An Overview of IMS Version 9

Alison Coughtrie
IBM Product Introduction Centre
Hursley, UK
alison_coughtrie@uk.ibm.com

Last updated: 29/06/2005

The IBM Information Management System (IMS) is unsurpassed in database and transaction processing availability and speed. With the demands of the open business environment, and a marketplace working in Web time, IMS delivers the integrity, capability, and performance customers have learned to expect from IMS.

IBM's Information Management System (IMS) helps you with the on demand business enablement, growth, availability, and systems management that the newer environments and cost measures require. The enhancements in IMS Database Manager (IMS DB) and IMS Transaction Manager (IMS TM) with V9 help you:

•Transform the way you do business with integrated information

•Build applications that tolerate the rigors of doing business on the Internet

•Run a scalable, available, safe, and easily managed environment

•Leverage everything you learn in the process and mine all your information to make better decisions.

With several dozen enhancements in this new version of IMS, this chart groups the items into their major categories, though there are items that overlap.

IMS V9 Highlights - Major Items

- **Availability**
  - Integrated Online Reorganization for HALDBs
- **System scalability and capacity**
  - Fast Path DEDB Area Open/Close Enhancements
  - Multi-Area Structure Support for SVSO DEDBs
- **Usability**
  - Type-2 Database Commands
- **Manageability**
  - System Generation and Install Enhancements
  - DBRC API
- **Application development**
  - Storage and retrieval of XML documents in existing and new IMS databases

IMS Version 9 contains over 50 enhancements, in all areas of the product, that address availability, scalability, capacity, usability, manageability, operations and application development requirements from it's huge customer base. All of the enhancements in this new version resulted from specific customer requirements. The DB items, including OLR will be presented first, followed by the XML and FP topics, followed by the TM, DBRC and Install enhancements. The major items are highlighted in this chart.

Improved database availability for HALDB databases, introduced in V7, is provided by the long-awaited Online Reorganization (OLR) enhancement. The database partition with an online reorganization underway is fully accessible to application updaters from V8 and V9.

The Fast Path users may see improvements in DEDB area open/close processing, especially those with very large numbers of DEDBs. Those Fast Path users in a sysplex shared VSO environment will have the ability in IMS V9 to define more than one area in the same Coupling Facility structure.

IMS Database command users will see major usability enhancements in the type-2 commands entered through the Operations Manager (OM) API.

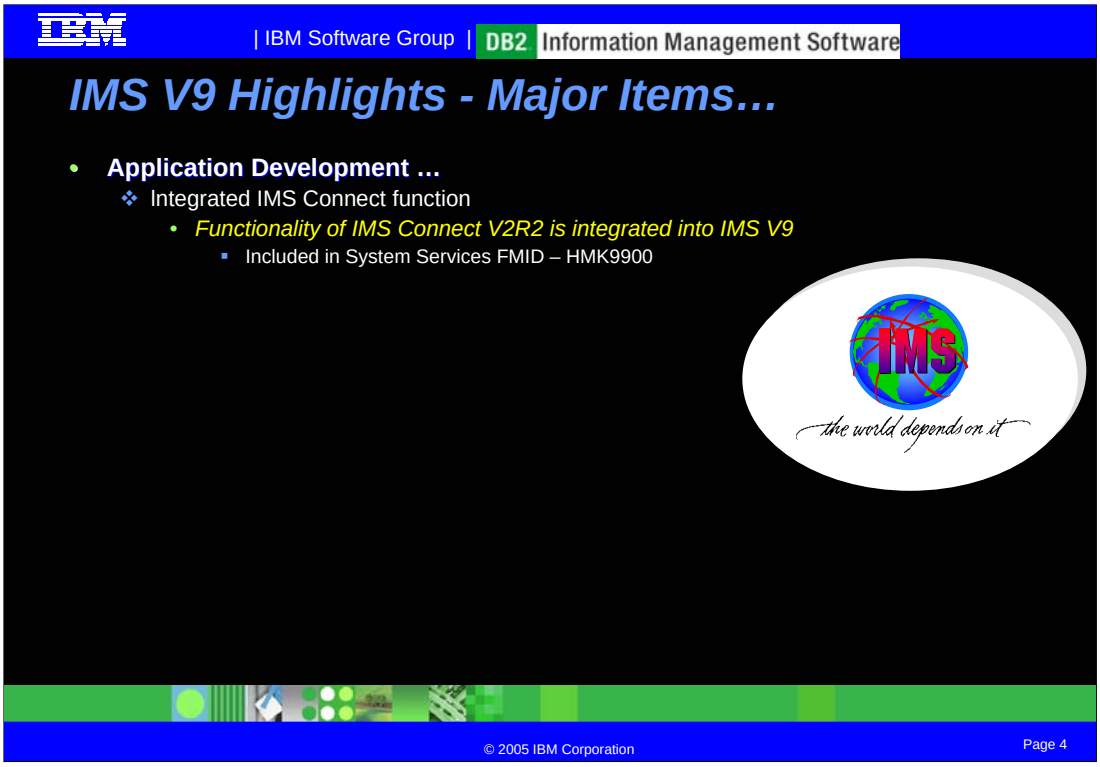
Definitions:

- **type-1 command:** A command, generally preceded by a leading slash character, that can be entered from any valid IMS command source. (New term for classic command.)
- **type-2 command:** A command that is entered only through the OM API. Type-2 commands are more flexible and can have a broader scope than type-1 commands. (New term for IMSplex command and enhanced command.)

Simplifying the IMS system definition process continues in IMS V9 with the removal of the conditional linkedit modules that will allow a single SDFSRESL to be shared by IMS systems that are defined differently, e.g. with FP, non-FP, DB/TM, etc.

DBRC will provide a new API that allows assembler programs to query the RECON.

IMS Java users will have new JDBC SQL calls and DLIModel utility support for retrieving existing IMS data in XML format as well as storing, indexing, searching, and retrieving valid XML documents into new or existing IMS databases.



Integrated IMS Connect Function

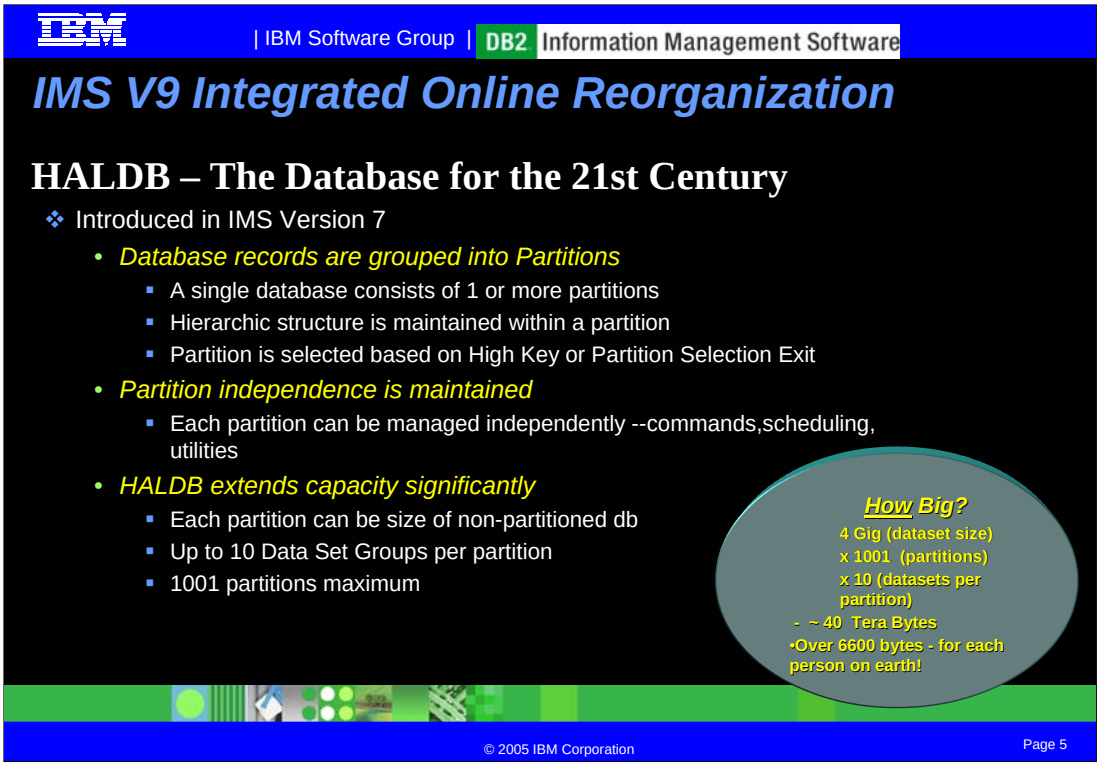
As of IMS Version 9, the functionality of the IMS Connect product (program number 5655-K52), Version 2.2, is included as part of IMS. This functionality is included in the IMS Systems Services function modification identifier (FMID), HMK9900. For a description of the IMS Connect product, see www.ibm.com/software/data/db2imstools.

You can use IMS Connect 2.2 to provide communications between one or more TCP/IP clients and one or more IMS Version 7 or IMS Version 8 clients. For communications with an IMS Version 9 client, you can use the Integrated Connect functionality with IMS Version 9. IMS Connect 2.2 will continue to be supported for IMS Version 7 and IMS Version 8 clients, but future enhancements to the Connect functionality will be made available only with IMS Version 9 or later.

Integrated IMS Connect function provides advanced security and transactional integrity in TCP/IP and local/390 access to IMS and the Internet. It supports high performance communications between one or more TCP/IP or local/390 clients, and one or more IMS systems. It provides commands to manage that environment and assist with workload balancing. It reduces the design/coding effort for client applications, and it provides easier access to IMS applications and operations.

Integrated IMS Connect function used with the WebSphere Development Tooling and the IMS Connector for Java, can significantly ease the development of on demand business solutions that access IMS transactions. These solutions can be deployed in IBM WebSphere Application Servers, allowing you to use Web applications, J2EE applications, or Web services to quickly transform static websites into sources of dynamic Web Content.

Integrated IMS Connect function, used with IBM DB2 Universal Database and the IMS Control Center, allows a single, graphical user interface to control both IMS and DB2, easing IMS operations.



The first topic to be presented is the Integrated HALDB Online Reorganization. HALDB was introduced in IMS Version 7 to allow customers with large databases to partition the database records into groups that can be processed independently by commands, scheduling, and utilities. Any size limitation for a database is eliminated, since the maximum size of a HALDB database is 4 gigabytes (data set size) X 1001 partitions X 10 datasets per partition or approximately 40 terabytes of data. HALDB is not just for large databases; it is for any size database where partitions allow you to take advantage of their increased availability, since they can be manipulated independently of each other.

Phase Two...in IMS Version 9

- **Integrated OnLine Reorganization of HALDBs**
  - Standard part of IMS V9
    - *Not a feature, product, tool etc.*
  - HALDB OLR provides 100% availability of the largest databases in the world!
  - OLR provides non-disruptive reorganization of HALDB PHDAM and PHIDAM partitions
    - *Partitions, partition or entire DB can be reorganized*
    - *Applications are unaffected*
      - They never get data unavailable conditions
      - Concurrent IMS updates are allowed while OLR is active
      - Concurrent data sharing updates are allowed
  - Planned data outage not required

Online reorganization (OLR) is essentially phase two of HALDB. IMS V9 OLR provides a fully integrated online reorganization by partition of HALDBs with concurrent online update and availability. This is totally non-disruptive. Users can adjust the pace of OLR to further minimize their online impact. Multiple partitions can be reorganized in parallel. This online reorganization function provides for enhanced database availability. Coordination is provided through the IMS DBRC facility.

Phase Two...in IMS Version 9...

- Runs in TM/DB or DBCTL System
  - Executes in DLISAS Address Space
- Secondary Indexes and Logical Relationships
  - Database with secondary indexes can be reorged
    - *PSINDEX itself cannot be reorged*
  - Database with logical relationships can be reorged
  - ILDS (ILEs) updated with new target RBAs
- Coordination is provided through the DBRC facility

A PSINDEX is always OLRNOCAP. The reorg. of a target DB does not require the rebuilding of its secondary indexes. After a reorg. of the primary DB, the secondary index will use the pointers in the ILDS to find their target segments. IDCAMs Repro can be used to reorg. a PSINDEX. Index Builder can be used to build the secondary index of a HALDB after a reorg.

active set of data sets

In cursor-active status, the paired input and output sets of data sets that the data is being copied from and to respectively. Otherwise, the single set of data sets that comprise the partition.

cursor

The boundary that delineates the end of the database records that have been copied.

cursor-active status

Paired data sets are active. Cursor ACTIVE=YES is displayed in the RECON listing.

HALDB OLR

Non-disruptive reorganization of HALDB PHDAM and PHIDAM partitions. Synonymous with online reorganization and HALDB Online Reorganization, OLR, OLREORG, and HALDB OLREORG.

HALDB OLREORG

See HALDB OLR.

HALDB Online Reorganization

See HALDB OLR.

hardened data

At an RSR tracking site, data that is guaranteed to have been written to DASD in the shadow database data sets up through a particular milestone.

inactive data sets

The old input data sets that the data was copied from.

milestone

A marker recognizable by the log router that delineates a point from which a restart can occur in IMS database tracking.

OLR See HALDB OLR.

OLR ACTIVE HARD COUNT

Represents the number of HALDB OLR cursor active calls that are pending in the tracking RECON.

OLR INACTIVE HARD COUNT

Represents the number of HALDB OLR cursor inactive calls that are pending in the tracking RECON.

OLREORG

See HALDB OLR.

online reorganization

See HALDB OLR.

owned

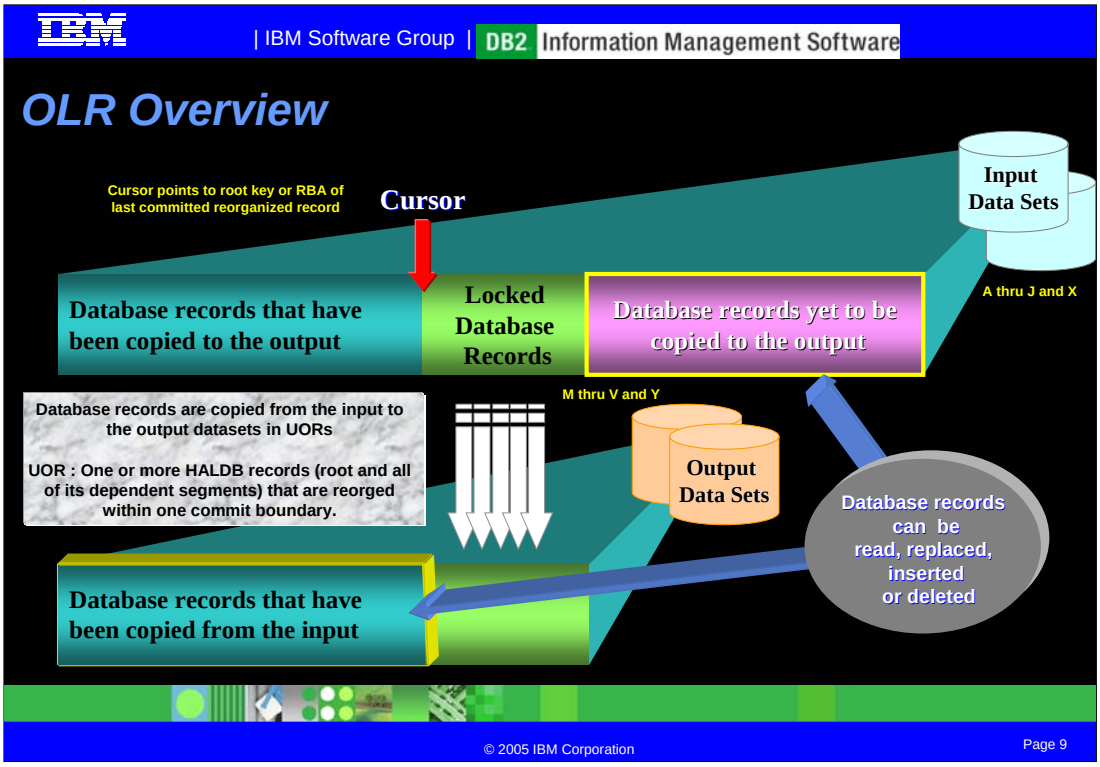
Status during which an IMS subsystem has exclusive control over a HALDB OLR.

SHAREPLEX

As used in this document, IMS SHAREPLEX is defined as the IMS Data Sharing within the boundaries of one set of DBRC RECONs. As such, this differs from MVS SYSPLEX and also from the term, IMSplex which is sometimes used in an IMS Shared Queues environment.

unit of reorganization

One or more HALDB records (root segment and all of its dependent segments) that are reorganized within one commit boundary.



Database records are copied from the input to the output datasets in units of reorganization (UOR). A UOR is one or more HALDB records (root segment and all of its dependent segments) that are reorganized within one commit boundary. Note that during the reorganization, database changes may still be made to the input datasets that have not been copied to the output data sets, and to parts of the output datasets to which data has already been copied. Consider the database records for the entire HALDB partition to be ordered from left to right in three categories:

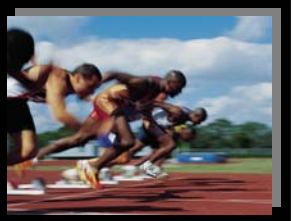
- Those records that have already been copied
- Those records that are currently being copied
- Those records that are still to be copied

A cursor is a key or RBA of a database record that has already been copied to the output data set; a cursor marks which records are still in the input data set. As the copying proceeds, this cursor moves forward through the database (from left to right in the diagram). The cursor is stored in the block after the first bit map.

IMS V9 Integrated HALDB Online Reorganization

Changed Terminology in IMS V9
type-1 = traditional, "classic" command
type-2 = IMSplex command, "enhanced" command

- Solution Highlights
  - INIT/TERM/QRY/UPD OLREORG commands
    - *type-1 and type-2 command support for OLR*
  - Dual data sets during cursor-based reorganization
    - *Allows concurrent data sharing updates while OLR is active - nondisruptive reorg*
    - *Utilities and DBRC support for dual data sets*
    - *Eliminates planned data availability outage*
  - Pacing of OLR via INIT command RATE parameter
    - ***INIT OLREORG...SET(RATE(100))...***
    - ***UPD OLREORG...SET(RATE(50))...***



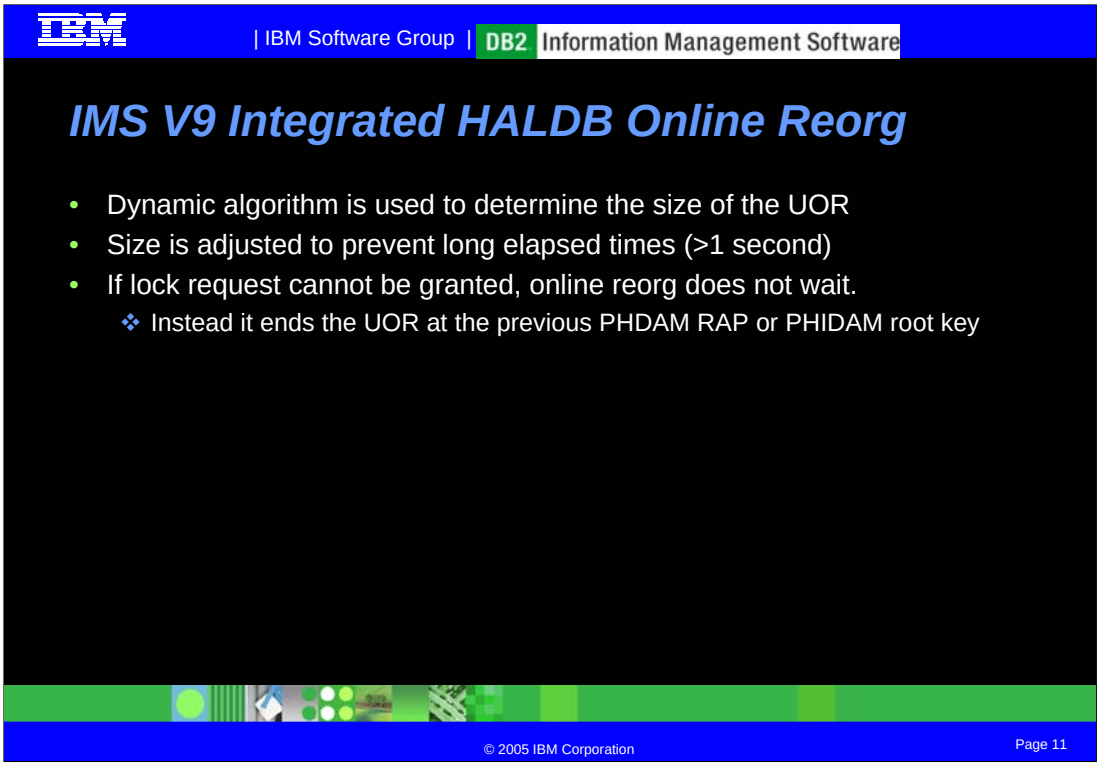
You can use type-2 commands (introduced in IMS V8) or type-1 (traditional) IMS commands to start and stop an online reorganization of a HALDB partition. Type-2 commands use the Operations Manager (OM) Application Programming Interface (API). An IMS Common Service Layer, which includes Operations Manager and Structured Call Interface (SCI) is required.

OLR is a cursor based reorganization on a paired set of data sets. A cursor marks the point at the end of the database records already copied. As the copying proceeds, this cursor moves forward through the database.

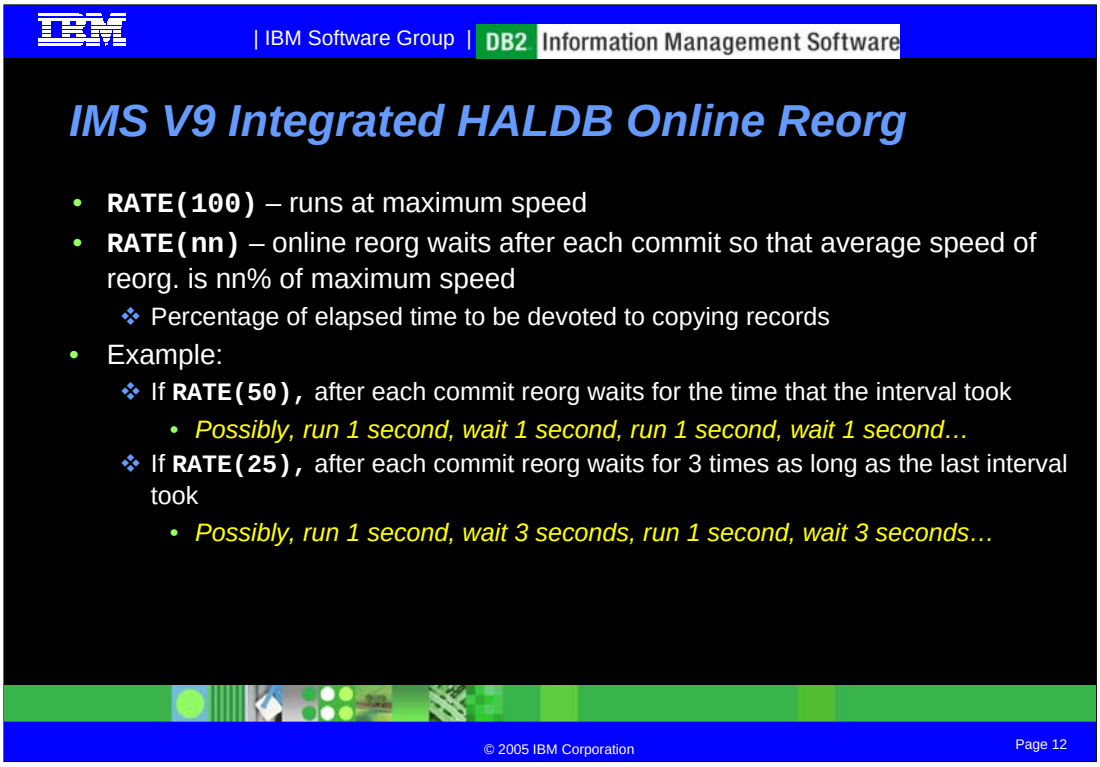
Dual Data sets: IMS performs a non-disruptive database reorganization using a pair of data sets. Database records are copied from the input to the output data sets in units of reorganization. Note that during the reorganization, database changes may still be made to the input data sets that have not yet been copied to the output data sets, and to parts of the output data sets to which data has already been copied.

This reorganization process does not support data definition changes. That is, it provides only reclustering and space distribution advantages. OLR limits the size of its UOR in an attempt to hold locks for no more than a second.

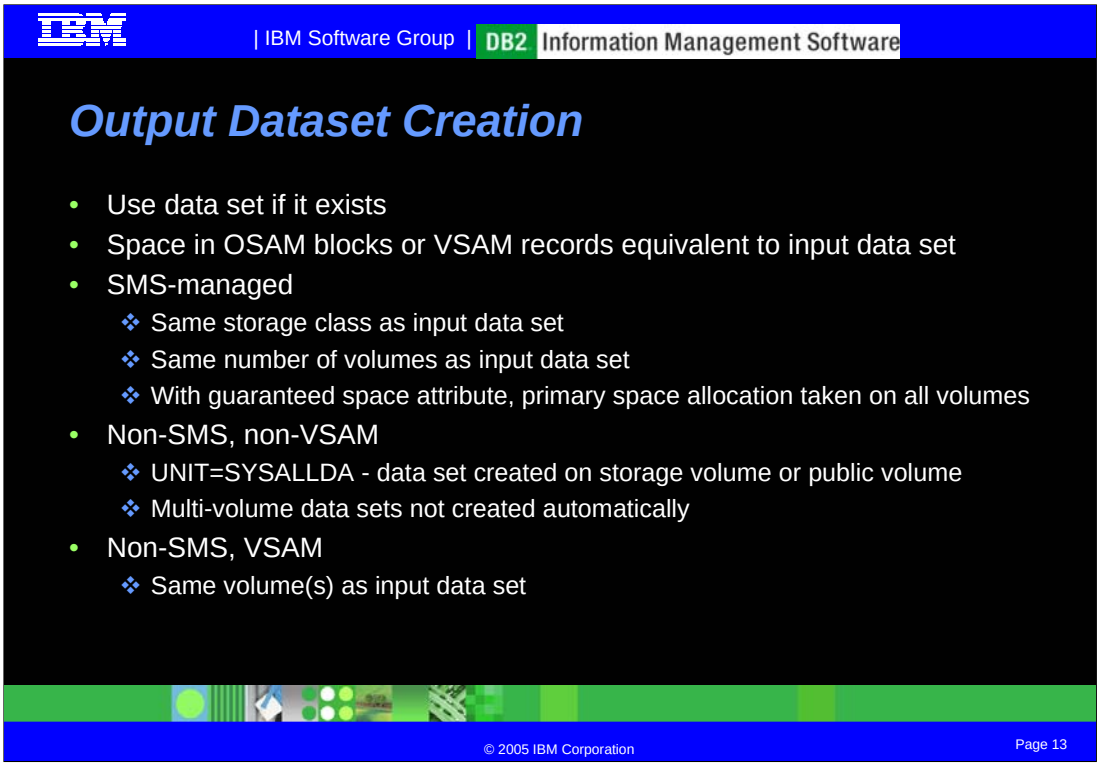
How fast an OLR runs is determined by the RATE specified on the INITIATE OLREORG or UPDATE OLREORG commands, system resources and system utilization. The amount of updates to the IMS log and system contention can also affect the speed at which an OLR runs. The RATE can be changed when adjustments are desired depending on system contention (slow down the OLR) or lack of system contention (speed up the OLR).



OLR limits the size of its UOR in an attempt to hold the locks for no more than a second.



An online reorganization's impact on the system is affected by the available system resources, by total system utilization (including other online reorganizations), by total logging volume, by log contention, and by the intensity at which this reorganization was set to run. These same factors also affect the speed at which the reorganization runs. You can use the RATE parameter as a dynamic throttling knob in the INITIATE OLREORG and UPDATE OLREORG commands to control the intensity at which the reorganization runs. This can affect not only the reorganization's speed but also its impact on the rest of the system. The value specified for RATE is the percentage of elapsed time to be devoted to copying records. The remaining time is an intentionally introduced delay in the copying process that minimizes the reorganization's impact on other IMS work and on the whole system. RATE(100) is the default that allows the online reorganization to run as fast as possible (depending on system resources, system contention, log contention, etc.) with no intentional delays. A RATE value of 50 specifies that 50% of the elapsed time should be spent copying records and the remaining 50% should be spent in a delay. This would cause the reorganization to run approximately twice as long as it would have run with RATE value of 100. A RATE value of 25 would cause the reorganization to take four times as long, and so on. The RATE value can be changed with the UPDATE OLREORG command at any time that adjustments are desired because of the amount of system contention. With higher system contention, the RATE value can be lowered to slow down the online reorganization and minimize its impact on other work. With less system activity, the RATE value can be raised, thus speeding up the reorganization.



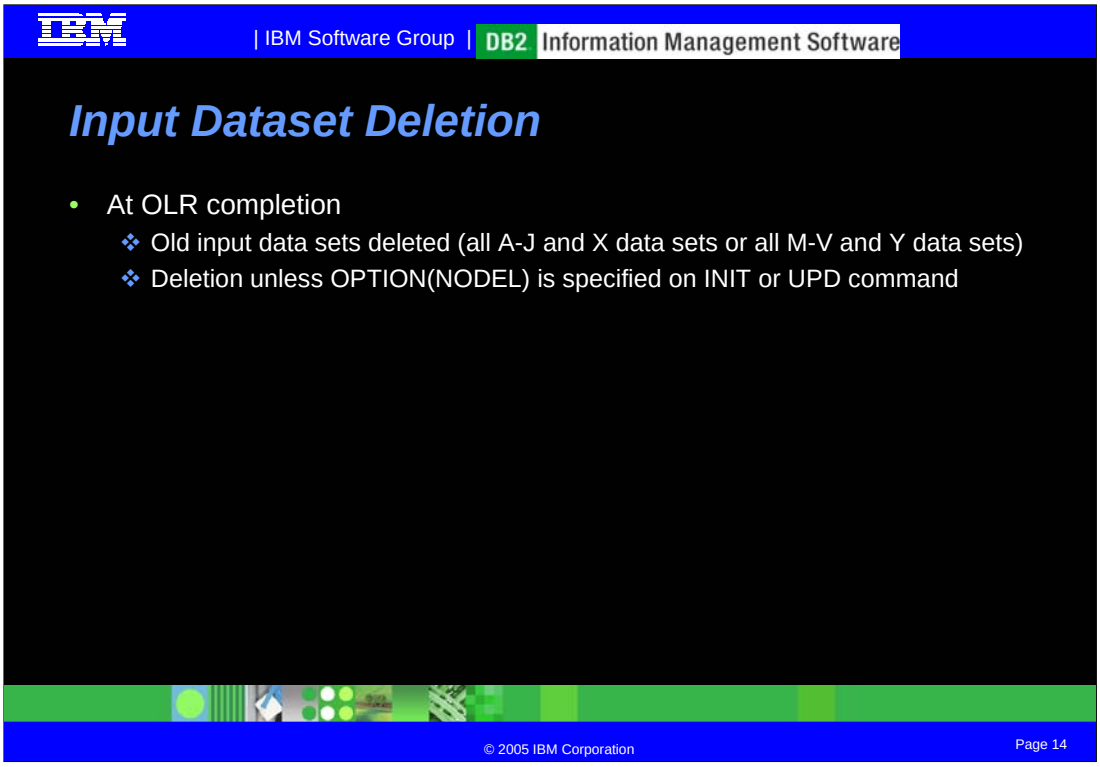
To simplify the OLR process for a HALDB partition, each of the output data sets can be created **automatically** by IMS. For each data set group defined in the DBD and for the primary index of a PHIDAM database, the output data set is created if it doesn't already exist as a cataloged data set. The indirect list data set is not created automatically because there is not a corresponding output version of it.

In order to reserve approximately the same amount of space that was reserved for the input data set regardless of the DASD types involved, the output data set's space is requested as a number of OSAM blocks or VSAM records.

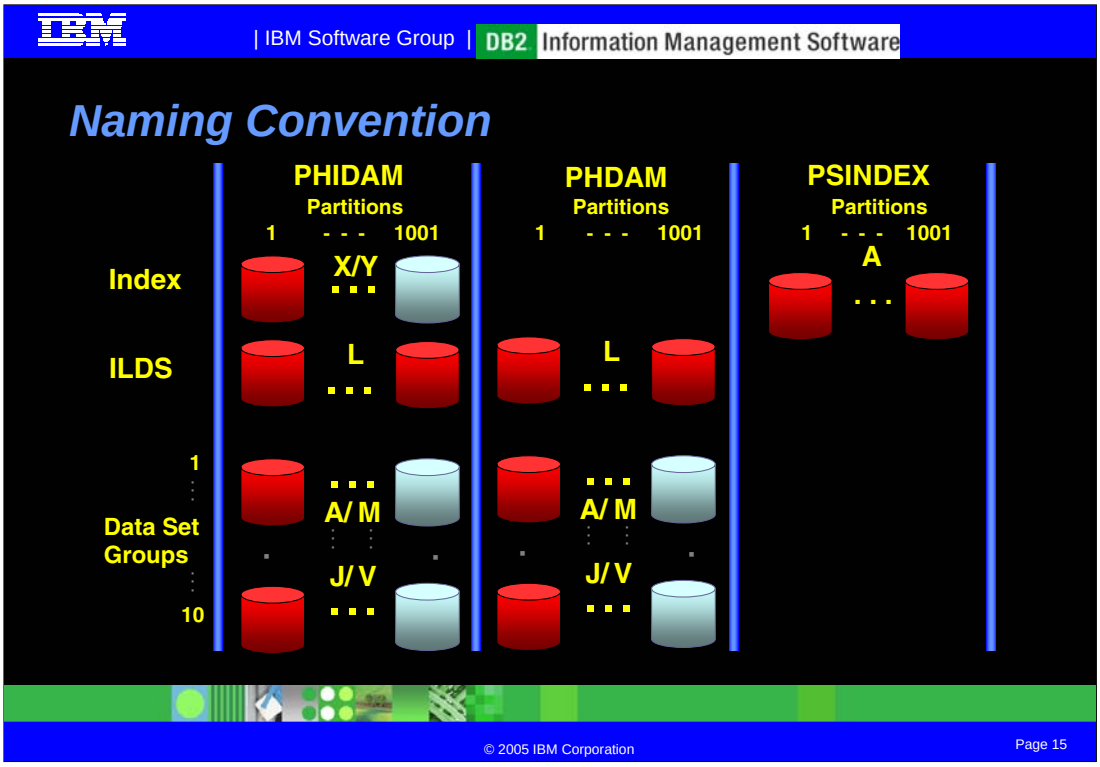
If an input data set is SMS-managed, then the corresponding output data set is SMS-managed as well, and the storage class used to create the output data set is that of the input data set. If the input data set has extents on only one DASD volume, the output data set is also created on a single volume. If the input data set has extents on more than one DASD volume, the output data set is created to allow the same number of volumes. If the storage class of the input data set has the guaranteed space attribute, the primary space allocation will be taken on each of the volumes when the output data set is created.

If an input data set is non SMS-managed and is non-VSAM, the output data set is created as though UNIT=SYSALLDA has been specified on a DD statement. This causes the data set to be created on a storage volume or, if no storage volume is available, on a public volume. Note that if the input data set is a multi-volume data set, the output data set will not be created automatically.

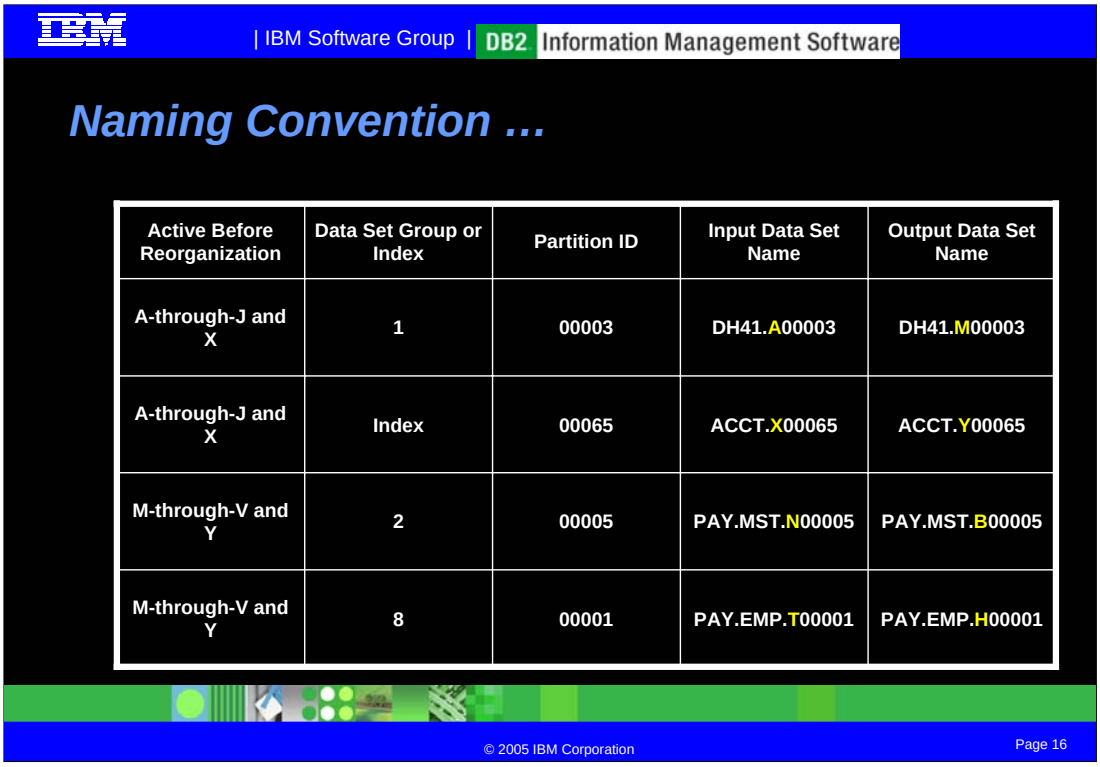
If an input data set is non SMS-managed and is VSAM, the output data set is created on the same volume or volumes as the corresponding input data set.



As part of the successful completion of an OLR, all of the input data sets involved in the reorganization become inactive. These data sets may be deleted automatically by IMS, regardless of whether they were created by IMS or were pre-existing data sets. This automatic data set deletion can be suppressed by specifying the NODEL option on the INIT OLREORG or UPD OLREORG command.



Online reorganization has extended the data definition and data set naming convention established for HALDB. Multiple data set groups in a HALDB database use the characters A-thru-J in the DDNAME statements and data set names of the supported ten data set groups, and the primary index for a PHIDAM database uses the character X in these names. This naming convention has been expanded for IMS Version 9 by implementing the characters M-thru-V and Y for an alternate (or paired) set of data sets.



Whether you create one of the output data sets, or you let IMS create it automatically, the data set name is identical to that of the corresponding input data set with the exception of the data set name type character. If the A-through-J and X data sets are the active set prior to this reorganization, the dataset name for the output data set is the corresponding data set in the M-through-V and Y set of datasets, and the opposite. The table above shows some data set name examples.

IMS V9 Integrated HALDB Online Reorganization

Entered through the OM API

- **Type-2 Commands**
  - **INITIATE OLREORG** ...
    - *Starts OLR and sets RATE and OPTION to delete the old input data sets (DEL)*
  - **TERMINATE OLREORG** ...
    - *Stops OLR or at next record boundary*
  - **UPDATE OLREORG** ...
    - *Change the RATE or delete/no delete OPTION*
  - **QUERY OLREORG** ...
    - *Show OLR RATE and number of bytes moved to output data set*
  - **QUERY DB STATUS(OLR)**
    - *Show OLR status for partitions with OLR in progress*

This chart summarizes the commands that control OLR. The complete syntax is not shown on the charts but described further in these notes. When these commands are issued from the OM API, the command output is defined in XML and is available to automation programs which communicate with OM.

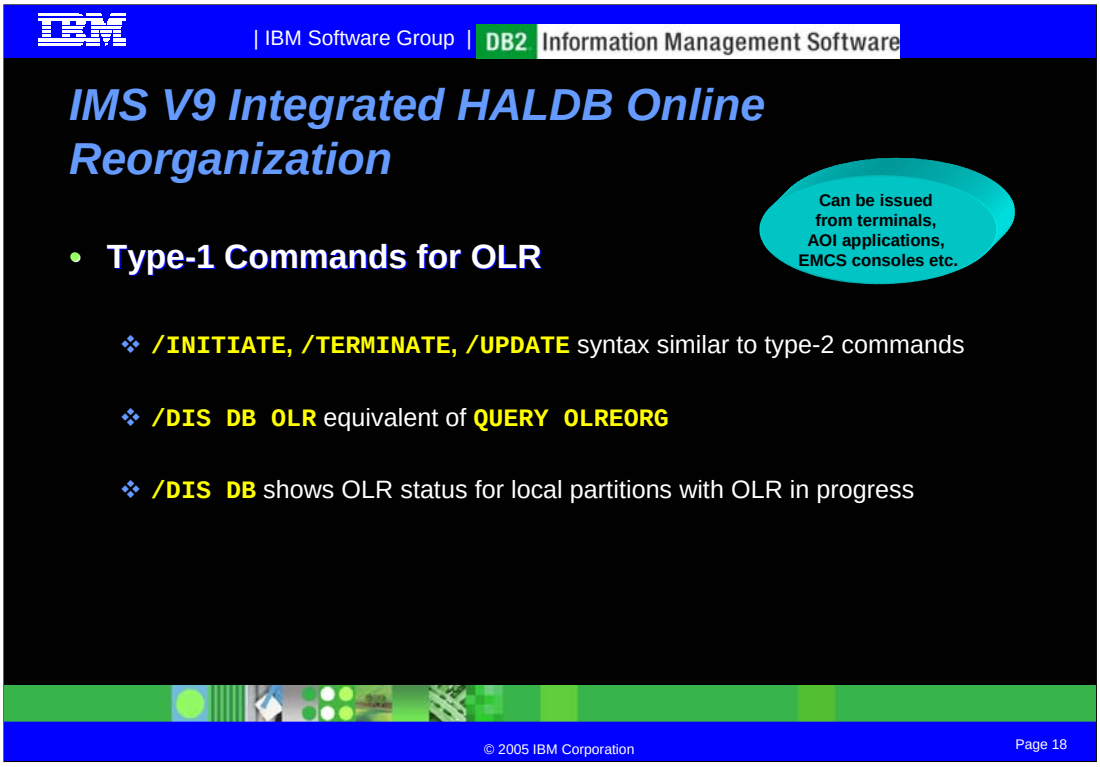
INITIATE OLREORG causes OLR to begin or resume on the partition or partitions specified. RATE(100) is the default which means run the OLR as fast as possible with no coded waits. OPTION(DEL) is the default and indicates the deletion of the old input data sets is to be attempted as part of the termination processing when the OLR completes. Note that one big advantage of specifying the NODEL option is to gain better control of data set placement.

TERMINATE OLREORG causes OLR to be stopped for the specified HALDB partition(s). OLR will no longer be active or have an owning IMS. OLR can be resumed on any IMS, with the INITIATE OLREORG command. OPTION(FORCE) indicates that the OLR for the named partition is to be stopped when the next record boundary is encountered. All moved data up to that point is committed to DASD and no backout is required. OPTION(ABORT) causes the OLR to be stopped immediately. Backout may be required depending on the state of the OLR.

UPDATE OLREORG allows the changing of the RATE and/or DELETE options for one or more OLRs.

QUERY OLREORG will return the status of one or more OLR tasks depending on the specified keywords. If NAME(partition name) is not specified, then the default NAME(*) will be used and the responses will only be for HALDB partitions that have OLR in progress. STATUS() selects which OLRs to return status for. STATUS(RUNNING) specifies that the output will be for the OLRs that are owned by each IMS (to which the command is routed) for the specified partition name(s). SHOW() specifies the output fields to be returned. SHOW(ALL) returns the rate at which the OLR is running and also the number of bytes moved to the output data set.

The existing QUERY DB command is enhanced to show OLR status. QUERY DB STATUS(OLR) will return all partitions with OLR in progress. No rate information is returned. This is returned in the QUERY OLREORG command output.



The commands to initiate, terminate and update OLR are supported as type-1 commands. /INITIATE OLREORG, /TERMINATE OLREORG and /UPDATE OLREORG may be issued from terminals, AOI applications, EMCS consoles and APPC/OTMA programs. They are also passed to the AOI user exits and logged to the secondary master. Note that /QUERY OLREORG is not supported.

The type-1 command syntax is the same as the syntax of the equivalent type-2 commands from the OM API. The type-1 commands are however preceded by a CRC, usually a '/'. Note that only one partition name can be specified on the type-1 commands.

When a command is issued from the OM API, the command response is encapsulated in XML tags. When the command is entered as a type-1 command, the command response is returned in the message format.

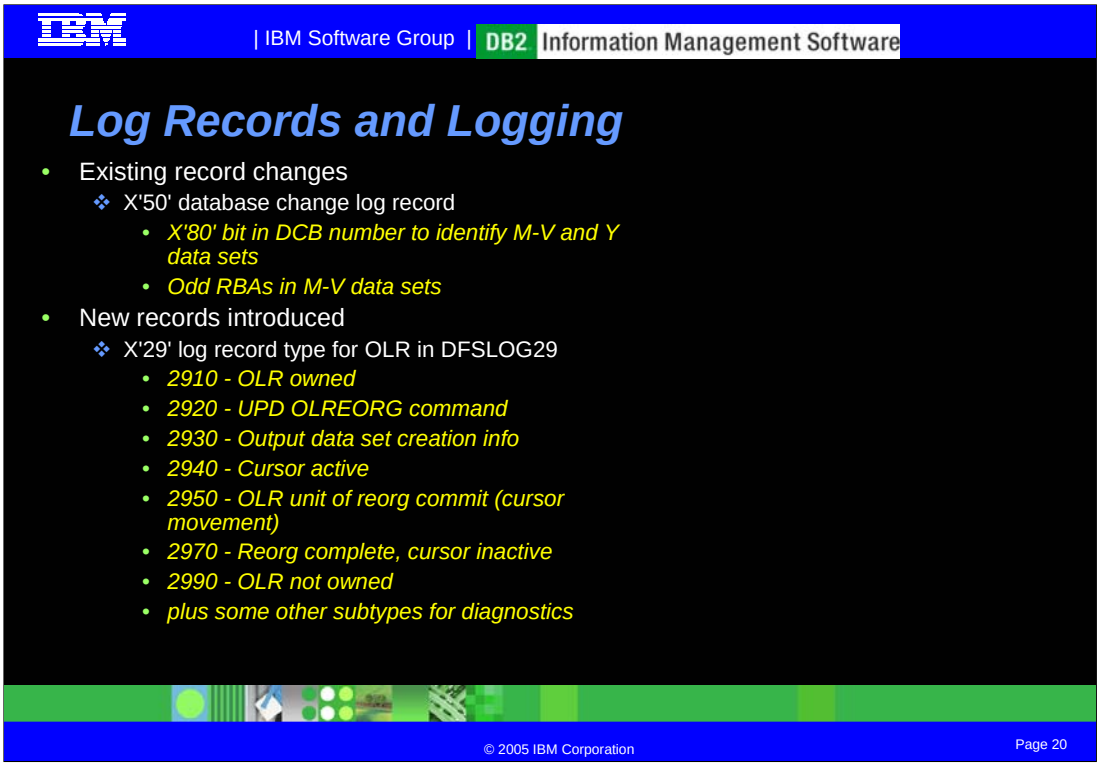
INITIATE command - if successful results in the DFS058I INITIATE COMMAND IN PROGRESS message followed by the DFS0725I message to the system console. TERMINATE and UPDATE commands - result in the DFS0725I message to the inputting terminal.

/DIS DB OLR command can be considered as the type-1 command flavor of QUERY OLREORG. /DIS DB OLR returns all partitions that have OLR in progress locally. The following information is returned: master DB name, partition name, OLR rate, number of bytes moved and OLR status.

The /DIS DB command is enhanced to show the OLR status for the specified partitions or databases which have OLR in progress locally.

IMS V9 Integrated HALDB Online Reorganization

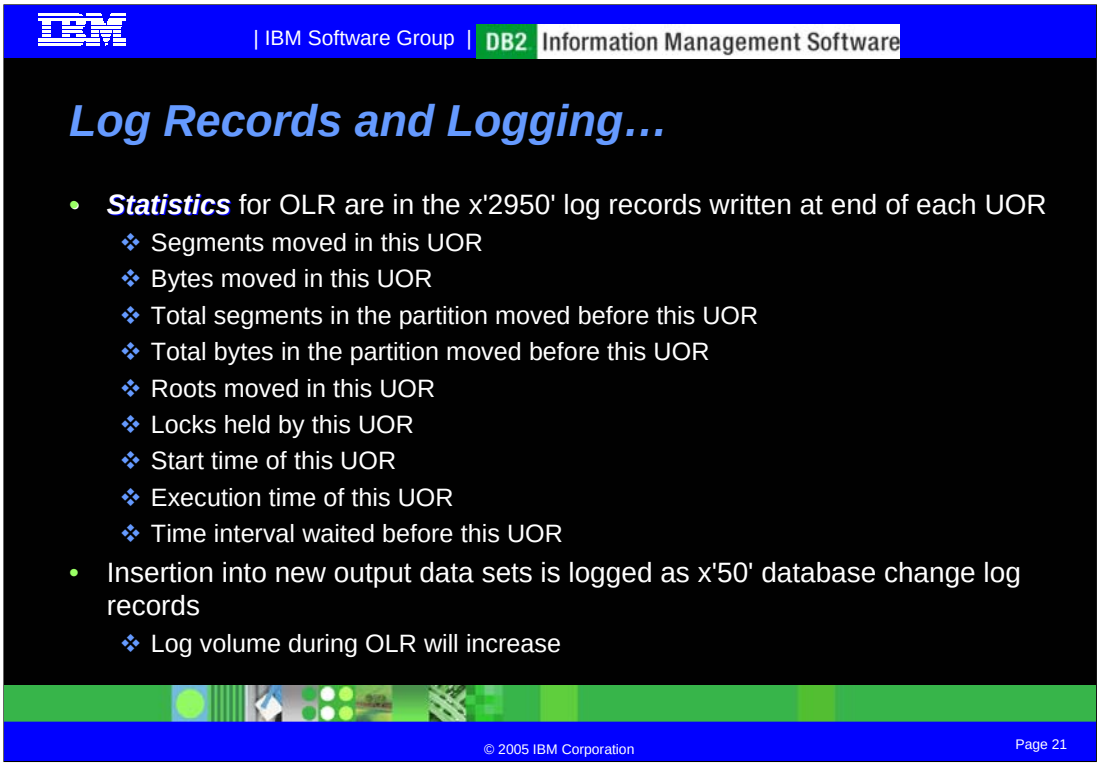
```
/INIT OLREORG NAME(PDHDOKA) SET(RATE(5))
DFS058I 12:08:07 INITIATE COMMAND IN PROGRESS
DFS2970I - OLR STARTED FOR NAME=PDHDOKA
DFS0725I INIT      OLREORG COMMAND FOR DB PDHDOKA  COMPLETE. CC=   0
/DIS DB OLR
DATABASE  PART      RATE         BYTES  STATUS
DBHDOK01  PDHDOKA      5         86776  RUNNING
*03099/120858*
/DIS DB PDHDOKA DBHDOK01
DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
PDHDOKA   PART                              UP   ALLOCS, OLR
DBHDOK01  PHDAM                             UP
PDHDOKA   PART                              UP   ALLOCS, OLR
PDHDOKB   PART                              UP   NOTOPEN
PDHDOKC   PART                              UP   NOTOPEN
PDHDOKD   PART                              UP   NOTOPEN
*03056/140011*
```



The x'50' database change log record has been changed. The DCB number field has x'80' bit on for M-V data sets. Odd RBAs are used in M-V data sets while even RBAs are used in A-J data sets. This means a 4GB data set size limit for HALDB partitions.

x'20'/ x'21' (DB open/close) log records and x'27' (DB extension) log records have also been changed. The DCB number field has x'80' bit on for M-V data sets.

New x'29' log record type is introduced to reflect the progress of OLR processing.



OLR statistics counters are included in the x'2950' record which is logged at the end of each unit of reorganization (UOR). The statistics include :

- Number of segments moved in this UOR
- Number of bytes moved in this UOR
- Total number of segments in the partition moved before this UOR
- Total number of bytes in the partition moved before this UOR
- Number of roots moved in this UOR
- Number of locks held by this UOR
- Start time of this UOR
- Execution time of this UOR
- Time interval waited before this UOR

The logging volume attributable to OLR is important to understand because OLR of a partition generates x'50' log records for insertion into the new output data sets. OLR of multiple partitions in parallel could generate sufficient log data to impact normal transaction processing. The OLR RATE can be changed when adjustments are desired depending on system contention.

IMS V9 Integrated HALDB Online Reorganization

- **OLR Migration Considerations**
  - OLR capable (OLRCAP) set via the **`CHANGE.DB`** DBRC command
  - V8 OLR Coexistence SPE required
    - *V8 data sharing updaters can access V9 DB being OLRed*
  - V7 ***will not*** access V9 OLR-capable HALDB or its partitions

Coexistence between V7 and V9

OLR can be run on V9 for all HALDBs that are marked OLR-capable.

V7 will not be able to see any HALDB that is marked OLR-capable.

Coexistence between V8 and V9

OLR can be run on V9 for all HALDBs that are marked OLR-capable.

With the OLR Coexistence SPE applied, V8 will be able to see all HALDBs (including HALDBs that are marked OLR-capable). V8 will also be able to share data with V9 while OLR is active. Once data sharing has begun between V8 and V9, the V9 utilities must be used except for Archive, Batch Backout and Log Recovery.

Without the OLR Coexistence SPE applied, V8 will not be able to see any HALDB that is marked OLR-capable.

IMS V9 Integrated HALDB Online Reorganization

- Other OLR considerations
  - *All data movement during OLR is logged*
  - *No data definition changes via OLR*
  - *Input and output data sets occupy same buffers*
    - OLR uses VSAM string
      - » Make sure there are enough by using STRINGNM on POOLID
  - *OSAM Sequential buffering is used by OLR*
    - SBONLINE in DFSVSMxx is required

❖ Utilities

- *Image Copy, Recovery, Change Accum operate on A-J and M-V data sets*
- *Image Copy not allowed while OLR is active*

The logging volume attributable to OLR is important to understand because OLR of a partition generates x'50' log records for insertion into the new output data sets. OLR of multiple partitions in parallel could generate sufficient log data to impact normal transaction processing. The OLR RATE can be changed when adjustments are desired depending on system contention.

In a Shareplex environment (SHRLEVEL 2 or 3), it is beneficial to dedicate a separate IMS to run OLR for performance reasons. You may also want to have dedicated buffer pools for those HALDBs that will be OLRed.

Change Accumulation Utility DFSUCUM0 will accumulate changes for HALDB partition A-J as well as M-V data sets. The DBRC GENJCL.CA command treats the start of OLR as though it were the most recent image copy and causes changes to the output data set to be accumulated from that point.

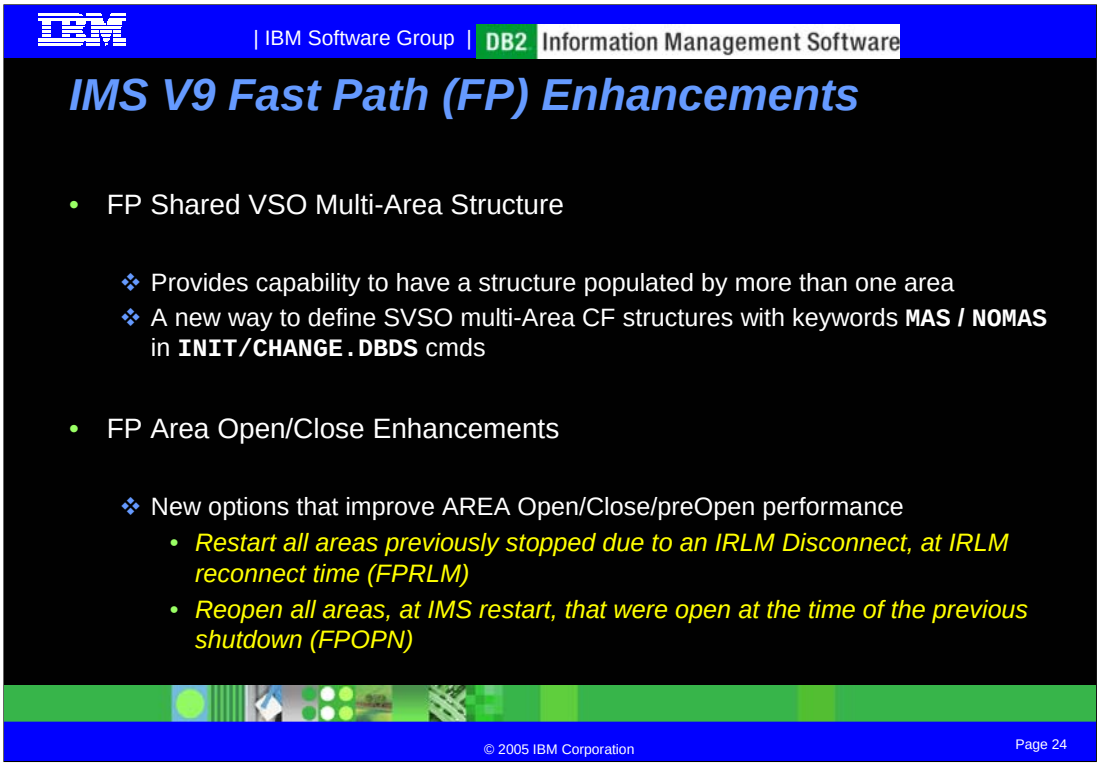
DB Recovery Utility DFSURDB0 will recover HALDB partition A-J data sets as well as M-V data sets. Timestamp recovery (TSR) to a point where a HALDB OLR was active is not allowed if OLR has completed. While running HALDB OLR you might have to recover the input and output data sets in order to restart the online reorganization, or you might use the TERMINATE OLREORG command to suspend the HALDB OLR. Once the cursor-active status for OLR is recorded in RECON, recovery of any of the output data sets with DFSURDB0 can be accomplished solely from DB change records (type x'50' log records) until OLR has completed and a subsequent image copy has been made. An empty output data set is considered to be the starting point from which DB change records can be applied. The RECON reflects the beginning of OLR as a starting point from which forward recovery of the output data sets can be done, even after OLR is finished. Until an image copy has been made for the output data set, the DBRC GENJCL.RECOV prepares recovery of an output data set from this point even though no physical image copy exists.

Image copy will copy the HALDB partition A-J or M-V data sets. The new function will be transparent to the user. The user does not need to be aware of which set of data sets is the active set and does not have to provide this information. Image Copy will determine if the M-V data sets are to be copied instead of the A-J data sets. While OLR is active for a partition, none of the three DB Image Copy utilities can copy any of the data sets for that partition. Even if the TERM OLREORG command is used to temporarily stop OLR, an image copy still cannot be made for any data set in that partition. The utility fails in this case, and IMS issues an error message for each request to copy HALDB partition A-thru-J data sets or M-thru-V data sets.

Since dynamic allocation is recommended, GENJCL no longer generates DD statements for the HALDB DBDS being copied. It is no longer necessary to code a DD statement in the JCL when copying partition data sets as they will be dynamically allocated.

If OLR fails, and IMS is in the middle of a unit of reorg, IMS will backout any changes required. The OLR will need to be restarted at a later time. An offline reorg is required only if there are data corruption-type problems.

Some installations could choose to dedicate a system to OLR. Logs produced by a system dedicated to OLR will only have log records from those databases using OLR. If a CA Group does not contain any of these databases, it will never need logs produced by this IMS system.



Fast Path Multiple Areas Shared VSO Structures

IMS Version 9 enhances the support for sharing VSO DEDB areas by allowing more than one VSO DEDB area in a coupling facility structure. Multiple DEDB areas can share a single coupling facility structure. These structures are called multi-area structures. Because of this support, a user does not need to define as many coupling facility structures as before. This enhancement enables users to have more shared VSO DEDB areas while using fewer coupling facility structures.

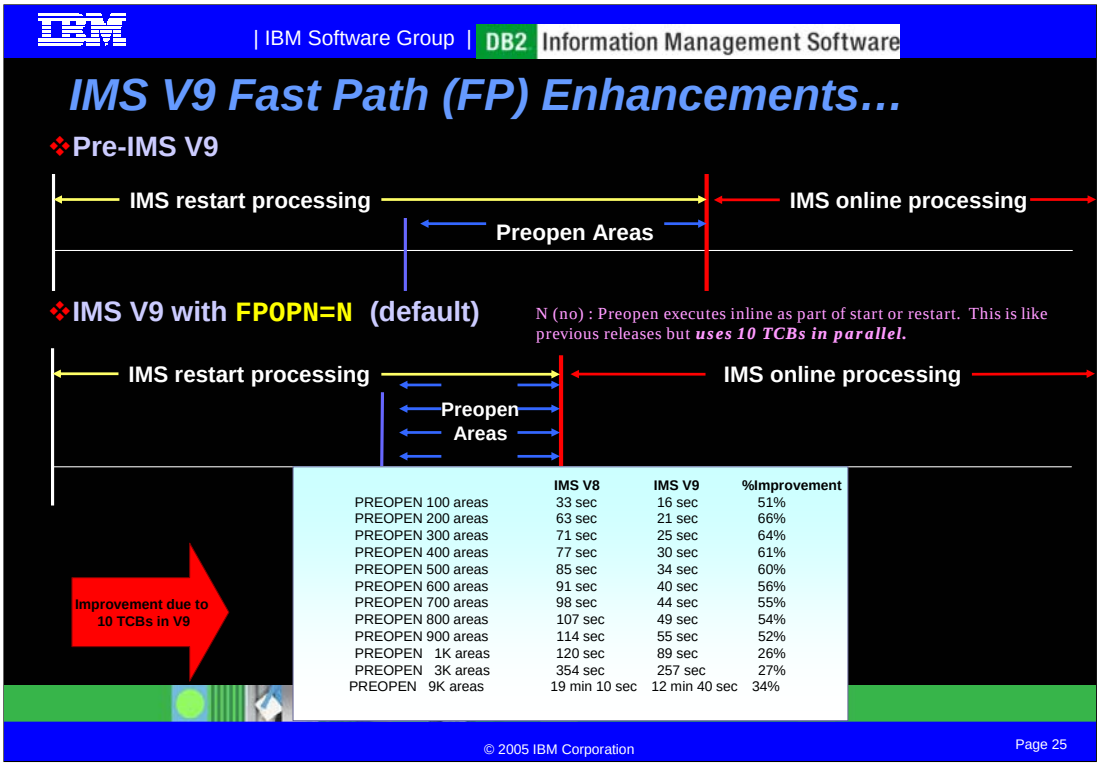
All areas in the same structure must have the same characteristics. The first area that is opened determines the characteristics of the structure, including the CI size. All areas opened in a structure must have the same CI size.

Fast Path Area Open/Close Enhancements

These enhancements provide the ability to reopen DEDB areas automatically after IMS warm or emergency restart or IRLM reconnect. At the end of IMS restart, all areas that were open at the time of the previous shutdown, will be reopened if these enhancements are enabled. This reopen processing will run asynchronously with normal online activity. However, if an area is not yet reopened when an IMS application program requests access, it will be reopened immediately as in prior releases. You enable this automatic reopen with a new parameter (FPOPN) for the IMS procedure.

During IRLM reconnect, all areas that were started when IRLM failed will be restarted if these enhancements are enabled. This restart processing will run asynchronously with IRLM reconnect and normal online activity. However, a user can still use the /STA AREA command to start an area immediately. Users enable this automatic restart with a new IMS startup parameter (FPRLM).

IMS Fast Path uses up to 10 task control blocks (TCBs) to open, preopen, or close DEDB areas, thus paralleling the processing for each task.



Fast Path Area Open/Close Enhancements

Restoring Open Areas During an Emergency Restart

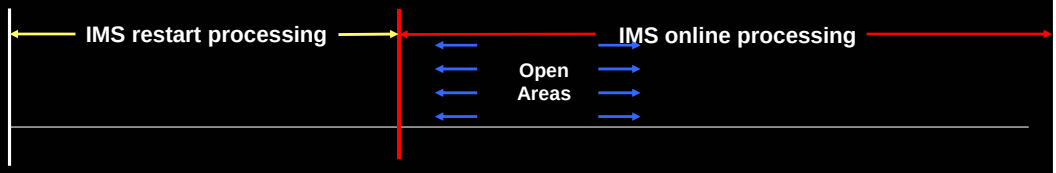
You have several options for how IMS reopens DEDB areas during an emergency restart. You can specify these options by using the FPOPN= keyword in the IMS procedure. The FPOPN option does not change any area options in DBRC. This includes PREOPEN, VSO, and PRELOAD. These options remain set in the RECON data set. This option does not affect DEDB forward recovery (REDO) processing. If an area requires forward recovery during an emergency restart, the area is opened, updated, and then closed.

The following are the FPOPN= parameter options:

•FPOPN=N : Specifies that areas should not be reopened automatically during IMS restart. Areas that are registered with DBRC for preopening will be preopened before the completion of an IMS start or restart as usual; other areas are not preopened. This is the default.

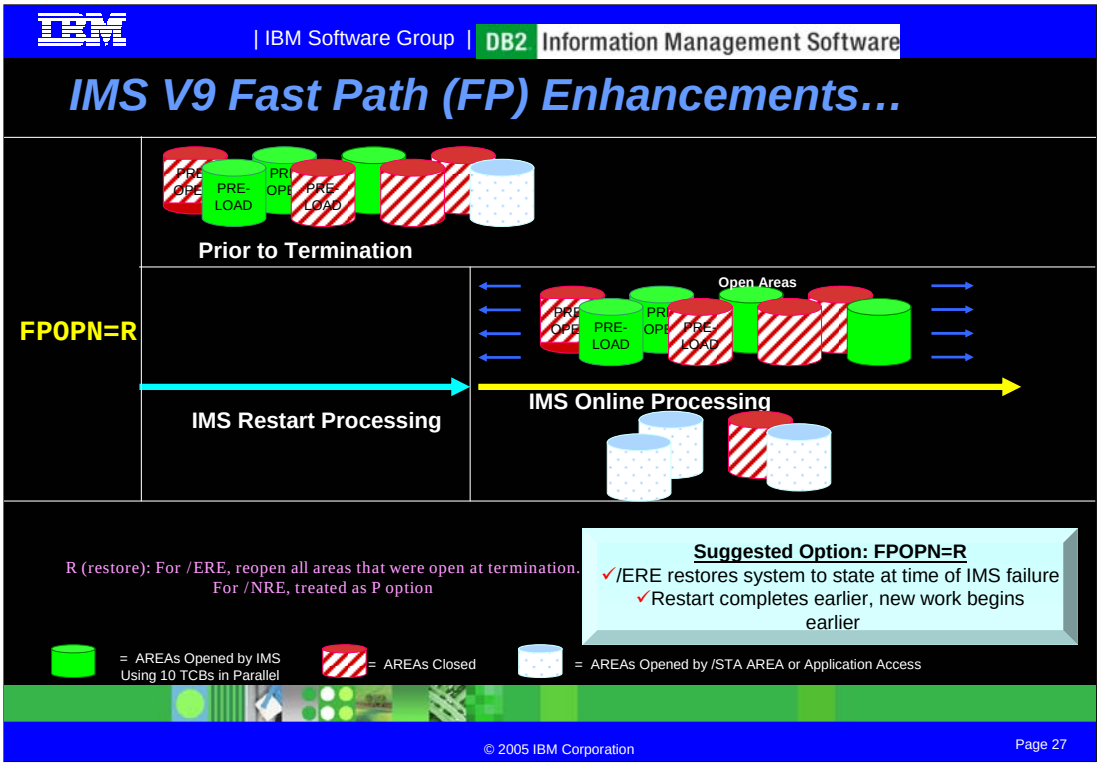
IMS V9 Fast Path (FP) Enhancements…

- IMS V9 with **FPOPN=P**

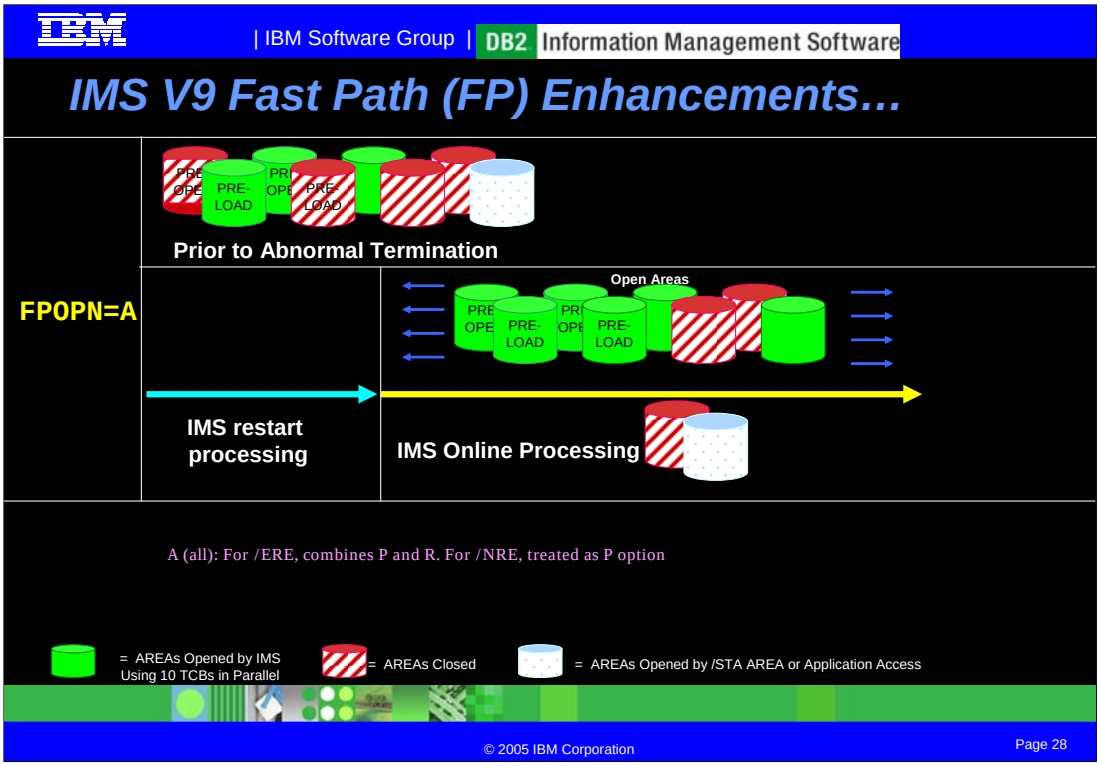


P (preopen): Initiate preopen (areas registered with DBRC to be preopened) at end of normal or emergency restart. Executes asynchronously with online activity

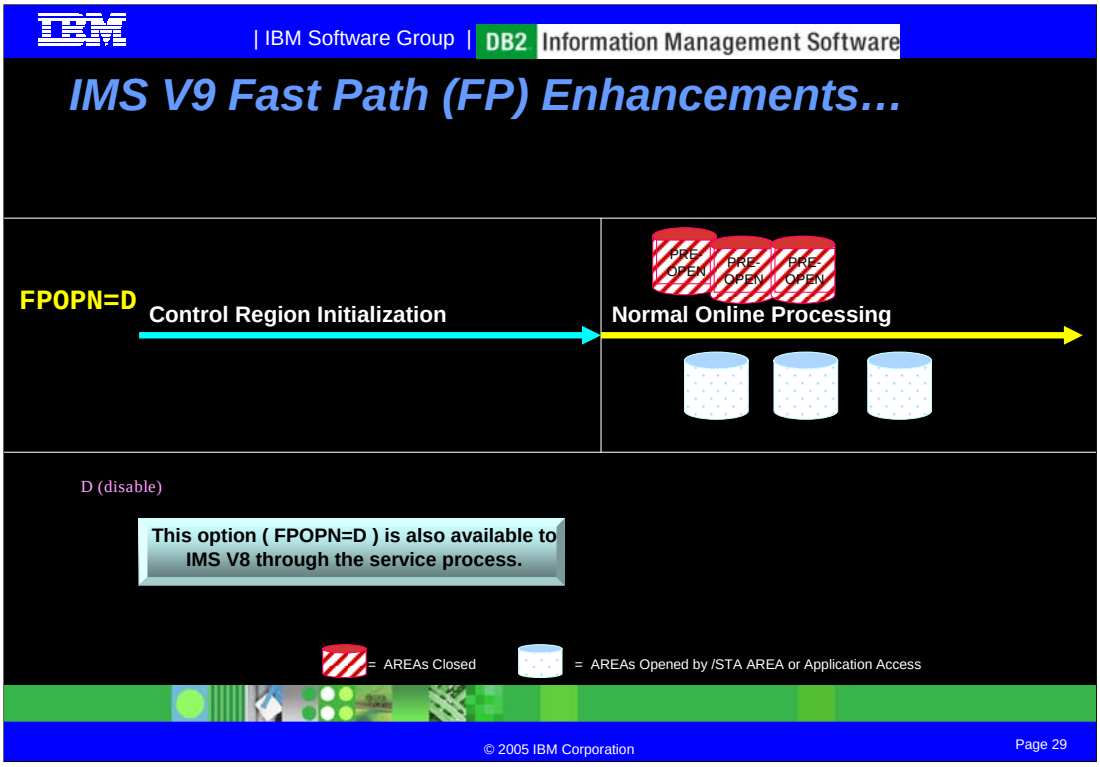
•FPOPN=P: Specifies that after an IMS normal or emergency restart, IMS should preopen those areas that are registered with DBRC to be preopened asynchronously and concurrently with normal IMS online processing.



•FPOPN=R: Specifies that during IMS emergency restart, IMS should automatically reopen all areas that were open at the time of the previous IMS failure. IMS reopens the areas after restart processing is complete. Reopening the areas occurs asynchronously and concurrently with normal IMS online processing. All DEDB areas that were closed at the time of the abnormal termination, including DEDB areas with a preopen status, will remain closed when you restart IMS.



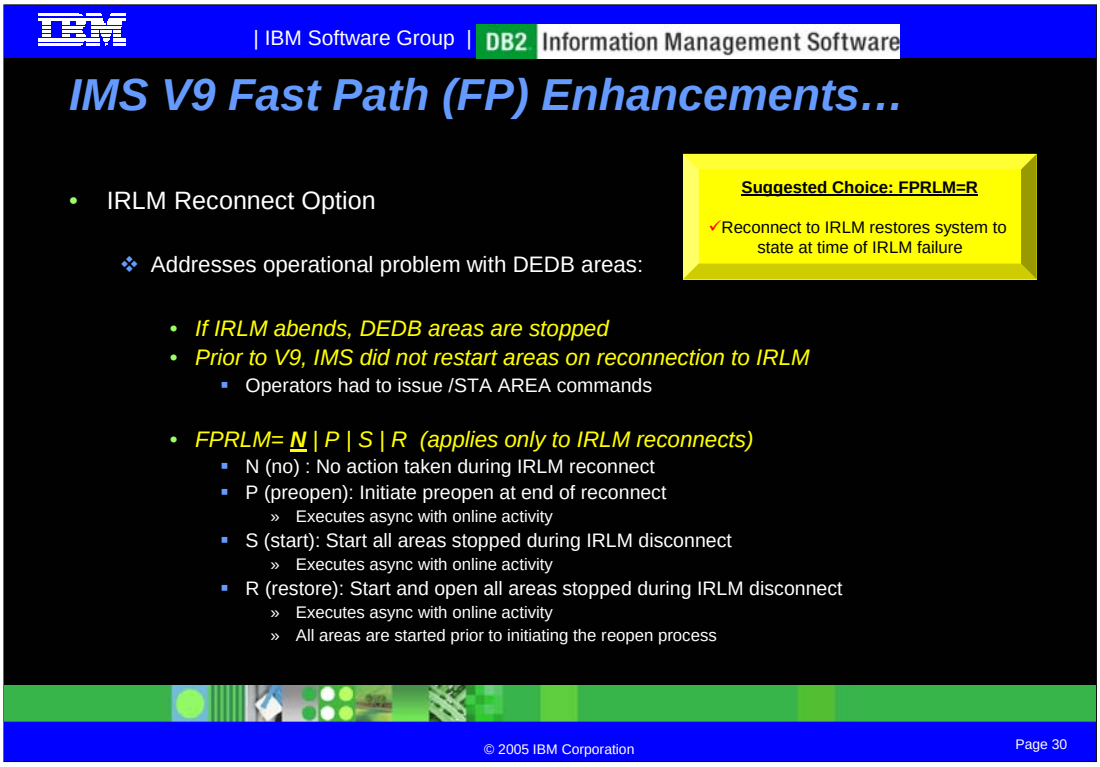
•After the startup process and asynchronous to the resumption of application processing, IMS opens all DEDB areas that have preopen status, even if they were closed at the time of the abnormal termination, in addition to any DEDB areas without preopen status that were open at the time of the abnormal termination. FPOPN=A specifies this.



Disabling the Preopen DEDB Area Process

You can disable the preopen process by specifying FPOPN=D in the IMS or DBC procedures. When the preopen process is disabled, DEDB areas with preopen status remain closed until they are first accessed by an application or they are manually opened with a /START AREA command. It applies to cold starts, warm starts, and emergency restarts.

Specifying FPOPN=D overrides, but does not change, the preopen specifications made with the DBRC commands INIT.DBDS and CHANGE.DBDS.



IRLM Reconnect option

Addresses operational problem with DEDB areas:

When IRLM fails, IMS does not fail. Instead, it quiesces its work and waits to reconnect to IRLM. It also stops all of the DEDB areas. Previous releases of IMS did not restart these areas when IMS reconnected to IRLM. Operators had to issue the /START AREA commands. IMS V9 gives users the option to cause IMS to automatically restart the areas when the reconnect is done.

The new IRLM reconnect options are controlled by the FPRLM start up parameter.

•The N option causes IMS to act as it did with previous releases. It does not start areas when the reconnection to IRLM is made.

•The P option applies to areas with the PREOPEN parameter. They are started and opened when IMS reconnects to IRLM regardless of their status prior to the IRLM abnormal termination.

•The S option causes IMS to start all of the areas which were stopped during the IRLM disconnect. The starts are done when IMS reconnects to the IRLM.

•The R option starts and opens only those areas which were stopped as part of the IRLM disconnect process. It does not start areas which were already stopped, including those with PREOPEN specified to DBRC. Message DFS3210I is issued after all of these areas have been started.

Additional IMS V9 Fast Path (FP) Enhancements

- EMHQ Structure now optional in IMS V9
- FP Serviceability/Usability Enhancements
  - Allow online forward recovery activation with /STA DB command
  - Respectively add AREAname & #processed CIs into various msgs & SCAN job output
  - Bypass the need for dataspace when no VSO
  - Add FP log rec DSECTs in ILOGREC
  - Add new FP Table Trace similar to DL/I table trace format
  - Add SDEP statistics to x'5937' log record
  - Add NBA/OBA values for the associated dependent region to x'5937' log record
  - Add EPST/EMHB/ESRT to x'6706' log record
  - ADD STOPAFTERALL # parameter to HSRE utility
  - Enhance FP HSSP diagnostics
  - FP Abend reduction (U3275, U2484, U0732, U3999)

Prior to IMS V9, an EMHQ structure and its associated datasets are required in a Shared Queues environment with Fast Path installed even if users only use DEDBs and/or local EMH processing. V9 allows the EMHQ structure to be optional.

There are numerous miscellaneous enhancements in the Fast Path area that satisfy serviceabililty and usability user requirements. They are listed here:

•The /START DB command now activates online forward recovery (OFR) for DEDBs on an RSR (Remote Site Recovery) tracking system. You no longer need to start each area individually.

•Add area name to DFS2634 DFS2635 and DFS2636 messages.

•Data spaces for VSO DEDB areas are now created when the VSO area is opened, rather than during IMS initialization. Data spaces are not allocated if VSO is not used.

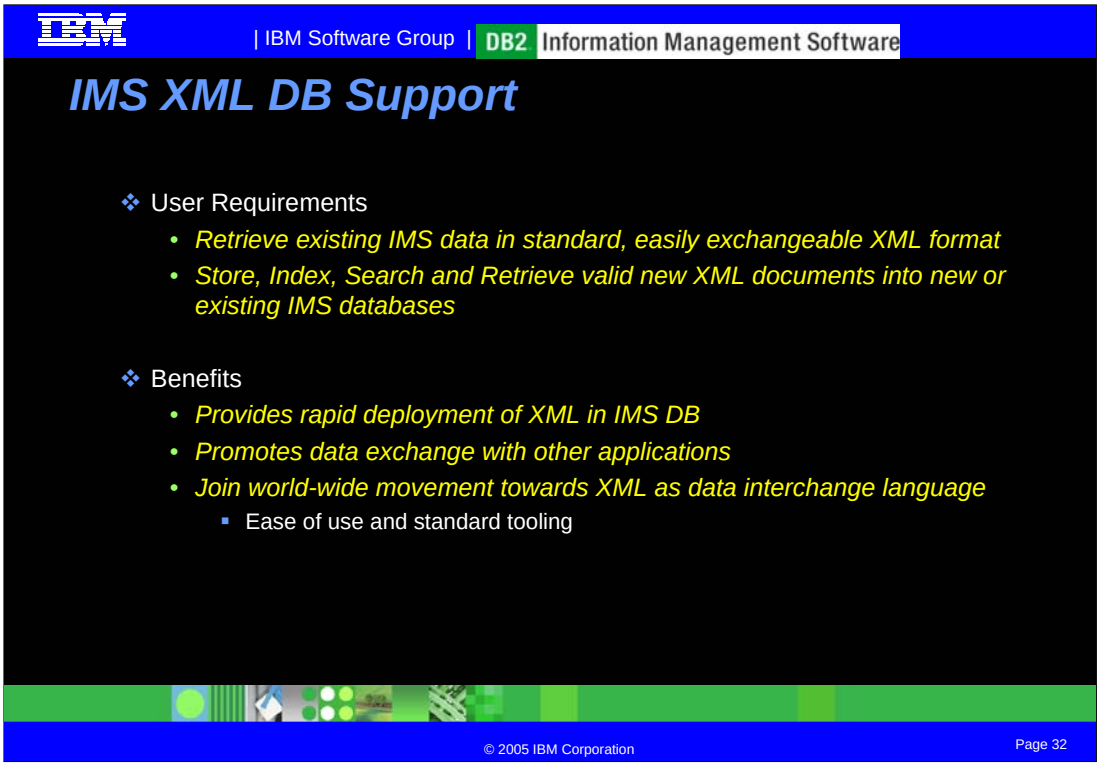
•Add the number of inserted SDEP CIs and SDEP segments to x'5937' log records.

•The FP dependent region blocks are logged to the x'6706' log record during dependent region abnormal termination if FP is in the system.

•Add STOPAFTERALL to High Speed DEDB Reorganization Utility to allow users to specify the number of UOW reorg failures, due to insufficient buffers for IOVF, before the utility is terminated.

•To enhance FP HSSP diagnostics add the IMS jobname of the HSSP job in the DMAC.

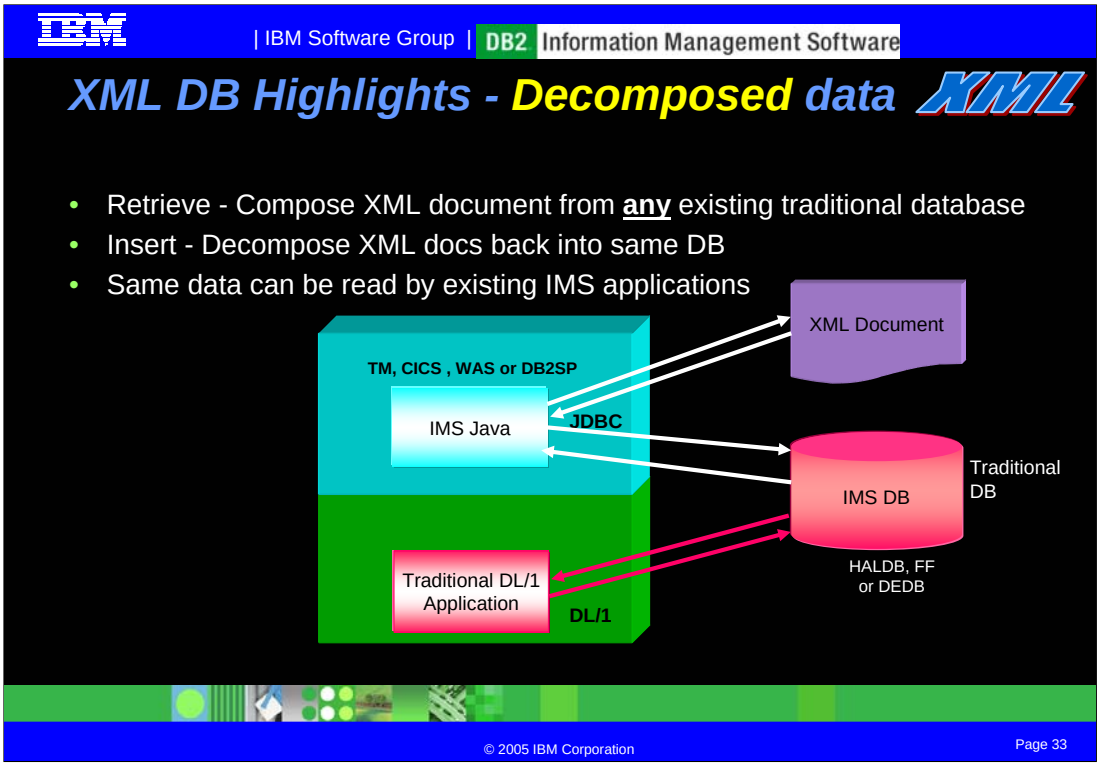
•Reducing Abends: U3275,U2484,U0732,U3999 abends are eliminated where appropriate.



The *eXtensible Markup Language* (XML) is a standardized, self-describing markup language for documents that contain structured or semi-structured information. XML is comprised of a set of tags (called *elements*) that are used to describe information in Unicode or EBCDIC text and organize that information hierarchically. XML technologies are used in the exchange and storage of application data, and to organize text-based documents.

IMS Java in IMS Version 9 supports the storage of XML in an IMS database. Some of the user requirements for this support include the need to retrieve existing IMS data in a standard, and easily exchangeable XML format and the ability to store, index, search, and retrieve valid, new XML documents into new or existing IMS databases.

There are many benefits to these new enhancements. You exploit the ease of use and Open standard tooling available with XML. Examples are transformation tools to other formats like PDF and HTML and validation tools that check that the XML schema matches the document. These allow you to rapidly deploy XML in an IMS DB. Other applications are able to access this data, promoting data exchange between applications. IMS joins the world-wide movement of using XML as a data interchange language, which is important for providing future transparent application integration.



With decomposed document storage, IMS converts the XML data to traditional IMS data types, and as a result, the data can be searched by IMS. With the XML DB enhancements, you can compose XML documents from traditional data in existing IMS databases. You can also receive XML documents, and decompose them and store their data back into existing IMS databases as traditional data. Tooling (the DLIModel utility) provides assistance to define the required metadata mapping between XML documents and IMS databases.

This support allows users to write Java applications to compose and receive XML documents in any of the following environments:

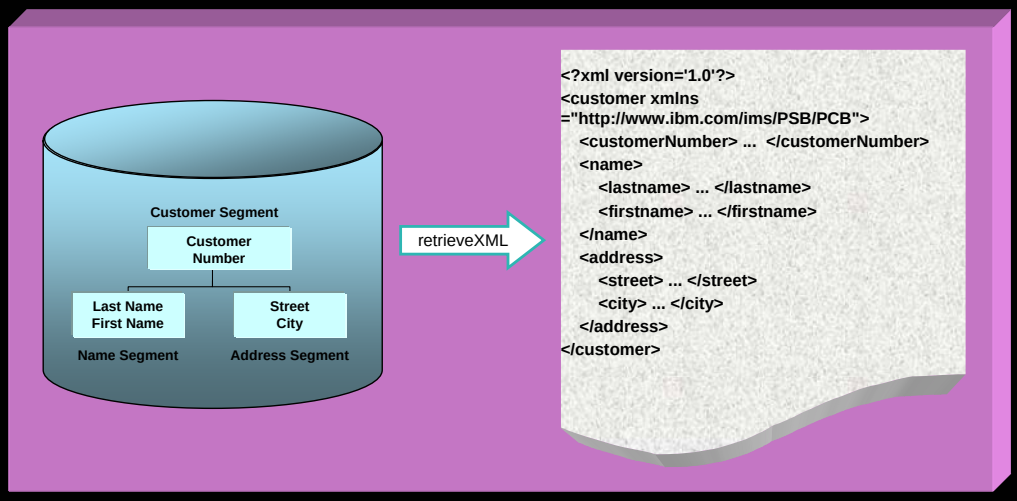
- IMS dependent region (JMP or JBP)
- WebSphere Application Server for z/OS
- WebSphere Application Server on a non-z/OS platform
- DB2 UDB for z/OS stored procedure
- CICS JCICS region

Decomposed XML storage

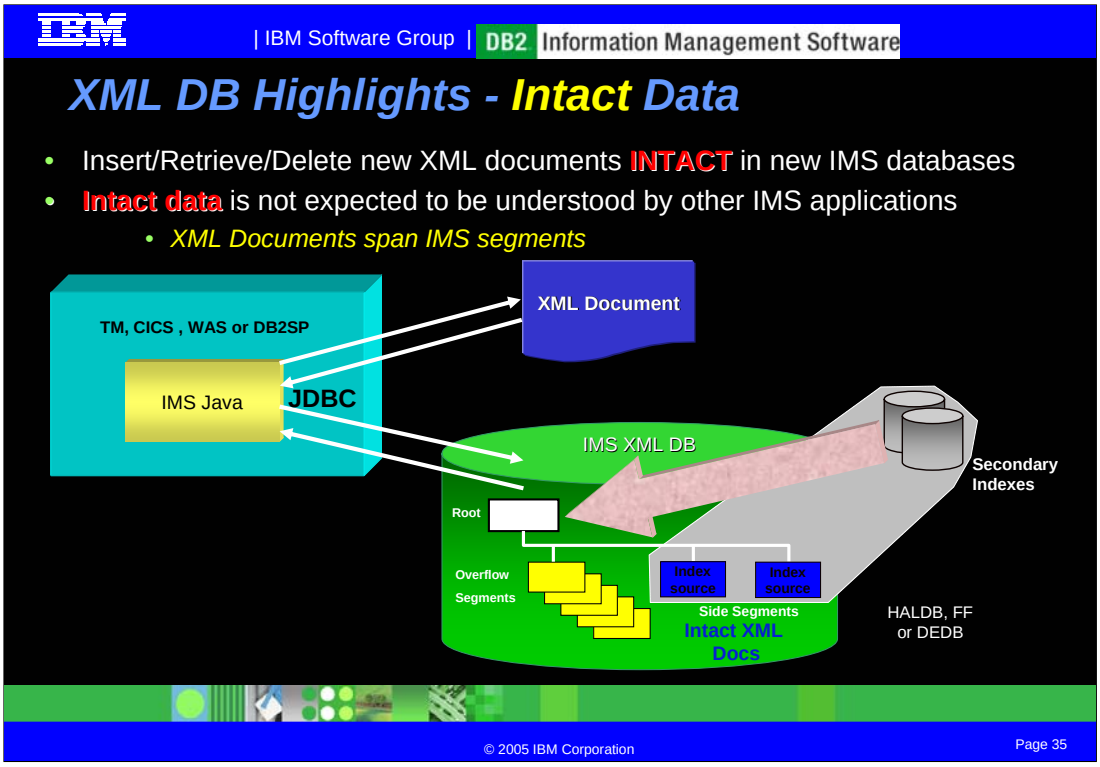
The XML tags are stripped from the XML document and only the data is extracted. The extracted data is converted into traditional IMS field types and inserted into the database. Use this approach in the following scenarios:

- XML applications and non-XML applications must access the same database
- Extensive searching of the database is needed
- A strict XML schema is available

Decomposed XML Materialization



The hierarchical nature of this "data-centric" document lends itself to storage in an IMS database. Examples of "data-centric" documents are invoices and bill of materials where the order of the data is not important and the size of the document is fixed. The retrieveXML JDBC call from the IMS Java application program causes IMS to materialize the decomposed XML data.



Intact XML storage

In this case the XML document is stored, with its XML structure and tags intact, in an IMS database designed exclusively for storing intact XML documents. In this case, only Java application programs can access the data in the database. Because the XML document does not have to be regenerated when the data is retrieved from the database, the retrieval of the XML data is typically faster than when it is stored without its XML tagging. Use this approach in the following scenarios:

- Faster storage and retrieval of XML documents are needed
- Less searching of the database is required
- The XML schema requires more flexibility

Intact XML storage requires a new IMS database or an extension of an existing database because the XML document must be stored in segments and fields that are specifically tailored for storing intact XML. To store all or part of an XML document intact in an IMS database, the database must define a base segment, which contains the root element of the intact XML sub-tree. The rest of the intact XML sub-tree is stored in overflow segments, which are child segments of the base segment.

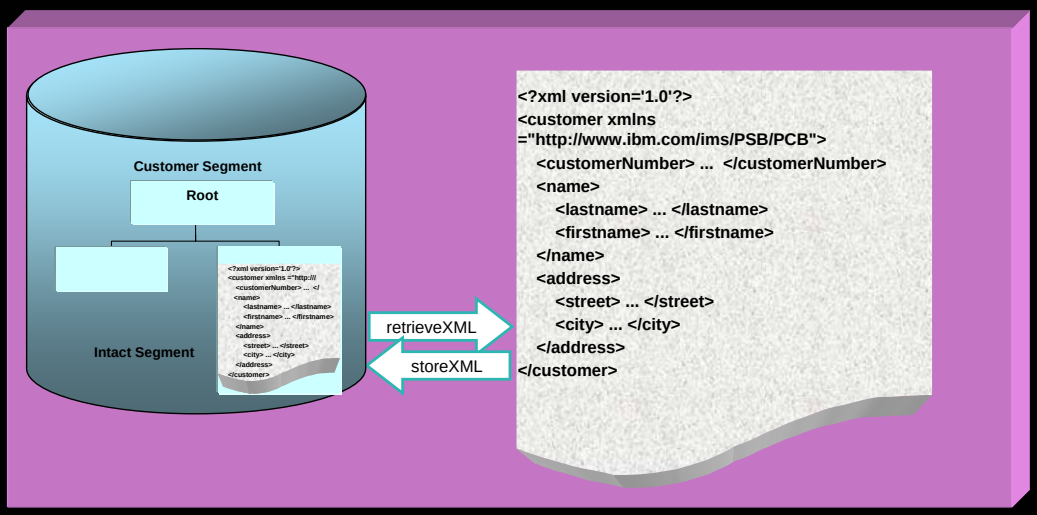
Side Segments for Secondary Indexing

IMS cannot search intact XML documents for specific elements within the document. However, you can create a side segment that contains specific XML element data. IMS stores the XML document intact, and also decomposes a specific piece of XML data into a standard IMS segment. This segment can then be searched with a secondary index.

This support allows users to write Java applications in any of the following environments:

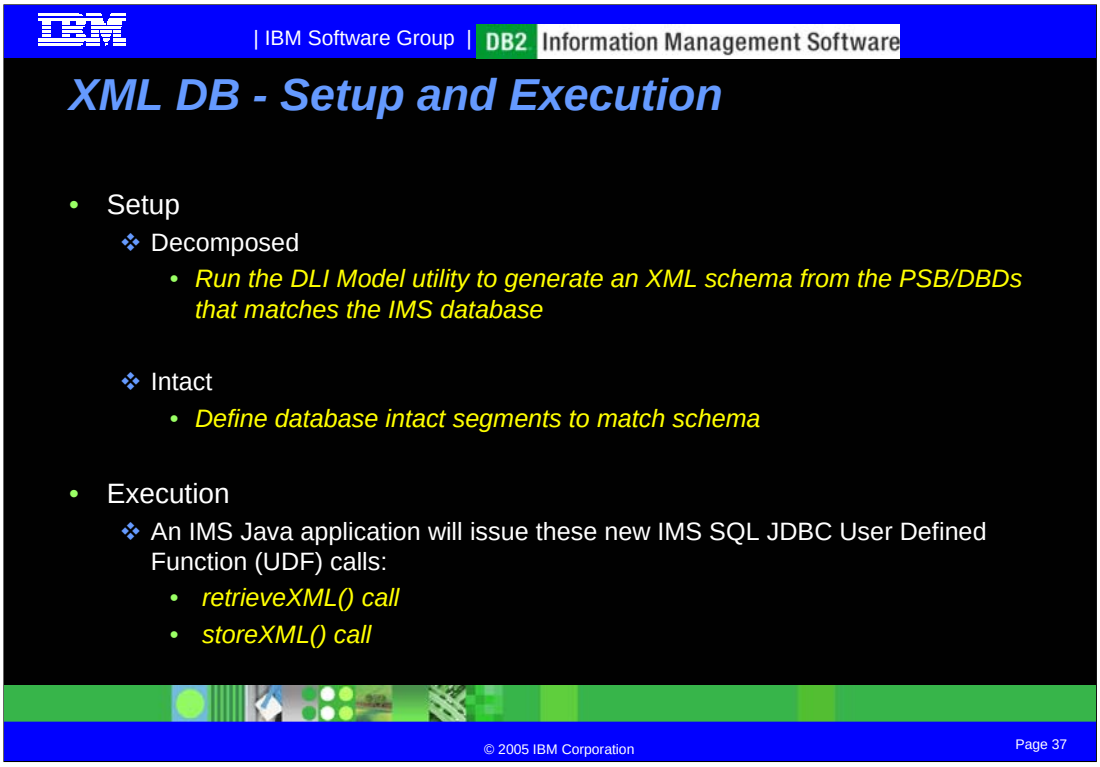
- IMS dependent region (JMP or JBP)
- WebSphere Application Server for z/OS
- WebSphere Application Server on a non-z/OS platform
- DB2 UDB for z/OS stored procedure
- CICS JCICS region

Intact XML document



Intact data is "document-centric", meaning that it is loosely structured, and the order of elements is significant. The data is mostly character data and the document is of unpredictable size, unlike "data-centric" content, which is of limited size with strongly-typed data elements. Examples of "document-centric" data are newspaper articles and manuals.

Storing intact data is desirable because the content of the document does not need to be processed as regular IMS data where retrieval speed is important. This mode is appropriate for "document-centric" documents, where you do not need to process the data in the document as normal IMS data, but you want to store the document centrally with the benefits and security provided by IMS.



The elements of the XML DB enhancement that provide the support for decomposed and intact data described in the prior charts are the following: the DLIModel utility that is part of IMS Java and two new User Defined Functions (UDF) to the IMS JDBC SQL interface available to an IMS Java application program: retrieveXML() and storeXML().

In order to setup decomposed data, use the DLIModel utility to setup an XML schema that matches the IMS database, using the PSBs and DBDs as input to the utility. New DLI Model control statement changes include these new options: GenXMLSchemas=retrieve/store/no and XMLSchemaPath=path.

For intact data, define the database intact segments to match the schema for the document.

XML database support adds the following new API calls:

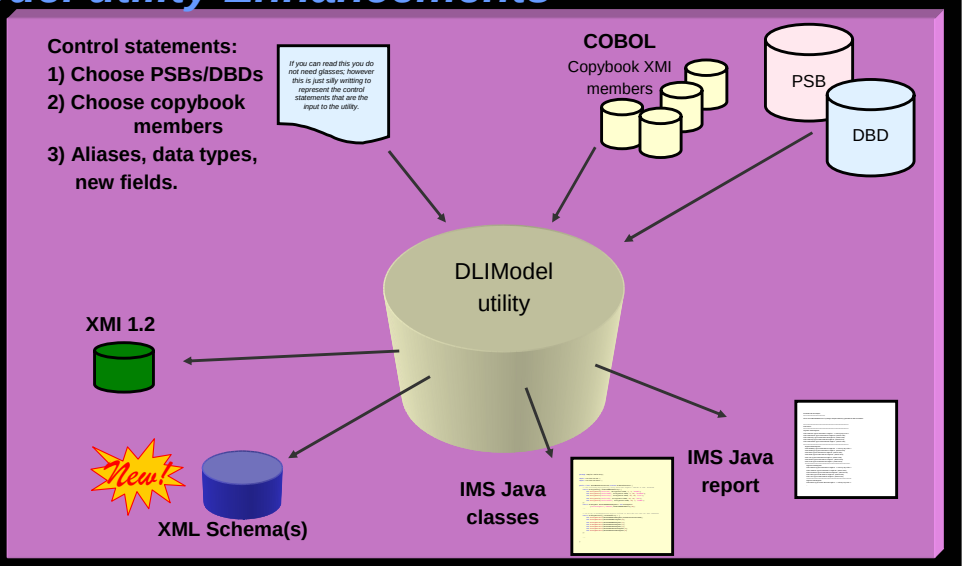
retrieveXML()

Retrieves an XML document from an IMS database. The required argument for this call is the desired IMS root segment, which must be in the path from the current segment. Mapping is controlled by the XML schema. An application program issues a JDBC SELECT statement to retrieve an XML document from the database. The SELECT statement must identify the XML root element segment.

storeXML()

Inserts the specified XML document into an IMS database at the current position in the database. The document is inserted starting at a root element segment. Mapping is controlled by the XML schema.

DLIModel utility Enhancements



IMS Version 9 XML database support is provided as an extension to IMS Java. This support allows users to write Java applications to compose and receive XML documents in all of the environments that are supported by IMS Java: IMS Java JMP or JBP application programs, WebSphere EJBs, CICS application programs, or DB2 stored procedures.

XML Schema Definition Language defines the legal building blocks of a valid XML document. XML database support allows you to use the XML Schema Definition Language to create an XML Schema that:

•Describes the mapping between a valid XML document and the underlying hierarchic IMS database.

•Describes the full set of valid incoming and outgoing XML documents.

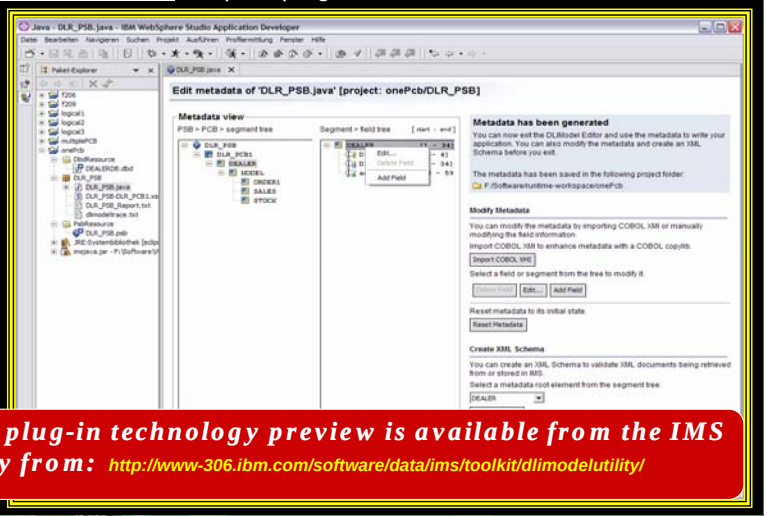
You can automatically create the XML Schema that represents a given IMS database view (defined by a PSB, PCB, and its referenced DBDs) using the IMS Java DLIModel utility.

The generated XML Schema is an XML document that describes an IMS database based on a PCB. An XML Schema is required to retrieve or store XML in IMS. IMS uses an XML Schema to validate an XML document that is either being stored into IMS or being retrieved from IMS. The XML Schema, not the application program, determines the structural layout of the XML in the database. The DLIDatabaseView subclass determines how the data is physically stored in the database. The DLIModel utility generates a Schema based on a PCB. The generated XML Schema must be made available at runtime in order to provide the XML structure of the data retrieved from the database or conversely of an incoming XML document being stored into IMS.

While storing XML documents we first verify that the document is valid to the XML Schema generated by the DLIModel utility. Invalid documents cannot be stored, and will throw an exception back to the user application. Validating against the XML Schema allows us to make sure that the structure of the XML document matches that of the underlying IMS database. If the structure of the IMS database changes (structural reorg for instance), the DLIModel utility will need to be rerun in order to produce a new XML Schema (the IMS Java DLIDatabaseView metadata will also need to be refreshed under such a case).

DLIModel Utility GUI

- DLIModel utility *Graphical User Interface* Eclipse "plug-in"
  - Allows you to generate
    - *Schema*
    - *IMS Java metadata classes*
    - *Java Report*



The DLIModel Utility plug-in technology preview is available from the IMS Home page or directly from: *http://www-306.ibm.com/software/data/ims/toolkit/dlimodelutility/*

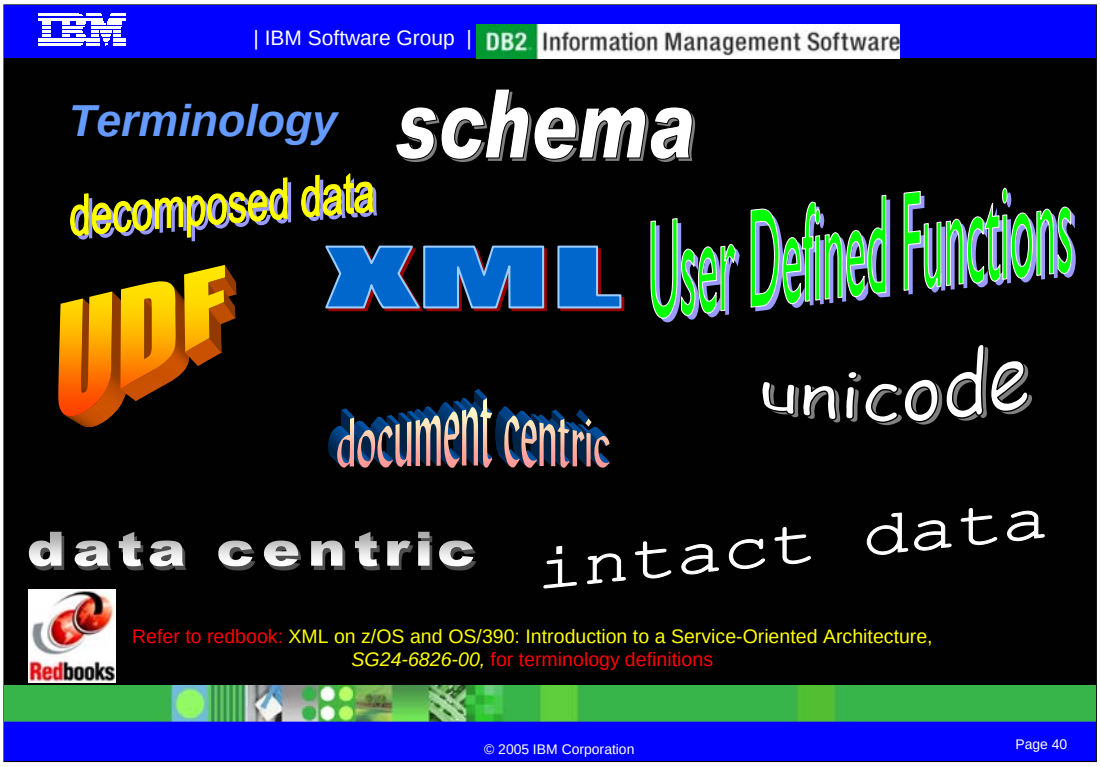
The DLIModel utility graphical user interface (GUI) provides a graphical version of the DLIModel utility built as an Eclipse plug-in. The DLIModel GUI simplifies IMS metadata generation, eases IMS Java and XML database application development, and offers a visual representation of IMS databases.

The plug-in works with WSAD(IE) 5.1+ or any freely available Eclipse of version 2.1.1+ and can be installed via the so called "Update Manager" coming with Eclipse/ WSAD(IE). This is a standard built-in component of Eclipse/WSAD(IE) allowing you to easily add new features, such as the DLIModel GUI, to your (base) product by just specifying a link to the download site of that particular feature. The install process will be described on our web site, but as it is the standard way of adding new features, it can also be found in the standard Eclipse/WSAD(IE) online help.

The help for this (technology preview) DLIModel GUI will only be available online and not in the IMS utilities manual.

The functionality provided by the GUI can basically also be used with IMS V7 and IMS V8 as well, except the generated XML Schema files can only be used with IMS V9. The IMS Java metadata class generated by the interface can also be used for IMS Java programs running in version 7 and version 8.

In its first release, the GUI basically provides all the functionality offered by the command line tool except for full XML support. That is, the control statements *XMLType*, *XMLStorageType*, *Overflow* (on Fields) and *SIDESEG* are not supported by the GUI. But it is possible to specify an XMLRootElement for the Schema. Moreover, no XMI output is generated for compatibility reasons. The GUI allows to generate Schema, the IMS Java metadata class and the Java Report (and the trace file).



Refer to the redbook for terminology definitions.

IMS Java Remote Database Services (RDS)

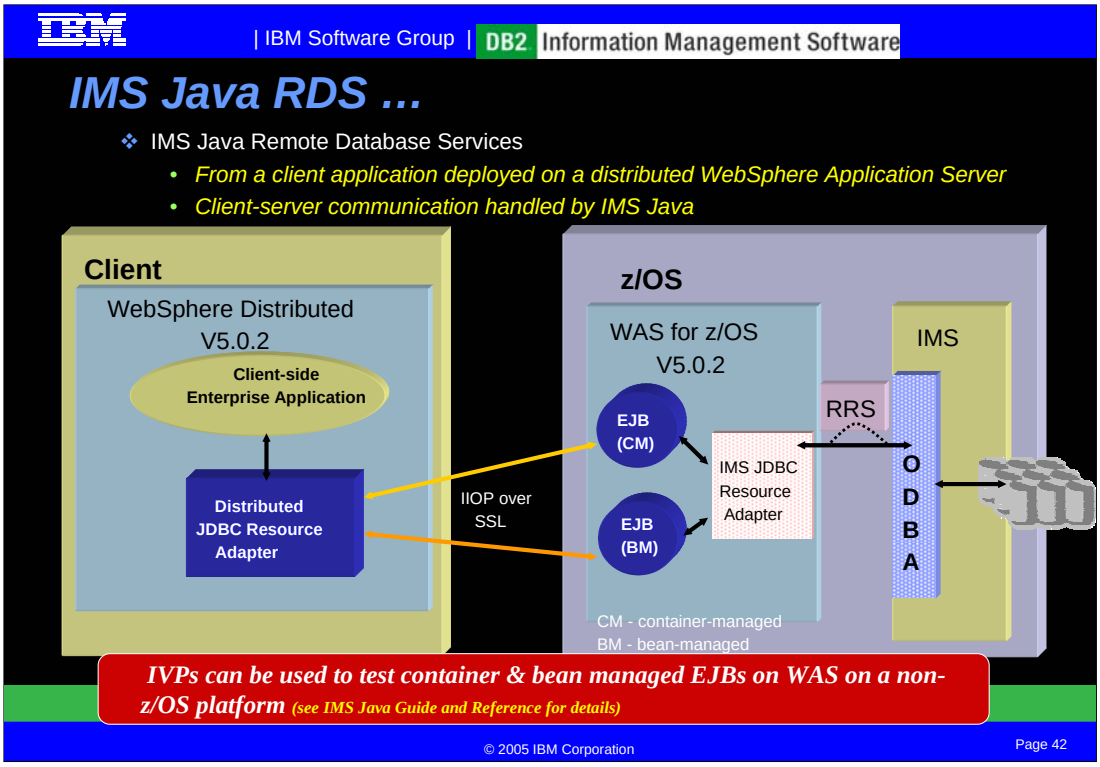
- Set of J2EE components that provide remote access to IMS data (IMS ODBA)
  - Consists of client-side and server-side components
  - Supports: Retrieve, Update, Delete, Insert
  - Provides an architected solution that allows remote Enterprise applications to issue JDBC calls to access IMS Databases
- Benefits
  - No need to develop or access a legacy z/OS application to have access to IMS data – ideal for application development in a WebSphere environment

JAVA J2EE Java 2 Platform, Enterprise Edition

In addition to the requirements for IMS Java, IMS Java Remote Database Services also requires:

•WebSphere Application Server for z/OS Version 5.0.2.3 or later. If you have WebSphere Application Server for z/OS V5.0.2, you must install either V5.0.2.1 or apply APAR PQ81944.

•WebSphere Application Server V5.0.2.2 with cumulative fixes that include PQ79485, or WebSphere Application Server V5.0.2.3 or later.



The IMS Java Remote Database Services enhancement in IMS V9 allows distributed database access to IMS database data that resides on a different platform from a JDBC application running as an Enterprise Java Bean (EJB) on a WebSphere Application Server. The need for the EJB to go over the network to access the IMS database data is transparent to the application itself. WebSphere works with the Resource Recovery Service (RRS), which is the transaction coordinator.

Unlike other Java solutions for IMS, you do not need to develop a z/OS application or access a legacy z/OS application to have access to IMS data. Therefore, IMS Java is an ideal solution for IMS application development in a WebSphere environment.

The following components are used for an enterprise application on a non-z/OS platform to access IMS DB:

non-z/OS platform The operating system that WebSphere Application Server V5 runs on.

WebSphere Application Server WebSphere Application Server V5.0.2 on which the client application runs.

EJB The enterprise application (an EJB in this case) that contains your business logic, and is deployed on WebSphere Application Server. This enterprise application can be either container managed or bean managed. The enterprise application can be transactional.

IMS distributed JDBC resource adapter The resource adapter that is deployed on the non-z/OS platform. It contains a type-3 JDBC driver.

IIOP (Internet Inter-ORB Protocol) IIOP is the protocol that can be used between WebSphere Application Server for z/OS and WebSphere Application Server running on another platform. IIOP allows the servers to exchange data. Data is securely transferred across the Internet using the SSL (Secure Sockets Layer) protocol.

WebSphere Application Server for z/OS WebSphere Application Server V5.0.2 for z/OS is required to manage transaction protocol and communication with RRS. It must reside on the same z/OS LPAR (logical partition) as IMS.

IMS Java EJB One of two IMS Java-supplied EJBs is the host-side component that facilitates communication with and passes transaction information to the IMS JDBC resource adapter. These EJBs act as listeners for remote requests. Depending on whether there is a transaction context on the non-z/OS platform, either a container-managed or bean-managed IMS Java EJB is used.

IMS JDBC resource adapter The IMS JDBC resource adapter that is deployed on the z/OS platform. It contains a type-3 JDBC driver.

ODBA Open Database Access is the IMS callable interface for access to IMS DB.

DRA The database resource adapter (DRA) is the bridge between the external subsystem and IMS.

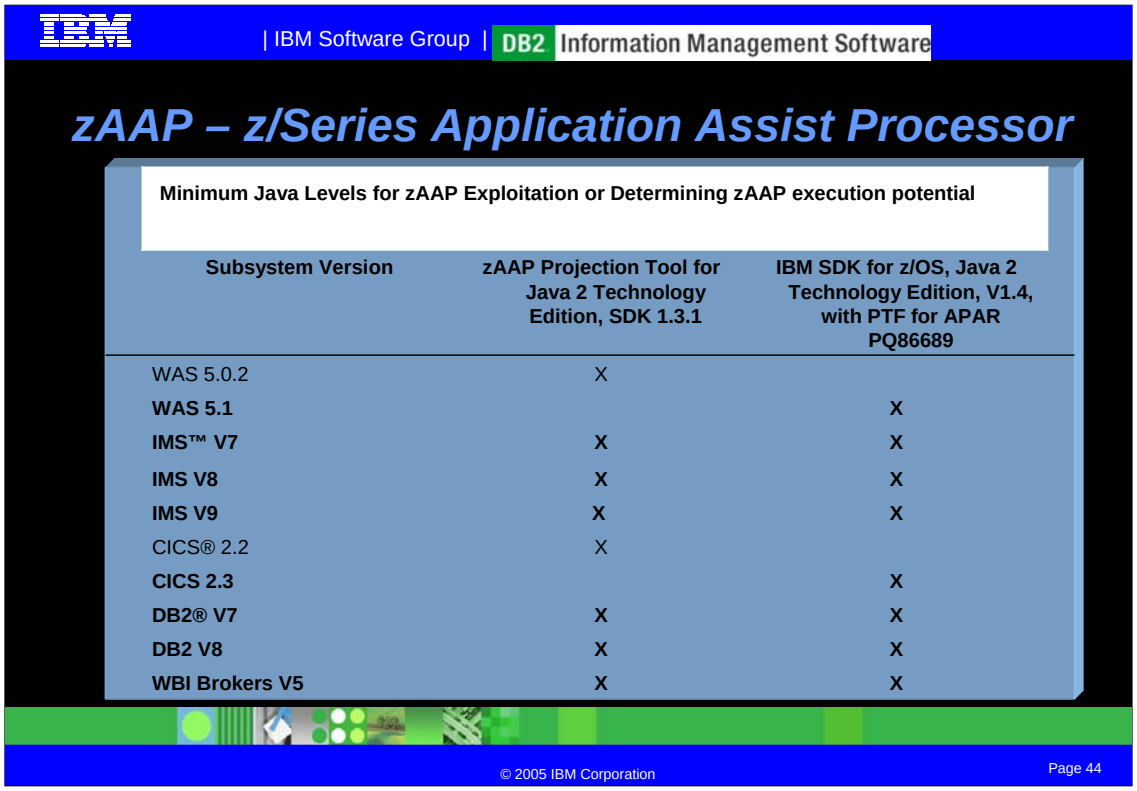
DL/I DL/I is the standard interface to IMS data.

zAAP – z/Series Application Assist Processor

- Specialized processing unit for z890 and z990 processors
  - Requires z/OS 1.6
- Provides a highly strategic z/OS Java execution environment
  - Increases system productivity by reducing the demands on general purpose processors, making capacity available for reallocation to other workloads.
  - Allows Java workloads to run at a lower total cost of ownership
  - Java workloads are transparently executed on the zAAP processors
    - *No requirement to change applications*

IMS JAVA support for the new IBM zSeries Application Assist Processor (zAAP) for the IBM zSeries 990 (z990) and IBM zSeries 890 (z890) servers allows Java workloads to transparently execute on the zAAP processors without requiring application change. This allows you to integrate and run Java workloads on the same server as your database at a significantly lower total cost of ownership than previously possible. This helps reduce the overall cost of computing for Java applications, increase system productivity by reducing the demands on general purpose processors, and makes capacity available for other workloads.

zAAPs are designed to operate asynchronously with the general CPs to execute Java programming under control of the IBM Java Virtual Machine (JVM). This is an important point as zAAPs can only help execute Java applications and application servers that use the IBM JVM. The IBM JVM processing cycles can be executed on the configured zAAPs with no anticipated modifications to the Java application(s). Execution of the JVM processing cycles on a zAAP is a function of the IBM Software Developers Kit (SDK) for z/OS, Java 2 Technology Edition, V1.4 with PTF (or later) for APAR PQ86689, z/OS 1.6, and the Processor Resource/Systems Manager (PR/SM).



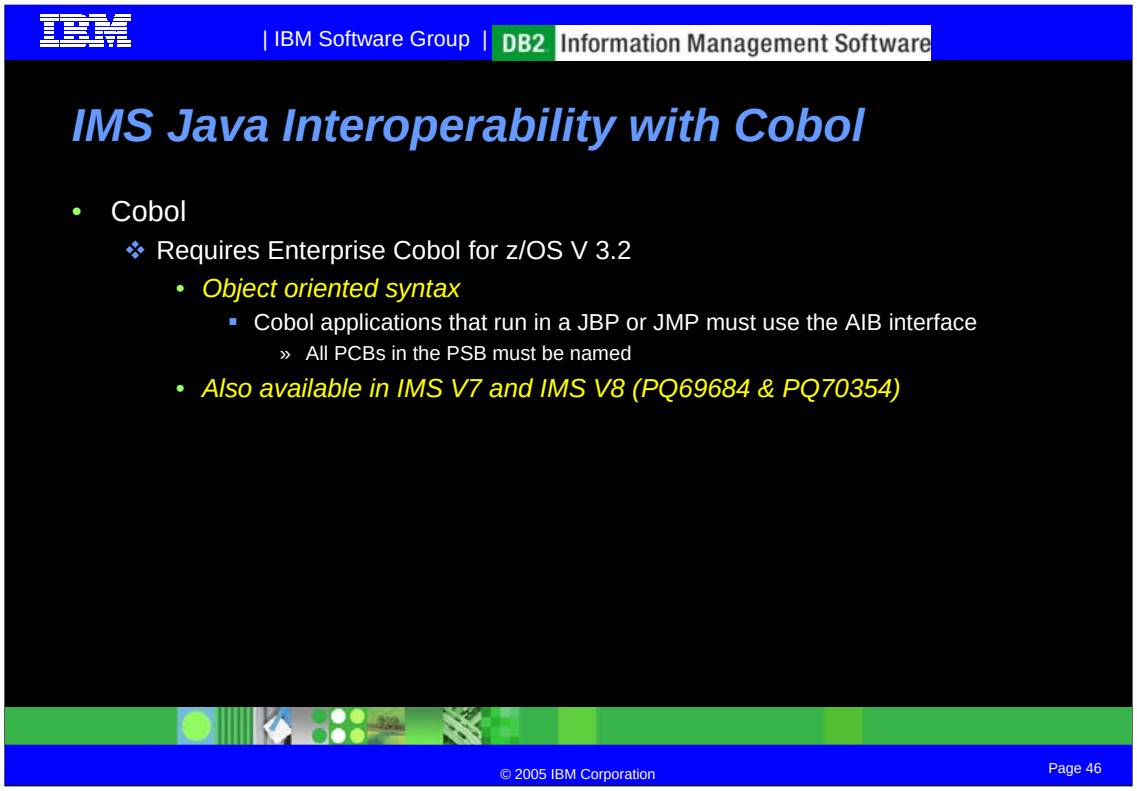
Before you have z/OS 1.6, Java SDK 1.4, or a z890 or z990 server, you can do some capacity planning to determine how many zAAPs you may need. There is a projection tool (zAAP Projection Tool) which is a modified Java SDK 1.3, that has some of the same functionality which has been incorporated into Java SDK 1.4 and higher. This tool will gather usage information about how much CPU time is spent executing Java code which could potentially execute on zAAPs. By running a Java workload that is representative of the production system operations, it will report, via the Java log, how much of that workload could be eligible for execution on zAAPs. This information is also useful in predicting the number of zAAPs that might be necessary in order to provide optimum zAAP configuration.

Here is a table outlining the subsystems and minimum Java levels dependencies for determining zAAP Java execution and exploitation potential. The items in bold identify some of the product levels that will exploit zAAPs. The items in normal font, while not able to exploit the benefits of zAAPs, show where the zAAP Projection Tool for Java 2 Technology Edition, SDK1.3.1 can be used to assist in zAAP capacity planning. For instance, WebSphere V5.0.2 cannot exploit zAAP, however the zAAP Projection Tool can be used to determine zAAP Java execution potential if the WebSphere workloads were migrated to the required level for zAAP exploitation.

Other IMS Java Enhancements…

- IMS-DB2 interoperability
  - Supports DB2 access from an IMS JMP or JBP region
    - *Part of base IMS V9*
    - *IMS V8*
      - PQ73326 (UQ80615)
      - Prereq PQ73897 (latest level of IMS Java)
      - Prereq PQ75284 (IMS code using DB2's attachment facility)
    - *DB2*
      - DB2 V8 (PQ74629)
      - DB2 V7 (PQ69861)
  - Uses DB2 RRS attach facility
    - *Requires SSM= and RRS=Y in the IMS startup definitions*

IMS-DB2 Interoperability provides the ability to connect to DB2 databases from within an IMS Java application running in either a JMP or JBP region. You can access both DB2 and IMS databases from within the same application. This enhancement uses the RRS attachment facility, and is also be available in IMS V8. JMP and JBP applications can access DB2 databases that are on the same z/OS image. For JMP or JBP applications to have DB2 access, you must attach DB2 to IMS using the DB2 Recoverable Resource Manager Services attachment facility (RRSAF). Unlike other dependent regions, JMP and JBP regions do not use the External Subsystem Attach Facility (ESAF). With RRSAF, the dependent region builds an attachment thread to DB2 using RRS. RRS coordinates the commits of the updates that the application program makes to both IMS and DB2 resources. IMS is a participant, not the coordinator, of these updates and commits.



With this support, you can:

•Call an object-oriented (OO) COBOL application from an IMS(TM) Java application by building the front-end application, which processes messages, in Java and the back end, which processes databases, in OO COBOL.

•Build an OO COBOL application containing a main routine that can invoke Java routines.

Restriction:

COBOL applications that run in an IMS Java dependent region must use the AIB interface, which requires that all PCBs in a PSB definition have a name.

IMS Java Enhancements …

- IMS Java Symbolic checkpoint/restart
  - Extends the IMS symbolic checkpoint/restart capability to the IMS Java environment
    - *Only applies to JBP (Java Batch Processing) applications*
    - *Take application checkpoints*
      - Restart from the last checkpoint after failure
    - *Save critical application Java objects as part of the checkpoint*
      - Restore those objects at restart
    - *Save database positions as part of the checkpoint*
      - Verify repositioning of those databases at restart

Similarly to BMP applications, JBP applications can use symbolic checkpoint and restart calls to restart the application after an abend. The primary methods for symbolic checkpoint and restart are:

IMSTransaction().checkpoint()

IMSTransaction().restart()

These methods perform analogous functions to the DL/I system service calls: (symbolic) CHKP and XRST. A JBP application connects to a database, makes a restart call, performs database processing, periodically checkpoints, and disconnects from the database at the end of the program. The program must issue a final commit before ending. On an initial application start, the IMSTransaction().restart() method notifies IMS that symbolic checkpoint and restart is to be enabled for the application. The application then issues periodic IMSTransaction().checkpoint() calls to take checkpoints. The IMSTransaction().checkpoint() method allows the application to provide a com.ibm.ims.application.SaveArea object that contains one or more other application Java objects whose state is to be saved with the checkpoint. If a restart is required, it is initiated in a similar way to BMP applications: the checkpoint ID is provided either with the IMSTransaction().restart() call (similar to providing the id to the XRST call in IMS), or with in the CKPTID= parameter of the JBP region JCL. The restart() method returns a SaveArea object that contains the application objects in the same order in which they were originally checkpointed.

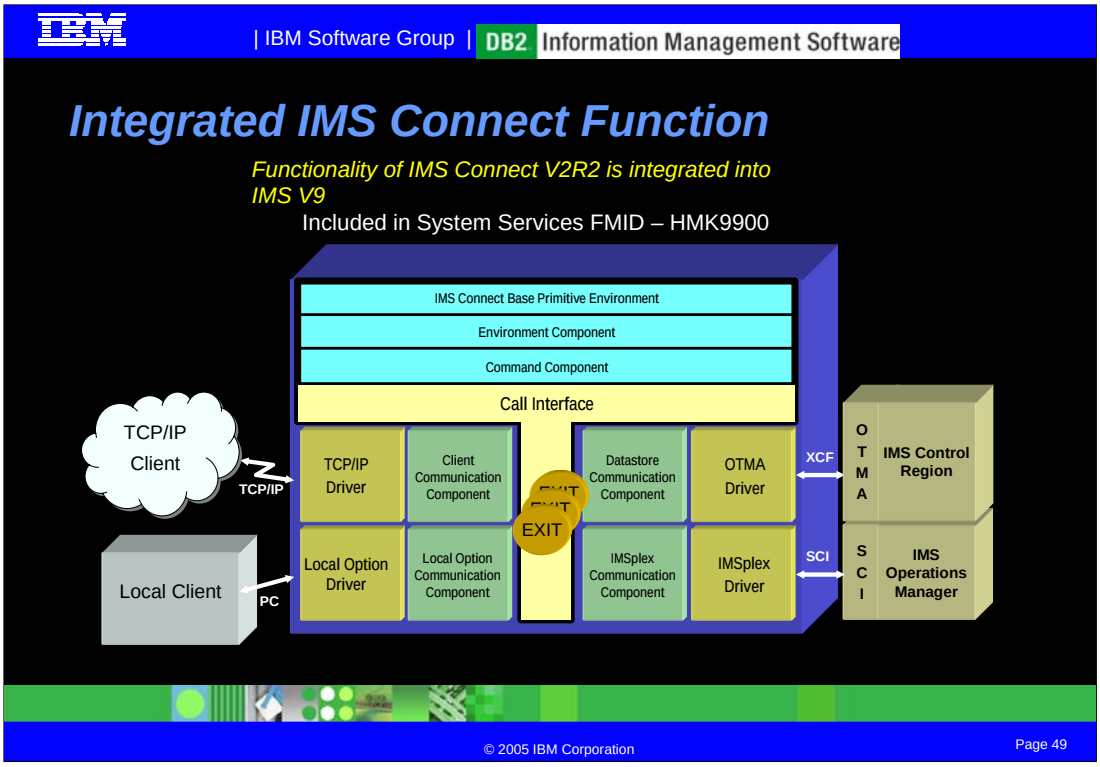
IMS Java Enhancements …

- IMS Java GSAM Support
  - Provides a Java interface to access GSAM
    - *Only applies to JBP (Java Batch Processing) applications*
      - Access (open and close) GSAM databases
      - Write to GSAM databases
      - Read from GSAM databases (sequentially and randomly)
- XPLINK Support with IBM SDK for z/OS V1.4.1
  - JMP and JBP regions MUST specify XPLINK=Y
    - *IBM LE Extra Performance Linkage runtime option*
  - Also supported in IMS V7 & V8 (PQ76609 & PQ77590)
- IMS Java SQL Enhancements
  - Select all fields within multiple segments using the asterisk (*) operator
    - *SELECT * FROM DealershipDB.Model*
    - *SELECT Dealer.*, Model.* FROM DealershipDB.Model*

IMS Version 9 adds the XPLINK= dependent region parameter for Java Message Processing (JMP) regions and Java Batch Processing (JBP) regions. This parameter allows IMS Java dependent regions to use the IBM Language Environment Extra Performance Linkage (XPLINK) runtime option. If you use IBM SDK for z/OS, Java Technology Edition, Version 1.4.1 or later, you must specify XPLINK=Y for a JMP region or a JBP region. However, if you use Version 1.3.1 of the SDK, you should specify XPLINK=N or allow IMS to use XPLINK=N as the default.

IMS JDBC provides added functionality to select all fields from multiple segments in an SQL SELECT clause. The asterisk (*) operator can now be used in conjunction with a segment name. For example: SELECT Dealer.*,Model.* FROM DealershipDB.Model

The IMS JDBC driver provides the ability to use non-searchable fields (fields that are not defined in a DBD) in an SQL WHERE clause, provided those fields are a subset of a searchable field. This item will be delivered through the IMS V9 service process.

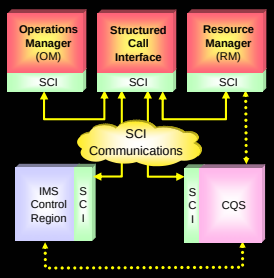


IMS Version 9 provides an integrated IMS Connect function that offers a functional replacement for the IMS Connect tool (program number 5655-K52). The integrated IMS Connect is included in the IMS System Services function modification identifier (FMID), HMK9900; the integrated IMS Connector for Java for z/OS is included with the IMS Java FMID, JMK9906; and the integrated IMS Connector for Java distributed can be downloaded from the IMS Web site: www.ibm.com/software/data/ims/

The integrated IMS Connect support allows high performance communications with advanced security and transactional integrity between one or more TCP/IP or local (z/OS) clients, and one or more IMS subsystems. This support provides commands to manage the environment and assist with workload balancing. For a description of the IMS Connect product, see www.ibm.com/software/data/db2imstools. You can use the integrated IMS Connect support with the WebSphere Development Tooling and the IMS Connector for Java to ease the development of on demand business solutions that access IMS transactions. You can also use the integrated IMS Connect support with the IMS Control Center for a graphical user interface to control both IMS and DB2.

You can use the base function of the integrated IMS Connect function with terminals, applications, and data running under IMS Version 7, Version 8, or Version 9. However, Distributed Operations support, as well as the separately orderable IMS Connect 2.2 tool functionality (such as distributed two-phase commit support), is supported only for IMS Version 8 and IMS Version 9. IBM continues to support the separately orderable IMS Connect tool for IMS Version 7, Version 8, and Version 9 use, thus easing IMS Connect tool migration to the new integrated IMS Connect function. Future enhancements to the IMS Connect functionality will be made available only with IMS Version 9 or later.

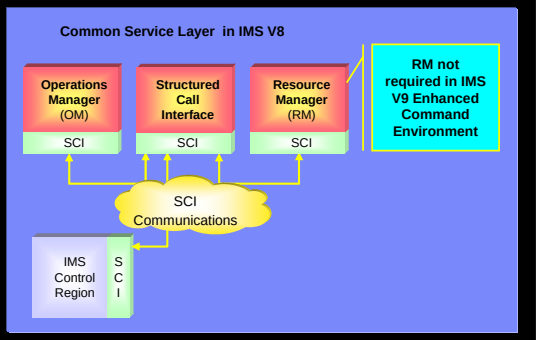
Common Service Layer



- IMS V8 introduced the CSL
  - Common Service Layer (CSL)
    - *An architecture, not an address space*
  - Three new address spaces built on the Base Primitive Environment (BPE)
    - ***Structured Call Interface (SCI)***
    - ***Operations Manager (OM)***
    - ***Resource Manager (RM)***
  - A new API provides communication between address spaces in the IMSplex
    - *the Structured Call Interface*
- Purpose
  - Improve the systems management capabilities of IMS systems running in an OS/390 or z/OS Parallel Sysplex

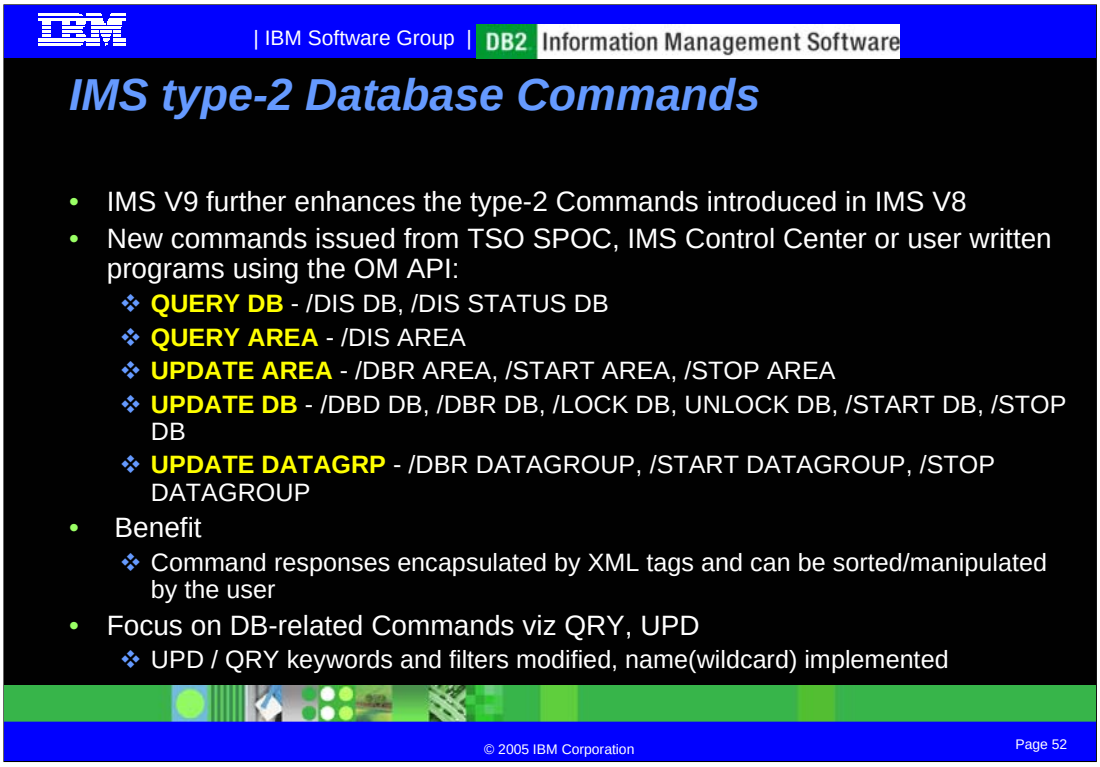
Enhanced Command Environment

- Resource Manager (RM) address space *not required* to support type-2 commands
- Allows the IMS control region to optionally start the SCI and OM address spaces




```
BROWSE    IMS.V9.DBDC.PROCLIB(DFSCGECE)
Command=➔_____________________________________
*--------------------------------------*
* IMS COMMON SERVICE LAYER MEMBER      *
*--------------------------------------*
CMSEC=N,              /* NO CMD AUTH CHECK*
IMSPLEX=PISC2,        /* IMSPLEX NAME     *
RMENV=N,
OMPROC=OMA,
SCIPROC=SCIA
```


IMS Version 9 provides the IMS Enhanced Command Environment, a simplified IMS configuration that does not require the Resource Manager. If you want to use the type-2 command format, but do not need the Resource Manager, you can group your systems into a generic IMSplex and use the IMS Enhanced Command Environment. (In IMS v8, all three address spaces of the Common Service Layer, OM, RM and the SCI are required to enter commands through the OM API.)



IMS Database Commands

The Operations Manager was delivered in IMS V8 and provided an OM API to allow IMS commands to be issued from OM. All existing commands, now referred to as 'type-1' IMS commands, can be issued from the OM API. A new command format called the 'type-2' commands was introduced. The type-2 commands are allowed only from the OM API and the command responses are encapsulated in XML tags. The output from all IMSs is consolidated and presented back to the SPOC or application that issued the OM API call. In IMS V8, type-2 commands for transaction resources were added to provide the user better ability to manage the IMSplex and also to provide a single system image. In IMS V9 the type-2 commands for database and area resources are added to provide the user the ability to better manage the IMSplex and provide a single system image.

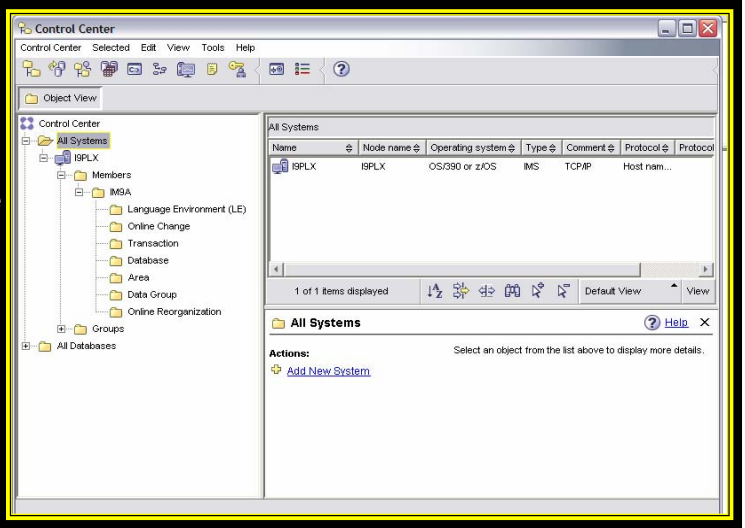
The following new type-2 commands are being added in IMS V9 to allow the user to query and update databases and to update DEDB areas: QUERY DB, QUERY AREA, UPDATE AREA, UPDATE DB, and UPDATE DATAGRP. The new type-2 database commands are allowed only from the OM API and the command response is returned to OM encapsulated in XML tags. In this release, these commands provide the same function as their similar type-1 commands - the input format and output format is different. Below is a list of the type-2 DB commands and the equivalent type-1 commands:

- QUERY DB - /DIS DB, /DIS STATUS DB
- QUERY AREA - /DIS AREA
- UPDATE AREA - /DBR AREA, /START AREA, /STOP AREA
- UPDATE DB - /DBD DB, /DBR DB, /LOCK DB, UNLOCK DB, /START DB, /STOP DB
- UPDATE DATAGRP - /DBR DATAGROUP, /START DATAGROUP, /STOP DATAGROUP

Additional filters, e.g. OPEN and OLR, are added to the UPD/QRY commands, as well as wildcard support for database and area names.

IMS Control Center

- Enhanced to support:
  - IMS V9 Resources
  - Command Shortcuts
  - Command editor
  - Results can be edited
  - Two sets of results can be compared
- Freely available with IBM DB2 V8.2 Admin Client



As from IMS V8 with the CSL, you can manage your IMS system/s using a graphical Control Center from a Windows workstation. The IMS Control Center is part of the IBM DB2 UDB Version 8 Admin Client. Using the IMS Control Center you can view members of the IMSplex and define groups of members. In addition, using the CC you can issue both type-1 and type-2 commands.

Added to the Control Center (requires IBM DB2 UDB V8.2) since the first release:

•V9 resources (areas, databases, OLR)

•Command dialog (expert dialog)

•Command editor

- syntax assist

•Command shortcuts

•Results panel

- compare two sets of command outputs
- direct edit from command output (e.g., change the class on the output page and run an Update under the covers and refresh results)

RACF Enhancements for Replacement of SMU

- ***IMS V9 is the last release to support SMU security***
- Migration to RACF (or equivalent) can begin in V8 or V9
- V9 includes RACF enhancements for SMU functions which previously had no alternative
  - Enhancements include areas of AGN, AOI, TCO, MSC link receive, signon verification, and /LOCK and /UNLOCK
  - V9 is a transition release where SMU is still available
- Migration off SMU will be easiest with V9
- Migration off SMU and from V8 to V9 follow-on (skipping V9) is still a valid option
  - Should begin migration off SMU in V8, especially from SMU LTERM-based security to RACF user-based security
  - Migration using V9 enhancements will have to occur during the V9 follow-on migration
- Migration references
  - Chapter 4, IMS V9 Administration Guide: System (SC18-7807)
  - Chapter 6, IMS V9 Implementation Guide redbook (SG24-6398)

Migration to RACF (or equivalent) can begin in V8 or V9. V9 includes RACF enhancements for SMU functions which previously had no alternative.

Enhancements include areas of AGN, AOI, TCO, MSC link receive, signon verification, and /LOCK and /UNLOCK.

AGN security is also referred to by the name Resource Access Security. Currently, AGN security is implemented with a combination of SMU and RACF or an IMS user exit (DFSISIS0). SMU is always required. The new support is called just Resource Access Security (RAS). It will no longer be called AGN. Either RACF, a new user exit, or both can be used to provide protection. Either AGN security, or the new RAS security can be used, but not both. If both are specified, then RAS will be used.

The ICMD call of type 2 AOI already uses the SAF interface (to RACF) and the DFSCCMD0 user exit for security. The CMD call of type 1 AOI currently uses SMU for security, but will have the option of using SAF and the DFSCCMD0 user exit. A new startup parameter, AOI1=A|N|R|C|S will indicate which security product will be used for authorization of commands for type 1 AOI, and the level of security. A new parameter is added to the system definition TRANSACT macro to specify if a transaction can issue a CMD call, and how the authorization will be based -- on the userid, the transaction code, or the command code.

For TCO in prior releases, IMS commands were defined with SMU to control which commands could be issued by a TCO script. With the new support, SAF/RACF and the DFSCCMD0 user exit can be used to authorize use of commands by a TCO script.

In prior IMS releases, when a message was received by IMS on an MSC link, sometimes RACF and a user exit were called, and sometimes SMU was called. With the V9 changes, a startup parameter controls who is called. It can be as it is today, or always RACF and a user exit, or no security. Another startup parameter, or a parameter from the DFSMSCE0 user exit, controls how the authorization will be based.

In addition to using SMU to specify which static terminals are required to signon, a new startup parameter and new system definition parameters allow alternative ways to make this specification.

For the /LOCK and /UNLOCK commands, passwords can be specified for some keywords and parameters, and are used for authorization using SMU. For certain parameters, authorization checking using SAF/RACF and a user exit are now possible.

V9 is a transition release where SMU is still available. Migration off SMU will be easiest with V9. Migration off SMU and from V8 to V9 follow-on is still a valid option. Should begin migration off SMU in V8, especially from SMU LTERM-based security to RACF user-based security. SMU security controls access to IMS resources based upon the terminal that entered the transaction or command. RACF security requires a user at the terminal to identify themselves, and based upon that identity, controls which IMS resources can be acessed by that user. The access by a user moves with the user when they move from terminal to terminal. The redbook reference below discusses the alternatives of this migration. Migration using V9 enhancements will have to occur during the V9 follow-on migration. Therefore, the migration off SMU security for AGN, type 1 AOI, TCO, MSC link receive, /LOCK and /UNLOCK, and signon verification will have to wait until the migration to the V9 follow-on, if V9 is skipped.

Migration references: Chapter 4, IMS V9 Administration Guide: System (SC18-7807); Chapter 6, IMS V9 Implementation Guide Redbook (SG24-6398)

New /DIAGNOSE Command eases Serviceability

- Log IMS resource data, on demand, to the OLDS or TRACE datasets as 6701 records
- Allow users to take a snap shot of IMS resources at any time without impacting IMS

- Benefits
  - Provides an alternative to console dumps for IMS problem diagnosis
  - Verify configuration/sysgen/startup options in key control blocks

- **`/DIAGNOSE SNAP BLOCK(xxx) NODE(yyy) TRAN(zzz) OPTION(OLDS¦TRACE)`**
  - Control blocks supported: ALL, CMDE, ESCD, LSCD, MWA, QSCD, SCD, SQM, TSCD

Customers would like the ability to obtain IMS diagnostic information without impacting operations. Currently, a console dump is required to gather the basic information necessary to analyze many types of ordinary problems, such as hung terminals or transactions. Even the simple verification of configuration, system generation, or startup information in key control blocks may require a dump. Taking a console dump in an IMS production environment, especially during peak hours, is a disruptive process that can have a negative impact on operations, service level agreements, and financial goals. Many customers are forced to schedule dumps during an off-peak period or are prevented from taking dumps at all in their production environment.

The new /DIAGNOSE command operates like a standard type-1 command. Uses the new type-2 command format. An operational consideration is that the OLDS needs to be switched to extract data if it has been captured on the OLDS rather than the trace datasets.

IMS V9 supports the control blocks shown: ALL CME ESCD LSCD MWA QSCD SCD SQM TSCD.

An OPTION() keyword is added to the /DIAGNOSE command syntax to provide a choice of output storage location for the data captured by the command. With this keyword, the user can route the captured storage information to either the OLDS dataset or the user-defined trace datasets. The OPTION() keyword is optional, and by default, the captured data will be stored in the OLDS dataset. Valid choices for the new keyword are: OPTION(OLDS) or OPTION(TRACE).

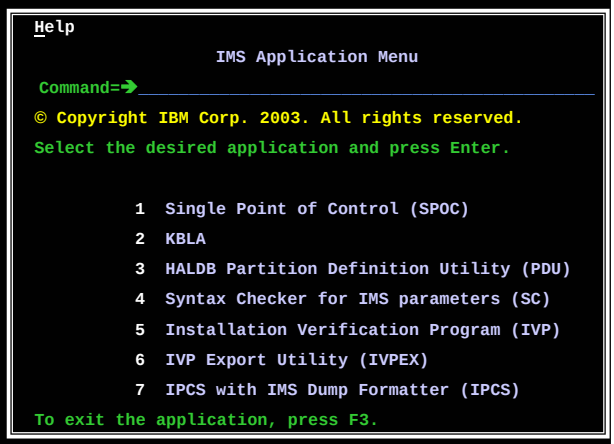
Tracing Enhancements

- From IMS V9, the DL/I trace and the Lock trace are both ***on by default***
  - Use the /TRACE command to deactivate either

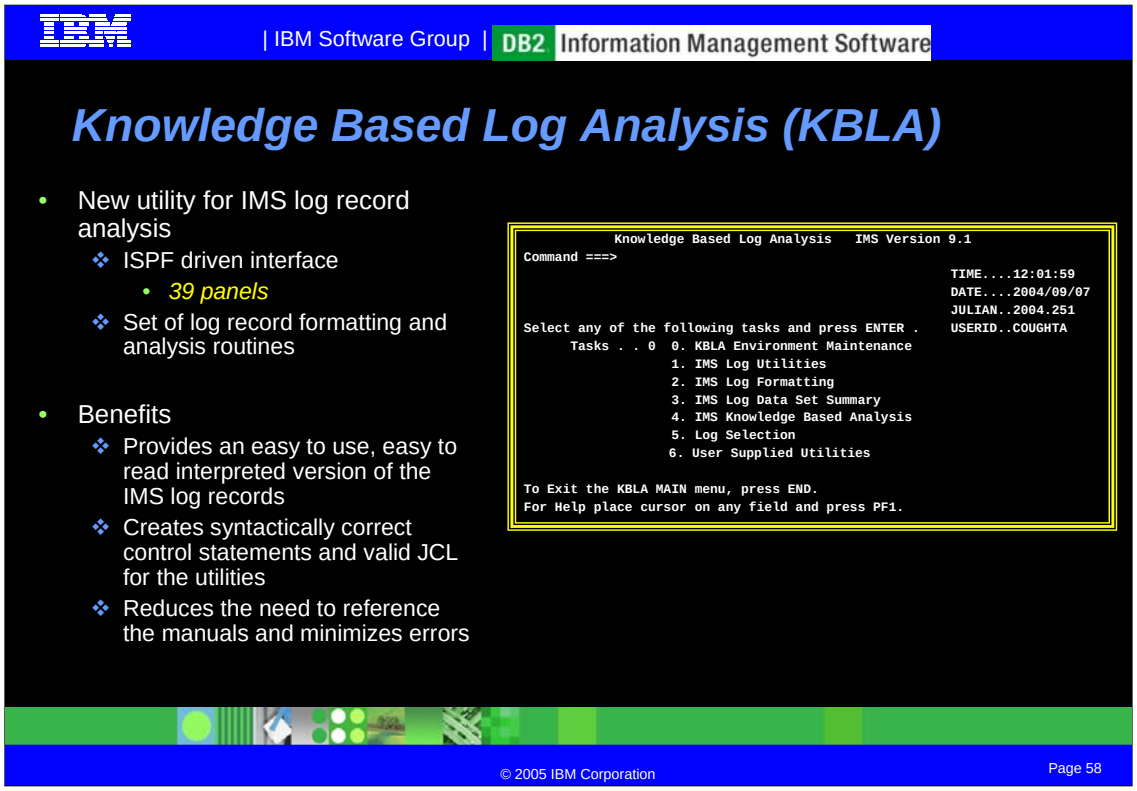
```
I9PLX                    IMS Single Point of Control
Command ===>  ____________________________________________________________

                              Plex .________Route . __________ Wait .      ______
Log for . . : DIS TRACE ALL                                        More:  -+
IM9A          NO LINE TRACES FOUND
IM9A          NO LINK TRACES FOUND
IM9A          NO NODE TRACES FOUND
IM9A          NO PSB TRACES FOUND
IM9A
IM9A          MONITOR IS ACTIVE:  LA SCHD APMQ APDB
IM9A          NO TRANSACTION TRACES FOUND
IM9A          NO PROGRAM TRACES FOUND
IM9A
IM9A          TABLE    TRACE   LOGGING  ENTRIES VOLUME
IM9A           RETR     ON       N/A   128/PST  N/A
IM9A           DL/I             OFF      2268
IM9A             DL/I   ON                      HIGH
IM9A             LOCK   ON                      HIGH
IM9A             PI     OFF                     OFF
IM9A          LATC     OFF      OFF      3024  OFF
 F1=Help      F3=Exit      F4=Showlist  F6=Expand    F9=Retrieve F12=Cancel
```

Additional Manageability Enhancements…

- **IMS Application Menu**
  - IMS menu for invocation of IMS components on TSO/E and ISPF
  - Provides a common interface to the following applications:
    - *Single Point of Control*
    - *Knowledge Based Log Analysis (KBLA)*
    - *HALDB Partition Definition Utility*
    - *Syntax Checker*
    - *Installation Verification Program*
    - *IVP Export Utility*
    - *IMS Dump Formatter*



IMS has an increasing number of applications that run on TSO and ISPF. Each application is started using its own unique command name and parameters. It becomes difficult for users to remember the invoking command for each application as more applications are made available. The IMS Application Menu provides a common interface to the applications. To facilitate their use, the user displays the IMS Application Menu and starts the applications from the menu.



Knowledge Based Log Analysis (KBLA) is a new facility for IMS log record analysis. The ISPF panel driven user interface is designed to simplify JCL job creation and to prevent JCL errors. All the necessary control statements are created in a syntactically correct form. KBLA reduces the need to reference the manuals and eliminates the work involved in creating jobs for extracting log records.

The utility also includes a new set of log formatting routines that allow the user to extract an interpreted version of the IMS log records, producing easy to read output. Key data for each log record is interpreted in plain English relieving the user from having to research the meaning of each flag or field.

Additional Manageability Enhancements

- Greater than 255 Transaction Scheduling Classes
  - Provides the ability to use more than 255 transaction scheduling classes, up to a value of 999
- LU Type 3 device logon option
  - Allows LU Type 3 devices to logon as ETO SLU1 or 3270P devices in order to make IMS available to these devices
- VSCR
  - Dynamic Allocation DSAB control blocks
  - Online Change modules
  - DBRC modules (DSPCINT1 and DSPURX00) moved into private and common storage above the 16MB line
- Enhanced IEBCOPY parm support on the Online Change Utility
  - Support IEBCOPY parameters WORK=,SIZE=, and LIST= on the Online Change Copy Utility (OLCUTL) to reduce potential outages by allowing overriding of default values
- Notify CQS outage to terminal users – DFS3649A (RC=436)
- Fewer SNAP dumps on the Log

You can assign scheduling-class values for a transaction. These values allow you to isolate, organize, or retrieve certain transactions based on the class value. Prior to Version 9, the maximum number of transaction scheduling-class values was 255. The maximum value is now 999.

Customers migrating to ETO (Extended Terminal Option) have difficulties with LU type 3 devices, because the bind fails due to IMS treating this as an invalid device type. IMS V9 provides a new option in DFSINTX0, the Initialization Exit routine to allow ETO LU type 3 devices, to logon as a SLU1 or 3270P device.

The VSCR enhancement ensures that the Data Set Association Blocks (DSABs) generated are allocated in above-the-line private storage, thereby providing enhanced Virtual Storage Constraint Relief. For example, if you have 20,000 Full Function and/or Fast Path data sets dynamically allocated, you could have over one megabyte of storage allocated above the line. Approximately 20 online change modules are moved from below-the-line private to above-the-line private. Also the online change modules are removed from the nucleus and become standalone modules. Two DBRC load modules (DSPCINT1 and DSPURX00) are moved into private and common storage above the 16MB line. This move will further reduce the amount of common storage area (CSA) that IMS uses, thus increasing the CSA available.

The IMS Online Change Copy utility (OLCUTL) now supports the IEBCOPY utility parameters WORK=, SIZE=, and LIST=. The IMS utility passes the values for these parameters to the IEBCOPY utility.

The WORK parameter specifies the number of bytes of virtual storage to request for a work area to hold directory entries, internal tables, and I/O buffers.

The SIZE parameter specifies the maximum number of bytes of virtual storage that IEBCOPY can use as a buffer.

The LIST parameter controls suppression of IEBCOPY IEB154I messages that are issued for each member that is successfully copied.

Prior to IMS V9 when CQS abended, users who tried to connect to that IMS were disconnected immediately after the DFS3649A message, with no notice sent to the terminals. IMS V9 will send error message DFS3649A with Rejected RC=436.

SVC dumps are now taken for some pseudo abends that used to cause the logging of snap dumps with associated buffer pools. This reduces logging volume considerably in certain cases.

IMS V9 External Subsystem Enhancements

- External Subsystem Resolve-In-Doubt Notify User Exit
  - Enhanced Recoverability with DB2 or other Subsystems
  - Optional Notify user exit, DFSFIDN0, called for each unresolved Unit of Work during /ERE, XRF takeover, FDBR
  - Allows customers to resolve their indoubts without waiting for IMS restart to complete
- FDBR ECSA Relief for DEDBs
  - Storage for DEDB buffer pools obtained from private region rather than ECSA
    - ***FPBUFF=LOCAL in DFSFDRxx***
    - ***Control blocks, DEDB & SVSO buffer pools***
- FDBR **DFS4172** Message issued as WTO
  - Eliminate need to reply to WTOR and ensure that active IMS system starts before FDBR

The Fast DB Recovery region is a separate IMS control region |that monitors an IMS, detects failure, and recovers any IMS-owned database |resources that are locked by the failed IMS, making them available for other |IMS systems without having to wait for the next full restart. For database |resources that are not IMS owned, such as DB2, the Fast DB Recovery region |provides an optional user exit, DFSFIDN0, for this purpose.

This enhancement provides the identity of indoubt IMS External Subsystem Attach Facility (ESAF) UOWs to an optional user exit routine (DFSFIDN0) during the recovery phase of an FDBR system or an emergency restart of an IMS system (including an XRF alternate subsystem), thus enabling you to resolve the indoubt UOW earlier than the restart of the IMS system.

As they scan their collections of IMS units of work for recovery, FDBR and IMS systems determine whether external-subsystem data was also changed as part of each unit of work. If so, they provide all of the following information to a user exit routine:

•The identity of this external-subsystem data

•The name of the external subsystem that was connected to the failed IMS

•The final resolution of the data

IMS calls the exit routine once for each indoubt ESAF unit of work. This is a synchronous call to the exit routine, so you should ensure that the exit routine does not impact the performance of the FDBR or the IMS system. The FDBR or IMS system attempts to load the user exit routine from the concatenation of the IMS.SDFSRESL libraries. The name of the exit routine must be DFSFIDN0. If the exit routine does not exist in the IMS.SDFSRESL libraries, then FDBR ignores the data and an IMS subsystem retains it until a connection is made to the external subsystem.A new Notify user exit, DFSFIDN0, can be coded in IMS V9, that allows customers to resolve in-doubt work without waiting for IMS restart to complete. Indoubt work is work that has completed synchpoint phase 1 (prepare) but has not completed synchpoint phase 2.

FDBR ECSA Relief for DEDBs

For IMS Version 9, IMS can obtain storage for DEDB buffer pools from the IMS private region rather than from ECSA. For the ACTIVEIMSID control statement of the DFSFDRxx member of IMS.PROCLIB, you can specify an optional parameter FPBUFF=LOCAL that specifies that the control blocks for Fast Path DEDB processing should be obtained from the FDBR control region private storage instead of from ECSA. This includes the DEDB common buffer pool as well as the SVSO buffer pools.

FDBR WTO Message

In IMS Version 8, when the IMS system that an FDBR system is tracking is not running, the FDBR system issued a write-to-operator-with-reply (WTOR) message that indicated that there was no active system; the reply required the operator to indicate what action to take. For IMS Version 9, this message, DFS4172I, is a write-to-operator (WTO) message, with no reply required. The FDBR system waits for the MVS Cross-System Coupling Facility (XCF) Group exit routine to indicate a change in status of a member of the group. That is, the FDBR system waits for the active IMS system to rejoin the group.

OTMA Enhancements

- Message level ACEE aging value override
  - Allows OTMA clients to set override at an input message level for a userid
    - *Default continues to be the global value in the OTMA client-bid*
  - Provides greater granularity and control
- OTMA trace table enhancement
  - New OTMT trace table entries for activity in and out of exits DFSYIOE0, DFSYPRX0, DFSYDRU0

IMS provides the following new functions for OTMA:

OTMA clients can now set the ACEE security aging value for user IDs in the OTMA message prefix without requiring the client to reconnect to OTMA. Specifies that the aging value in the message-control information message prefix is to be used for this userid. This aging value will override the member aging value specified in the client-bid time for this message. If this option is not set, member aging value will be used to refresh the cached ACEE for the userid. If the userid aging value is less than 300 seconds (5 minutes), IMS always creates a non-cached ACEE.

New OTMA trace table entries for trace activity in and out of the exits as X'5A05'-X'5A0A' are supported in IMS Version 9. OTMA supports the creation of new OTMT trace table entries in the effort to standardize OTMA traces and conform to the tracing that is done for other OTMA modules. The existing exit entries do not follow the OTMA table entry standard that is documented in the Diagnosis Guide and Reference manual. This enhancement provides customers and IBM service support personnel with an easier way to check the return code set by the user exits. Whenever an OTMA user exit is called, an OTMT table trace record is cut to trace the input parameters associated with the exit. After the call to the exit, another OTMT table trace entry is cut to trace the return code set by the exit. This enhancement applies to all the OTMA exit routines, DFSYIOE0, DFSYPRX0, and DFSYDRU0.

OTMA Enhancements …

- DFSYPRX0 Pre-routing Exit enhancement
  - Allows specification of the OTMA destination TPIPE name
    - *Minimizes the need for the OTMA Destination Resolution Exit DFSYDRU0*
  - With this enhancement, all information that is set in DFSYDRU0 can be set in a combination of startup parameters and DFSYPRX0
- Input message ZZ protection
  - Allows values set by remote program to pass through untouched
    - *Preservation of the Z2 field of the LLZZ input message*
    - *Can be set or modified by OTMA Input/Output Edit Exit DFSYIOE0*

•The DFSYPRX0 user exit routine for OTMA can set the OTMA destination TPIPE name.

•OTMA will no longer zero out the Z2 data of the OTMA input message and DFSYIOE0 can set the Z2 data. In prior releases, OTMA always zeroed out the values in the ZZ field of the input LLZZ application data. In IMS Version 9, the bit settings are untouched and sent in to IMS. If necessary, OTMA exit DFSYIOE0 which has access to all messages inbound and outbound, can set or modify the value. This could be important to IMS applications that were originally coded for sensitivity to MFS options. The ZZ field in (non-OTMA) MFS messages contains the MFS formatting option being used to format the messages to and from the IMS application program. Although OTMA does not invoke MFS, the existing IMS application program may be sensitive to this information and expect to see a value in the field. Remote clients, understanding that they are communicating with a program that looks for this field, can set the bit prior to transmission to IMS.

OTMA Enhancements …

- Ability to Delete Undeliverable CM0 Output
  - When IMS Connect is disconnected from the XCF group
    - *IMS Connect applications have an option to inform OTMA to purge CM0 IOPCB output*
- /DISPLAY TMEMBER TPIPE
  - Command is enhanced to show the wait status for an expected ACK or NAK for CM0 messages
- Also delivered in IMS V8
  - PQ87087

See the IMS Newsletter, Winter 2004 edition, "IMS Connect, CM0 Persistent Sockets for IMS Connect", pg 13, for more information

IMS Connect 2.1 APAR PQ87160 adds a new option, Purge Not Deliverable, to CM0. This option allows the client application to request CM0, synch level confirm, with Purge Not Deliverable. If an output message cannot be delivered to the client, IMS Connect responds with an ACK to IMS OTMA to request that the message be removed from the client message queue. This function allows the client application to execute CM0 synch level confirm and request that the output be deleted if the message cannot be delivered. It also means that if the message is delivered to the client application, then an ACK/NAK is required; however, if the message cannot be delivered, the client application requests that the message be deleted because it is not needed. This function is extended to IMS OTMA and requires IMS V8.1 APAR PQ87087 or V9.1 APAR PQ87088 be applied to the corresponding IMS level.

OTMA Enhancements …

- New “Auto-One” option for OTMA Resume TPIPE protocol
  - After receiving the Resume TPIPE command from IMS Connect, if there are messages on the asynchronous hold queue, OTMA sends one message
  - If there are currently no messages, OTMA will send one message when it arrives
  - Once an message is delivered, the RESUME TPIPE option will be reset to "NO-AUTO”
  - Also delivered in IMS V8
    - *PQ79040*

- MQ/OTMA CM0 Non-Persistent Message Performance Enhancement
  - Removed unnecessary "OTMA sequence number" logic for processing CM0 non-persistent output messages
    - *For this type of message, there is no need to track the sequence number*
  - Customer feedback on the reduced elapsed time:
    - “Wow - what a difference .... it goes down from 0.491 sec/trx before to 0.248 sec/trx with the change. Pretty consistently *half the elapsed time.*"
  - Also delivered in IMS V8
    - *PQ81398*

A new RESUME TPIPE option for CM0 persistent or transaction sockets, SINGLE with WAIT, allows an IMS Connect client to retrieve one message from the IMS OTMA hold queue. If there are no messages when the resume Tpipe is processed by IMS

OTMA, IMS OTMA waits for a message. After a message is available in the hold queue, IMS OTMA delivers the message to

IMS Connect. See IMS V7 APAR PQ78912 or V8 APAR PQ79040 for more details.

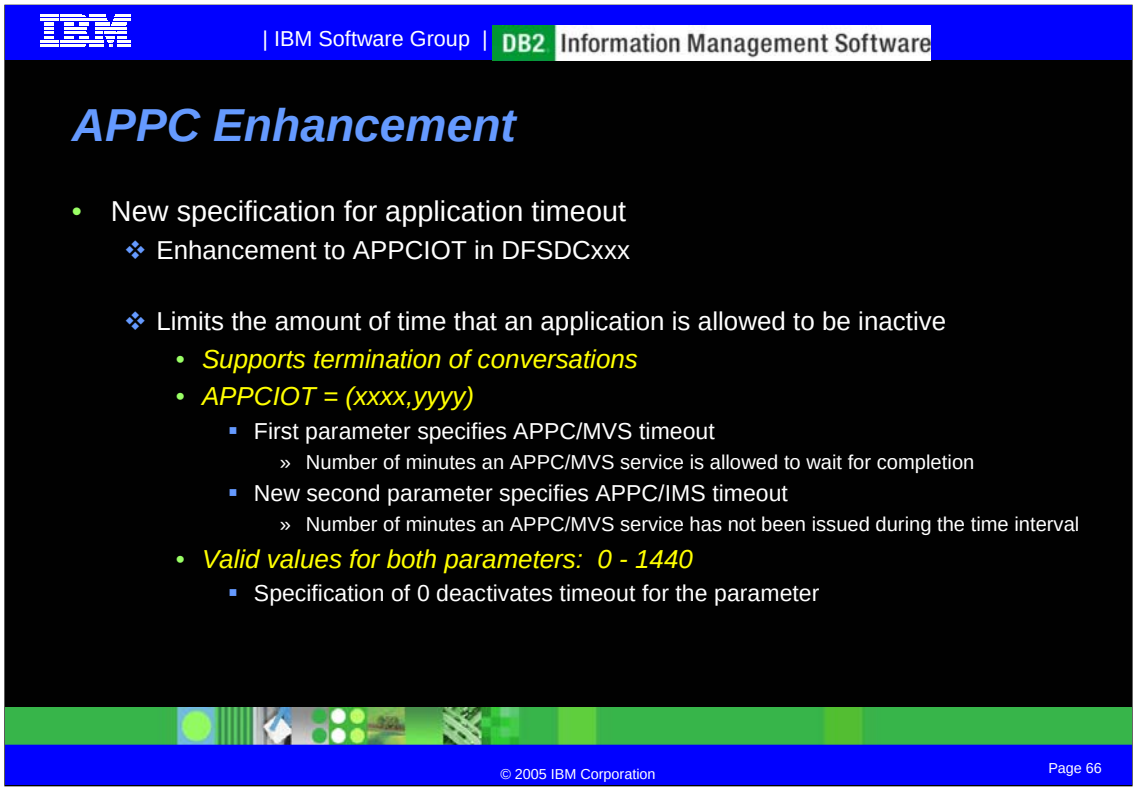
/EXIT Command for APPC/OTMA

- Ability for IMS operators to terminate IMS conversations originated by APPC or OTMA
  - /EXIT can also be issued from APPC/OTMA devices
  - Enhancement to /EXIT and /DISPLAY CONVERSATION (for OTMA) commands

/EXIT CONV convid LUNAME luname

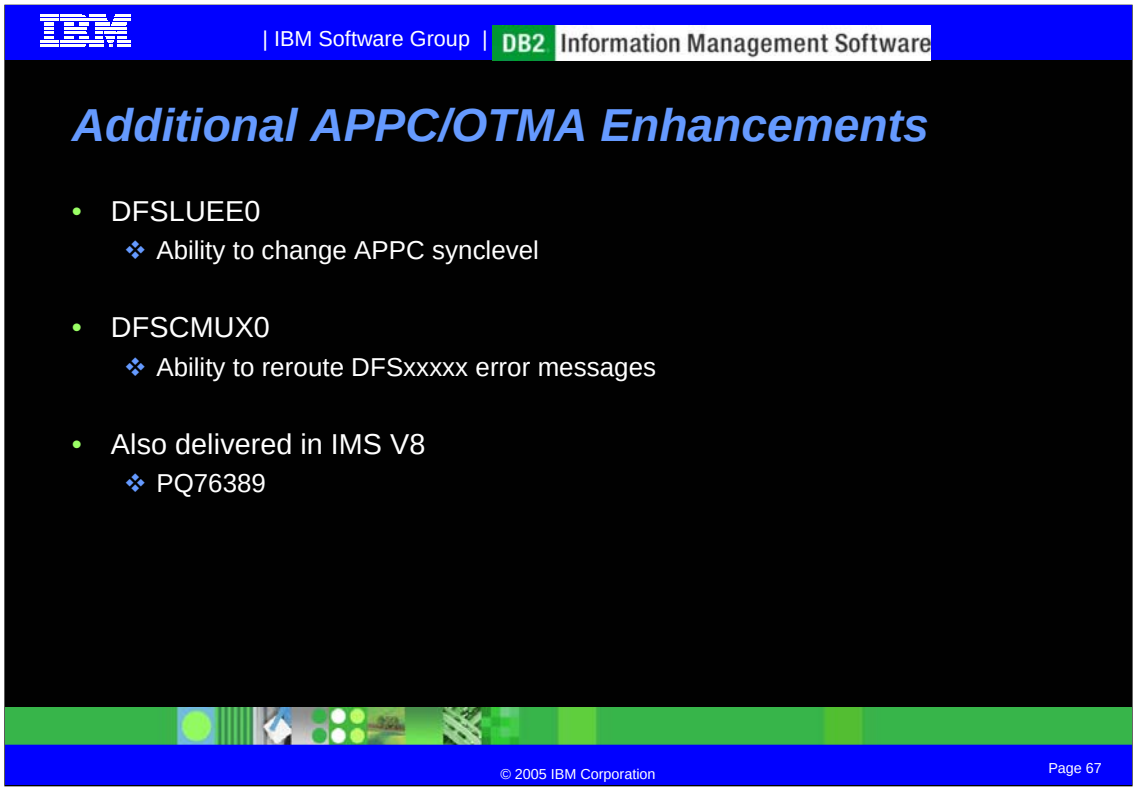
/EXIT CONV convid TMEMBER tmembername TPIPE tpipename

- Benefit
  - Greater control of IMS conversations in APPC/OTMA environments
    - *Allows IMS conversational processing to be terminated for processing such as Online Change*



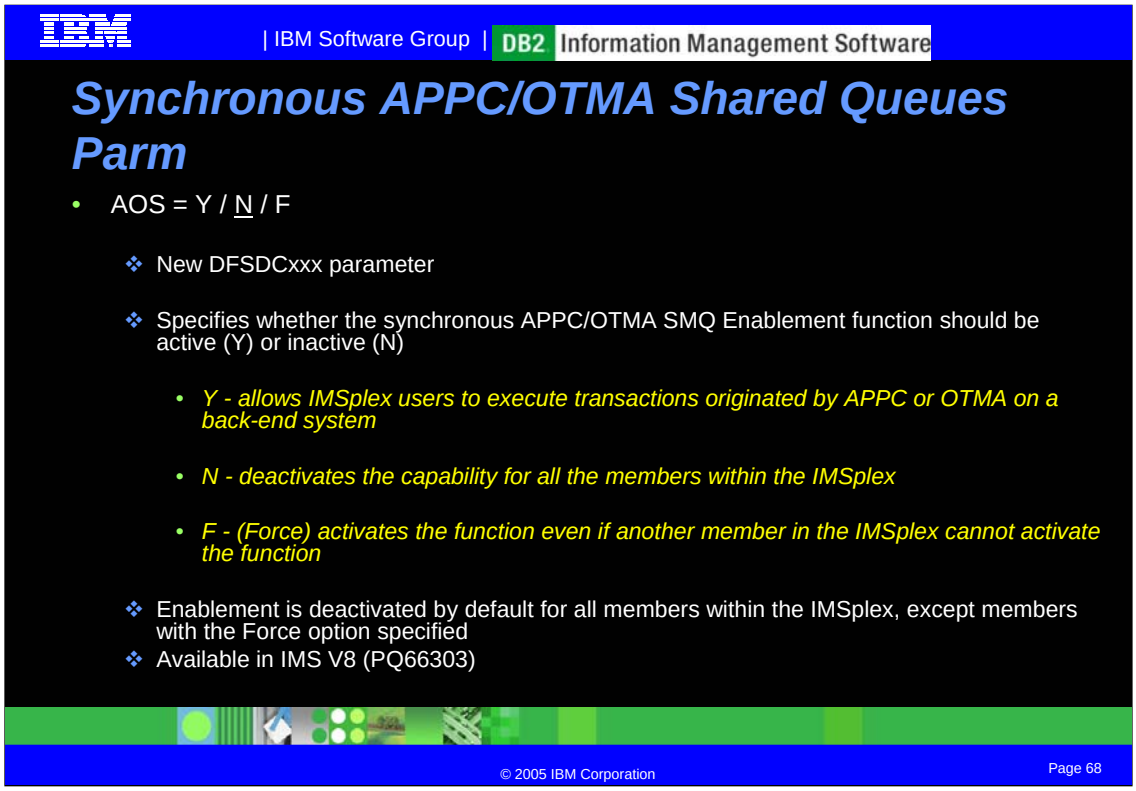
The first parameter specifies the APPC/MVS time-out value in minutes. This is the number of minutes an APPC/MVS service is allowed to wait for completion.

The second parameter specifies the APPC/IMS time-out expression. This is the number of minutes an application is allowed to be inactive and is for synchronous conversations only. Inactive means that the application was not able to respond within the time-out limit. Note: There are cases where it is not determined if an APPC Conversation is synchronous or asynchronous. This happens in program to program switch cases where it is unpredictable which program will continue the synchronous APPC conversation.



User exit DFSLUEE0 has been enhanced to get the synclevel information in the parameter list. A new flag will inform the user exit if this is an outbound conversation. For outbound conversations the user exit is able to change the synclevel.

User exit DFSCMUX0 has been enhanced to receive DFS messages and be able to re-route these messages. A new flag will inform the user exit that this is a DFS message.



NOTE: Specifying N will deactivate APPC/OTMA SMQ Enablement for all the members within the IMSPLEX except for members specified F (force). APPC/OTMA SMQ Enablement uses RRS Multi-System Cascaded Transaction (MSCT) support to synchronize between the IMS systems. If RRS is not active on one system then the systems which have specified F (force) will still queue incoming transactions without any affinity. Therefore if the system without RRS tries to process one of these transactions it will abend the application with U0711. This option was added to address a specific customer scenario where the front-end has no databases, and if one back-end RRS goes down then all the distributed sync point transactions would have affinity to the front-end which is not able to process them.

Other Usability Enhancements

- HALDB Enhancements:
  - Specific Partition Initialization w/o DBRC Commands
    - *Initialize HALDB partitions in a single job step*
    - *Process a list of HALDB partition names to be initialized*
  - Enhancement to allow Batch job to run without DBRC when using a PSB that references a HALDB (provided the HALDB is not referenced)
  - HALDB Control Statement enhanced to allow 20 control statements
  - HALDB Partition Selection Exit enhancements
  - HD Unload (DFSURGU0) and HD Reload (DFSURGL0) enhanced to generate statistics on a partition basis

With IMS Version 8, the HALDB Specific Partition Initialization Utility (DFSUPNT0) requires that a DBRC CHANGE.DBDS command be issued for any partitions requiring initialization. When it executes then all of the partitions of the specified HALDB, recorded in RECON as “init required”, are initialized. It is desirable to be able to run this utility and unconditionally initialize individual partitions without using DBRC commands. With IMS V8 all of the partitions belonging to a master database can be unconditionally initialized but not individual partitions. In IMS V9 DFSUPNT0 provides the ability to initialize specified HALDB partitions without having previously issued DBRC CHANGE.DB commands.

DBRC Option enhancement for HALDB allows a batch program that references a HALDB to run without DBRC so long as no database calls refer to the HALDB during execution. This provides more flexibility in running a single program in different environments, thus reducing application developer and operations efforts that might be needed to convert programs and ensure their running only in specific environments. This enhancement is also being provided through the IMS V7 and IMS V8 service process.

HALDB Processing Control Statement enhancement increases the number of HALDB single partition processing control statements from 10 to 20 for usability. This allows a customer's application to specify up to 20 DB PCBs that are allowed to perform single partition processing in a HALDB. This enhancement is also being provided through the IMS V7 and IMS V8 service process.

HALDB Partition Selection Exit DSECT enhancement allows customers to use the DFSPSEIB macro to get the partition selection exit DSECTs. This makes HALDB with a user-written partition selection exit easier to use. Customers no longer have to create their own DSECTs for the partition selection exit parameters. This enhancement is also being provided through the IMS V7 and IMS V8 service process.

HALDB Partition Selection Exit Customization enhancement makes HALDB with a user-written partition selection exit more customizable and easier to use. This allows a partition selection exit to run differently, depending on IMS and the application running. This enhancement is being provided through the IMS V9 service process.

HD Unload and Reload utilities (DFSURGU0 and DFSURGL0) are enhanced to generate statistics for each HALDB partition, previously they had statistics for the entire HALDB database. HD Reload has been enhanced to read partition statistics in the unload file and to support concatenated unload files. This enhancement is also available through the IMS V7 and V8 service process.

Other Usability Enhancements

- Command Recognition Character Registration
  - Unique CRCs within a system to allow routing of command to the appropriate subsystem for execution in a sysplex
- DFSVSMxx option added for Parallel DB Open
- Enhanced DFS0832I for non-HALDB FF pseudo abends
- DFSDCxxx option STM=Y/N for sharing resources with Sysplex Terminal Management
- Automatic adjustment of the global database directory table

CRC Registration: With IMS Version 9, during initialization of the IMS control region, IMS uses the CRC= execution parameter to register the IMS command recognition character (CRC) with the sysplex (if IMS is running in a sysplex) or with the z/OS system. If the CRC is not unique, IMS issues message DFS1946W and continues with initialization. Thus, all IMS systems in the sysplex should have a unique CRC. With a unique CRC, a console operator can issue IMS commands from any system in the sysplex and z/OS will route it to the appropriate system for execution. However, note that IMS checks this only to see if it is unique within a system, not across the sysplex, in spite of what the text of the DFS1946W might imply and the explanation in the IMS Version 9: Messages and Codes Volume 2, GC18-7828 explicitly states. This is done intentionally since you may want to have the same CRC on different LPARs in the same sysplex, because it allows a single command to be processed by all the subsystems having the same CRC. A DISPLAY OPDATA command on the z/OS console shows you all recognition characters used in your environment.

A new control statement can be specified in the DFSVSMxx Proclib member to allow the disablement of full function Parallel Database Open processing (introduced in V8) if desired. Parallel DB Open processing is active by default. To disable the Parallel DB Open function, you must define the NOPDBO (No Parallel DB Open) keyword in the DFSVSMxx proclib member.

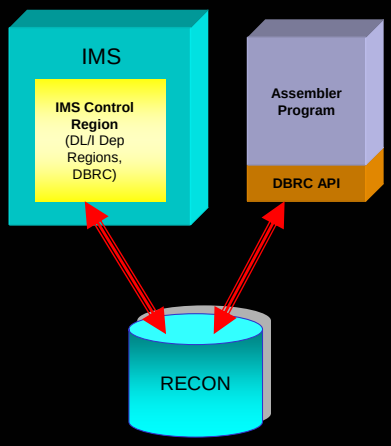
DFS0832I message was introduced in V7 for HALDB abends such as U0880 and U3498. It is issued prior to the abends to indicate which database or partition was affected. The message is being enhanced in V9 to be issued also for non-HALDB abends set by modules DFSDLR00, DFSDDLE0, DFSDXMT0, DFSDLD00 or DFSRCHB0. This enhancement addresses a long-time requirement from customers who need to easily identify which database (HALDB or non-HALDB) is affected by pseudo abends such as U085x.

Sharing TM Resources in an IMSplex with RM IMS Version 9 adds a new STM= keyword to the DFSDCxxx PROCLIB member. STM=YES (the default value) indicates that IMS is to share IMS Transaction Manager resources when a resource structure is being used by Resource Manager (RM) in an IMSplex. These resources include terminals, users, user IDs, LTERMs, MSNAMEs, and APPC descriptors. Including the STM=YES keyword is equivalent to prior-release IMS function, which enabled TM resource sharing automatically if a structure exists. STM=NO indicates that IMS is not to share IMS Transaction Manager resources when a resource structure is being used by RM in an IMSplex. The STM= keyword does not affect transactions (static and CPI-C) because IMS transactions are not considered IMS Transaction Manager resources. When you define a resource structure, transactions are shared in RM, regardless of what you specify for the STM= keyword.

The autonomic adjustment of the global database directory table results in more effective management of the table, thus improving performance during IMS data sharing processing. With this enhancement IMS will build the table and adjust the size based on the highest global DMB number in the system. This will ensure the GDMB table will have enough room for all the registered databases. No CBTS scan will be performed and this will improve the performance through the data sharing process. This enhancement is being provided through the IMS V7, IMS V8, and IMS V9 (PQ90072) service processes.

DBRC API

- Stage a supported, Assembler interface to access information in the RECON
- DSPAPI Functions supported
  - Start/Stop the API environment
  - RECON Query
    - *RECON status, database, groups, log (RLDS, SLDS, OLDS), subsystem, and backout*
  - Release buffer storage after Query



A new DBRC API is introduced in V9 that allows assembler application programs to use a standard, supported (release independent) interface to invoke DBRC to get information in the RECON. This support will be done in stages, with V9 introducing the RECON Query function.

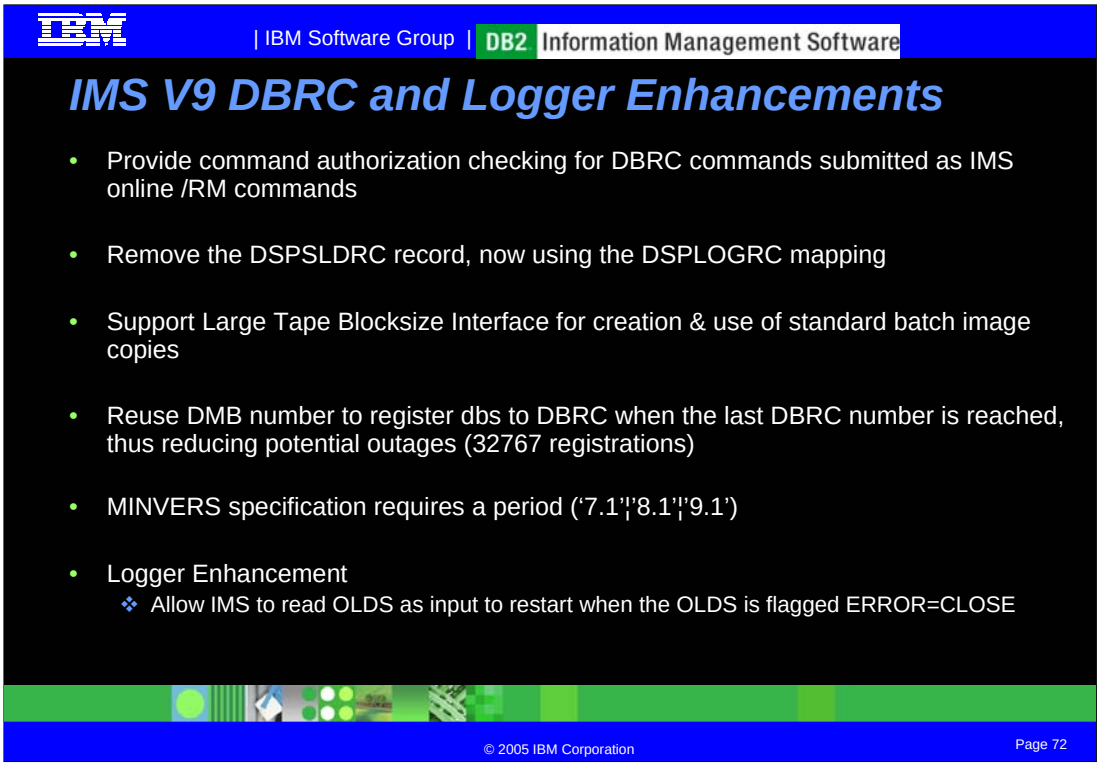
The benefit of a supported API is that it solves difficulties with the current methods of access, employed by user programs, which are subject to change or undocumented. These include the use of the Recovery Control Utility, DSPURX00, DFSBRLSC, and VSAM itself to access the RECON.

The interface consists of a new DBRC macro, DSPAPI, which contains several functions available to the application program. One of the DFSAPI functions starts the API environment, when the connection between DBRC and the RECON is made (DSPAPI FUC=STARTAPI). Stopping the API environment terminates this connection (DSPAPI FUNC=STOPAPI).

The main Query function in the API, DSPAPI FUC=QUERY TYPE=, allows the program to retrieve the following information from the RECON: RECON status, database, groups, log (RLDS, SLDS, OLDS), subsystem, and backout.

Private storage that is obtained during a Query call must be released by the program using the DSPAPI FUNC=RELBUF call.

The functions provided by the DSPAPI macro have a version associated with them because parameter lists can change from one release to the next. To use the parameters associated with a version, you must specify that version number or a later version. If you specify an earlier version level, the parameter is not accepted by the macro and an MNOTE error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.



IMS Version 9 provides the following enhancements for DBRC:

•Command authorization that is currently available for batch DBRC commands is now also available to the online (/RM) commands. Specifically, the ability to use a security product (like RACF), a security exit routine, or both.

•Two DBRC load modules (DSPCINT1 and DSPURX00) are now in private and common storage above the 16MB line. This move further reduces the amount of common storage area (CSA) that IMS uses, thus increasing the available CSA.

•DBRC uses PRILOG RECON record mapping instead of PRISLD RECON record mapping.

•Large Tape Blocksize support – the DFSMS access method added support for >32K blocksizes providing customers with a performance advantage. Support for this is now added to the Image Copy and Recovery Utilities.

•Reuse of database management block (DMB) slots to register databases to DBRC when the last DMB number is reached reduces potential outages.

•The MINVERS parameter format now includes a period. DBRC considers this a special character and requires it to be enclosed in single quotes. The new format is: MINVERS('7.1'|'8.1'|'9.1')

•Logger enhancement ERROR=CLOSE: DBRC can inform IMS during restart which log in an online log data set (OLDS) to read next, even though the OLDS might not yet be closed. You must still process the OLDS using the Log Recovery utility (DFSULTR0) before you can archive the OLDS or use it as input to any utility.

•Logger enhancement to check the second WADS for additional data before closing the OLDS during a restart: Customers using dual WADS would like IMS to compare the two incase one contains additional data but both are readable and usable (no I/O errors). So IMS will check the secondary WADS even though the primary WADS has no error, allowing it to check for any additional data that may have been written only to the secondary WADS.

IMS Resource Definition Manageability Staging

- Requirement:
  - Reduce IMS System Generation time and effort

- History:
  - **IMS V4** stopped using sysgen to support major new function
  - **IMS V5/6** removed conditional assembly modules
  - **IMS V7** put non-conditional link-edit modules under SMP control
  - **IMS V8** removed RSR Feature install checking, provided Resource Manager/Global Online Change, Syntax Checker, Packaging, Installation, and IVP enhancements
  - **IMS V9** further reduces time and effort

In the more recent IMS versions we have been focused on reducing the IMS System Generation time and effort:

•IMS V4 stopped using sysgen to support major new functions by introducing ETO.

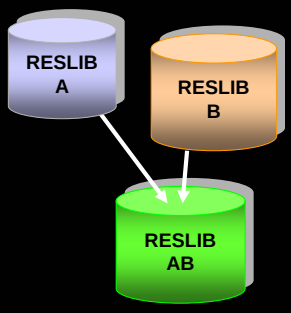
•IMS V5/6 removed conditional assembly modules.

•IMS V7 put non-conditional link-edit modules under SMP control.

•IMS V8 provided the Resource Manager/Global Online Change, Syntax Checker, Packaging/Installation/IVP enhancements, and removed the RSR RLT/DLT features install checking.

IMS V9 Resource Definition Manageability

- Remove the restriction of maintaining RESLIBs for different system definitions
  - *V9 removes conditional linkedit load modules*
    - *Simplify IMS nucleus and VSCR*
  - *V9 supports generic DFSVC000 module*
    - Ability to override values with Control Region parameters
  - *Results: FP, Non-FP, DB/DC, DBCTL or DCCTL Systems can use the same RESLIB in V9*
- Simplify Extended Terminal Option (ETO) feature install
  - *In V9, IMSCTRL ETOFEAT is ignored*
  - *ETO Feature checking occurs at execution time rather than during system definition*



There are substantial changes to IMS system definition in IMS V9 to further simplify this process. In the short term, we are continuing to reduce the IMS system generation effort by:

Removing the restriction of maintaining RESLIBs for different system definitions

With IMS Version 9, you do not need separate IMS SDFSRESL (i.e. RESLIB) libraries for sysgen of any of the following environments: DB/DC, DBCTL, DCCTL or Fast Path. You can use separate IMS SDFSRESLs or combine them, as needed for your environment. IMS V9 introduces enhancements to facilitate the use of a single shared SDFSRESL library. This is accomplished through two capabilities: the removal of sysgen conditional binds and support for a generic DFSVC000 module through the ability to override required values through new IMS Control Region parameters. Three new control region parameters, SVC2 (type 2 SVC number), MCS (IMS route code), DESC (IMS descriptor code), replace the equivalent IMSCTF and IMSCTRL system definition parameters. The IMSGROUP (defines the IMSID) control region parameter must be added to each IMS/DBC/DCC procedure.

Removing conditional link edit modules

Removing approximately 33 Online Change modules from the nucleus Linkedit step of IMS system definition and placing them in their own standalone load modules. Approximately 20 of these modules are moved to above-the-line private storage. This is another step towards simplifying system definition and allows you to save below-the-line private storage.

Removing the ETO feature install checking.

With IMS Version 9, IMS system definition no longer validates that the ETO feature has been installed. During IMS system definition, IMS ignores the first two sub-parameters that are specified on the ETOFEAT keyword of the IMSCTRL macro. Instead, during initialization of the IMS Transaction Manager, IMS validates that the ETO feature is installed when you specify the ETO=Y startup parameter. If the feature is not installed, IMS issues a DFS3676I error message, followed by an ABENDU3476. IMS builds ETO descriptors, if requested, during system definition. If you need to build ETO descriptors, run the LGEN first (Stage 1 and Stage 2). Then, run a standard CTLBLKS system definition and specify ETOFEAT=(,,ONLY).

IMS V9 Resource Definition Manageability

- Other Install Improvements
  - Install Type 4 DBRC SVC dynamically
    - *Prior to V9 any modification to Type 4 SVC requires re-ipl of MVS*
  - Resource Manager Cleanup
    - *V9 dynamically installs IMS' new Resource Cleanup routine DFSMRC20 using z/OS RESMGR service*
    - *Eliminates SUPERZAPping a z/OS module*
  - Syntax Checker enhancements
  - Installation Verification Program Release support

Other install improvements include the following:

DBRC Type 4 SVC Dynamic Install

The Dynamic SVC (DFSUSVC0) utility dynamically updates the DBRC Type 4 SVC module. Thus, you can apply maintenance to the DBRC Type 4 SVC module without having to re-IPL z/OS after each update. Prior to IMS Version 9, only the IMS Type 2 SVC module could be dynamically updated. Any updates to the DBRC Type 4 SVC required re-IPLing z/OS.

z/OS RESMGR Services

IMS dynamically installs its resource manager cleanup routine; you do not need to install the DFSMRCL0 module as part of the IMS installation. Registration of the IMS resource manager cleanup routine with the operating system is done automatically during IMS startup. With the invocation of the z/OS RESMGR service to dynamically register the new IMS Resource Cleanup routine during IMS initialization, IMS V9 users no longer need to install DFSMRCL0 into the z/OS system as a part of IMS installation. This means that DFSMRCL0 no longer needs to be bound into LPALIB nor does it need to be defined as a static z/OS resource manager in CSECT IEAVTRML of module IGC0001C. The RESMGR service actually calls a stub module DFSMRC00 which calls the new IMS Resource Cleanup Module DFSMRC20. This allows for transparent application maintenance. DFSMRC20 also contains a version number. If a dynamic update is done, the old copy will remain until the next IPL.

The Syntax Checker and Installation Verification (IVP) enhancements in IMS V9 are described in the next two charts.

IMS V9 Installation Verification Program (IVP) Enhancements

- Add a Common Queue Server sample application
- Add an Enhanced Command Environment Sample Application
- Resolve the problem of IVP jobs not being able to run in a JES3 environment
- Dataset allocation cleanup - allow SMS allocation
- Enhanced IMS Dump Formatter Setup
- A new (required) high level variable, IXUVSMHQ, for VSAM data sets
- Export and Import IVP Variables between IMS releases - DFSIVPEX
- IVP Usability Changes - customized help text for IVP steps
- Miscellaneous Items

For IMS Version 9, the IMS Installation Verification Program (IVP) provides the following enhancements:

•The “E” IVP steps support a CQS sample application.

•The “O” IVP steps include an Enhanced Command Environment Sample Application.

•The IVP help text for the IVP steps:

- •Uses the variables from variable gathering.
- •Uses color to emphasize important information.

•Support for JES3 job classes is improved, and problems encountered by the IVP jobs that can’t run in a JES3 environment have been eliminated.

•SMS Storage Class and Management Class parameters are available for allocating all data sets.

•A new and separate high level qualifier variable, IXUVSMHQ, for VSAM data sets is introduced.

•IVP variables can be exported and imported between IMS releases for easier migration. You can use a new IVP utility, DFSIVPEX, to export and import the variables. Two new options, EXP (export) and IMP (import), have been added to the variable gathering panel.

•To complete and test the install of the IPCS for IMS dumps, IMS Version 9 introduces the following new steps:

IV_D215T

IV_D216T

IV_D217T

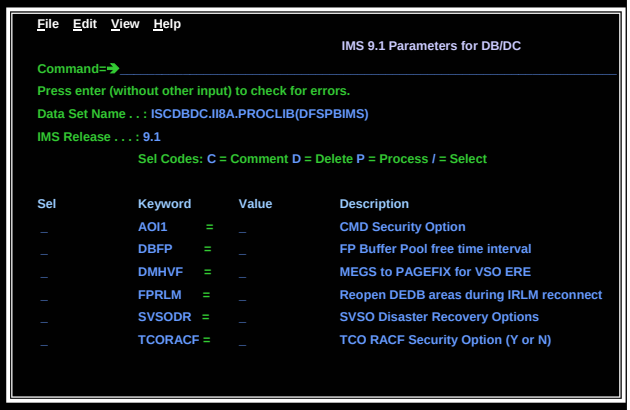
IV_H311T

IV_J309T

IV_G309T

IMS V9 Syntax Checker Enhancements

- Focus on syntax, parameter validity and consistency
  - *Display Default value of keywords*
  - *New Select Command: "P" Process*
- Update of the DFSPBxxx members for new V9 parameters
- New Members processed by Syntax Checker
  - *DFSDCxx Proclib member*
  - *DFSSQxx Proclib member*



IMS V8 DFSPBIMS processed by Syntax Checker, showing parameters that are new in IMS V9

The Syntax Checker provides an ISPF user interface that enables you to define, verify, and validate parameters and their values in IMS.PROCLIB members. With IMS Version 9, you can:

- View default values for parameters.
- There is a new Select command “P” for Process which enables you to Interrupt the processing of the current member and begin processing the selected member.
- Update the DFSPBxxx members for new Version 9 parameters.
- Process the DFSDCxxx and DFSSQxxx members. (Data Communications Options member, and the CQS address space member for Shared Message Queues.)

IMS V9 Migration Considerations

- IMS V9 supports migration from V7 and V8
  - *V7 and V8 DBRC Migration/Coexistence SPEs*
  - *V8 OLR Coexistence SPE*
    - PQ78493, PQ78758, PQ78916, and PQ78917
      - » The SPEs have many pre-reqs including PQ73858/UQ81733 PDU SPE ***(this is a large SPE and affects over 100 parts in IMS)***

- V7 migration considerations to V8 (or V9)
  - *OTMA Compatibility SPE for prefix change PQ58631*
  - *MSC, QCF, Shared Qs Compatibility SPE PQ32932*
  - *DFSMSCE0 user exit replaces old MSC exits*
  - *UQ99327 needed for IMS V7 Change Accumulation migration to IMS V8*
  - *IMS V7 with PQ63491 to use IC2's with the "SameDataset" option created on IMS V9*
  - *IMS V8 was the last version to support the V5 format for SDEP CIs.*

IMS V7 and/or IMS V8 require DBRC Migration/Coexistence SPE in order to access and process a RECON that has been migrated (upgraded) to the V9 format.

IMS V8 data sharing users who need to access a HALDB that is defined via DBRC in V9 as OLR-capable must install the OLR Coexistence SPE.

IMS V7 users migrating to either IMS V8 or IMS V9 should be mindful of the following V7 SPEs and other considerations, if applicable:

•OTMA Compatibility SPE (PQ58631) addresses an OTMA prefix changed that was made in IMS V8.

•MSC Compatibility SPE (PQ32932).

•IMS V7 is the last release to support the following four MSC user exits: Terminal Routing (DFSCMTR0), Input Message Routing (DFSNPRT0), Link Receive Routing (DFSCMLR0/DFSCMLR1), and Program Routing (DFSCMPR0). V7 introduced a new exit, DFSMSCE0, which combined these exits.

IMS V9 Migration Considerations…

- IMS V9 is the last release to support SMU, BTAM
- IMS V9 is the last release to support Integrated IMS Connect user message exits, HWSIMSO0 and HWSIMSO1
  - These two user message exits will not be available in future
  - Recommend move to HWSSMPL1 or HWSSMPL0

Migration options are documented in the IMS Connect Guide and Reference, SC18-7260-01

IMS V9 is the last release to support the Security Maintenance Utility (SMU) and the BTAM access method for terminal device support.

IMS Connect Version 9 is the final and last release with which HWSIMSO0 and HWSIMSO1 is shipped. It is recommended that you move to HWSSMPL1. You may also move to HWSSMPL0. There are two ways of migrating to HWSSMPL0 or HWSSMPL1.

1. Change the client code to support HWSSMPL0 or HWSSMPL1 by changing the ASCII or EBCDIC settings in your code to one of the following options:

Change *IRMREQ* to *SAMPLE* to move from HWSIMSO0 to HWSSMPL0.

Change *IRMRE1* to *SAMPL1* to move from HWSIMSO1 to HWSSMPL1.

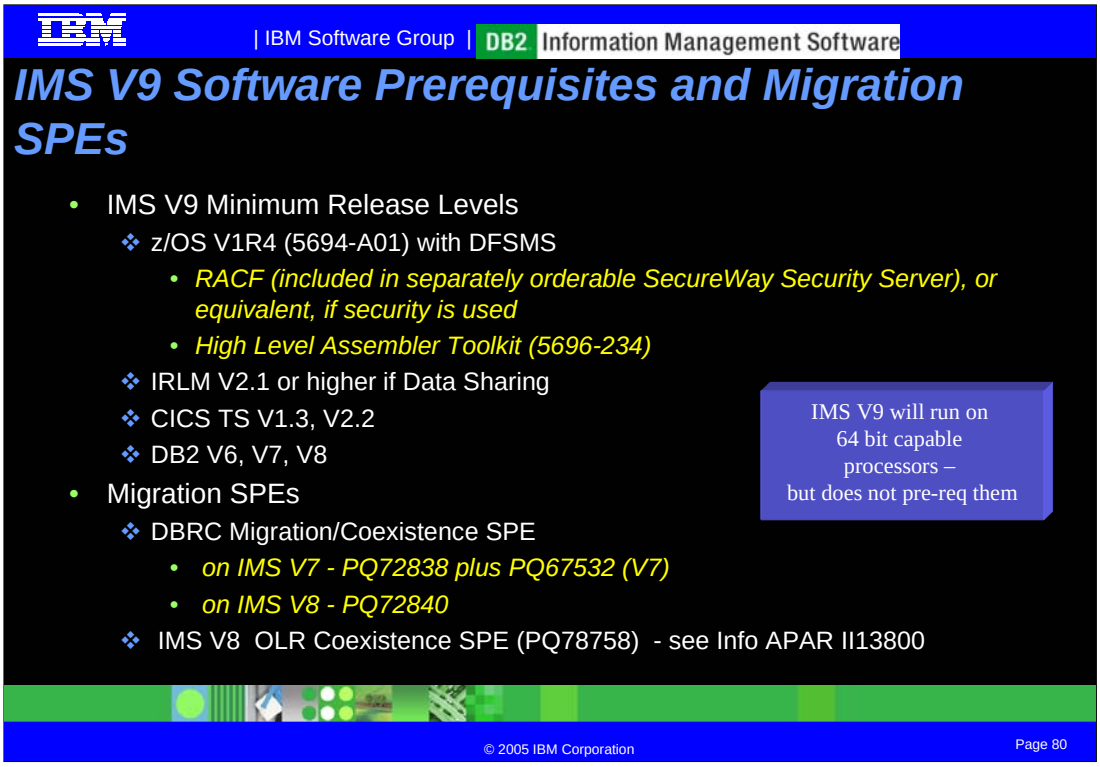
Change *IRMREQ* to *SAMPL1* to move from HWSIMSO0 to HWSSMPL1. You must also modify the client code to accept the fullword output message length field preceding the LLZZ data.

2. Change either HWSSMPL1, HWSSMPL0, or both to accept the current settings as they exist in the client code by changing the ASCII or EBCDIC setting in the user message exit. You can select one of the following options:

Change *SAMPLE* to *IRMREQ* to move from HWSIMSO0 to HWSSMPL0.

Change *SAMPL1* to *IRMRE1* to move from HWSIMSO1 to HWSSMPL1.

Change *SAMPLE* to *IRMRE1* to move from HWSIMSO0 to HWSSMPL1. You must also modify the client code to accept the fullword output message length field preceding the LLZZ data.



IMS V9 operates under z/OS V1R4 (5694-A01) configurations, or subsequent versions, releases and modification levels unless otherwise stated, and requires the following minimum version, or release or modification levels:

•z/OS V1R4 (5694-A01) with DFSMS (a base element of z/OS V1R4).

•RACF (included in a separately orderable SecureWay Security Server feature of z/OS V1R4), or equivalent, if security is used

•IBM High-level Assembler Toolkit (5696-234), a separately orderable feature of z/OS.

DFSMS Macros

The IMS open and close module DFSZD110 (GSAM and BSAM) uses the DFSMS macros EZCTGPL and IEZCTGFL. Beginning with DFSMS 1.5, macros IEZCTGPL and IEZCTGFL are provided on the optional source tape only. If DFSZAD110 needs to be assembled, these macros must be available. Note: DFSZD110 does not need to be assembled to process PTFs. It needs to be assembled only when processing any APARs or USERMODs that affect it.

IMS also operates in a virtual machine (VM) under control of z/OS V1R4 and is intended for use in a customer program development, testing and non-XRF production environment with some restrictions.

Both IRLM 2.1 and 2.2 support IMS V9, and IMS data sharing partners can be on either or the same releases.

DBRC Migration/Coexistence SPEs will also be provided for Migration from IMS V7 or V8. The coexistence SPEs are needed for the changes in DBRC in V9 that were made to support OLR, multi-area structure support for SVSO, and reusing the dmb number when registering dbs to DBRC. (The prior IMS versions will not recognize the MAS/NOMAS settings in DBRC or the ability to support >32K dbs, so the coexistence SPE is required for those reasons, in addition to the OLR changes in V9-olr capable setting for HALDB partitions). PQ67532 (V7) is a performance fix for a problem that's introduced when DBRC starts segmenting large records (16M RECON Record Size enhancement).

IMS V8 data sharing partners will need the OLR coexistence SPE to access a HALDB which has been marked as OLR Capable by IMS V9.

IMS V9 can be used with all IBM processors capable of running z/OS V1R4 or later.

IMS V9 Prerequisites…

- Hardware Prerequisites
  - >32K Tape Blocksize support requires the large block interface on 3480 Magnetic Tape, 3490 and 3490E Magnetic Tape Subsystems, or 3590 devices
- Other Software Prerequisites
  - RACF Enhancements for SMU replacement
    - *Requires an enhancement to RACF (z/OS 1.6) to add new security classes as default classes*
      - Optionally these can be added manually
  - Dynamic Allocation DSAB VSCR requires z/OS 1.5
  - IMS-DB2 Interoperability from a Java Dependent Region requires z/OS Resource Recovery Services and:
    - *PQ69861 (DB2 V7)*
    - *PQ74629 (DB2 V8)*
  - IMS Control Center for IMS requires:
    - *IMS Connect V1.2 or later & DB2 UDB Version 8, with Fixpak 5 or later*
    - *DB2 UDB Version 8.2, with Fixpak 6 or later is required for multi-version support (V8 & V9 in the same IMSplex)*
  - IMS Java Remote Database Services requires:
    - *IBM SDK for z/OS, Java 2 Technology Edition, V1.3.1 or later*
    - *The IMS DB subsystem and the programs that comprise the server tier must all run in the same logical partition (LPAR)*
    - *The server-tier EJB must run in WebSphere Application Server for z/OS, Version 5 or later. The middle-tier EJB can run in any WebSphere Application Server.*

To use the “RACF Enhancements to Replace SMU” , the RACF product must include the following four new RACF security classes as default classes:

•IIMS - Program Specification Block (PSB)

•JIMS - Grouping class for PSB

•LIMS - Logical terminal (LTERM)

•MIMS - Grouping class for LTERM

If these RACF resource classes are not already included in the default list of classes for RACF, or if you want to use class names that are different from the default names, you need to define them in the resource class descriptor table (CDT). To define a new resource class, run the RACF resource class macro, ICHERCDE. The procedure for adding classes to the class descriptor table is described in "Adding Installation-Defined Classes" in the z/OS Security Server RACF: System Programmer's Guide.

When IMS recognizes it is running on a z/OS 1.5 or later system, it will ensure that DSAB blocks created as part of the allocations for FF or FP databases, will be above the line.

Support for interoperation between Java and COBOL when running in a JMP or JBP region requires IBM Enterprise COBOL for z/OS and OS/390 Version 3 Release 2, or later.

IMS Java and the DLIModel utility require Xalan-Java version 2.6.0 or later from the Apache Software Foundation (www.apache.org) XML Project (xml.apache.org), or equivalent code function. The Apache XML Project is a collaborative software development project that licenses open source software at no charge. The following open source files (or equivalent code function) are required by IMS Java and the DLIModel utility: **xercesImpl.jar, xalan.jar, xml-apis.jar.**

IMS Java Remote Database Services JDBC access to IMS DB from WebSphere Application Server (on a non-z/OS platform) requires:

•WebSphere Application Server for z/OS Version 5.0.2.3 or later. If you have WebSphere Application Server for z/OS V5.0.2, you must install either V5.0.2.1 or apply APAR PQ81944.

•WebSphere Application Server V5.0.2.2 with cumulative fixes that include PQ79485, or WebSphere Application Server V5.0.2.3 or later.

To use IMS Java Remote Database Services to access IMS databases from applications that run on WebSphere Application Server in a non-z/OS environment, you must download IMS Java files from the IMS Java Web site. These downloaded files are required in addition to the files that are installed as part of the SMP/E installation of the IMS Java FMID. To download the required IMS Java files, go to the IMS Web site at www.ibm.com/ims and link to the IMS Java Web page for more information.

Packaging

- IMS V9 product number - 5655-J38

FMID	Feature Description
HMK9900	System Services
JMK9901	Database Manager
JMK9902	Transaction Manager
JMK9903	Extended Terminal Option (ETO)
JMK9904	Recovery Level Tracking (RSR)
JMK9905	Database Level Tracking (RSR)
JMK9906	IMS Java
HIR2101	IRLM 2.1
HIR2102	IRLM 2.2

- Integrated IMS Connect function
  - Included in the IMS Systems Services FMID, HMK9900
- IMS Java Remote Database Services
  - Download IMS Java Files from IMS Website
    - http://www.ibm.com/ims

This chart shows the FMIDs for the components that make up the IMS V9 product.

IMS V9 supports IRLM V2, either V2.1 or V2.2. Members of an IMS data sharing environment can be on either IRLM 2.1 or 2.2, and both releases can coexist with each other.

Note: The release/s of IRLM distributed with IMS V9 at GA time depend on the order type as below:

- •CBPDO or ProductPac==> customer would receive both IRLM 2.1 and IRLM 2.2, and choose which they prefer to install
- •ServerPac, FunctionPac or SystemPac ==> customer would receive IRLM 2.2
- •Standalone ==> customer would receive both IRLM 2.1 and IRLM 2.2 and choose which they prefer to install
- •CustomPac ==> CustomPac is made up of three offerings: FunctionPac, ProductPac®, and SystemPac®. Which IRLM version you get with CustomPac depends on which offering you order.
- •SystemPac and FunctionPac ==> customer would receive IRLM V2.2

Integrated IMS Connect function: The IMS Connector for Java 9.1.0.1 runtime component for z/OS is delivered with the base of IMS. The IMS Connector for Java 9.1.0.1 development component which ships with WebSphere Studio Application Developer Integration Edition (WSADIE) will be delivered as an update to WSADIE 5.1.1 in the same timeframe as the IMS V9 General Availability.

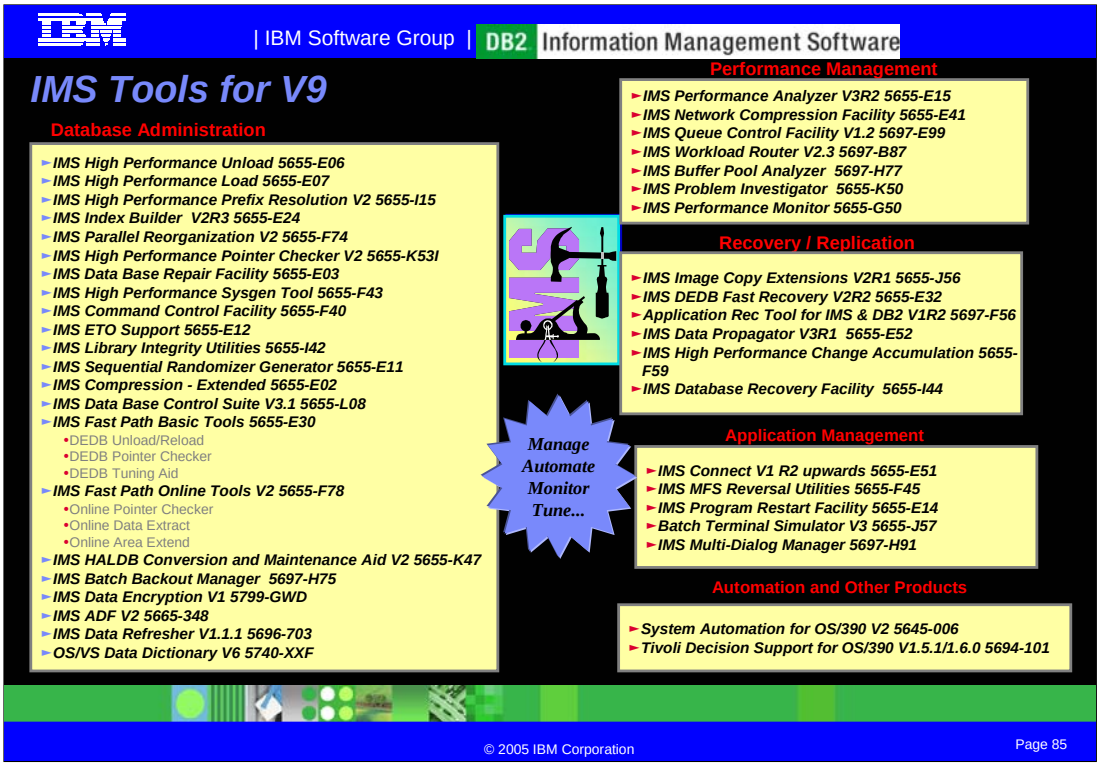
IMS Java Remote Database Services: To use IMS Java RDS to access IMS databases from applications that run on WebSphere Application Server on a non-z/OS, you must download IMS Java files from the IMS Java Web site. These files are required in addition to the files that are installed as part of the SMP/E installation of the IMS Java FMID. To download the required IMS Java files, go to the IMS Web site at http://www.ibm.com/ims and link to the IMS Java Web page for more information.

Some Key Dates

Product	Date	Announcement Letter
IMS V9 (5655-J38)	September 9, 2003 October 29, 2004 *New!*	ZP03-0375 IMS V9 Announced *ZP04-0409 IMS V9 GA*
IMS V8 (5655-C56)	October 16, 2001 October 25, 2002	ZP01-0518 IMS V8 Announced ZP02-0447 IMS V8 GA
IMS V7 (5655-B01)	October 27, 2000 September 8, 2004 August 3, 2004 *New!*	ZP00-0488 IMS V7 GA ZP04-0249 IMS V7 Withdrawn from Marketing *ZP04-0312 Withdrawal from Service announced for November 8, 2005*
IMS V6 (5655-158)	December 26, 1997 September 4, 2002 *September 30, 2003*	ZP97-0463 IMS V6 GA ZP02-0255 IMS V6 Withdrawn from Marketing *ZP02-0343 IMS V6 Withdrawn from Service*

IMS V9 and IBM Tools

- Web Matrix - Available at:
  http://www-1.ibm.com/support/docview.wss?rs=434&context=SSZJXP&uid=swg21167251&loc=en_US&cs=utf-8&lang=en+en
- Preventive Service Planning (PSP) Buckets
  - Have been updated with IMS V9 requirements
  - Available at:
    https://techsupport.services.ibm.com/server/390.psp390



The above information is correct as at 29/09/2003 but should be reverified at General Availability.

In addition to IMS, IBM has been providing a wide range of price/performance, competitive Systems Management tools for IMS. This shows a summary of the IBM IMS tools currently available. The tools provide support for speeding up and reporting on performance, extend the functions of and assist with testing of IMS, and provide system tools for querying, validating, managing, and tuning the IMS Database. These include for example tools necessary to maintain and repair databases. Many tools serve multiple purposes. IBM offers tool functionality like IMS Control Suite that is not available from any other vendor. IBM offers high performance tools that are competitive within the industry at an affordable price. In fact when taken together "price/performance and functionality", IBMs IMS tool can be considered the best in the industry.

We have over 30 products to support all aspects of IMS usage.

- Utilities for full function and fast path database provide a high performance solution that improves IMS availability.
- Administrative tools make managing large and small IMS systems easier and faster.
- Performance management tools help you tune IMS systems and avoid outages.
- Recovery and replication tools enable fast and effective transfer of data from transactional to informational systems.
- And application management tools make application runtime environments more effective.

Documentation

- IMS V9 Information:
  - Unlicensed publications included in DB2 Information Management Software Information Center for z/OS Solutions
    - **http://publib.boulder.ibm.com/infocenter/dzichelp**
  - Licensed and unlicensed publications available in PDF and Bookmanager formats from the IMS Website
    - **http://www.ibm.com/ims**
    - *Licensed IMS books require a valid IMS license or customer number to download*
  - New book
    - IMS V9: IMS Connect Guide and Reference
  - DLIModel utility info. moved to *IMS Version 9: Utilities Reference: System*.
  - *"An Introduction to IMS"* – ISBN # 0-13-185671-5
    - *Available now from Pearson Publishing*

The IMS Version 9 information is now available in the DB2 Information Management Software Information Center for z/OS Solutions, which is available at http://publib.boulder.ibm.com/infocenter/dzichelp. The DB2 Information Management Software Information Center for z/OS Solutions provides a graphical user interface for centralized access to the product information for IMS, IMS Tools, DB2 Universal Database (UDB) for z/OS, DB2 Tools, and DB2 Query Management Facility (QMF™).

IMS Version 9: IMS Connect Guide and Reference

The library includes new information: *IMS Version 9: IMS Connect Guide and Reference*. This information is available in softcopy format only, as part of the DB2 Information Management Software Information Center for z/OS Solutions, and in PDF and BookManager formats. IMS Version 9 provides an integrated IMS Connect function, which offers a functional replacement for the IMS Connect tool (program number 5655-K52). In this information, the term *IMS Connect* refers to the integrated IMS Connect function that is part of IMS Version 9, unless otherwise indicated.

The information formerly titled *IMS Version 8: IMS Java User's Guide* is now titled *IMS Version 9: IMS Java Guide and Reference*. This information is available in softcopy format only, as part of the DB2 Information Management Software Information Center for z/OS Solutions, and in PDF and BookManager formats.

To complement the IMS Version 9 library, a new book, from IBM Press is available to complement the IMS Version 9 Library. An Introduction to IMS: Your Complete Guide to IBM's Information Management System, by Dean H. Meltz, Rick Long, Mark Harrington, Robert Hain, and Geoff Nicholls (ISBN 0-13-185671-5), is the first externally published IMS book to include information about V9. It features key coverage on security, message format services, system recovery, and Java programming. Written by the team responsible for IMS' development and deployment, this unique book is updated to include the new enhancements available with V9. This book is available now from Pearson Publishing.

IMS Information

- Information is available at http://www.ibm.com/ims
  - Presentations/Papers, Newsletters, Redbooks, Fact Sheets, Announce Letters, additional documentation including:
    - *IMS V9 Fact Sheet – GC18-7697*
    - *IMS V9 Release Planning Guide – GC17-7831-01 (October 2004)*
    - *IMS V9 Installation and Migration Tips*
    - *Converting IMS/SMU Security to RACF with IMS Version 9*
    - *IMS V9 Performance Summary*
    - *IMS V9 Value Brochure*
  - **IMS Examples Exchange**
    - *This resource enables you to exchange IMS examples with other IMS users*
  - **The IMS Information Roadmap is now available**
    - *This roadmap outlines the information and educational resources available for IMS*
  - **IMS Favorites CD-ROM (LK3T-7144)**
    - *You can order the IMS Favorites CD-ROM (LK3T-7144) from the IBM Publications Center. This CD contains:*
      - The IMS/ESA Version 7, IMS Version 8, and **IMS Version 9** licensed/unlicensed softcopy libraries in BookManager and PDF formats
      - The IBM DB2 and IMS Tools library in BookManager and PDF formats
      - Selected IMS redbooks in PDF format

A wide range of IMS information is available:

Additional documentation and information is available from the IMS home page at http://www.ibm.com/ims.

IMS Information…

- IMS Redbooks
  - SG24-6398 IMS V9 Implementation Guide: A Technical Overview
  - SG24-6945 The Complete IMS HALDB Guide
  - SG24-6404 IMS Performance and Tuning Update
  - SG24-6514 IMS e-business Connectors: A guide to IMS Connectivity
  - SG24-6533 Ensuring Data Integrity Using IMS Tools
  - SG24-6574 IMS Installation and Maintenance Processes
  - SG24-6866 Using IMS Data Management Tools for Fast Path Databases
  - SG24-6908 IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology
  - SG24-6928 IMS in the Parallel Sysplex Volume II: Planning the IMSplex
  - SG24-6929 IMS in the Parallel Sysplex Volume III: IMSplex Implementation and Operations
- *IMS Education (EMEA) at:* http://www.ims-society.org
  - *US at:* http://www.ibm.com/services/learning
- *Upcoming 2005 Conferences:*
  - IMS Technical Conference – October 10-13, 2005 San Jose, California, US
  - EMEA IMS Technical Symposium – November 14-17, 2005 Koenigswinter, Germany

A wide range of IMS information is available:

Additional documentation and information is available from the IMS home page at http://www.ibm.com/ims.

The IBM International Technical Support Organization has been producing redbooks and redpieces with additional information, available at http:// www.redbooks.ibm.com. A number of IMS Technical Conferences are also being provided on an ongoing basis.

IMS Version 9

- Providing quality through on-demand solutions

Integration

Manageability

Scalability

IMS V9 helps in the evolution to on demand business through integration, openness, autonomic computing, and virtualization, providing:

INTEGRATION with other IBM products and with other products and platforms within the industry, supporting IBM's commitment to open standards that benefit you, and developing supporting tools for application development and connectivity.

MANAGEABILITY through staging users to Autonomic Computing, easing use, eliminating/reducing outages, and minimizing the education curve for users of IMS.

SCALABILITY through virtualization in assuring flexibility for growth and expansion in a homogeneous environment while utilizing many heterogeneous data and application sources and new hardware and software facilities to optimize performance, capacity, availability, and recovery. This includes new levels of enhanced availability for IMS High Availability Large Databases (HALDB), introduced in V7, with fully integrated Online Reorganization support that provides concurrent online update and availability of data.


IBM
| IBM Software Group |
DB2
Information Management Software
Thank You
© 2005 IBM Corporation
Page 90