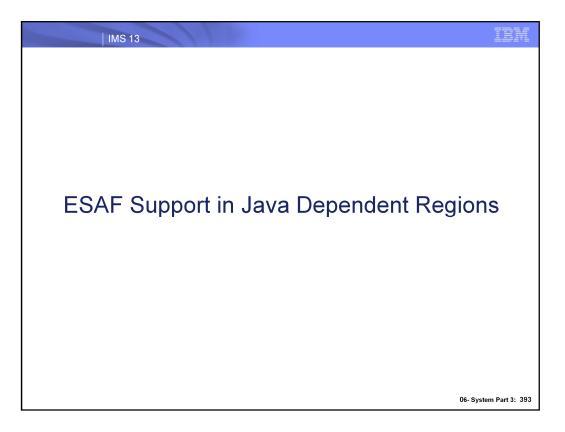
IMS 13
IMS zIIP Utilization Enhancement – User Controls (Default)
 Default mode
 No CONDSRB statement coded in the BPE Configuration PROCLIB member
– Or
CONDSRB(COND,)
 Uses SRB mode if at least one zIIP online at BPE initialization, otherwise uses TCB mode
06- System Part 3: 390



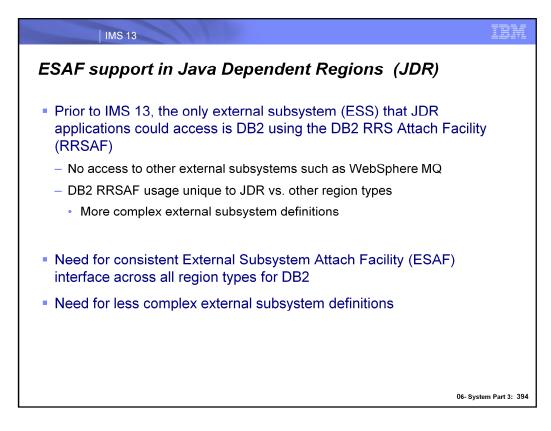
z/OS APAR OA39392 (PTF UA66823) is strongly recommended for z/OS 1.13 when executing the IMS Connect or IMS ODBM address spaces in an environment where BPE conditional SRB threads are run in SRB mode (i.e., when CONDSRB=COND in the BPE configuration PROCLIB member and one or more zIIPs is present, or when CONDSRB=ALWAYS). APAR OA39392 provides a new API for abnormally terminating SRBs. BPE SRB support internally makes use of this API during abnormal termination to attempt to ensure that all BPE SRBs terminate. The APAR is only needed on z/OS 1.13; later versions of z/OS have this support in the base.

Note that it is possible to run with BPE conditional SRB threads in SRB mode without the SRB abnormal termination API provided by OA39392. However, if a conditional SRB thread is not able to terminate normally (perhaps because it is in an infinite loop), the BPE address space may hang during termination waiting for the SRB, and may have to be FORCED to cause it to exit from the system. Forcing an address space can prevent cleanup of system resources, which may require a z/OS IPL to correct. For this reason, IBM recommends that you specify CONDSRB=NEVER for any IMS V13 IMS Connect or IMS ODBM address spaces that execute on a z/OS without SRB abnormal termination support.



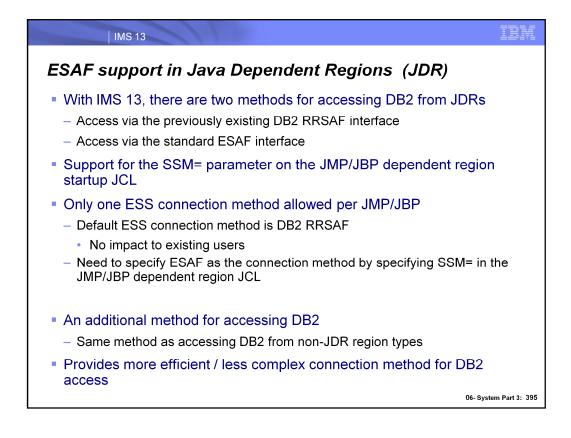


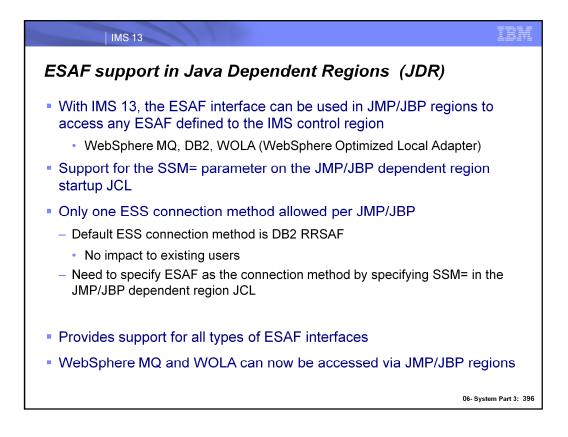
This section discusses new support for using the ESAF (External Subsystem Access Facility) in Java dependent regions.

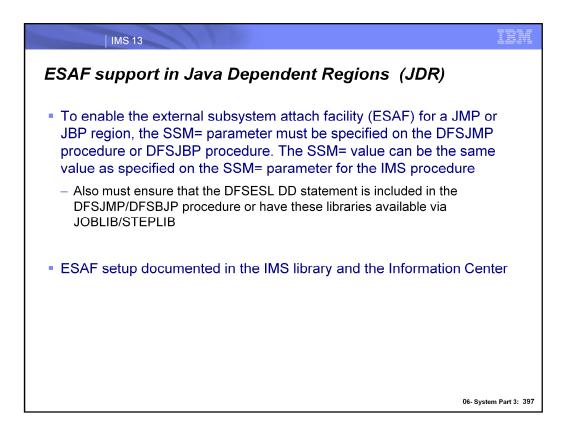


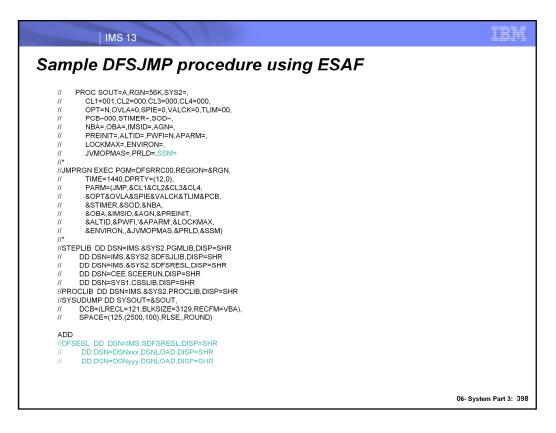
There are two enhancements for support of ESAF in Java Dependent Regions:

- 1. new method for DB2 access
- 2. access to any ESAF

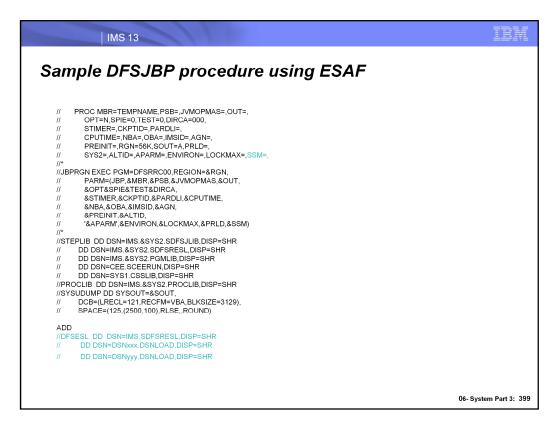




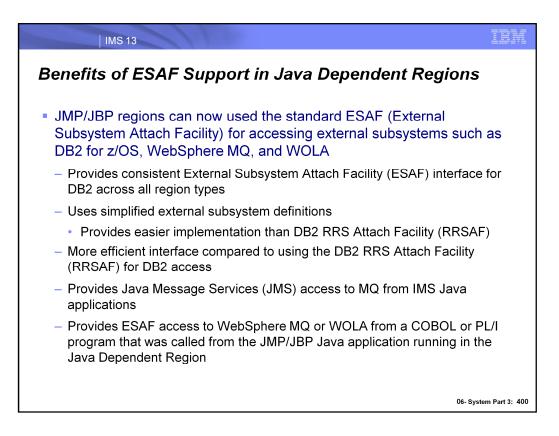


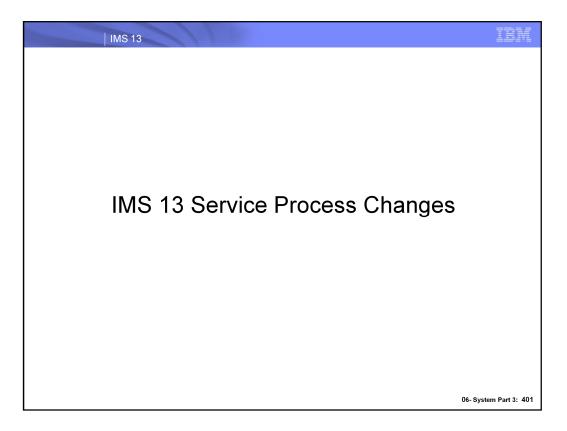


Here is a sample DFSJMP procedure for using ESAF.

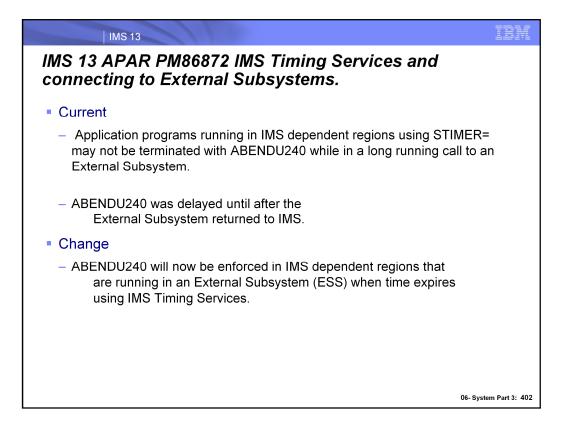


Here is a sample DFSJBP procedure for using ESAF.





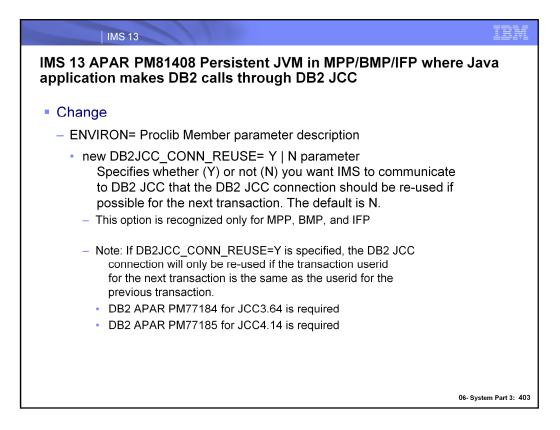
This section discusses new support for using the ESAF (External Subsystem Access Facility) in Java dependent regions.



IMS timing is established in BMP regions using CPUTIME= and the STIMEr= parameters on the BMP startup. MPP regions estalish timing using the STIMER= parameter on the MPP startup and the PROCLIM=parameter of the TRANSACT macro for the transaction being processed.

Prior to this change, ABENDU240 was delayed until after the External Subsystem returned to IMS. Now when the timeout routines detect processing in an ESS, they will schedule the ABENDU240.

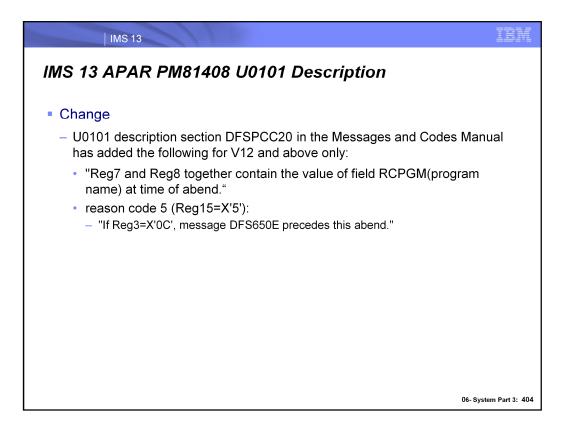
It should be noted: IMS establishes a time limit using the STIMER macro and the TASK parameter to indicate that the time interval is only decreased when the associated task is running. Any ESS that WAITs or SUSPENDs in the IMS dependent region will therefore not decrease the time interval during these events and will not drive STIMER exit routines to issue abends.



The new DB2JCC_CONN_REUSE= Y | N specified whether (Y) or (N) you want IMS to communicate to DB2 JCC that the DB2 JCC connection should be re-used if possible for the next transaction. The default is N.

This option is recognized only for MPP, BMP, and IFP regions.

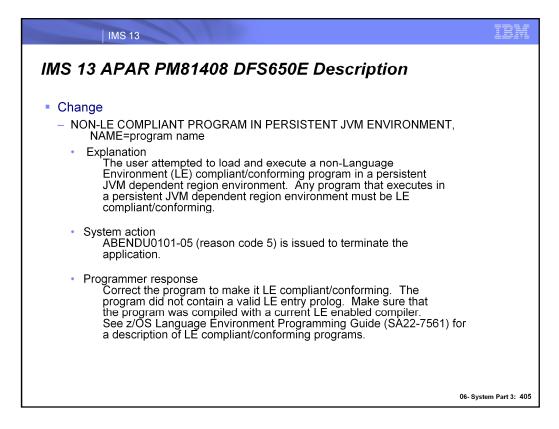
Note: if If DB2JCC_CONN_REUSE=Y is specified, the DB2 JCC connection will only be re-used if the transaction userid for the next transaction is the same as the userid for the previous transaction.



The Messages and Codes, Volume 3: IMS Abend Codes manuals for IMS 12 and IMS 13 have added the following information:

Under the U0101 description, under section "For DFSPCC20": • Reg7 and Reg8 together contain the value of field RCPGM (program name) at time of abend

Under the U0101 description, under section "For DFSPCC20": Addition to the description of reason code 5 only (Reg15=X'5'):



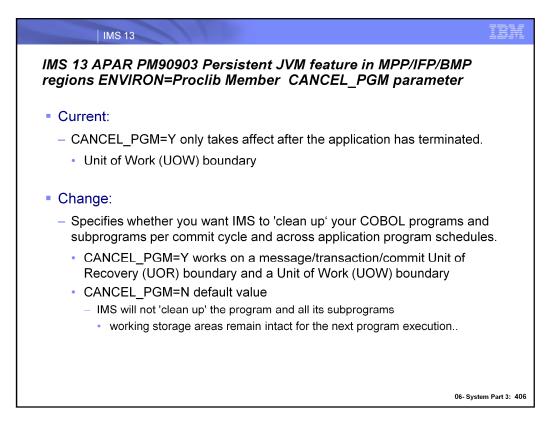
Messages and Codes, Volume 1: DFS Messages:

DFS650E NON-LE COMPLIANT PROGRAM IN PERSISTENT JVM ENVIRONMENT, NAME=program name

Explanation: The user attempted to load and execute a non-Language Environment (LE) compliant/conforming program in a persistent JVM dependent region environment. Any program that executes in a persistent JVM dependent region environment must be LE compliant/conforming.

System action: ABENDU0101-05 (reason code 5) is issued to terminate the application.

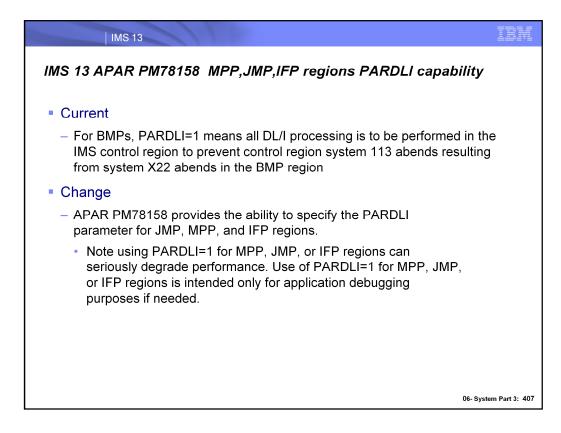
Programmer response: Correct the program to make it LE compliant/conforming. The program did not contain a valid LE entry prolog. Make sure that the program was compiled with a current LE enabled compiler. See z/OS Language Environment Programming Guide (SA22-7561) for a description of LE compliant/conforming programs.



Currently, CANCEL_PGM=Y only works on an application schedule boundary (ie. Unit of Work (UOW) boundary) which means CANCEL_PGM=Y only takes affect after the application has terminated.

CANCEL_PGM=Y must also work on a message/transaction/commit boundary (ie. Unit of Recovery (UOR) boundary) which means CANCEL_PGM=Y must take effect on a syncpoint commit boundary while the application is executing. For example, within the application's message GU loop for a message driven application, CANCEL_PGM=Y will be honored. As another example, for a non-message driven BMP, CANCEL_PGM=Y will be honored per syncpoint commit cycle (ie. per SYNC call) while the application is executing.

The one difference in this model is that on the UOR boundary, CANCEL_PGM=Y will not act upon the currently executing "main" application in the dependent region which is the actual application that IMS schedules. On a the UOW boundary, CANCEL_PGM=Y does act upon the "main" application.



PARDLI=0 | 1 specifies the parallel DL/I option.

- 0: DL/I processing is to be performed within the region. This is the default.
- 1: All DL/I processing for this region is to be performed in the IMS control region. If data apture (EXIT= on the DBD statement) is enabled and this is a delete, replace or insert call, PARDLI=1 is ignored for the DL/I call.

PARDLI=1 prevents control region system 113 abends resulting from system X22 abends in the region. If PARDLI=1, parallel DL/I is disabled. This can degrade performance. Caution: Using PARDLI=1 for MPP, JMP, or IFP regions can seriously degrade performance. Use of PARDLI=1 for MPP, JMP, or IFP regions is intended only for application debugging purposes if needed.

