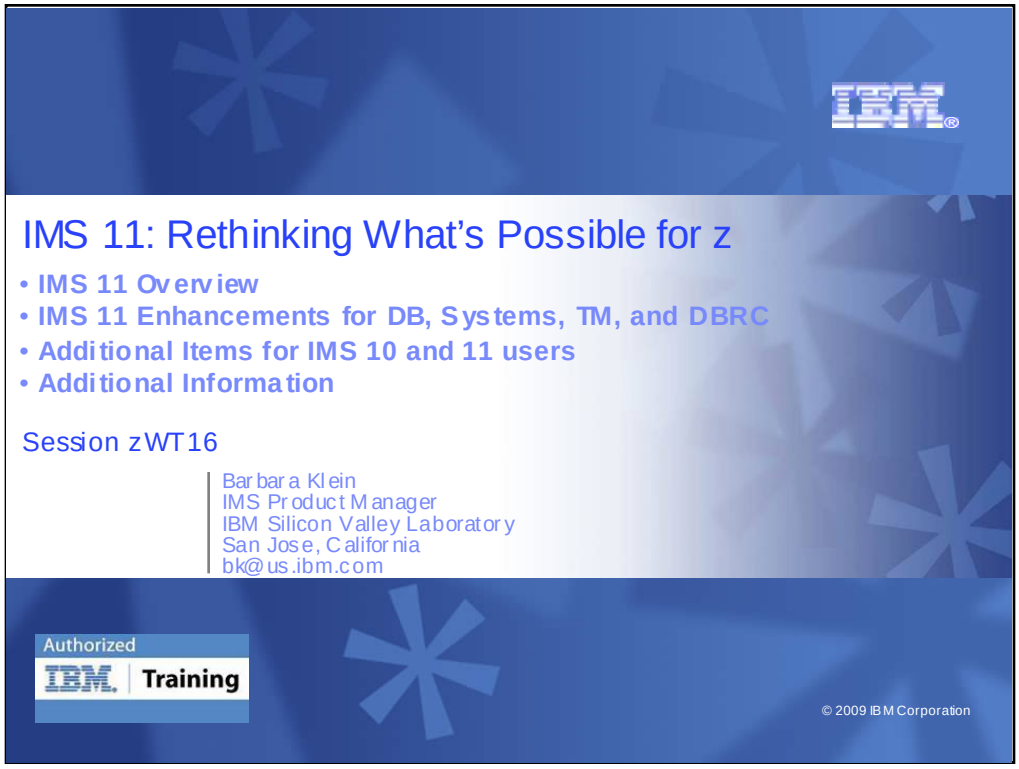


IMS 11 is being made generally available Oct 30, 2009. I'll be taking your through a high level overview of it, the enhancements within, additional items for IMS 10 as well as IMS 11 users, and some additional information on migration, coexistence considerations.

Key Message: IBM continues to enhance IMS

IMS 11 continues to address customer requirements for lowering Costs in the continuing journey.

IMS 11 provides solutions to ease integration with new technology for a service oriented architecture, focusing on:

•IMS Open DB for direct distributed TCP/IP access to IMS data provides cost efficient, enabled application growth, and improves resilience,

•Enhanced IMS Connect, the TCP/IP gateway to IMS transactions, operations, and now data, improves IMS flexibility, availability, resilience, and security. And IMS Open Transaction Manager Access (OTMA) enhancements improve storage utilization, reduce dependency on RRS, enhance Security.

•Broadened Java and XML support and tools eases IMS application development and connectivity, and enhances IMS web services to assist developers with business transformation. broadened Java and XML support and tools to ease development and access of IMS data. Enhanced Websphere and Rational Tools for IMS application development and enablement are providing SOAP, XML, Web Services and other standards support enhancements, for example enhanced MFS BPEL support for WebSphere Integration Developer (WID) and Websphere Process Servers (WPS), and Mashup support for IMS Data and enhanced IMS Transaction access,

IMS 11 provides solutions to help simplify installation and management, focusing on:

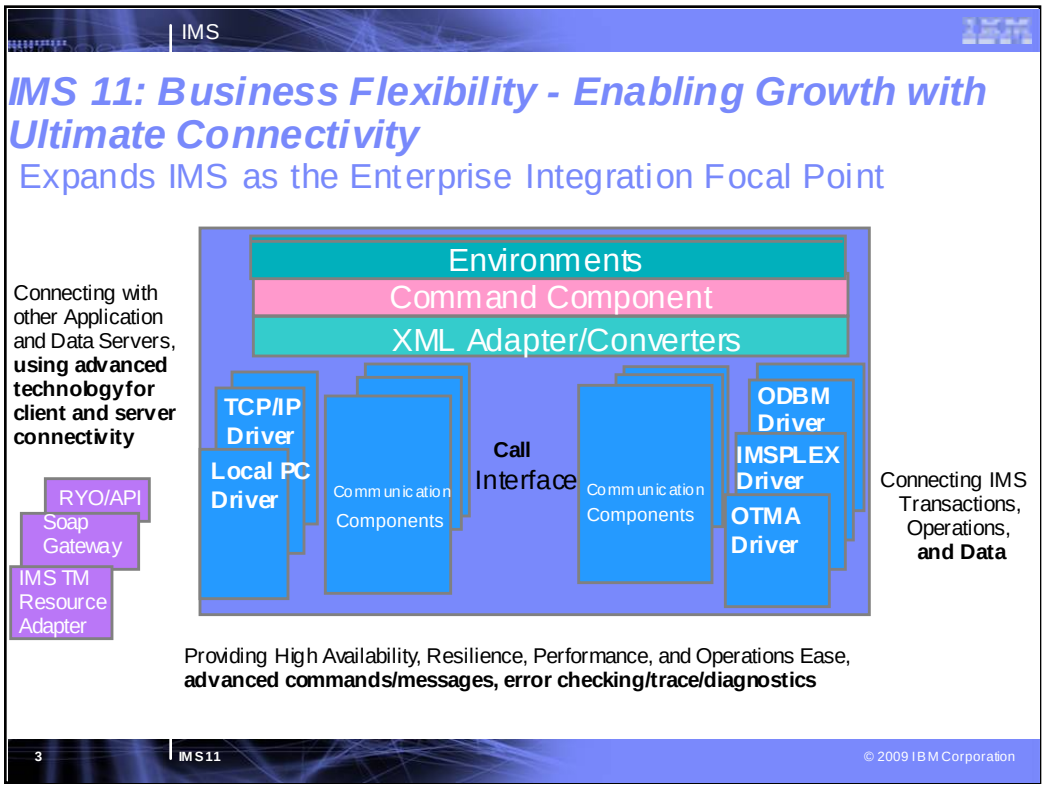
•Enhanced commands and User exits that simplify operations and improve availability

•Syntax checking that ease installation

IMS 11 provides enhancements for high performance, scalable, reliable, secure solutions, focusing on:

•IMS Fast Path Buffer Manager, Application Control Block library, and Local System Queue Area storage reduction utilizing 64-bit storage to improve availability and overall system performance

•Security enhancements



Key Message: IMS provides and continues to enhance the integrated IMS Connect function.

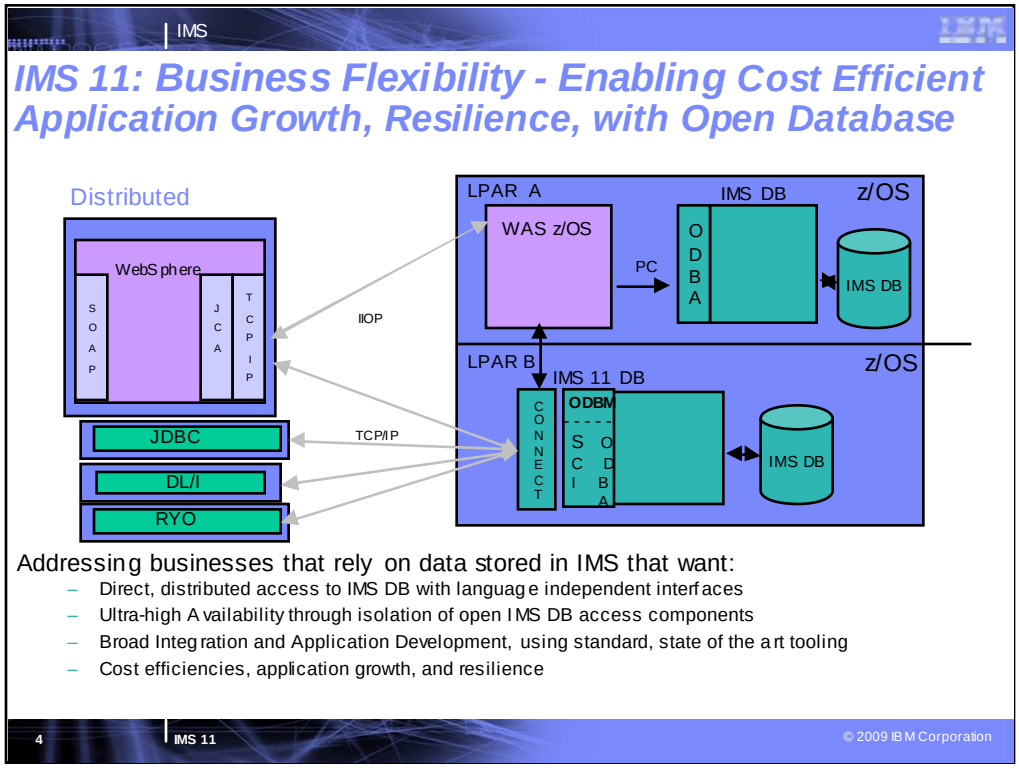
IMS Connect function is part of the overall restructure of IMS for the 21st Century and is architected as the base for all future IMS Connectivity. Much of the function of IMS Connect can also be used with earlier IMS Versions so you can start to take advantage of it before migrating your networks/applications/databases to IMS V9. The structure of IMS Connect is designed such that drivers can be interchangeable. That is, alternatives for the TCP/IP front end or OTMA back end interfaces are already being provided. These are allowing IMS to exploit newer, additional, and enhanced protocols and/or interfaces. Along with IMS Connect is provided the IMS Connector for Java for access from Java applications, SOAP Gateway and parsers, and samples for other language access as well.

With IMS Version 8, IMS extended its use of XCF for use by other IBM subsystems, such as IMS Connect , for distributed operations access through the Structured Call Interface to the Operations Manager from the DB2 Version 8 Control Center as a single point of control.

With IMS Version 9 this function was integrated in.

With this structure IMS 11 Connect is evolving to address other connectivity requirements -- distributed database access to IMS DB, enhanced client connections,

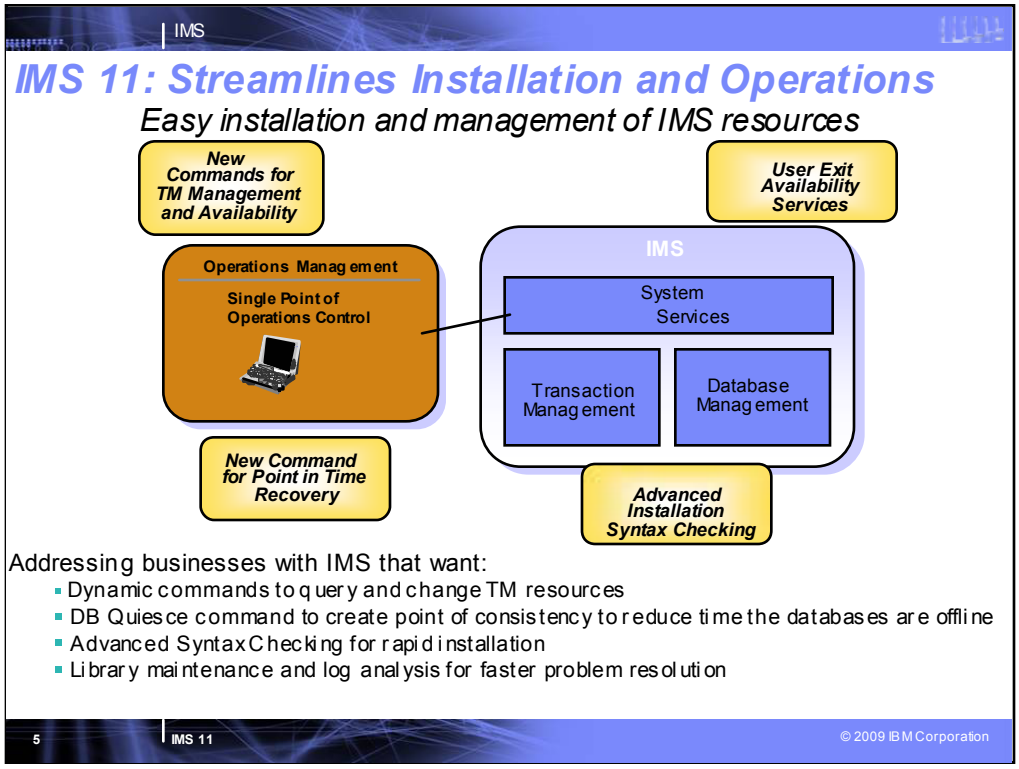
.



Key Message: IMS 11 is also addressing requirements for extended distributed database access.

LPAR A is the existing solution for IMS database access, with access from distributed environments requiring a transaction interface through IMS TM or another z/OS subsystem; for example, WebSphere z/OS can be the TCP/IP endpoint on the mainframe with IMS libraries using the Open Database Access (ODBA) API to access IMS databases.

IMS 11 provides direct access to IMS data, leveraging the integrated IMS Connect function as the TCP/IP endpoint on the mainframe for data, as well as transactions and operations provided earlier. This nicely positions IMS Connect as the complete gateway to IMS resources…both TM and DB related. This is shown in LPAR B. This separate LPAR approach also provides for failure isolation between IMS DB and subsystems, such as WAS z/OS, enhancing resilience, cost efficiencies, and enabling application growth.



Key Message: IMS 11 simplifies Operations and eases installation

IMS 11 is easing the installation and management of IMS resources with

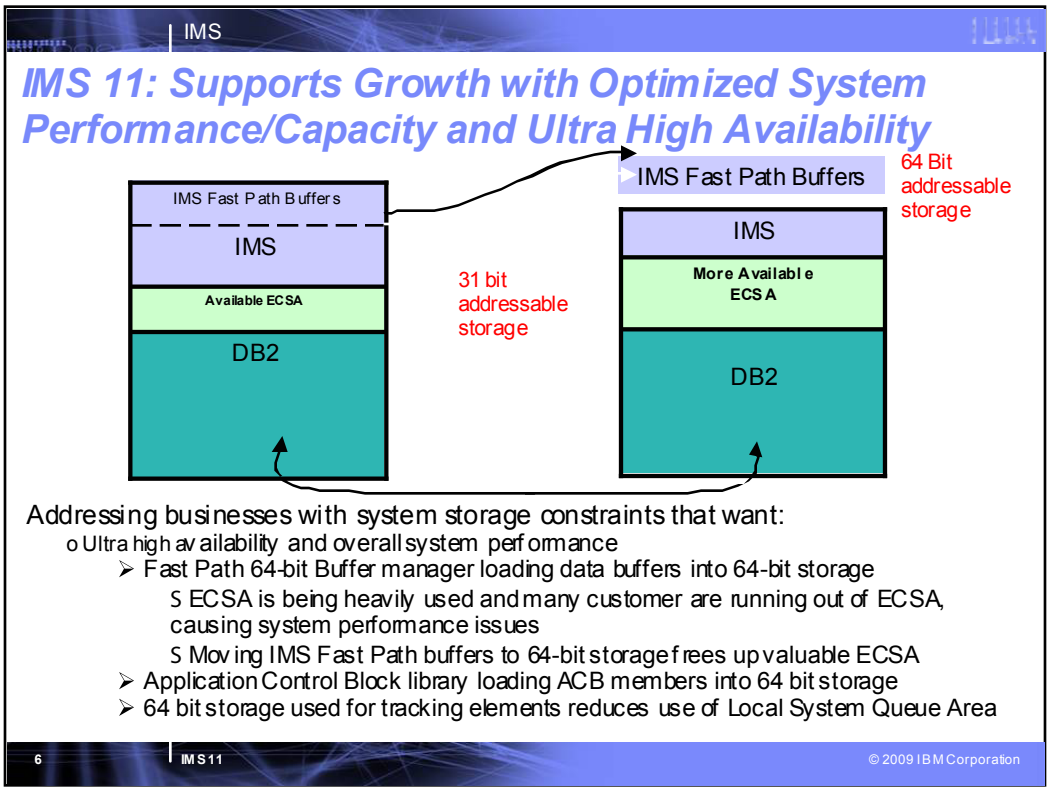
Dynamic commands to query and change IMS TM resources

An IMS DB Quiesce command to create point of consistency to reduce time the database are offline

User exit services enhancements to define multiple instances and to dynamically refresh modules

Enhanced Syntax Checking for simplifying installation

RAS enhancements to minimize library maintenance mistakes and enhance log analysis



Key Message: IMS 11 is also enabling growth

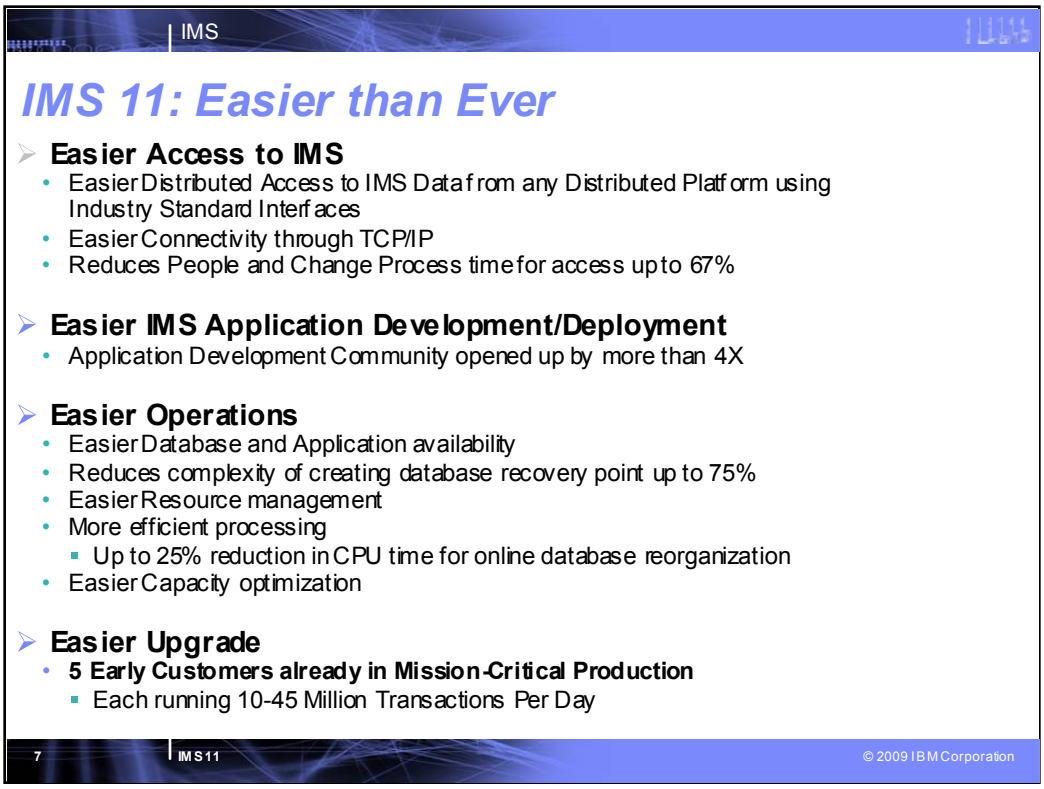
IMS 11 enables growth with 64 bit support, improving availability and overall system performance with Fast Path 64-bit Buffer manager loading data buffers into 64-bit storage

ECSA is being heavily used and many customers are running out of ECSA, causing system performance issues

Freeing up IMS Fast Path buffers to 64-bit storage frees up valuable ECSA

Application Control Block library loading ACB members into 64 bit storage improves storage utilization and performance

64 bit storage used for tracking elements reduces use of Local System Queue Area



IMS 11 is providing:

Easier Access to IMS using Open Database and Connect

•Easier access to IMS Data from any distributed platform using Industry Standard Interfaces

•Easier Connectivity through TCP/IP

•Reduced People and Change Process time for data access up to 67% - In the past, many companies had to replicate IMS data to relational structures in order to gain standard access to the data. With IMS 11, all applications can now access IMS data using standard interfaces; therefore a significant portion of data replication and change data capture processes can be either simplified or eliminated all together. With IMS 11, data can be accessed directly from the application requiring the data; no intermediate steps are needed. Therefore we anticipate customers can eliminate up to 67% of their current data replication processing.

•67% estimate based on comparison of old process vs. new process. From a 3 step old process (Create copy of IMS Data, convert IMS data to relational format; load IMS data into relational structure) to a 1 step new process (access IMS data directly). A customer example is of a southwestern US state, whose IMS data is the master data. They have written most new applications on .NET and each night take an image copy of IMS data and load to SQL server. .NET applications access data from SQL Server. With IMS 11, .NET applications can access IMS data directly, eliminating nightly data replication process. .NET applications now have access to most current master data from IMS. Applications can also start doing update (was read only in the past)

Easier Operations

•Easier Database and Application availability

•Reduces complexity of creating database recovery point up to 75% - With the new Quiese commend, application recovery points can be created more easily. In the past the process of creating a recovery point was a manual process: 1) stop db resources, 2) check to ensure all db resources are stopped, 3) start db resources, and 4) ensure all db resources are restarted. With IMS 11, 1 Quiece command does all of this – go from 4 steps to 1 step.

•Up to 25% reduction in CPU time for online database reorganization processing. Specific measurements were done at the SVL lab that resulted for IMS 11 compared to IMS 10: for root only databases, OLR logging reduced by 20%, locking reduced by 65%, CPU usage reduced by 18%; and for multisegment databases, OLR logging reduced by 60%, locking reduced by 80%, CPU usage reduced by 24%

•Easier Resource management

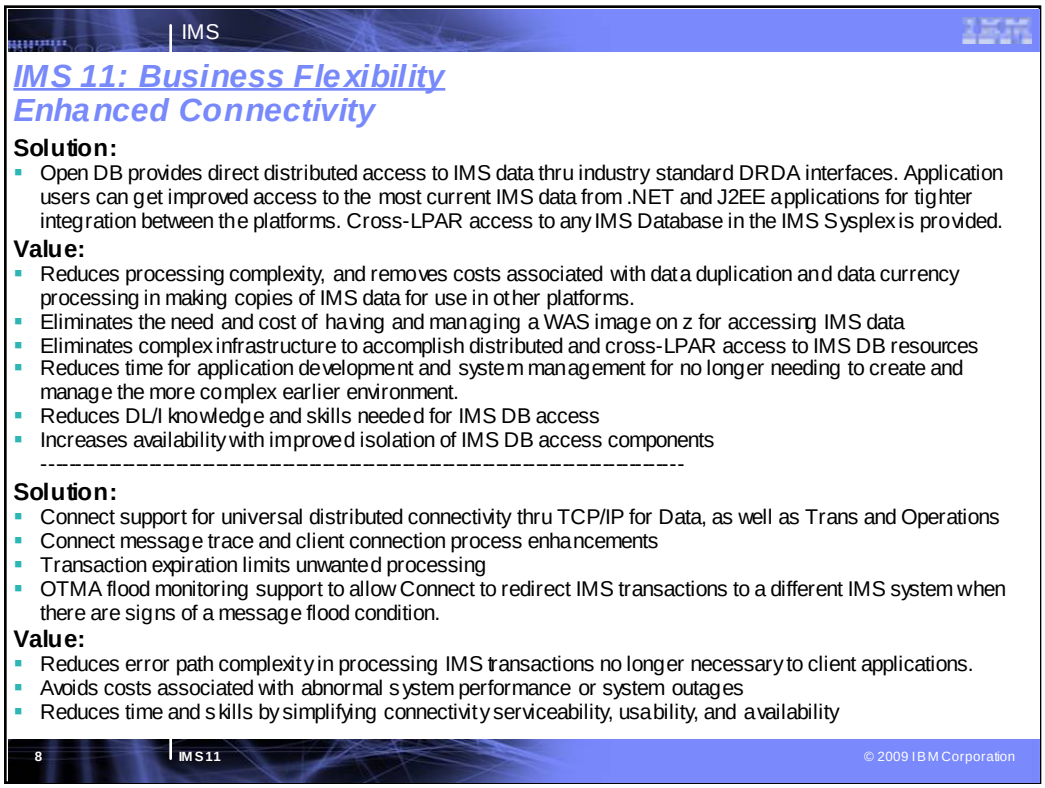
•Easier Capacity optimization

Easier Application Development/Deployment

Application Development Community opened up by 4X. By support JAVA, organizations can now leverage a larger portion of their developer talent pool for IMS application development and support. According to a Gartner survey, estimated number of application developers in 2007 were 3M Java, 2.5M VB .Net, 2.0 M C#, 1.5 M Cobol, and .5M C/C++. Traditionally, IMS supports Cobol and C/C++ (2M programmers). With IMS 11, IMS now supports an estimated 9.5M programmers – 400% increase in IMS developer skills

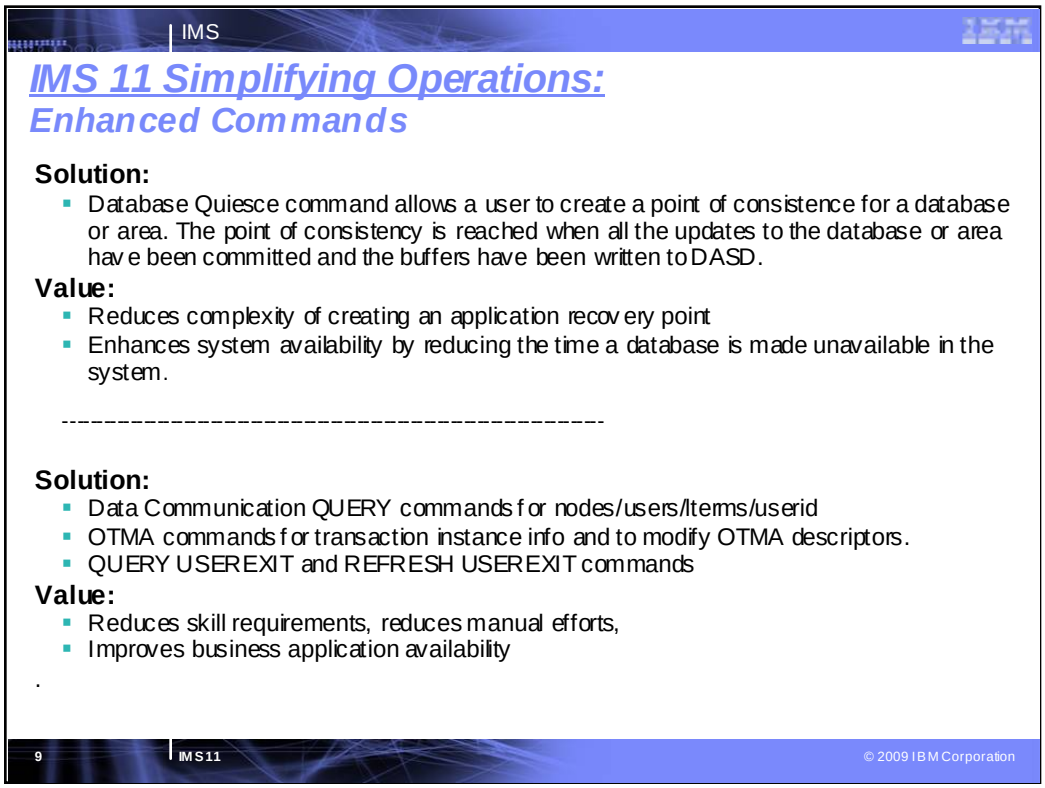
40% of Early Support Customers already in Production. 5 are already in mission critical production, each running 10-45 Million Transactions Per Day

The next few charts show some of the IMS 11 items contributing to this.



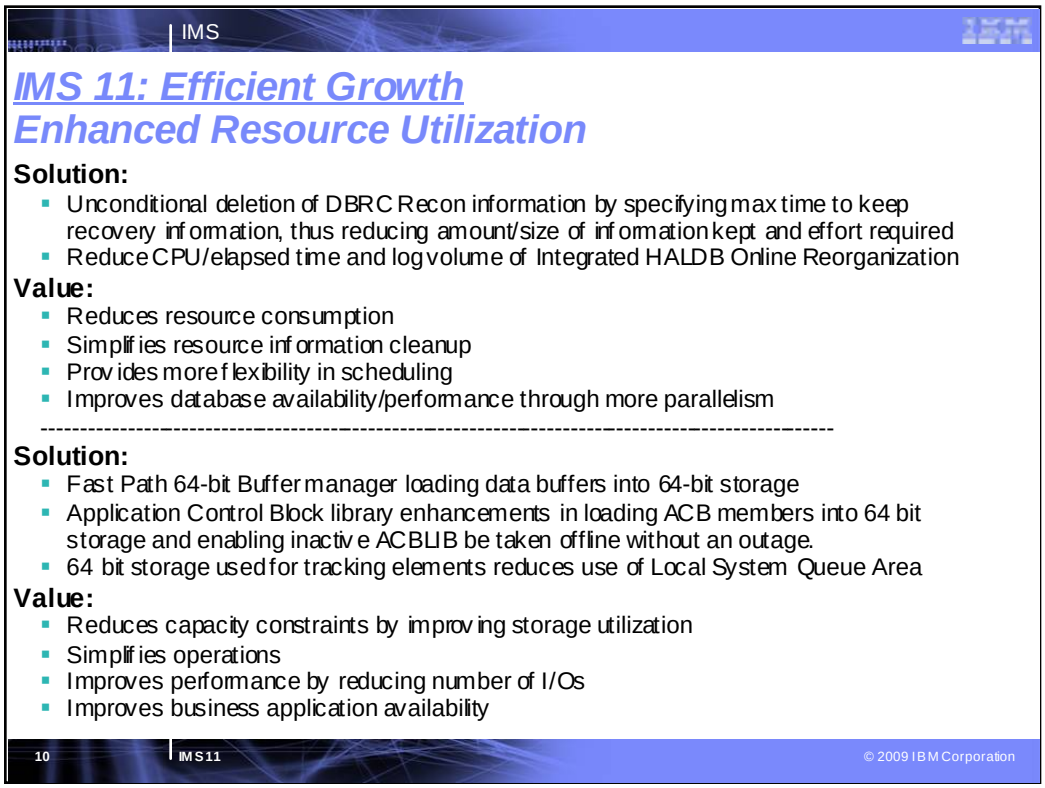
Key Message: IBM is enhancing IMS connectivity

IMS connectivity enhancements simplify the process of connecting to IMS applications and data. These would reduce skill requirements, reduce manual efforts, and improve business application and data availability.



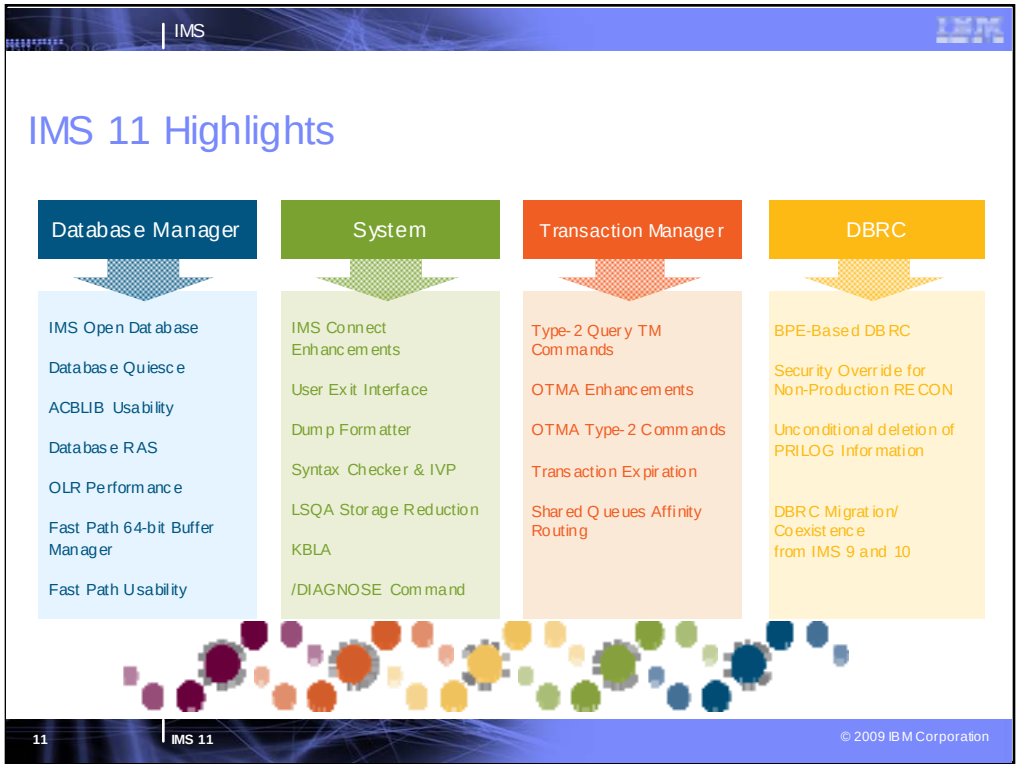
Key Message: IBM is enhancing IMS to ease manageability

IMS Manageability enhancements can simplify the process to create/modify database resources and enhance availability when integrating or modifying existing business functions. Extensions provide new Commands that can reduce skill requirements, reduce manual efforts, and improve system and application availability.



Key Message: IBM is enhancing IMS Scalability

IMS scalability enhancements increase availability, recovery, capacity and performance of the IMS environment. DBRC continues to be enhanced for performance/capacity with multithreading capability. And the Integrated HALDB Online Reorg function is being enhanced to reduce CPU time, elapsed time and log volume. And Fast Path buffer management enhancements can improve usability, flexibility and ease ECSA usage. This additional parallelism and reduced resource consumption would improve database availability and performance.



This shows many of the IMS 11 enhancements in the DB, Systems, TM and DBRC areas

First let's talk about the DB enhancements.

IMS Open Database is a new function in IMS 11 taking on the challenge of modernizing IMS DB access and application development.

It addresses two significant bottlenecks for business growth:

•Connectivity – IMS DB has been historically grounded to the mainframe…certainly there are ways to get to it but none straightforward and simple.

•Programmatic access – even when connectivity isn't an issue – the skills aren't readily available to develop new application workload. DLI isn't industry standard and skills aren't plentiful.

With IMS 11 we are rolling out a complete suite of Universal drivers in support of IMS database connectivity and programmatic access. The intent is to access IMS in a uniform way using the most relevant industry standards from any platform and from within the most strategic runtimes. A standards-based approach opens a lot of growth and expansion opportunity. The fundamental communication protocol we will use to communicate with IMS Connect will be the industry standard Distributed Relational Database Architecture (DRDA) protocol. Single Universal driver in support of both type-4 and type-2 connectivity in all supported runtimes – there's no need to learn another driver's semantics to toggle between environments and desired connectivity – it's all built in to the framework. Distribution of resources within an IMSplex is included. The idea is to extend the reach of IMS by extending the data. IMS DB metadata is exposed via the standard JDBC API and therefore can be consumed and visualized by JDBC tooling. By allowing inspection of metadata, the next step is query. Query syntax uses standard query language syntax.

IMS Open Database – IMS Open Database offers direct distributed access to IMS database resources. The distributed nature is two-fold. At the IMSplex level, it allows cross-LPAR access to any IMS database in the IMSplex. At the pure distributed level, it allows non-mainframe (e.g., Windows OS) access directly to IMS database resources through industry standard interfaces. This enhancement extends IMS Connect as the gateway to IMS DB. It adds a new Common Service Layer address space which manages connections to the IMS ODBA interface. This enhancement improves application access to IMS.

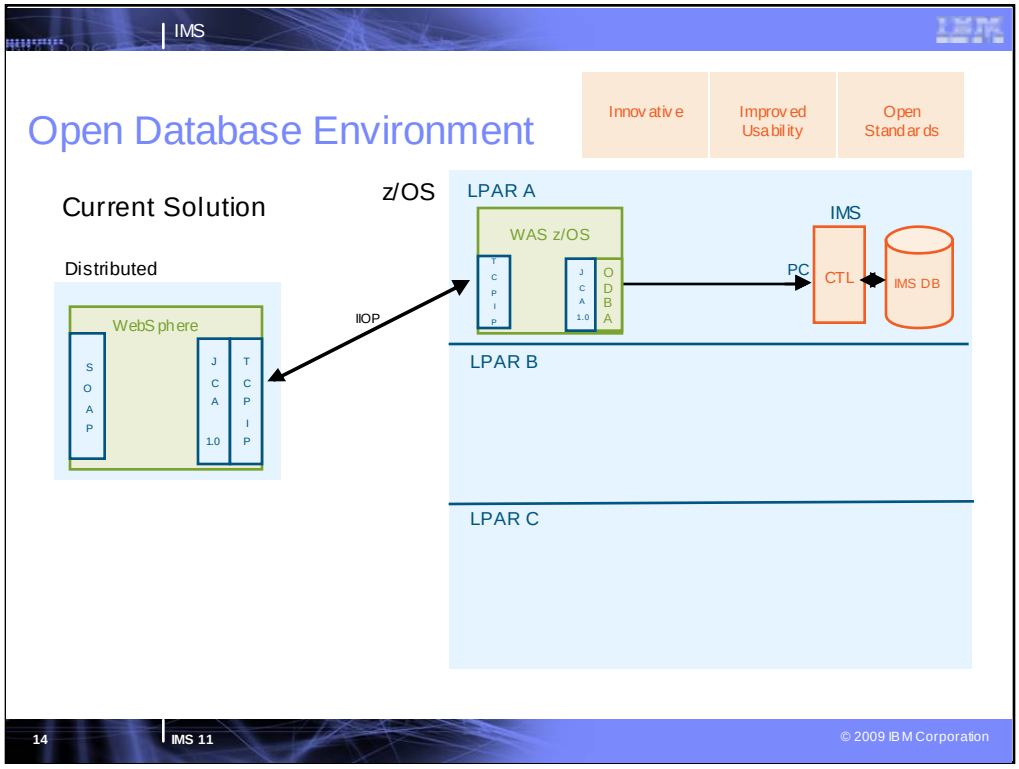
IMS has seen an increased number of requests for distributed access to all database types. IMS Connect is currently the gateway to IMS TM. It will also become the gateway to IMS DB.

Distribution of database assets comes in two flavours

Distribution within an IMSplex. Applications on one LPAR can access an IMS database on another LPAR

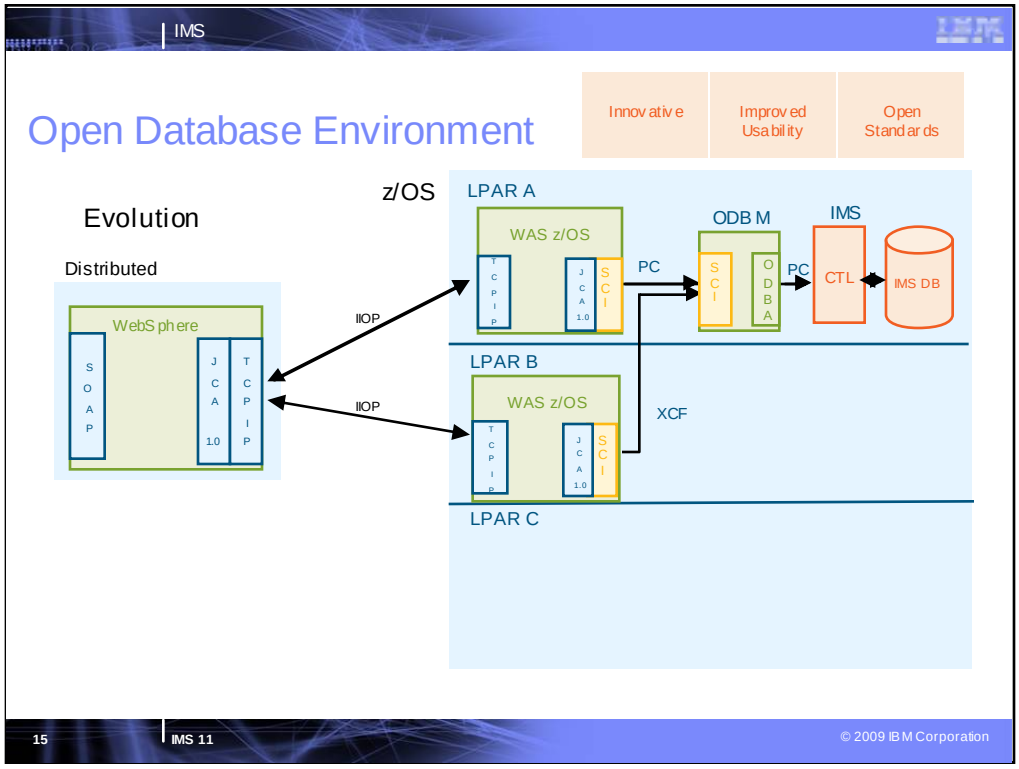
Distribution to non-System z platforms. Applications on a non-System z platform can have direct IMS DB access without needing an IMS transaction to proxy the data.

The Universal drivers (JCA, JDBC, DLI) will allow both distributed as well as local (CICS, IMS, WAS z, DB2 z) access to IMS databases.

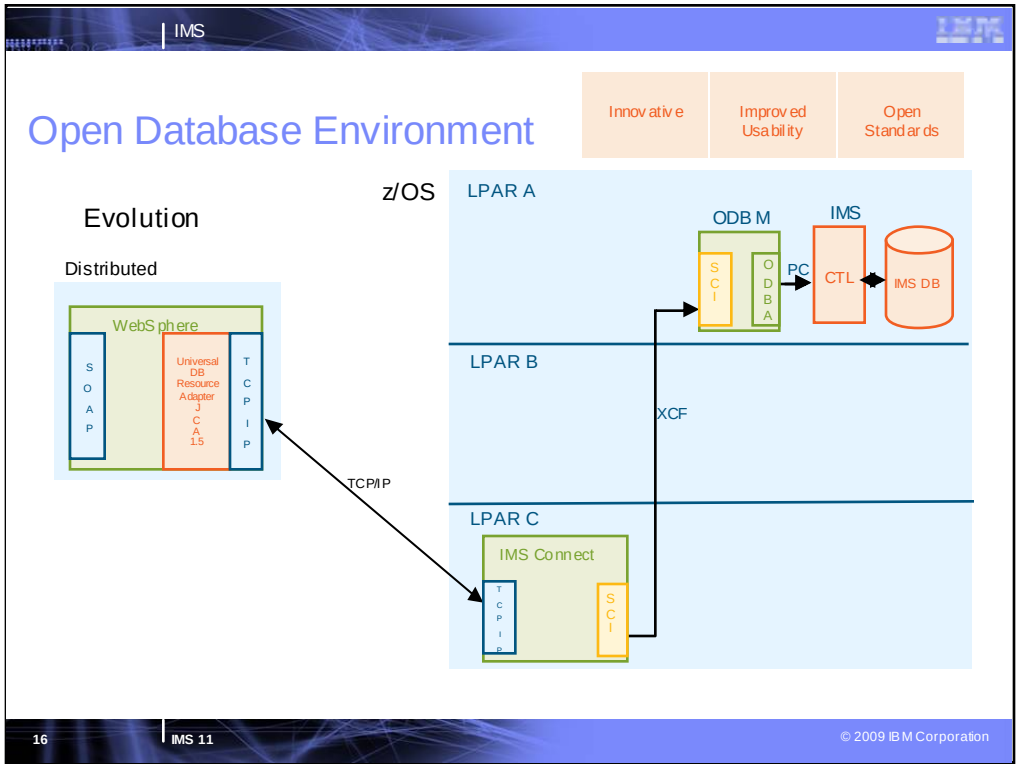


The intent of the following 4 charts is to show the current topology and illustrate the evolution to the new one, pointing out the enhancements at each step. As a point of fact, WAS z/OS cannot take advantage of ODBM's cross-LPAR feature unless WAS itself embraces SCI. Applications can use the out-of-the-box compatibility mode to use AERTDLI and have those calls routed to an ODBM which will still prevent the U113 abend – but WAS and the ODBM address space will still need to be on the same LPAR. It is just an illustrative example showing what can be possible with WAS z/OS as an ODBM client.

The current solution (whether or not we are talking about distributed or local access to IMS DB) leverages ODBA as the API to access IMS database resources. ODBA is capable of making address space to address space calls (PC calls) in the same logical partition. The net effect of this is that the ODBA modules need to be on the same LPAR as the IMS CTL region. These modules (ODBA) are loaded in the address space of the application, which is in turn loaded in the address space of the container. In this case the container is WebSphere AS. The result of this is that the WAS installation has to be on the same LPAR as the IMS DB itself. There is no isolation.

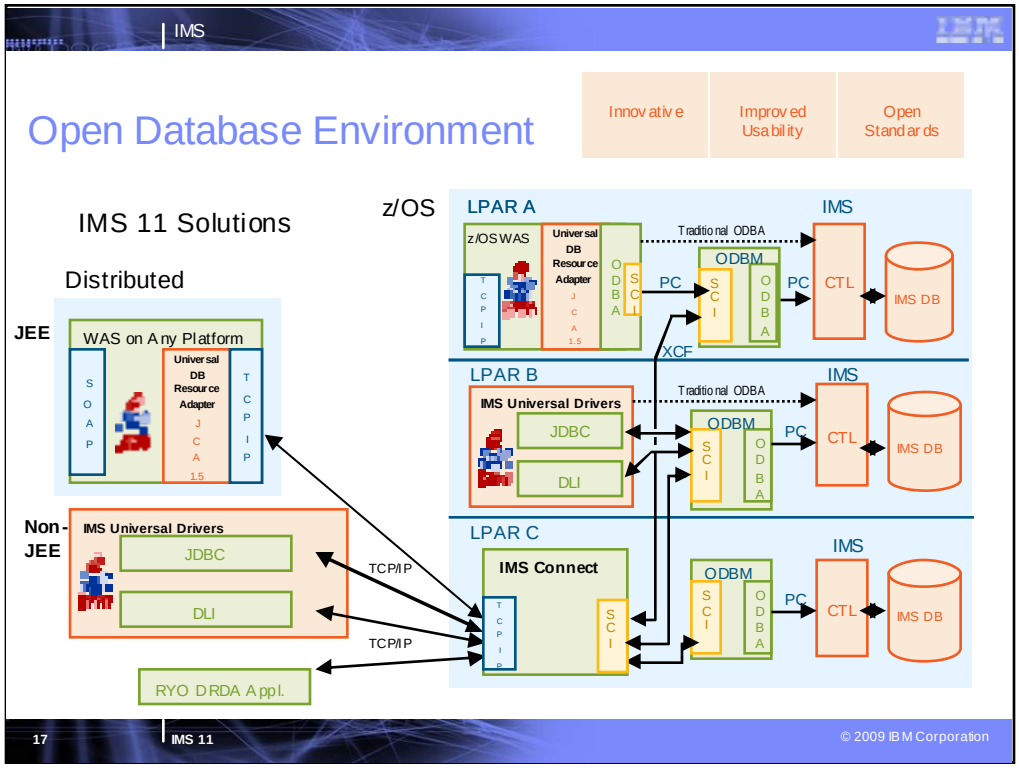


By leveraging SCI, the applications can be on any LPAR in an IMSplex. SCI uses either PC or XCF calls to communicate with other SCI components. XCF allows calls to go across LPARs in an IMSplex. This allows applications (and their containers) to be isolated on their own LPARs.



What we are doing is creating a new CSL address space to house the ODBA modules. This interface will use SCI as its communication mechanism. The ODBA modules are no longer tightly coupled with the applications themselves (and therefore the containers).

By leveraging SCI, the applications can be on any LPAR in an IMSplex. SCI uses either PC or XCF calls to communicate with other SCI components. XCF allows calls to go across LPARs in an IMSplex. This allows applications (and their containers) to be isolated on their own LPARs.



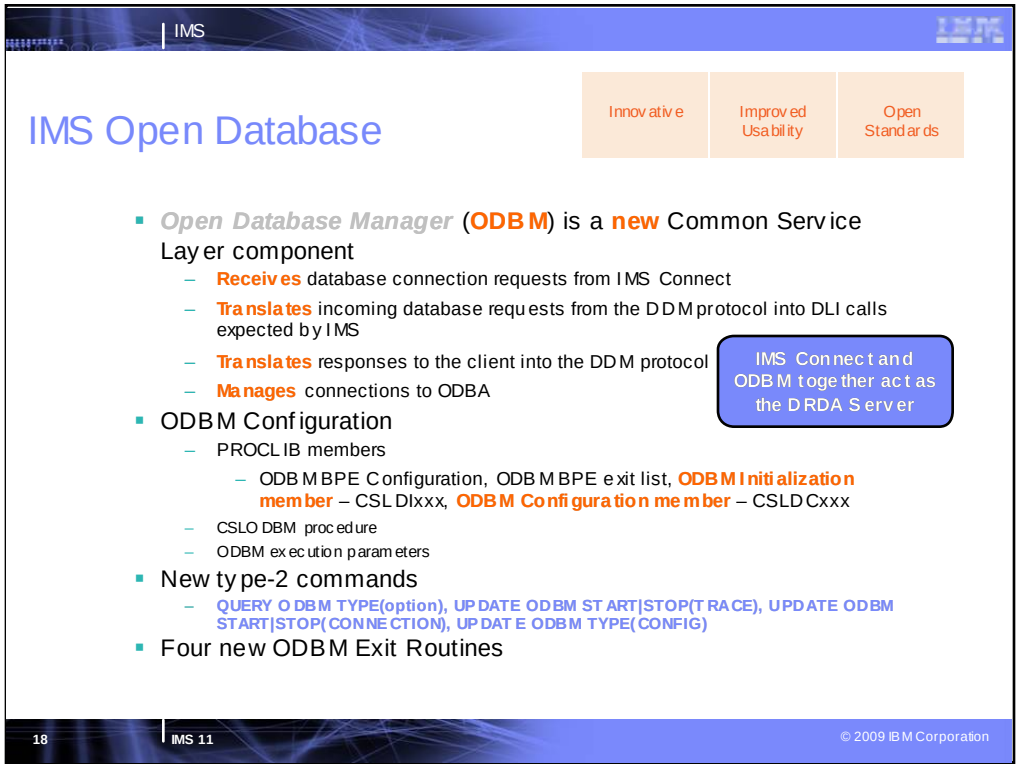
This leads us to our real goal, which is to leverage IMS Connect as the complete gateway solution for IMS TM, OM, and now DB. IMS Connect will be augmented to be an ODBM client. This will allow distributed applications to leverage the TCP/IP protocol to communicate with IMS Connect, which can then access any database in the entire IMSplex.

IMS Connect becomes the IMS Gateway to both IMS TM and IMS DB.

WebSphere and DB2 Stored Procedures no longer have to be on the same LPAR with IMS when they interface with the IMS ODBM (Open Database Manager) address space. The ODBM address space must be on the same LPAR with IMS due to the use of the ODBA (Open Database Access) interface.

Distributed clients would now have the option of going directly to IMS Connect for IMS DB requests.

Existing DB Resource Adapter applications are unaffected by Open Database. In order to **exploit** Open Database from existing DB Resource Adapter applications, a migration to the JCA 1.5 programming model would have to be done.



One ODBM instance must be defined in the IMSplex to use ODBM functions. Each z/OS image can have more than one ODBM. If multiple instances of ODBM are defined in the IMSplex, any ODBM instance can perform work from any z/OS image in the IMSplex.

ODBM routes the database connection requests received from IMS Connect to the IMS systems that are managing the requested database. Before establishing the connection to the IMS system, ODBM translates the incoming database requests from the DDM protocol submitted by the IMS-provided connectors and user-written DRDA applications into the DL/I calls expected by IMS. When ODBM returns the IMS output to the client, ODBM translates the response to the DDM protocol. From the ODBM perspective, application programs that interact directly with ODBM, such as IMS Connect, are ODBM clients. Users can create their own ODBM clients by using the new ODBM CSLDMI API. ODBM client application programs can access databases that are managed by IMS DB on any LPAR in an IMSplex.

Use the CSLDIxxx IMS.PROCLIB member to specify parameters that initialize the ODBM address space. Certain parameters within CSLDIxxx can be overridden with the ODBM execution parameters.

Use the CSLDCxxx IMS.PROCLIB member to configure the ODBM datastore connections to IMS systems.

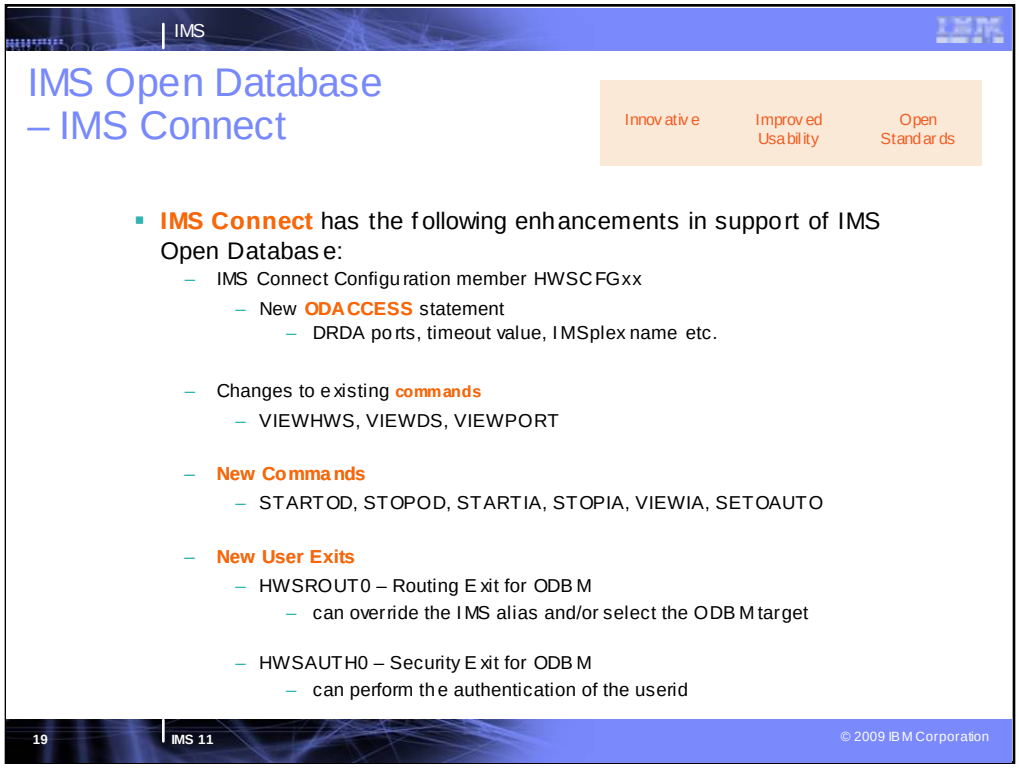
The following IMS exit routines are new:

•CSL ODBM Initialization and Termination user exit routine - Called when an ODBM address space or IMSplex initializes or terminates.

•CSL ODBM Input user exit routine - Called when ODBM receives a call to the ODBM callable interface.

•CSL ODBM Output user exit routine - Called when ODBM returns the results from a call to the ODBM callable interface.

•CSL ODBM Client Connect and Disconnect user exit routine - Called when a client registers to or de-registers from ODBM.



IMS Connect, which previously provided access to only IMS Transaction Manager (IMS TM), through Open Transaction Manager Access (OTMA), now also provides access to IMS Database Manager (IMS DB), through ODBM. IMS Connect can also provide workload distribution by routing database connection requests to ODBM based on an alias name that is submitted by the client application program. To the client application, the alias name represents the IMS system, or data store, to which the application program connects. Depending on the value of the alias name submitted, IMS Connect either routes the incoming connection request to a specific ODBM instance or distributes the incoming connection request to any available instance of ODBM in an IMSplex.

IMS Connect will invoke the routing user exit first to allow the exit to select an ODBM and/or override the IMS ALIAS.

•If the routing exit selects an ODBM, IMS Connect will use that ODBM and will not perform the round robin routing method.

•If the routing exit does not select an ODBM, IMS Connect will select an ODBM and perform the round robin routing method based upon the explicit ALIAS name as well as the blanked ALIAS name.

•If the routing exit overrides the IMS ALIAS, IMS Connect will use that IMS ALIAS.

•IMS Connect will validate the ALIAS and the ODBM upon returning from the exit.

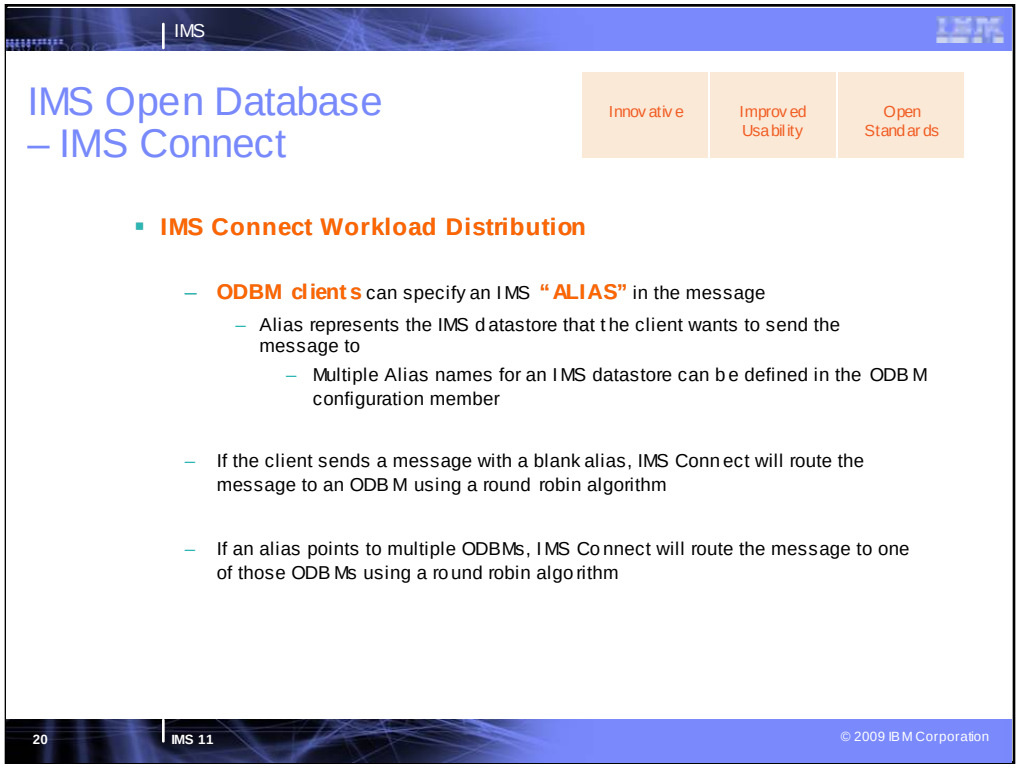
IMS Connect will always call the new security user exit HWSAUTH0 independently of the RACF= parameter in the IMS Connect configuration member.

•IMS Connect calls HWSAUTH0 before invoking any installed security facility such as RACF.

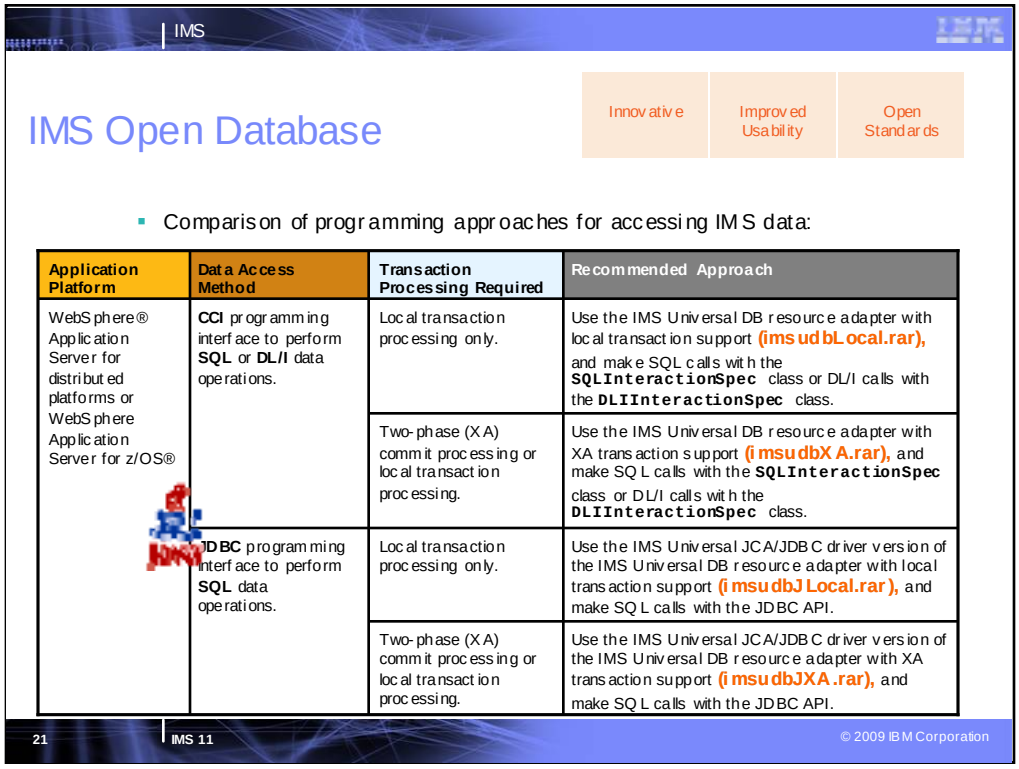
•HWSAUTH0 will perform the authentication of the userid/passticket of the ODBM client.

•HWSAUTH0 can override the input userid with a different userid and to provides a RACF groupid to be authenticated further by IMS Connect. HWSAUTH0 can be refreshable.

•Password is passed in the clear.



IMS Connect can also provide workload distribution by routing database connection requests to ODBM based on an alias name that is submitted by the client application program. To the client application, the alias name represents the IMS system, or data store, to which the application program connects. Depending on the value of the alias name submitted, IMS Connect either routes the incoming connection request to a specific ODBM instance or distributes the incoming connection request to any available instance of ODBM in an IMSplex.

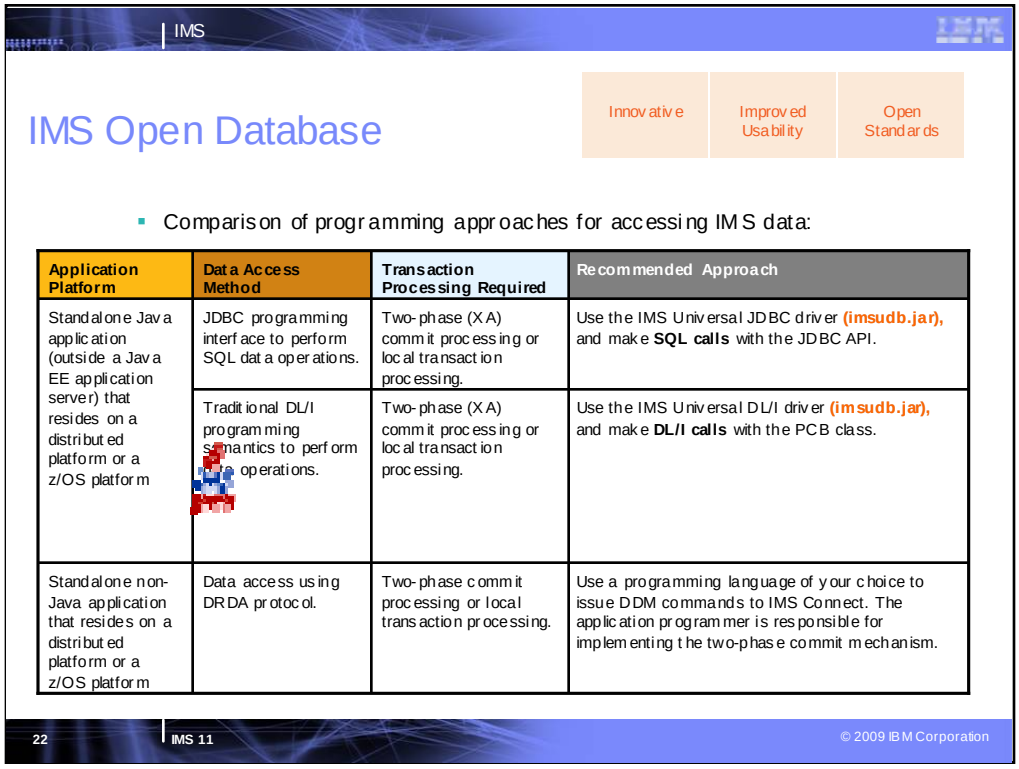


We are providing different API layers that leverage the DRDA protocol. The DLI Java API is an API that mimics the traditional DLI API plus a few more advanced concepts. We are also offering XA capabilities for applications leveraging our framework. (Definition of XA is below.)

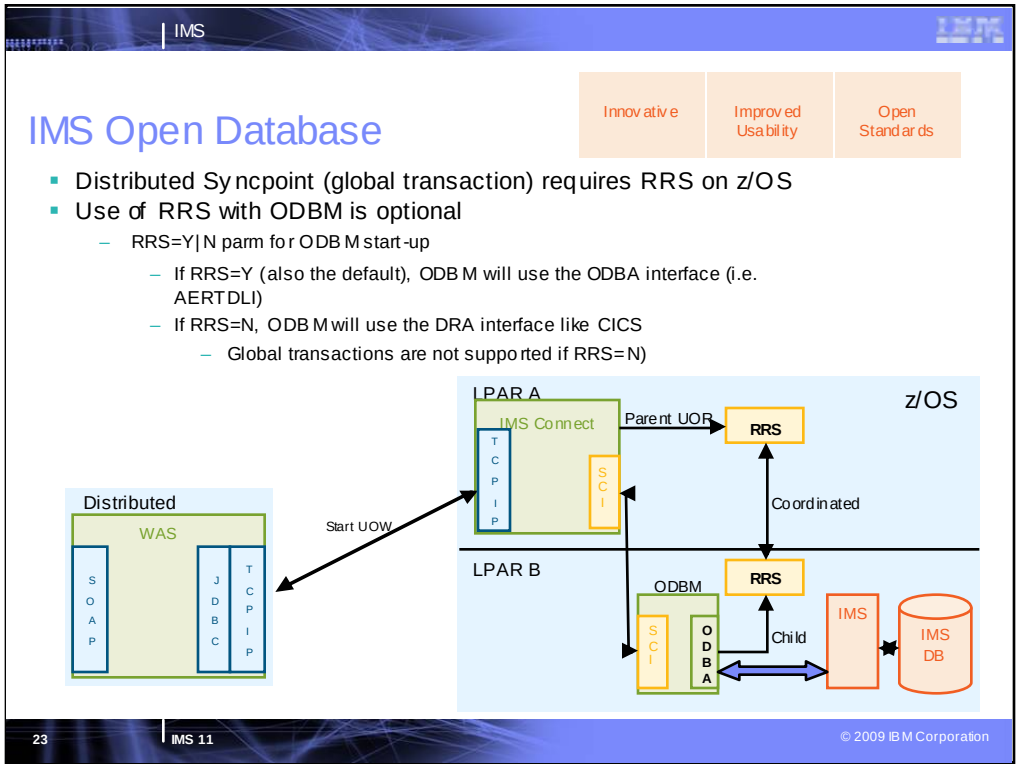
For Open Database, we are offering a type 4 JDBC driver built on top of this framework. Ultimately we will offer a J2EE (Java Enterprise Edition) platform solution which is packaged as a JCA (Java Connection Architecture) 1.5 resource adapter for deployment in WebSphere Application Server (which is a J2EE server). (Editorial note: Sun dropped the '2' so it is really now JEE and not J2EE but most folks still keep the '2'.) **This JCA resource adapter will leverage all of the robust features of the J2EE platform** (connection management, transaction management, security management, XA support, logging, etc.) and will allow for direct distributed access to IMS database resources.

These interfaces will provide a base set of APIs that future clients (such as XQuery) can be built on top of (we have an XQuery implementation today, but now with ODB these APIs will be distributed).

The XA standard is an X/Open specification for distributed transaction processing (DTP). It describes the interface between the global transaction manager and the local resource manager. We are the resource manager (from JDBC driver through IMS connect, ODBM and IMS).



Here is a comparison of programming approaches for accessing IMS data using the IMS 11 Open Database function



RRS=Y must be specified in the configuration for a global transaction.

This slide shows how Open Database achieves cross LPAR transaction management.

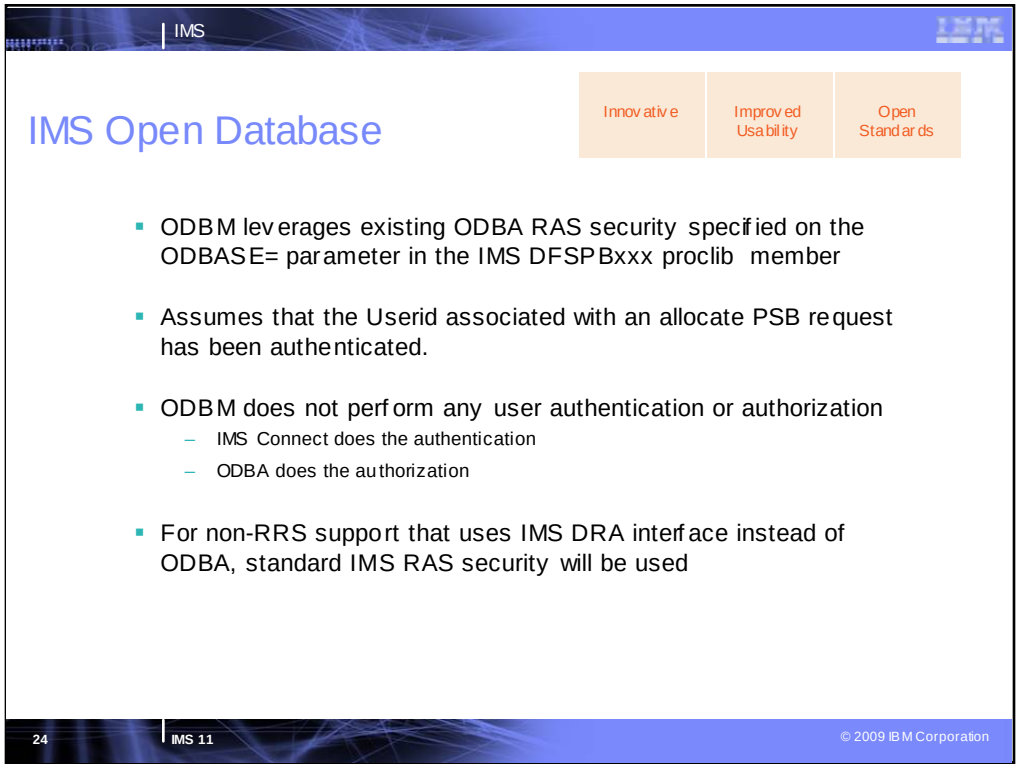
When a client establishes a connection through IIMS Connect to ODBM several things are done to establish a coordinated Unit of Work.

First IMS Connect creates the parent Unit of Recovery.

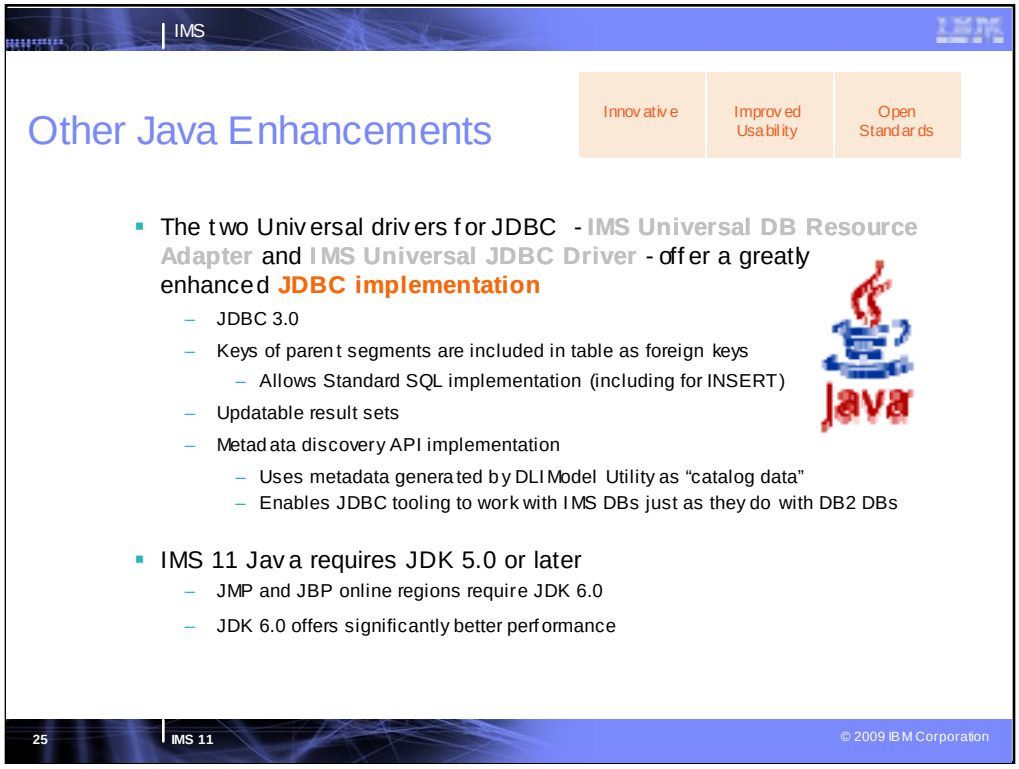
IMS Connect then sends the UR token to ODBM.

ODBM then expresses interest in the UR as a child.

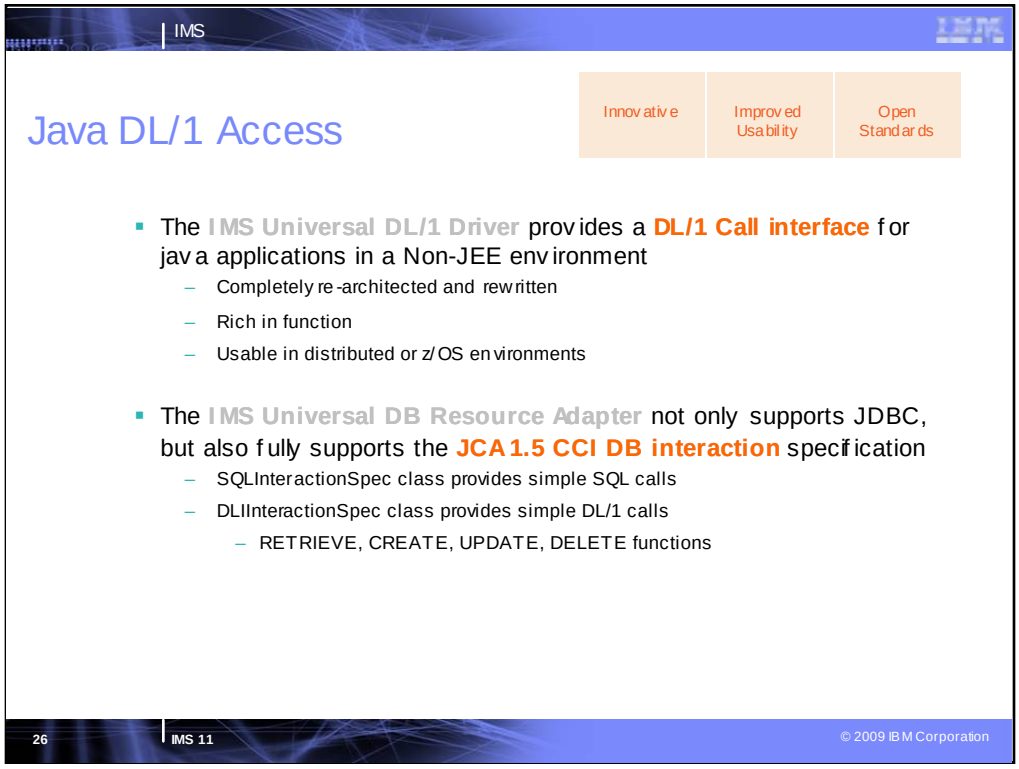
At this point we have a coordinated Unit of Recovery established.



For each request to access IMS data, connection information on the IMS host system, port number, and a valid user ID and password must be supplied in order to establish communication with IMS. A socket connection is first established to connect to the host IMS Connect system. When IMS Connect receives the request, it proceeds to authenticate the user based on the supplied user ID and password. After successful authentication, necessary information on the socket, such as PSB name and IMS alias (database subsystem) is sent to ODBA in order to allocate the PSB to connect to the database. An actual connection to an IMS database is only established when a PSB is allocated. Authorization for a particular PSB is done by the ODBA component during the allocation of a PSB.



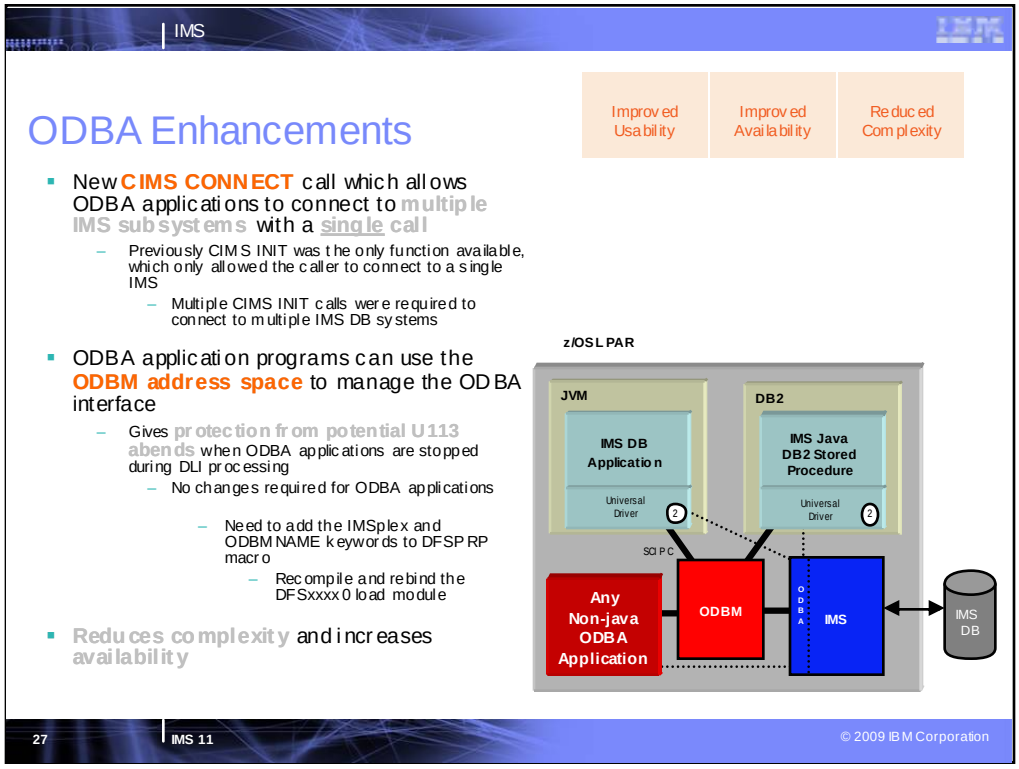
Other Java enhancements in IMS 11 include the two universal drivers for JDBC offering a greatly enhanced JDBC implementation, and JDK 5.0 or later support, with JDK 6.0 offering significantly better performance.



For Java DL/I access:

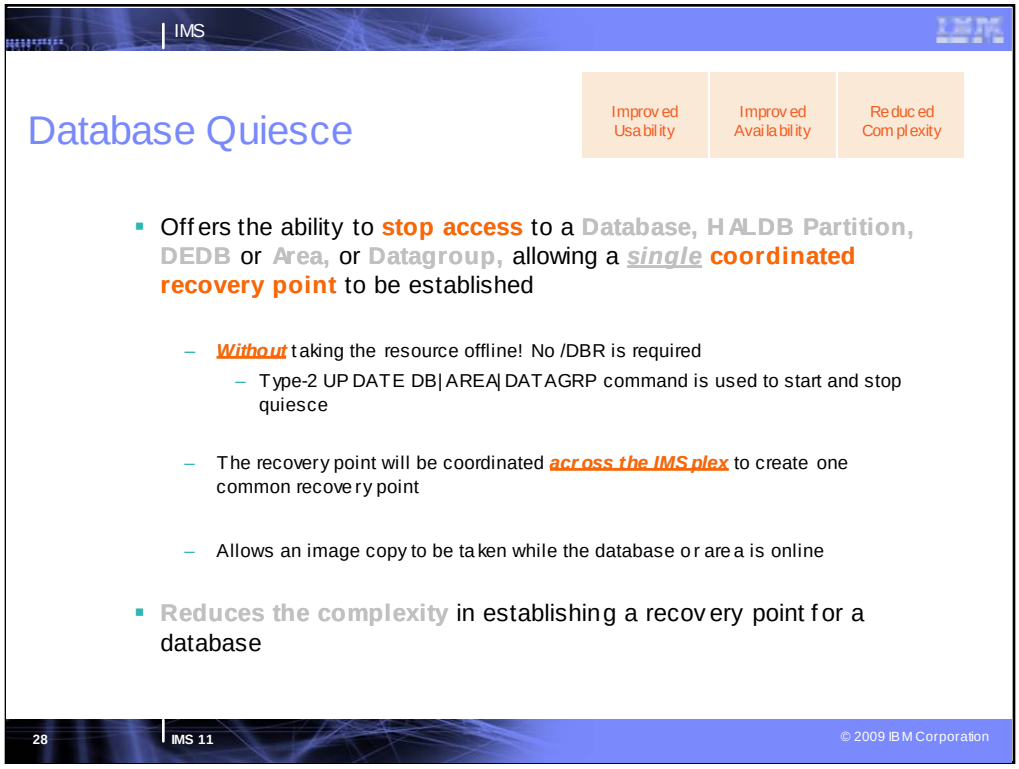
•The IMS universal DL/I driver provides a DL/I call interface for Java applications.

•The IMS universal DB Resource Adapter supports JDBC ad JCA 1.6 CCI DB Interaction.



The ODBA interface has a new command for IMS Version 11: CIMS CONNECT. This command will initialize the ODBA interface and will connect to multiple IMS DB systems. Prior to IMS Version 11, ODBA applications used the CIMS INIT command to initialize the ODBA interface and connect to a single IMS DB system. If the application wanted to connect to multiple IMS DB systems, it had to issue multiple CIMS INIT commands, one for each IMS DB system. The ODBA interface is enhanced in IMS Version 11 to support a new command, CIMS CONNECT, which allows the ODBA application to initialize the ODBA interface and connect to multiple IMS DB systems with a single command.

Existing ODBA applications can also use ODBM to protect IMS from abends that are caused by the unexpected termination of the ODBA applications during DL/I processing. ODBM support for ODBA application programs requires no changes to the application program, but does require minor changes to the DFSPRP macro, which defines the connections to IMS DB, and a recompile and rebinding of the DFSxxxx0 load module. You can modify your existing ODBA application servers to use ODBM by adding the IMSPLEX and ODBMNAME keywords to the DFSPRP macro. After adding these keywords, you must recompile and rebind the DFSxxxx0 load module (xxxx is the DRA start-up table name specified on the APSB call in the AIBRSNM2 field of the AIB). FUNCLV=2 should be specified when using the IMS 11 DFSPRP macro that contains these two new parameters. FUNCLV=2 is set as the default in the shipped IMS 11 DFSPRP macro.

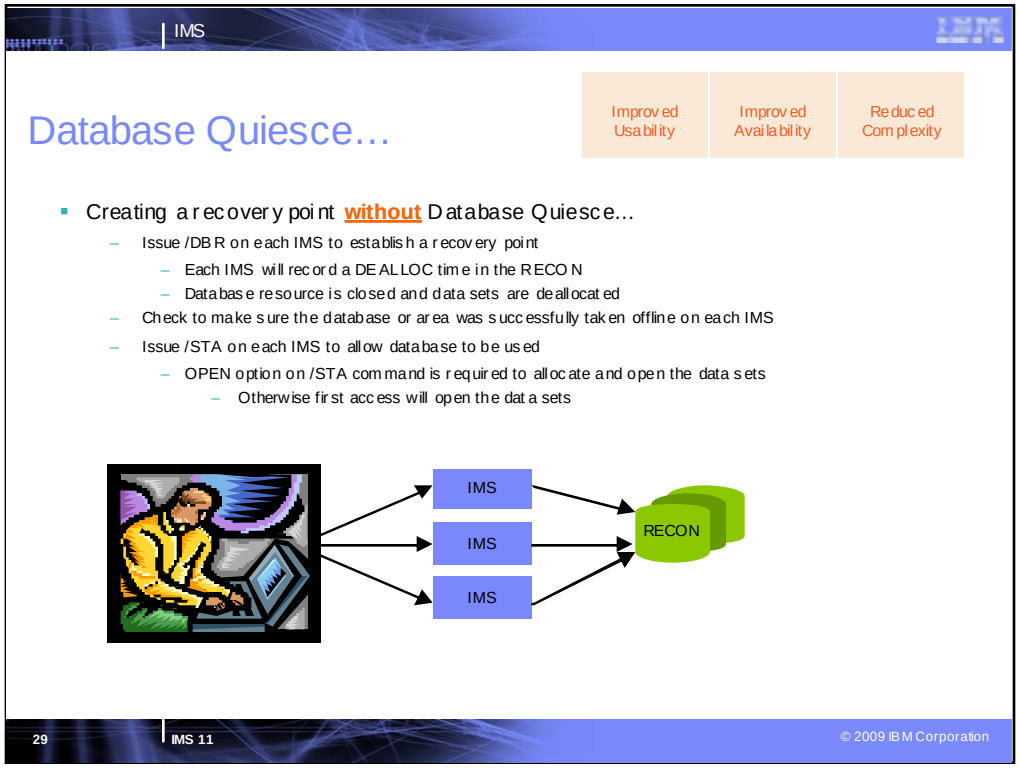


Database quiesce enables you to create a coordinated recovery point across an IMSplex for IMS Fast Path data entry databases (DEDBs), Fast Path areas, full-function databases (including High Availability Large Databases - HALDBs), and database groups, without taking your resources offline or causing applications to encounter an unavailable database.

When the quiesce function is invoked, the point of consistency is reached when all updates that are in progress are committed. After all updates are committed, and stored on DASD, the database data sets reflect the current database information because no updates are pending. At this time, the point of consistency has been reached locally on a single IMS. This process must occur on every IMS in the IMSplex that is actively using the database. When all activity has been quiesced, a new recovery point is recorded.

You can create a point of consistency on a database without making the database unavailable to applications by using a new form of the existing type-2 UPDATE command (for example, UPDATE DB START (QUIESCE)). This command quiesces all work in progress so that a new recovery point can be created for a database while it is actively in use in the IMSplex.

This enhancement should reduce the complexity of establishing a recovery point for a database. The reduced complexity should allow more frequent recovery points to be created for a database resource. We see value to your business with this enhancement by requiring less complexity in establishing a recovery point. Creating frequent recovery points could reduce the impact of an outage on a database since there would be less data to be processed during recovery.



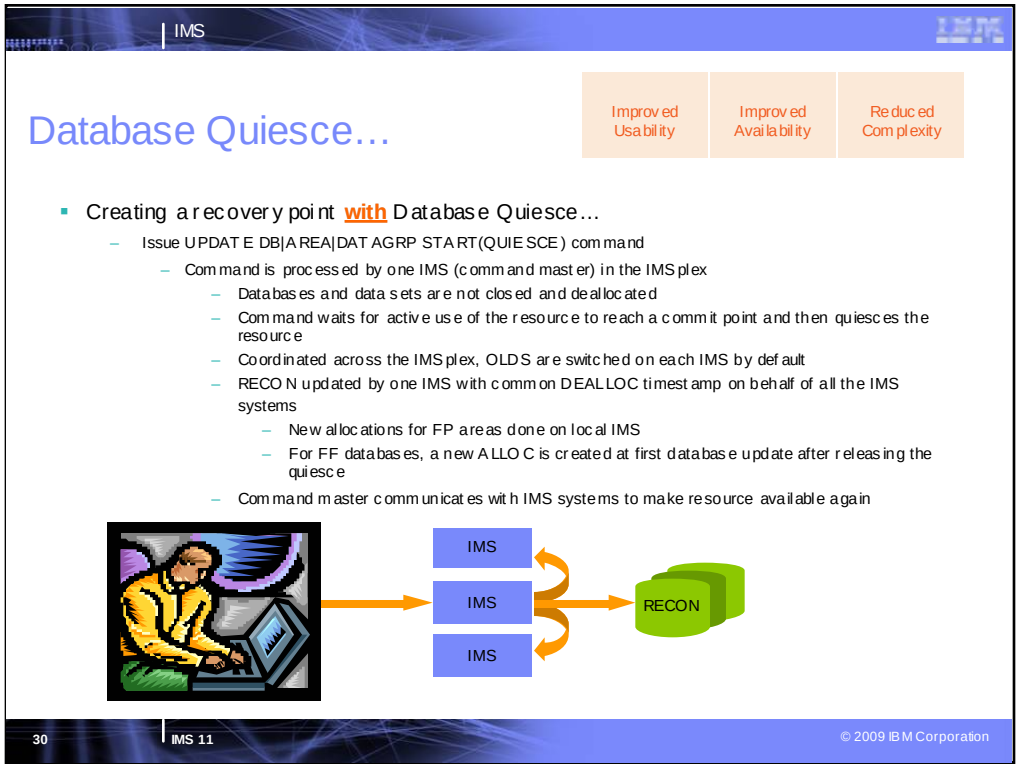
Before going into the detail of quiesce, let's look at an illustration of the current process required to create a recovery point for a database resource. In this picture there is a simple IMSplex with three different IMS systems in the DBRC sharing group. The database resource is in use on the three different IMS systems.

First a user needs to take the resource offline. To do this the /DBR command is issued for the resource and routed to each IMS or issued on each IMS. On each IMS, the command checks for activity on the resource, if there is none then the command can be processed. Each IMS will go to the RECON and record a DEALLOC time. The database resource is closed and the data sets are deallocated.

Next the user needs to check to make sure that the command was successful on each IMS system.

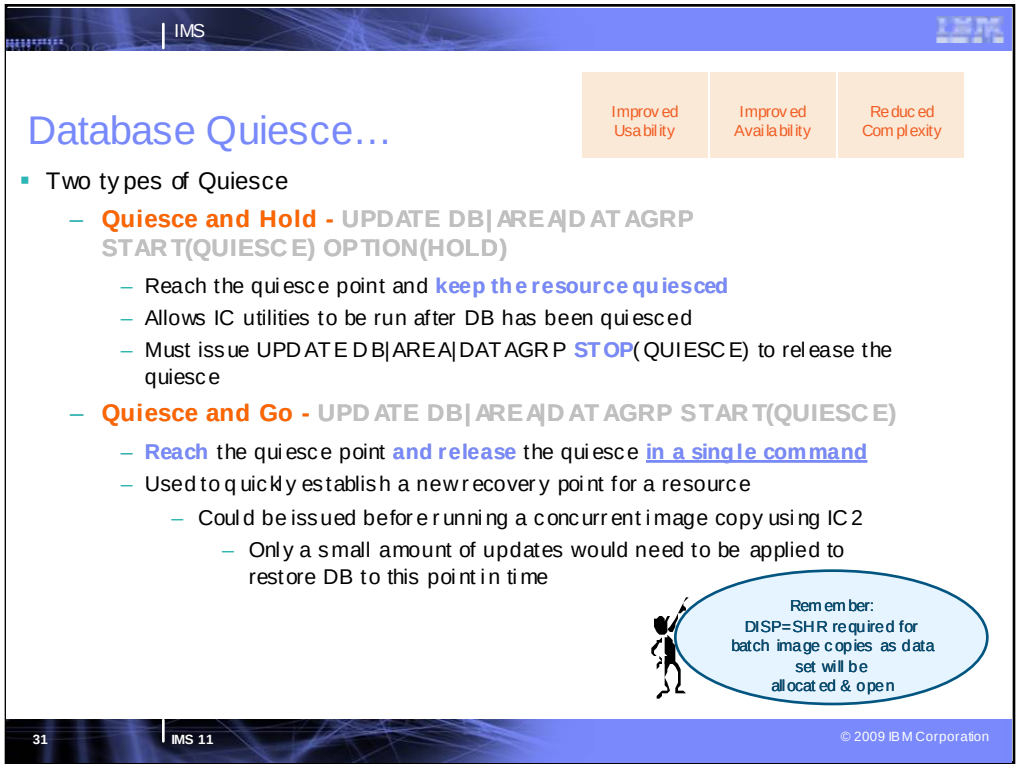
Now a recovery point has been created for the database.

Finally the user needs to make the resource available online. To do this the /STA command is issued for the resource and routed to each IMS or issued on each IMS. On each IMS, the command makes the database available. The database data sets are not allocated, and they are closed after the /DBR command. The OPEN option would be required to allocate and open the data sets on the /STA command, otherwise the first access will open the data sets.



This is an illustration of the process to create a recovery point for a database resource with Database Quiesce. In this picture there is a simple IMSplex with three different IMS systems in the DBRC sharing group. The database resource is in use on the three different IMS systems.

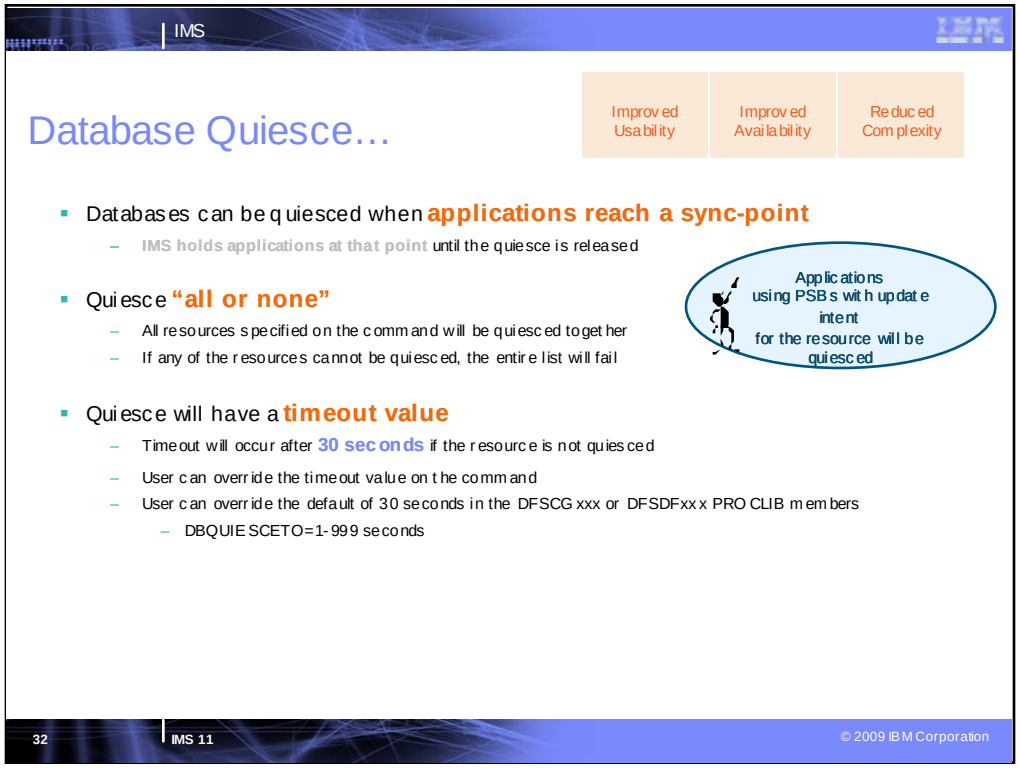
To create the recovery point issue the quiesce command. The command will be processed by a single IMS, the command master. That command master will communicate with the other IMS systems to get the resource quiesced. If there is current activity on the resource the command will wait for a point of consistency to be reached. After that point the access on the database is stopped. The IMS systems will communicate back that the resource is quiesced. The command master will update the RECON on behalf of all the IMS systems, creating a common DEALLOC timestamp. The OLDS are switched on each IMS (as the default). After the quiesce point is reached, the quiesce on the resource is released. When the quiesce is released then a new ALLOC will be recorded for DEDB areas. No ALLOC is recorded for Full Function databases, instead, the first update access to the database will create a new ALLOC in the RECON. The command master communicates with all IMS systems to make the resource available again.



The quiesce function will be implemented with two different forms of quiesce. A quiesce and hold and a quiesce and go type of command.

The quiesce and hold will allow the database resource to be quiesced and stay in the quiesce state. This would allow utilities like IC2 to be run against the quiesced resource. To release the quiesce, a second command must be issued. This would be similar to issuing a /DBR or /DBD today then running utilities and then issuing a /STA command to allow normal access to the resource to resume. This process will create a recovery point for the database which doesn't require any logs to be applied to recover the database. Note that in order for image copies to be allowed to execute when the database or area is in a QUIESCE HELD state, the DISP for the dataset to be copied must be DISP=SHR (instead of DISP=OLD) for batch image copies since the dataset is open and allocated to the online IMS subsystems.

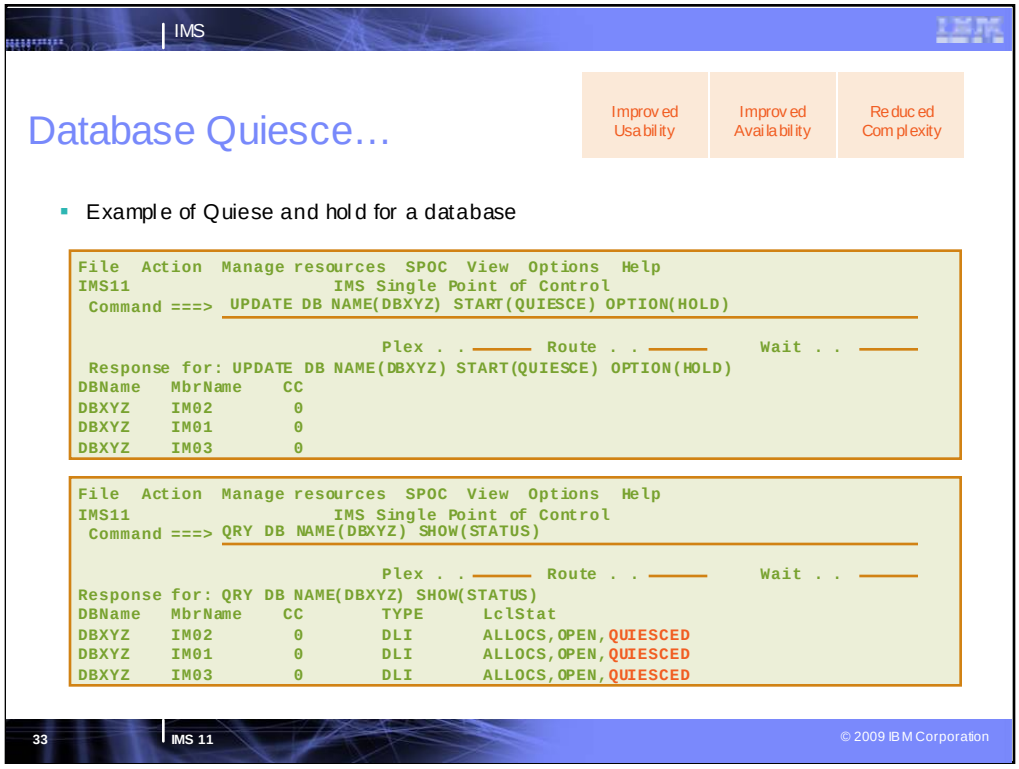
The quiesce and go is a way of quickly creating a new recovery point for a database resource and then making the resource available again immediately in a single command execution. The quiesce and go may be used for example when a recovery point is created outside of the normal recovery period, when there is more activity on the IMS system. After the quiesce and go has completed, a concurrent image copy of the database could be started using IC2. This process will create a recovery point for the database which only requires a small amount of updates to be applied, from the point after the quiesce until the image copy, to restore the database to this point in time.



When the quiesce function is invoked, the point of consistency is reached when all updates that are in progress are committed. After all updates are committed, and stored on DASD, the database data sets reflect the current database information because no updates are pending. At this time, the point of consistency has been reached locally on a single IMS. This process must occur on every IMS in the IMSplex that is actively using the database. When all activity has been quiesced, a new recovery point is recorded. During the time that the database is quiesced, only read-only applications can access the database. All other applications are held at the time when they attempt to access the database with a DL/I call, until the quiesce is released. Batch jobs with update access or above will fail authorization while the database is quiesced.

When the quiesce is released, the applications are able to access the database again. The quiesce leaves the database data sets in the same state that they were in at the start of the quiesce process.

The DFSCGxxx PROCLIB member and the COMMON_SERVICE_LAYER section of the DFSDFxxx PROCLIB member are updated with the new DBQUIESCETO keyword, where you can specify the maximum amount of time (from one to 999 seconds) to allow a database quiesce process to complete before timing out. The default value is 30 seconds.



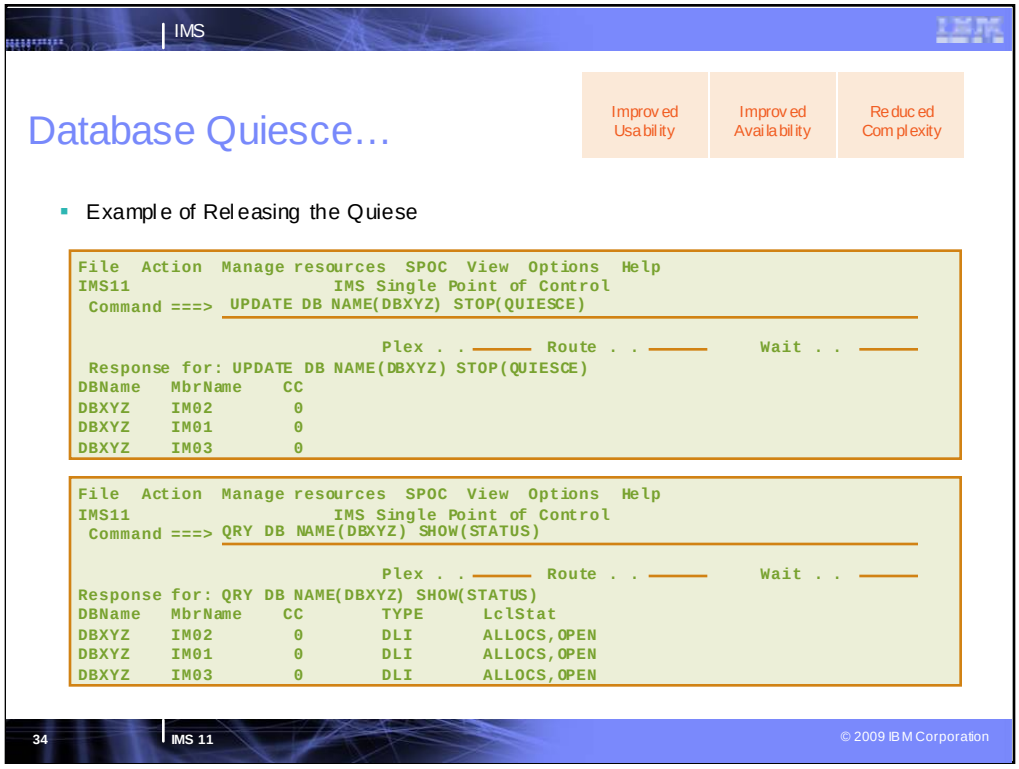
There are three IMS systems in the IMSplex, IM01, IM02, and IM03.

The TSO SPOC was used to issue the quiesce and hold command.

The output of the command shows it was successful on three IMS systems.

After the quiesce and hold was performed a QUERY command was issued to display the status of the database resource.

The command shows that the resource is quiesced, but notice that the database is still allocated and open. This is because the quiesce doesn't deallocate the database data sets or close the database data sets.

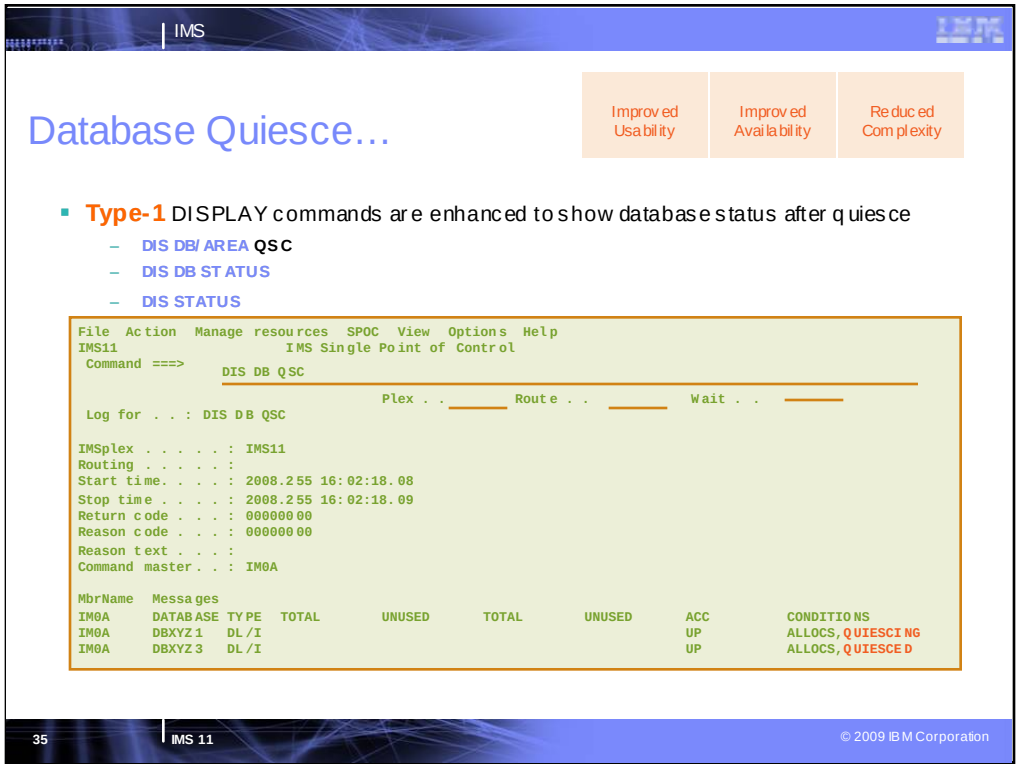


Now release the quiesce on the resource by issuing another UPDATE command.

The output of the command shows it was successful on three IMS systems.

After the quiesce and hold was released, a QUERY command was issued to display the status of the database resource.

The command shows that the resource is allocated and open.



The following Type-1 commands are enhanced to show quiesce status for a database resource:

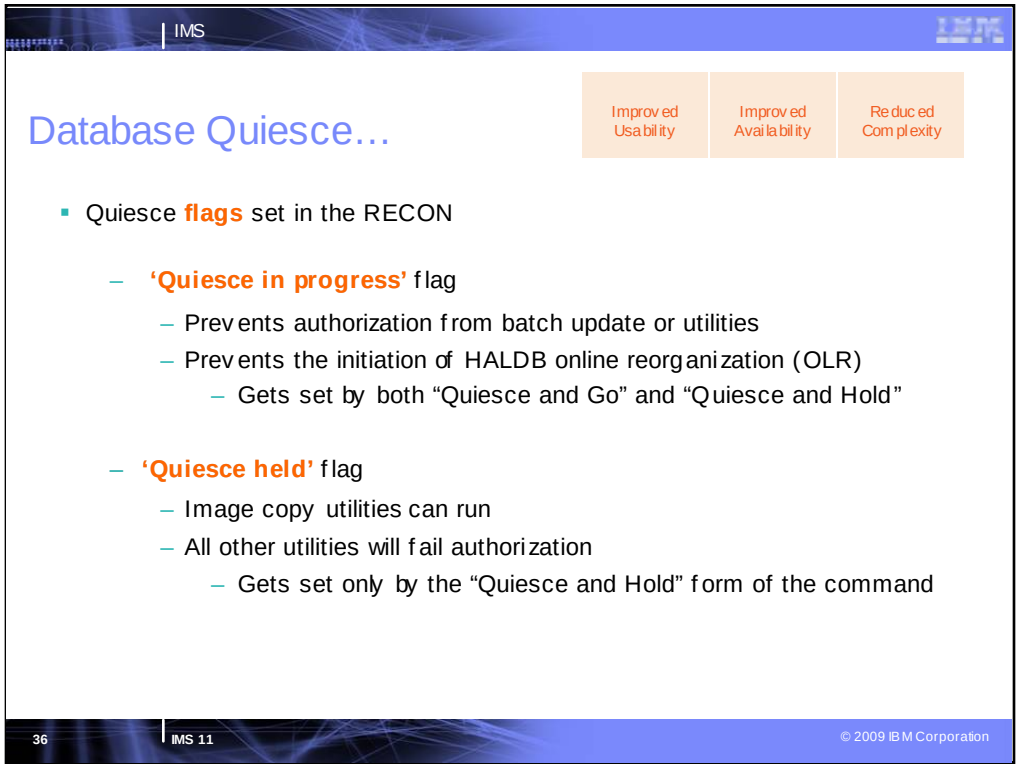
•DIS AREA with QSC attribute – displays all FP areas that have status of 'quiescing' (indicating quiesce in progress) or 'quiesced' (indicating the resource is quiesced)

•DIS DB with QSC attribute – displays all databases that have status of 'quiescing' (indicating quiesce in progress) or 'quiesced' (indicating the resource is quiesced)

•DIS DB STATUS

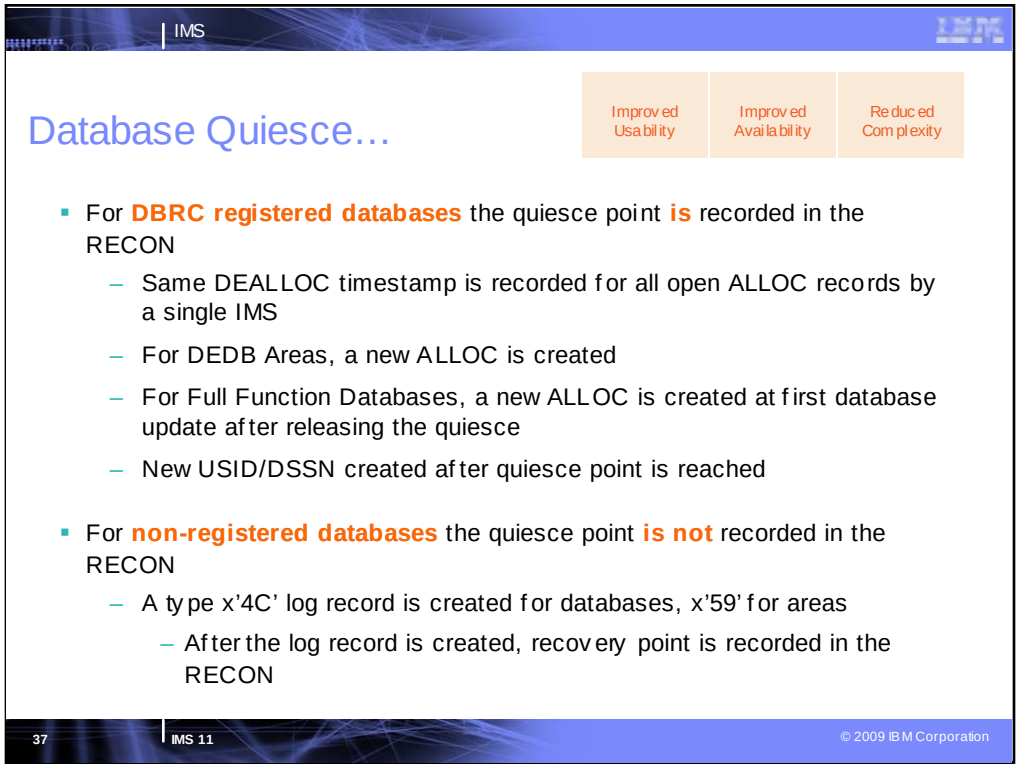
•DIS STATUS

The command specified in the example is to show database resources that have status of 'quiescing' or 'quiesced'.



When quiesce is started, with either form of the command, the 'Quiesce in progress' flag in the RECON will be set on. While the flag is on, any new batch update or utility authorization of the database or area will be prevented. All online IMS subsystems including update access and batch jobs requesting read access will be allowed authorization. All utilities will fail authorization. Batch jobs with read access only and online IMS subsystems with read access only will be allowed access to the database or area. IMS systems which didn't participate in the quiesce will be allowed authorization but will be prevented from accessing the database or area while the flag is on in the RECON data sets.

Also the quiesce function will not be allowed during integrated HALDB online reorganization (OLR).



The quiesce point will be a common DEALLOC time in the RECON for all open ALLOCs.

When the quiesce is achieved any open ALLOC record in the RECON data sets is updated. The open ALLOC record will be closed with a DEALLOC time which corresponds to the point the quiesce was achieved across the IMSplex. The closed ALLOC record will show that the deallocation was the result of a quiesce command.

When the quiesce is reached, the command master will update the open ALLOCs in the RECON on behalf of all the IMS systems.

When the quiesce is released (either quiesce and go or release of quiesce after quiesce and hold) then a new ALLOC will be recorded for DEDB Areas. No ALLOC is recorded for Full function databases, instead the first access to the database will create a new ALLOC in the RECON.

If the database is not registered with DBRC, then the assumption is that it is not being shared with other systems. Since it is not being shared, there is no coordination. Also since there is no RECON to update the quiesce point can only be recorded on the log. When the quiesce has caused a point of consistency to be reached, a log record is created. The type X'4C' log record is created for databases. The type X'59' log record is created for areas. The log records are used by each participant in the quiesce to note that the database has been quiesced. After the log record is created, the recovery point is recorded in the RECON data sets.

When a quiesce and hold had been issued, the quiesce status is maintained across restarts of IMS. A call is made to DBRC during restart processing to recover the status.

The quiesce point will be a common DEALLOC time in the RECON for all open ALLOCs.

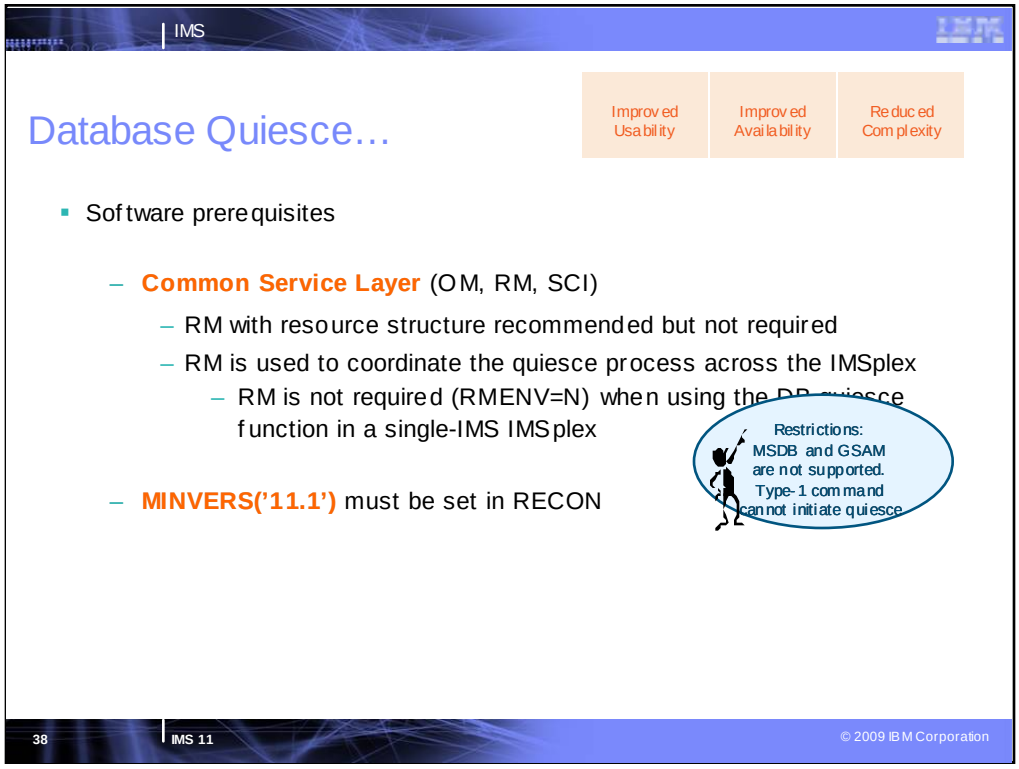
When the quiesce is achieved any open ALLOC record in the RECON data sets is updated. The open ALLOC record will be closed with a DEALLOC time which corresponds to the point the quiesce was achieved across the IMSplex. The closed ALLOC record will show that the deallocation was the result of a quiesce command.

When the quiesce is reached, the command master will update the open ALLOCs in the RECON on behalf of all the IMS systems.

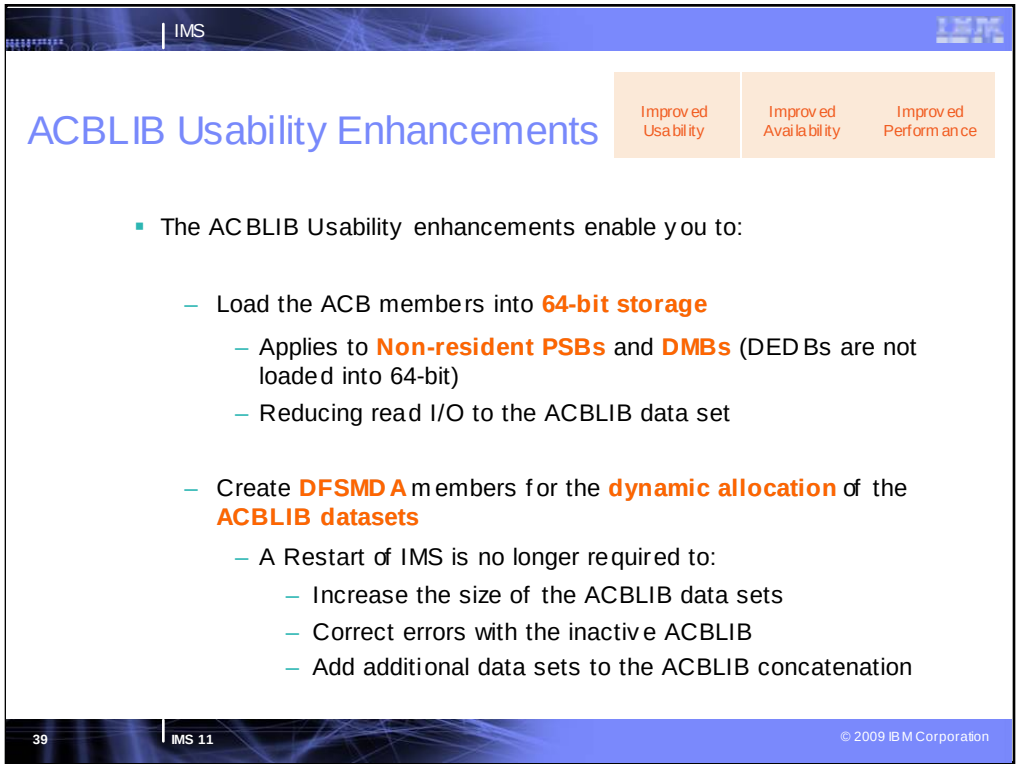
When the quiesce is released (either quiesce and go or release of quiesce after quiesce and hold) then a new ALLOC will be recorded for DEDB Areas. No ALLOC is recorded for Full function databases, instead the first access to the database will create a new ALLOC in the RECON.

If the database is not registered with DBRC, then the assumption is that it is not being shared with other systems. Since it is not being shared, there is no coordination. Also since there is no RECON to update the quiesce point can only be recorded on the log. When the quiesce has caused a point of consistency to be reached, a log record is created. The type X'4C' log record is created for databases. The type X'59' log record is created for areas. The log records are used by each participant in the quiesce to note that the database has been quiesced. After the log record is created, the recovery point is recorded in the RECON data sets.

When a quiesce and hold had been issued, the quiesce status is maintained across restarts of IMS. A call is made to DBRC during restart processing to recover the status.



The CSL Resource Manager (RM) is required when using the DB quiesce function in a multi-IMS IMSplex. An RM resource structure is recommended, but not required. RM is not required (RMENV=N) when using the DB quiesce function in a single-IMS IMSplex.



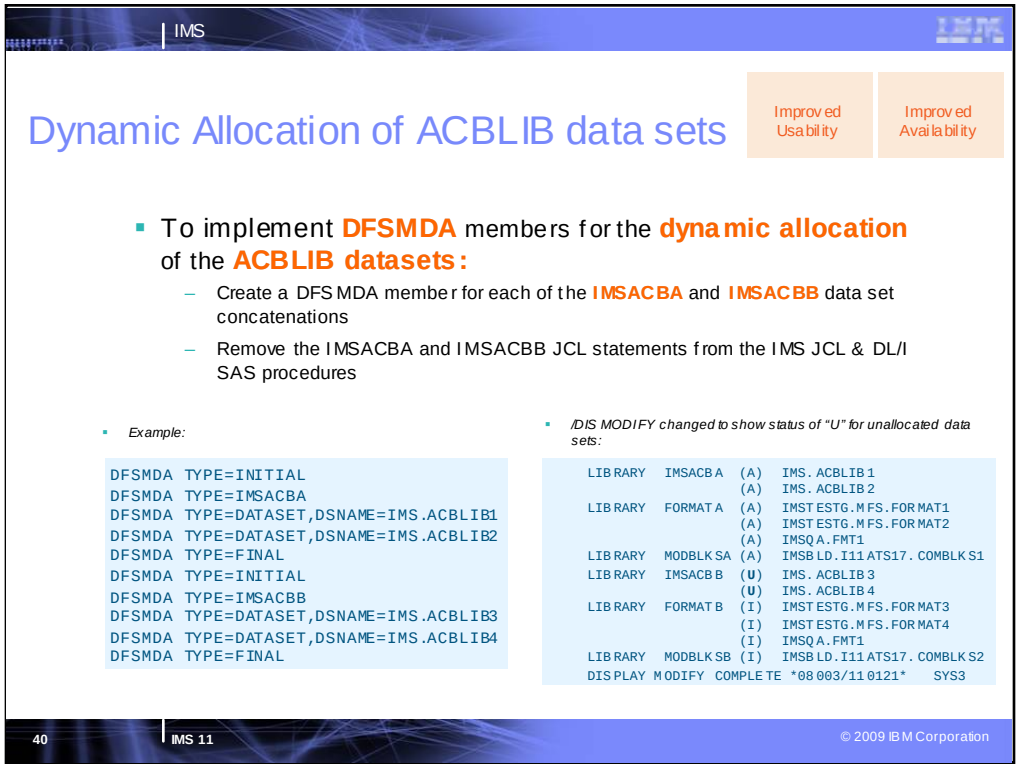
These enhancements add options to:

•Load the ACB members into 64-bit storage. At application scheduling time, the ACB members are retrieved from 64-bit storage rather than from the ACBLIB data set (unless they are not currently in the 64-bit cache). This option reduces the amount of read I/O to the ACBLIB data set. By changing the resident option for a PSB, the user could have a smaller resident pool and also realise some CSA reduction. There will be no performance degradation at scheduling since the PSB is retrieved from the 64-bit pool.

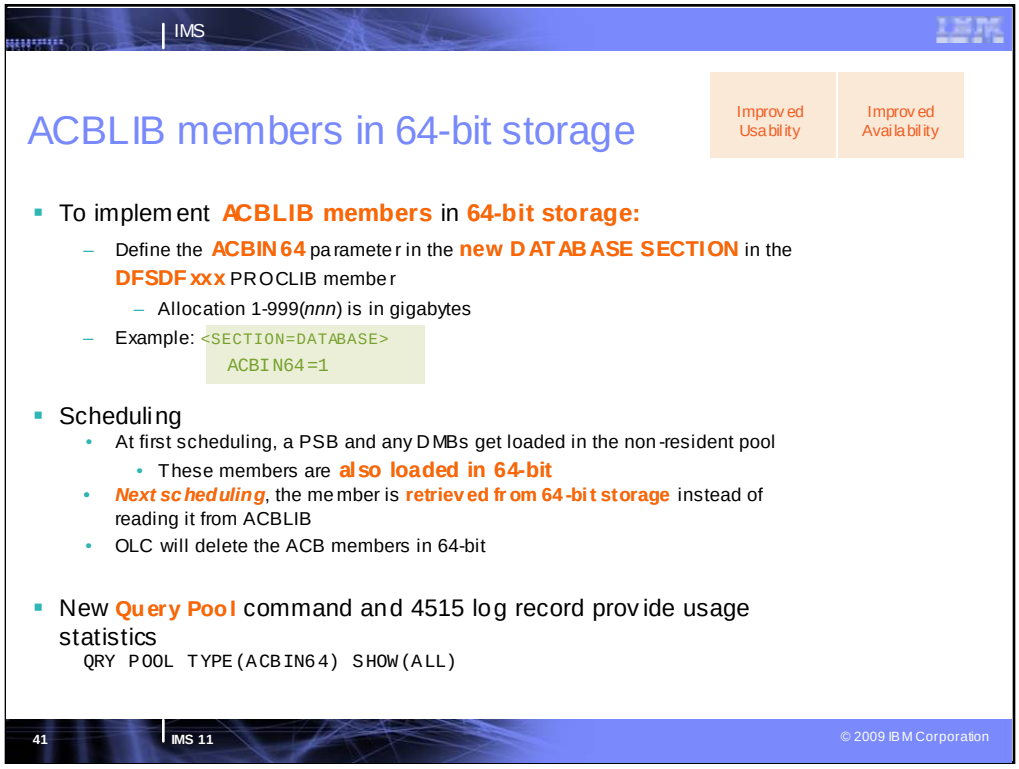
•Use the Dynamic Allocation macro (DFSMDA) macro to create DFSMDA members for the dynamic allocation of the ACBLIB data sets. You can specify how the ACBLIB data sets are allocated with either JCL or DFSMDA members. DFSMDA members provide the following benefits:

- •You can increase the size of the ACBLIB data sets without stopping and restarting IMS.
- •You can correct errors with the inactive ACBLIB without stopping and restarting IMS.
- •You can add additional data sets to the ACBLIB concatenation without stopping and restarting IMS.

The dynamic allocation of the ACBLIB data sets and the 64-bit storage pool for ACB members is supported in the following online IMS system configurations: DB/TM, DBCTL, DCCTL, XRF, and FDBR. Dynamic allocation of ACBLIB datasets and the 64-bit storage pool for ACB members are not supported in IMS batch.

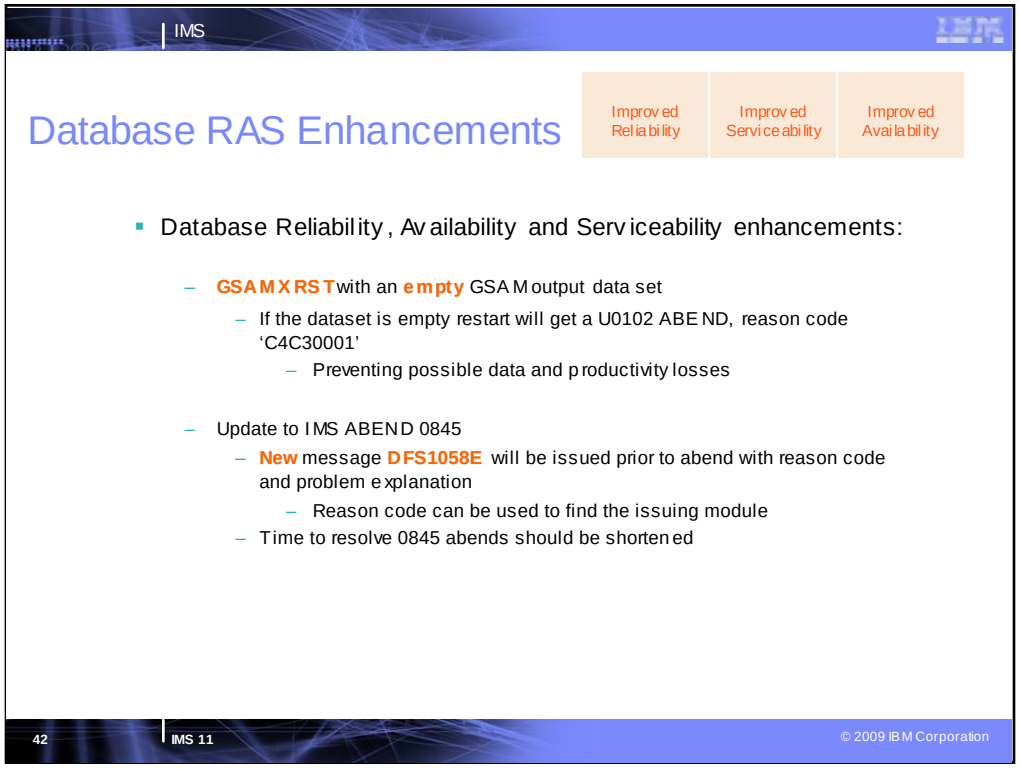


The /DIS MODIFY command is changed to show the inactive ACBLIB libraries when MDA members are used. IMS will dynamically allocate the Active ACBLIB during initialization and will deallocate it during Online Change. The inactive ACBLIB datasets are not allocated until an OLC process. The datasets will show a status of "U". This indicates that these datasets are unallocated.



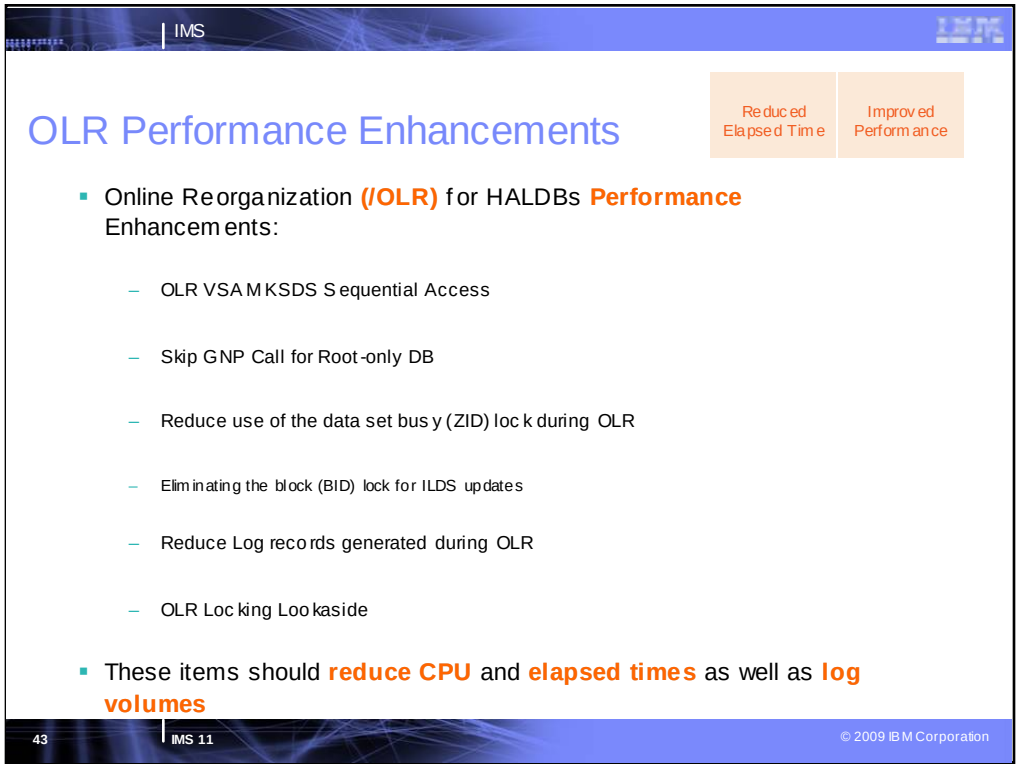
The allocation quantity is in gigabytes.

The QUERY POOL command is a new command for IMS 11. It will be used to display statistics for IMS storage pools. This command will be used to display the storage utilization statistics for the 64-bit ACB pool. It will also display statistics for other IMS pools such as the Fastpath buffer pool. The 4515 log record contains the same statistics that are displayed on the QRY POOL command.



This improvement prevents you from accidentally pointing to an empty GSAM dataset when you need to do an extended restart (the XRST command). During GSAM XRST processing, a check is made to determine if the GSAM output dataset to be repositioned is empty. If the dataset is empty and the abending job had previously inserted records to the dataset, the restart job will issue an U102 abend with the new reason code 'C4C30001'.

Prior to issuing a U0845 abend, the DFS1058E message will be issued, listing the reason code and a short explanation of the problem. The reason code will also be returned in Reg 1 of the 'Registers at Abend'. The reason codes have been changed and can now be used to identify the abending module.



OLR VSAM KSDS Sequential Access: OLR is enhanced to take advantage of the VSAM sequential access option when issuing KSDS GET requests to retrieve sequentially from the input data set(s). This results in reduction in CPU and elapsed time.

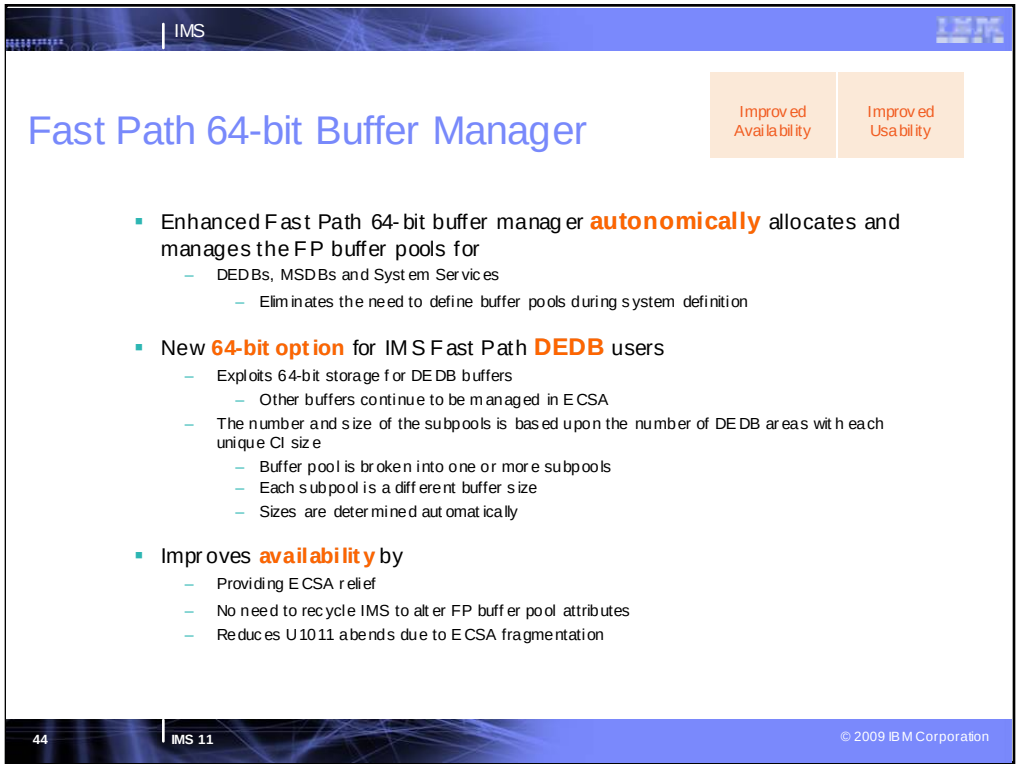
Skip GNP Call for Root-only DB: For a root-only database, there is no reason to issue the GNP call since there is no dependent segment to be read. Skipping the GNP call saves CPU and elapsed time.

Reduce use of the data set busy (ZID) lock during OLR: For a PHIDAM partition undergoing reorganization by OLR the updates for the primary index (KSDS) can be saved for insertion at the end of the unit of reorganization (UOR). By saving all the KSDS updates until the end of the UOR, the usage of the ZID lock for the primary index can be UOR changed. The change will be to obtain the ZID once before starting to insert all the saved primary index updates, and then released once. When the UOR covers many roots, many ZID lock requests will be eliminated which will result in a reduction of CPU usage and elapsed time.

Eliminating the block (BID) lock for ILDS updates: For a HALDB partition with logical relationships the updates for the indirect list dataset (ILDS) by OLR do not need to obtain the block (BID) lock. The BID lock is used for serialization of the updates to a block across IMS data sharers. However the design of HALDB doesn't allow the ILDS to be updated by more than one IMS at a time. Therefore the BID lock does not need to be obtained. However the ZID lock needs to be obtained in order to notify other IMS data sharers in the case of CI/CA splits. To reduce the use of the ZID lock during OLR, the updates for the ILDS can be saved for insertion at the end of the UOR. The ZID lock will be obtained once before starting to insert all the saved ILDS updates, and then released once. When the UOR covers many to the ILDS, many ZID lock request will be eliminated which will result in a reduction of CPU usage and elapsed time.

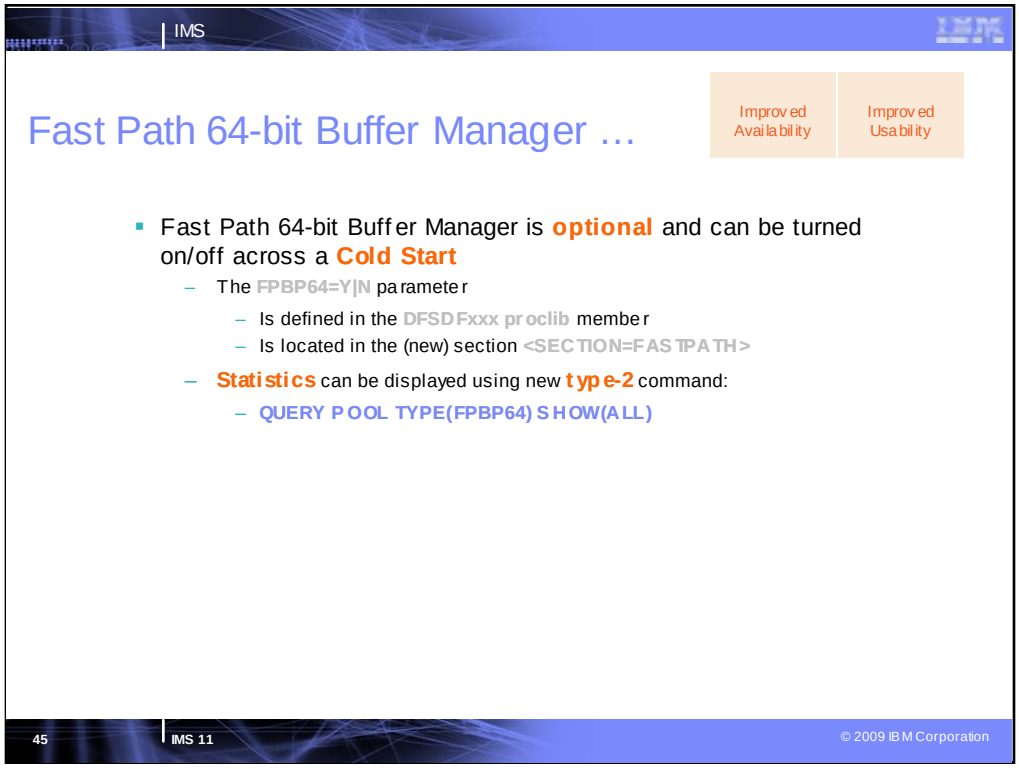
Log reduction: For a HALDB partition undergoing OLR, the database update log records (type '50'x) will be consolidated when possible into full block updates. By combining all the updates for a full block into a single type '50'x log record, many of the small type '50'x log records will be eliminated. This will result in a reduction in the log volume generated by OLR.

Lock reduction: Whenever a lock request is made by the OLR owning region, a lookaside operation is performed. The lookaside function will determine if this is a new, or already owned, lock. If this is a new lock, normal processing to call IRLM to get the lock is performed. If this is a lock that is already owned, the lock manager returns to the caller. The savings in path length by optimizing the lock manager out of the call flow results in reduction in CPU and elapsed time.



The Fast Path 64-bit buffer manager autonomically controls the number and size of Fast Path buffer pools, including buffer pools for data entry databases (DEDBs), main storage databases (MSDBs), and system services, which eliminates the need for users to manually set buffer pool specifications during system definition. The Fast Path 64-bit buffer manager also places the DEDB buffer pools above the bar in 64-bit storage, which reduces the usage of ECSA storage. Fast Path buffer pools for MSDB databases, buffers for inserting sequential dependent (SDEP) segments, buffers for system services, and buffer headers continue to be managed in ECSA storage. The existing mechanism for managing Fast Path buffer pools, which places all Fast Path buffer pools in 32-bit ECSA storage below the bar, requires the number and size of buffer pools to be set during system definition by using the DBBF, DBFX, and BSIZ execution parameters. After IMS is started, the number and size of the Fast Path buffer pools cannot be changed without stopping and restarting IMS.

If a database is added to the online IMS system and none of the active buffer subpools can accommodate the CI size of the database, the Fast Path 64-bit buffer manager allocates a new buffer subpool of the appropriate size.



The Fast Path 64-bit buffer manager is enabled by the parameter, FPBP64, in the <SECTION_FASTPATH> of the DFSDFxxx PROCLIB member.

When the Fast Path 64-bit buffer manager is enabled:

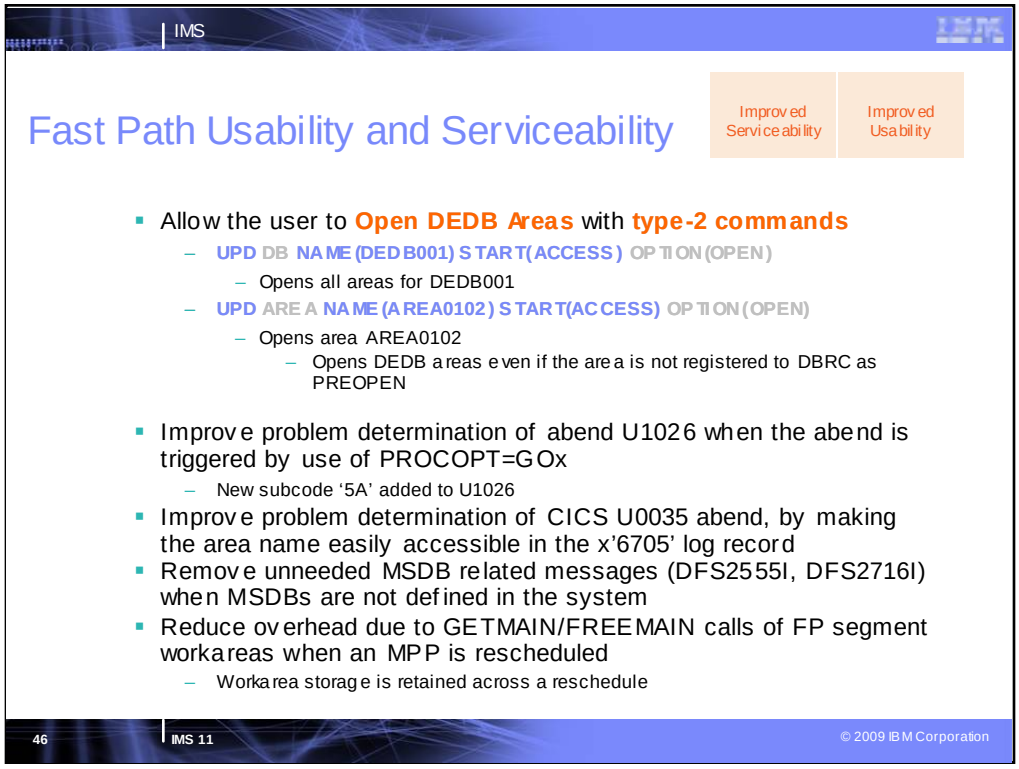
•Database administrators can improve the performance of their DEDB databases by changing the CI size of databases, without having to adjust the buffer sizes to match. If no buffer subpools are active that can accommodate a new or changed CI size, the Fast Path 64-bit buffer manager automatically allocates a buffer subpool with the correct CI size.

•Users can display statistics for Fast Path buffers by issuing the new IMS type-2 command QUERY POOL TYPE(FPBP64) SHOW(ALL).

•Multiple application programs can access the overflow buffers (specified with the OBA parameter) in parallel.

The values specified on the overflow buffer allocation (OBA) parameter and on the normal buffer allocation (NBA) parameters affect the Fast Path 64-bit buffer manager. Combined together, they determine the maximum number of buffers that the Fast Path 64-bit buffer manager can allocate to a dependent region.

Users can display statistics for Fast Path buffers by issuing the new IMS type-2 command QUERY POOL TYPE(FPBP64) SHOW(ALL).



The IMS Fast Path usability and serviceability enhancements provide IMS customers with additional ways to open DEDB areas by using UPDATE DB and UPDATE AREA type-2 commands. Currently /STA area does not open DEDB area logically. With this enhancement, UPD AREA and UPD DB commands can open area logically and save time at first access of applications. By using the UPDATE commands, Fast Path customers can open DEDB areas even if the area is not registered to DBRC as PREOPEN. The DEDB area open process still requires DBRC authorization. The following commands are affected by this enhancement:

•The OPTION(OPEN) keyword on the UPDATE DB command is enhanced to work with Fast Path DEDB areas. Previously, this keyword worked only with full-function databases.

•A new keyword, OPTION(OPEN), is added to the syntax and parameter descriptions of the UPDATE AREA command.

Several enhancements are related to troubleshooting and enable both IMS customers and IMS Service to more efficiently resolve problems that are related to Fast Path.

Problem determination time is reduced by adding a new subcode '5A' to ABEND1026 when it's issued due to PROCOPT=GOx.

Add area name to CICS abendu0035 for problem determination - In analyzing a CICS U0035 which was due to a DEDB area being closed out from under a CICS thread, it was more difficult than needed to find the Area Name. In IMS 11 the area name is added to the x'6705' log record.

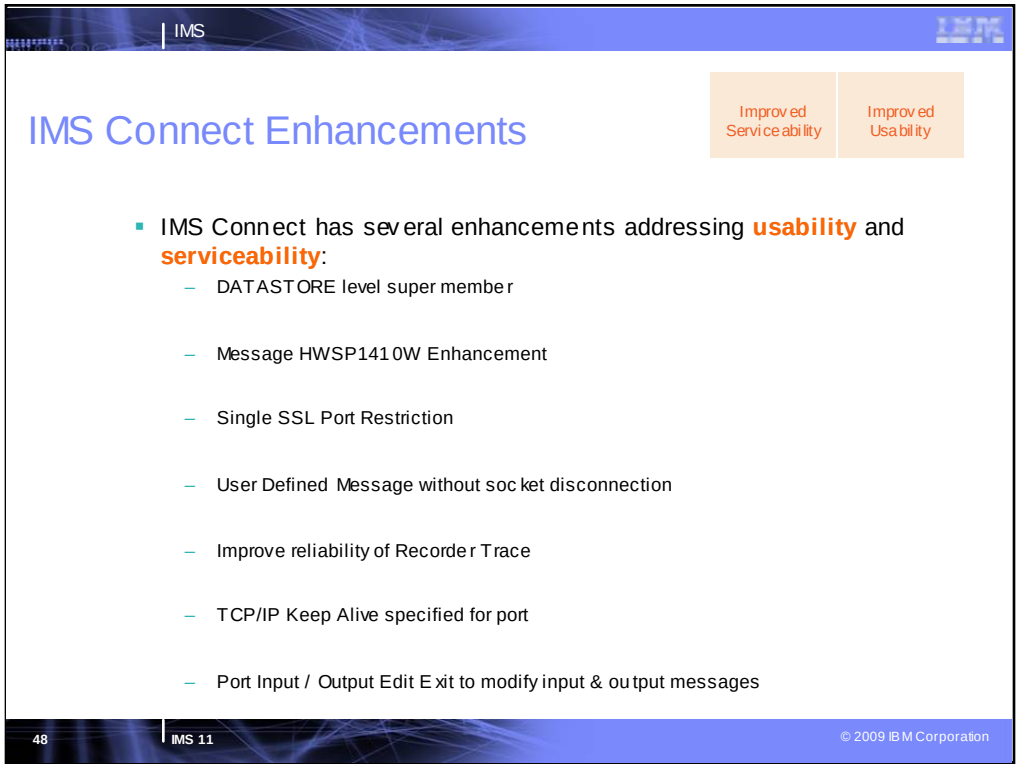
Remove MSDB related messages when no MSDBs - Remove these MSDB related messages. When RG suffix does not have MSDB suffix or there is no MSDB defined in MSDBINIT, these messages are issued.
DFS2555I NO SUFFIX FOR MSDB MEMBER DEFINED PC

DFS2591I NO MSDB HEADERS FOUND, IMAGE COPY LOAD IGNORED PC

DFS2716I NO MSDBS FOUND - NO MSDB CHECKPOINT TAKEN PC

Reduce GETMAIN/FREEMAIN calls during MPP reschedule process - During each schedule of a dependent region, storage is obtained for a DBFSEG workarea. For IFPs and BMPs, this is acceptable. But for MPPs, which can continue to reschedule while the region is up, this causes performance hits due to the FREEMAIN/GETMAIN overhead. This enhancement changes the processing so the DBFSEG storage is retained across a reschedule. The storage is released if the dependent region terminates, or the existing DBFSEG storage is not large enough for the new DBFSEG blocks.

The IMS System Enhancements support by IMS TM and IMS DB users



IMS Connect Enhancements – several enhancements for user requirements, usability and serviceability.

DATASTORE level super member – The current IMS Connect support for the OTMA super member feature allows customers to specify only one super member name for each instance of IMS Connect. All datastore connections from IMS Connect must use the same super member name. Customers would like IMS Connect to support multiple super member names, so that, within a single instance of IMS Connect, different groups of client application programs can use different super members. With this enhancement, the super member name can be specified at the connection level so that customers can group client applications by the super member name that they share.

Message HWSP1410W Enhancement - IMS Connect reports errors when freeing storage with message HWSP1410W. This message reports the error return code, type of storage and the module that encountered the error. To further aid problem determination, the address of the storage is added to this message.

Single SSL Port Restriction - IMS Connect does not support multiple SSL ports in an IMS Connect instance. Through this enhancement, IMS Connect will restrict users from specifying more than one SSL port in the IMS Connect configuration member to prevent user errors that will result in potential failure in the Language Environment.

User Defined Message without socket disconnection – This allows customers to modify the user message exit to send a user defined message back to the client application as a reply to the input message with the request to keep the socket connected.

Recorder Trace:

•Improved IMS Connect product reliability by reducing system outages caused by an ABENDS0C4 in the Recorder Trace process.

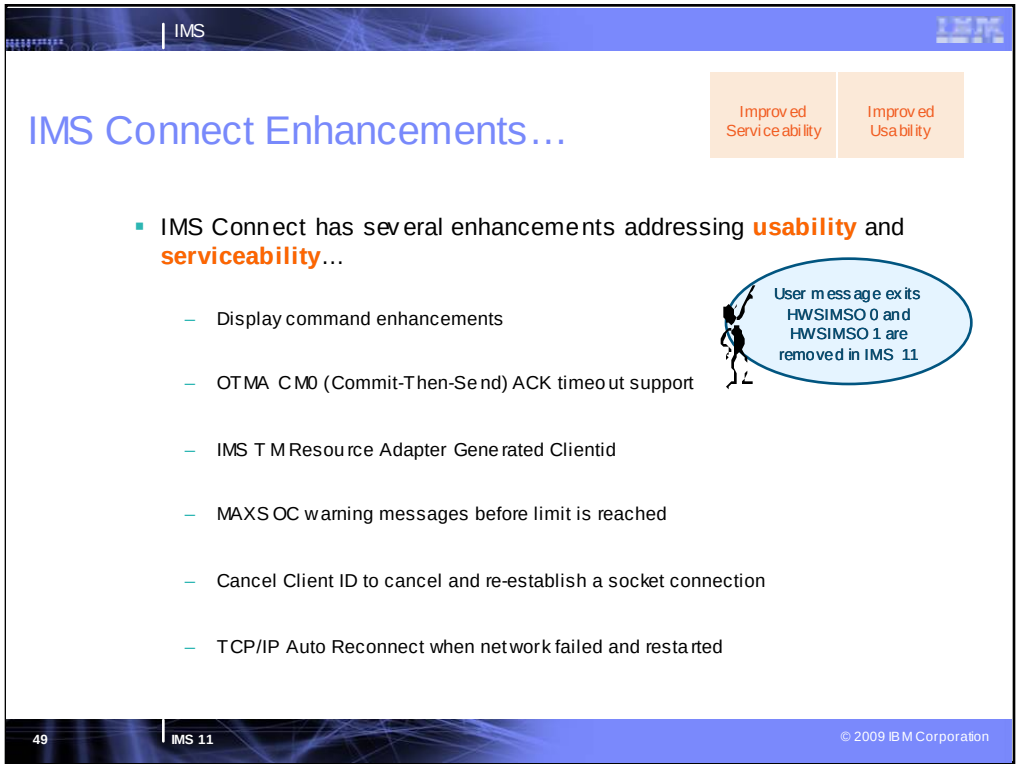
•Eliminate Recorder Trace dataset full conditions and the resulting loss of diagnostic data.

•Improve the reliability of diagnostic information to shorten or streamline the problem determination process.

•Allow the user more flexibility and control over the amount of diagnostic data recorded.

TCP/IP Keep Alive - Allow IMS Connect customers to set the TCP/IP KeepAlive value at the Port level to override the default stack value.

Port Input/Output Edit Exit - Provide IMS Connect customers with the ability to modify input messages received from TCP/IP before IMS Connect processes them and output messages after IMS Connect has completed processing before they are passed to TCP/IP.



IMS Connect Enhancements – several enhancements for user requirements, usability and serviceability…

Display Command Enhancements – Provide a "summary" version of the VIEWHWS & QUERY MEMBER commands that will only report the summary information for PORTs without the potentially large amounts of individual socket displays. The addition of the Datastore last accessed to the client information displayed will allow customers to determine to which IMS that request was sent.

OTMA CM0 ACK Timeout Support - Enable IMS Connect customers to specify the name of the Queue to be used by OTMA for rerouting any CM0 messages that timeout, while waiting for the ACK rather than using the OTMA default queue name.

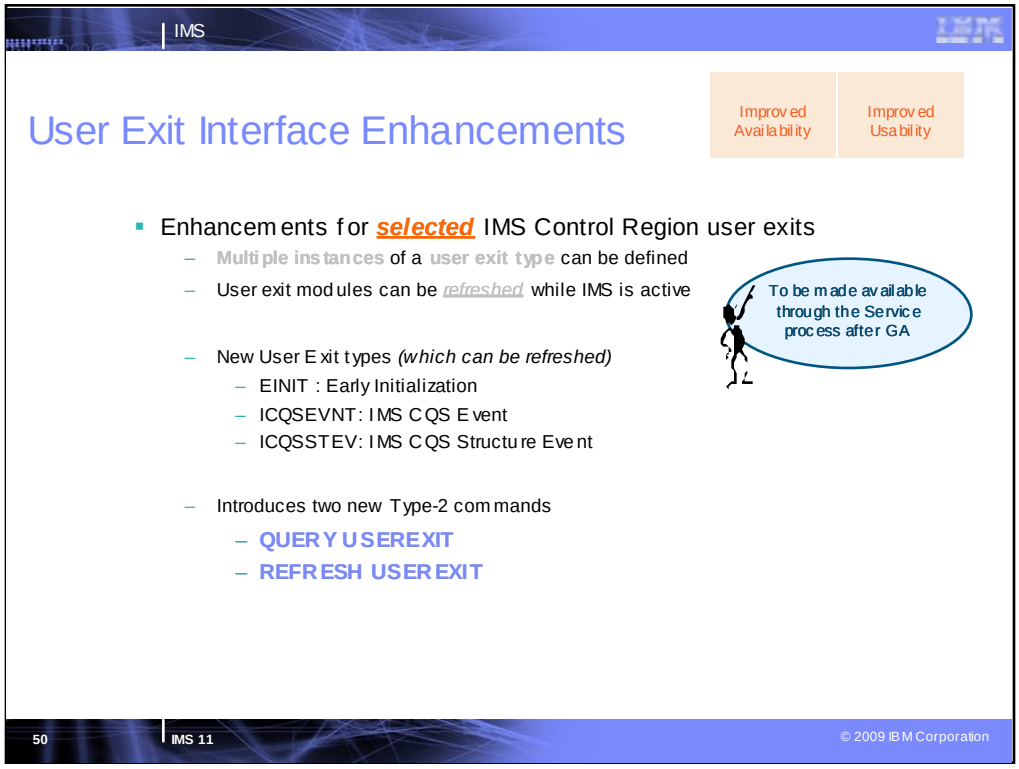
IMS TM Resource Adapter Generated Clientid - Eliminate the requirement that customers use different IMS Connect PORTs for instances of distributed WAS when using ITMRA (IC4J) Shareable Persistent sockets. Generate the HWSxxxxx clientid if none is provided. To be exploited by corresponding IMS TM Resource Adapter change.

MAXSOC Warning - Provide IMS Connect customers with warning messages before the max socket limit is reached. Provide the ability to specify the warning level to start issuing the warning messages.

Cancel Client ID - Provide IMS Connect customers with the ability to cancel an existing socket connection and establish a new socket connection with the same Client Id.

TCP/IP Auto Reconnect - Eliminate operator intervention to re-establish the connection to the TCP/IP network where the network has failed and restarted during an instance of the IMS Connect.

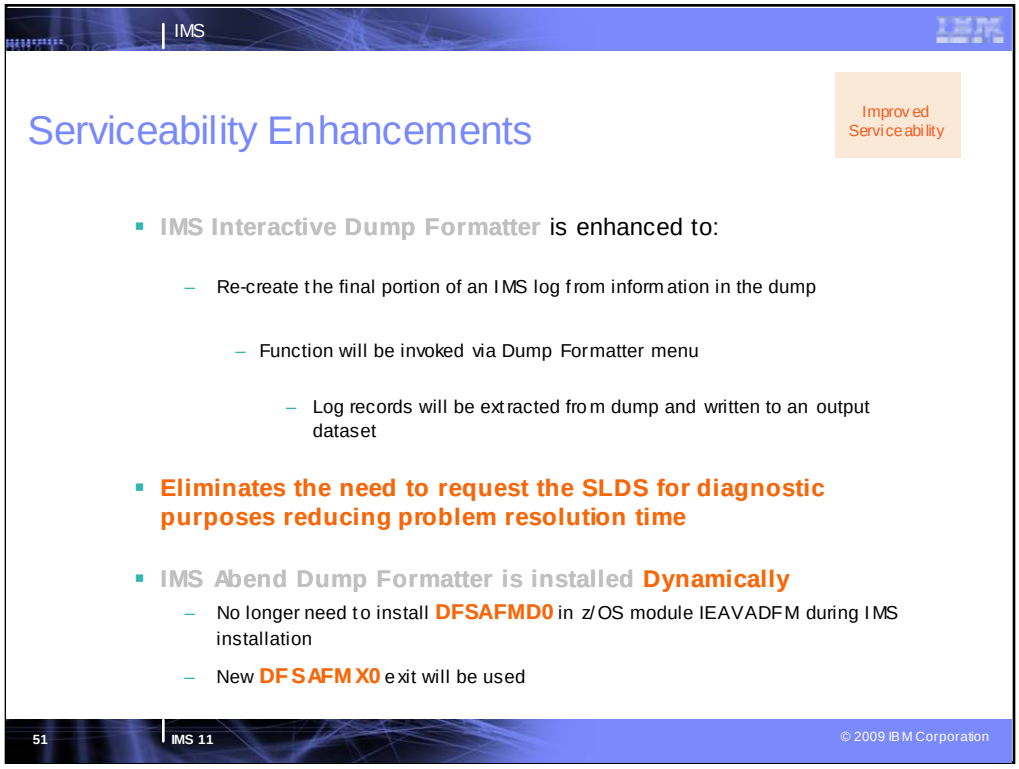
Removal of Obsolete User Message Exits - Eliminate the non-source (OCO) User Message Exits (HWSIMSO0 & HWSIMSO1) that have been functionally stabilized and are not modifiable by customers. Exits are replaced by HWSSMPL0/1.



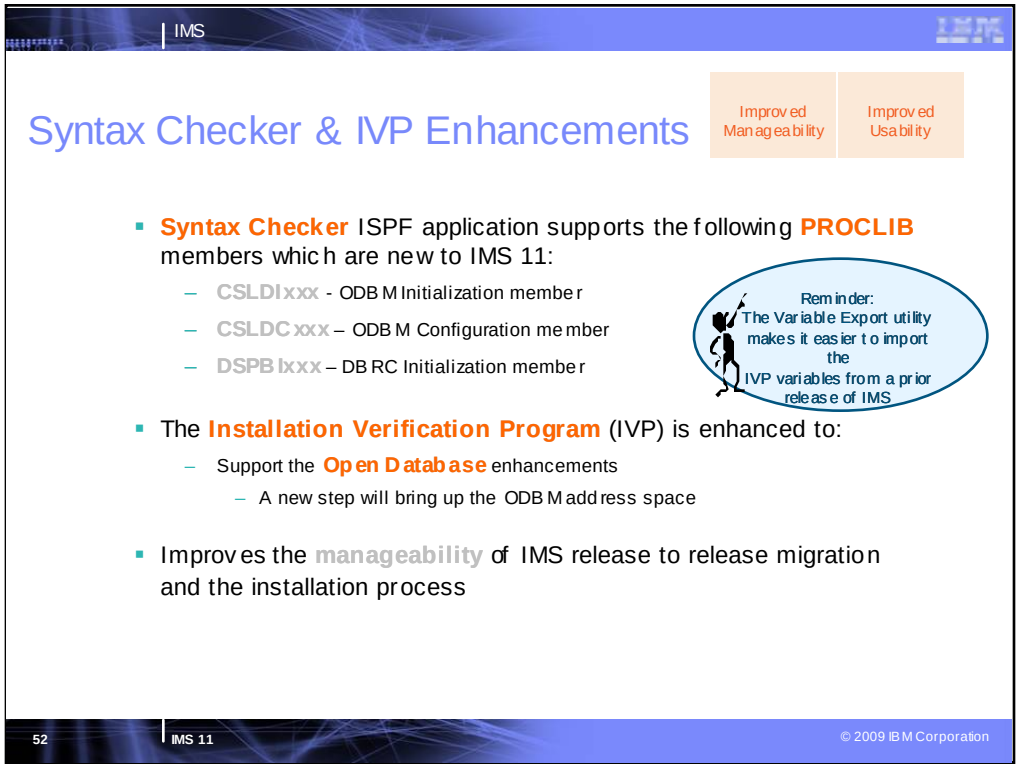
The User exit enhancements line item introduces the following new functions in IMS Version 11 and applies to the new exit routines introduced in IMS 11 (Early Initialization, CQS Event and CQS Structure Event) and the Restart exit which was introduced in IMS 10:

•A way to refresh exit routines online by using the new REFRESH USEREXIT command. Use the type-2 REFRESH USEREXIT command to bring new or modified exit routines online (and delete the old exit routines) without requiring that IMS be stopped and restarted. After the new or modified exit routines are online, any subsequent calls to those exit types will result in the execution of the new or modified exit routines. Only exit types that are specified on the TYPE parameter in the USER_EXITS section of the DFSDFxxx member of the IMS PROCLIB data set are eligible for refreshing. As part of processing the REFRESH USEREXIT command, IMS loads the new or modified exit routines and brings them online before deleting the old exit routines. If there is a failure in processing the command, the old exit routines remain online and are not deleted.

•A way to query information about certain exit routines by using the new QUERY USEREXIT command. Use the type-2 QUERY USEREXIT command to display information about the exit routines that are defined in the USER_EXITS section of the DFSDFxxx member of the IMS PROCLIB data set.



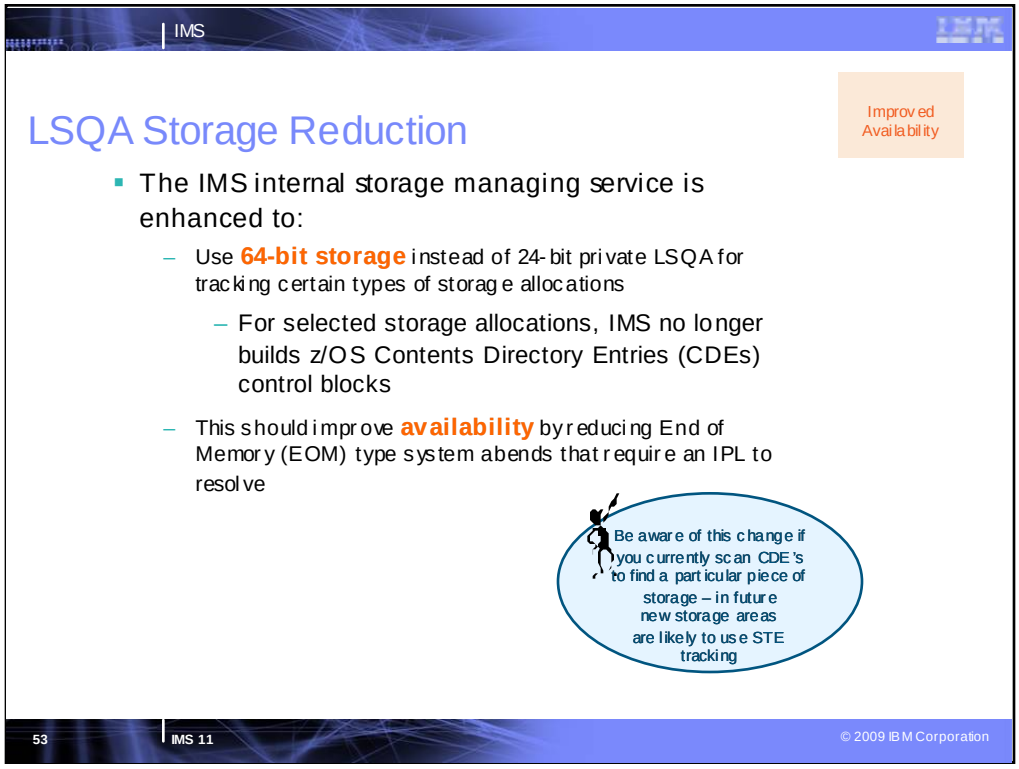
IMS Interactive Dump Formatter is enhanced in IMS Version 11 to re-create the final part of an IMS log from the information that is available in an IMS dump, thus eliminating the need to request the final system log data set (SLDS) for diagnostic purposes. When an IMS customer is working with an IBM Software Support representative, frequently the customer creates a memory dump and the service representative needs IMS log data. To get this log data, the customer requests the final SLDS, compresses (terses) it, and transfers the file to IBM by using FTP. The Service person decompresses (unterses) the data set and then analyzes it. The enhancements built into the IMS Interactive Dump Formatter in IMS Version 11 can build a log data set from the log records that reside in the dump's log buffers, thus avoiding the requests for the SLDS and the extra work associated with these requests.



The Syntax Checker ISPF application assists with IMS release-to-release migrations by providing the ability to convert supported IMS.PROCLIB members from one release to the other.

The IVP is enhanced to support the Open Database enhancements. In a new step, the ODBM address space will be brought up.

In IMS Version 10 and later, the Variable Export utility can be directly accessed as an option from the IVP Phase Selection panel, which makes it easier to import the IVP variables from a prior release of IMS.



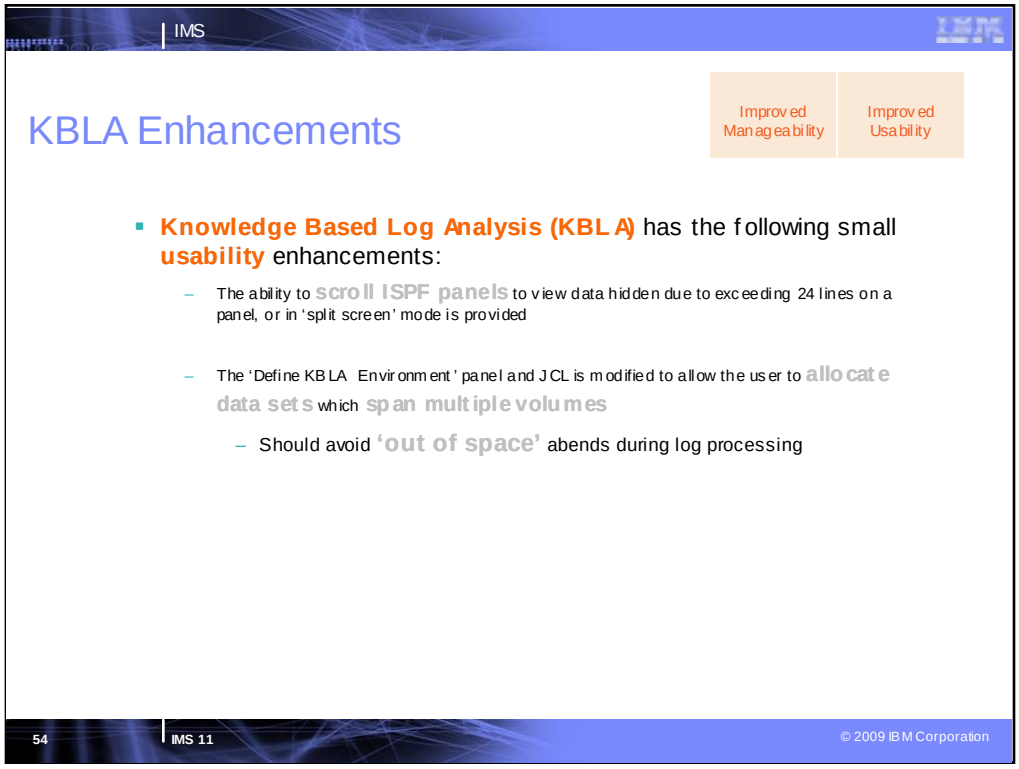
The IMS internal storage managing service, IMODULE, is updated to use 64-bit storage, instead of 24-bit private storage, for certain IMS functions. This enhancement should reduce the occurrences of end-of-memory (EOM) type IMS abends that require an IPL of the z/OS system to resolve.

IMS provides an internal IMS service called IMODULE, which IMS modules use to allocate and release storage, and load and delete modules. Today, IMODULE keeps track of both storage areas and modules by building a control block structure that is defined by z/OS – CDEs and related blocks. These z/OS blocks must architecturally reside in 24-bit private storage (LSQA storage). 24-bit private storage is a limited resource (limited to a maximum of sixteen megabytes, but more practically in the range of eight to ten megabytes). With the large size of today's address spaces, it is possible to allocate more storage areas in 31-bit storage than it is possible to track using 24-bit CDE structures. When this happens in the IMS CTL or DLI address spaces, it is often the case that z/OS itself cannot get enough storage to perform recovery/termination manager (RTM) processing for the address space. This leads to "end-of-memory" (EOM) type abends, where IMS is unable to cleanup its allocated common storage. This often requires a z/OS IPL to clear up the "orphaned" common storage so that IMS can be restarted on that z/OS.

This line item creates a new internal IMS service and control block structure for tracking storage. The use of this new tracking service for any given storage request is controlled by coding a new optional parameter on the IMODULE macro. The storage tracking elements (STEs) are built in 64-bit storage for the CTL region and DLI region (no LSQA storage is used). Other region types continue to use the CDE tracking technique.

The IMODULE GETMAIN storage requests for an internal type of IMS block called a "BCB IPAGE" are changed to use the new parameter to request tracking by IMS STEs, rather than by z/OS CDEs. BCB IPAGE storage is heavily used for many IMS internal processes and control block structures. Often, run-away conditions lead to the allocation of many IPAGEs of storage, and can lead to the out-of-storage and end-of-memory conditions. Thus, moving this one type of IMS storage to be tracked by STEs should address many of the common scenarios that end up leading to EOM and z/OS IPL situations. However, note that the design of the new IMODULE service is such that other IMS storage areas could easily be changed in the future to be tracked with STEs, should the need arise.

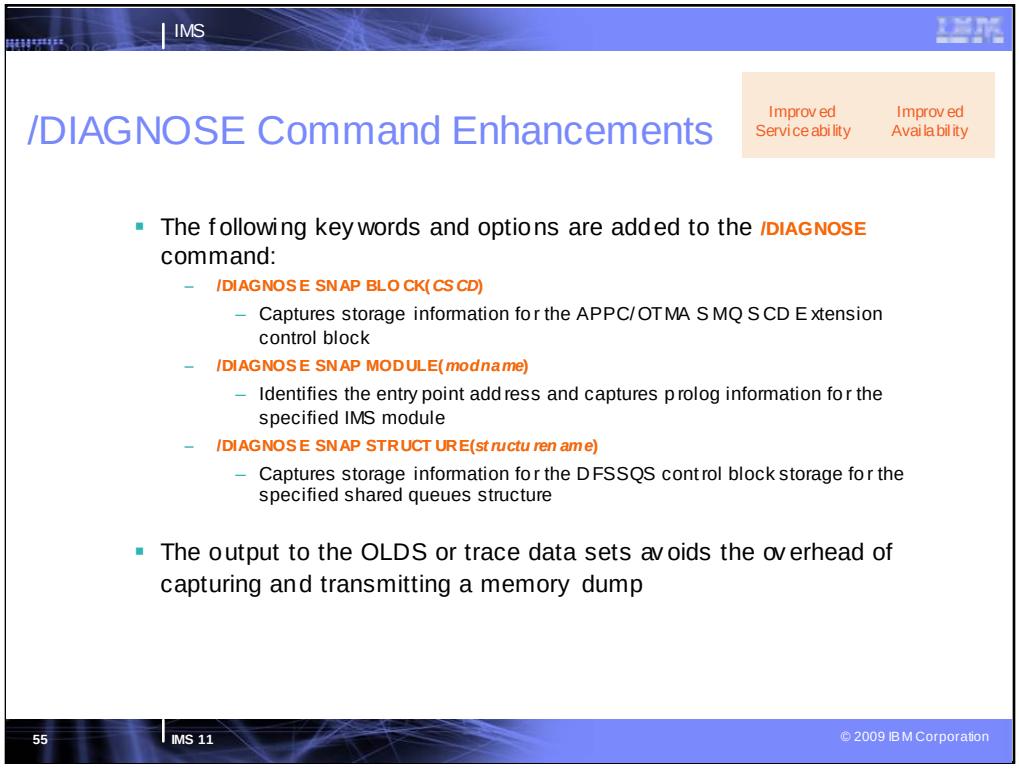
For this line item, only DFSBCB IPAGE storage will be changed to be tracked with STEs; however, future use of STE tracking – particularly with new storage areas – will certainly occur.



When you upgrade the KBLA utilities to support IMS Version 11, the following functions are added:

•Data entry panel scrolling

•Multi-volume KBLA output data set allocation



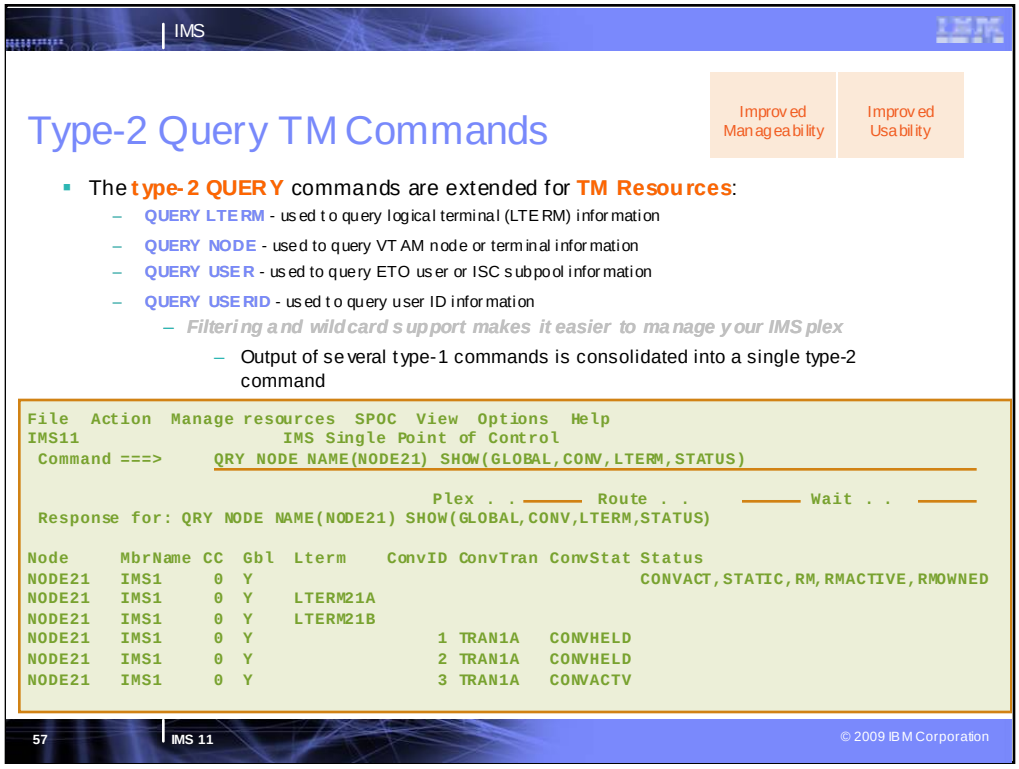
The /DIAGNOSE command takes a snap shot of IMS system resources at any time without impacting availability. The output produced can be quickly transmitted to IBM Software Support, thus avoiding the overhead of capturing and transferring a memory dump. The following keywords and options are new to the /DIAGNOSE command:

•BLOCK(CSCD) Captures storage information for the APPC/OTMA SMQ SCD Extension control block.

•MODULE(modname) Identifies the entry point address and captures prolog information for the specified IMS module. The prolog information contains the current maintenance level for a module and can help you to determine if any maintenance is missing.

•STRUCTURE(structurename) Captures storage information for the DFSSQS control block storage for the specified shared queues structure.

There are also a number of specific IMS TM enhancements.



This enhancement provides type-2 command support for data communications resources. It provides QUERY command support for NODE, LTERM, USER and USERID. This enhancement improves ease of use because the QUERY commands are enhanced commands which allow wildcards and filters that provide options not available to type-1 commands. The new TM type-2 QUERY commands are an alternative, not a replacement, for existing type-1 commands. The new commands enable you to manage your IMSplex environment more efficiently.

The new TM type-2 QUERY commands and their usage are:

QUERY LTERM - used to query logical terminal (LTERM) information

QUERY NODE - used to query VTAM node or terminal information

QUERY USER - used to query ETO user or ISC subpool information

QUERY USERID - used to query user ID information

The diagnostic information that can be generated by using a single TM type-2 command is comparable to the diagnostic information that can be generated by consolidating the output of several existing type-1 commands. In general, equivalent type-1 and type-2 commands operate in the same manner. Issuing type-2 commands requires a Common Service Layer with a minimum of an OM and SCI.

Query Node example:

There are two IMS systems in the IMSplex: IMS1 and IMS2.

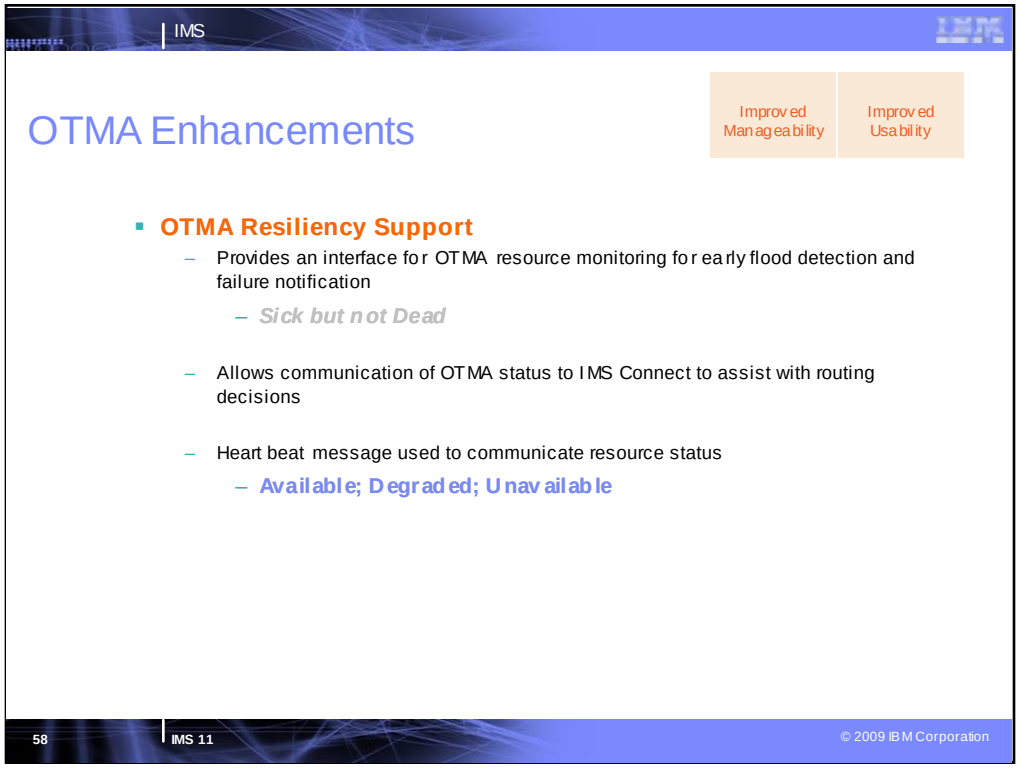
The Resource Manager (RM) is maintaining status in the resource structure (STM=YES), and Shared queues are enabled.

IMS1 is the command master, and because SHOW(GLOBAL) was specified, IMS1 is the only system that processes the command.

Any other IMS ignores the command (RC=4, RSN=x1000). NODE21 exists in the resource structure. IMS1 displays a global line which shows that the node is active in the IMSplex (RM, RMACTIVE, and RMOWNED status), and has a conversation active. There are two lterms assigned to the node, and are displayed on separate output lines. There are three conversations associated with the node, and are displayed on separate output lines.

IMS1 displays an additional global line for each assigned LTERM, so 2 lines are displayed for the 2 lterms LTERM21A and LTERM21B.

IMS1 displays an additional global line for each active conversation for the node. 3 lines are displayed, because the node has 2 held conversations and 1 active conversation.



OTMA Resiliency Support - When an OTMA becomes unable to effectively process the work submitted by an OTMA client (that is, sick, but not dead), the client application is usually unaware of the situation and continues sending work to the IMS. This situation could lead to flooded transactions and can result in an IMS outage that disrupts all client applications.

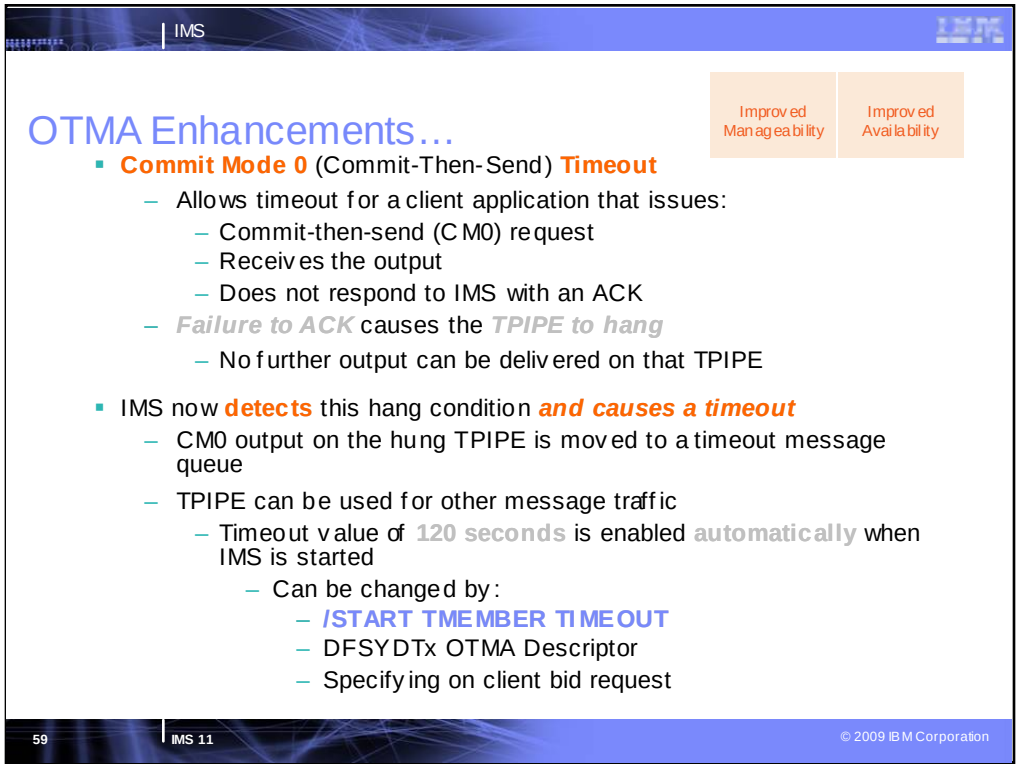
OTMA Resiliency Support provides a client-server protocol that allows OTMA to detect the sick-but-not-dead situations related to OTMA processing, identify the resources involved, take server actions if needed, and send out a protocol message to the client. This protocol message would then be processed by an OTMA client, such as IMS Connect or MQSeries, where the transaction can be redirected to another IMS. If there is no other suitable IMS to reroute the transaction to, the OTMA client could mark the IMS system "temporary unavailable, try again later".

IMS Connect utilizes this OTMA function by processing these protocol messages, updating its data store entry, and recording new data store events for warning and severe status so that IMS Connect vendor applications and user exits can access the information and redirect the transaction requests to a different IMS if needed.

Example 1: IMS Connect submits a lot of transactions to OTMA. However, OTMA in IMS has experienced a slow-down issue and failed to process the transaction in a timely manner. Once OTMA detects that the message flood level for the IMS Connect reaches 80% of the threshold, the OTMA resource monitor will send an action message to inform IMS Connect. In addition, OTMA periodically sends out a "heart beat" message with the resource info. to IMS Connect. In this case, the "heart beat" message will indicate a warning condition with the flood status if the flood condition still exists for the IMS Connect.

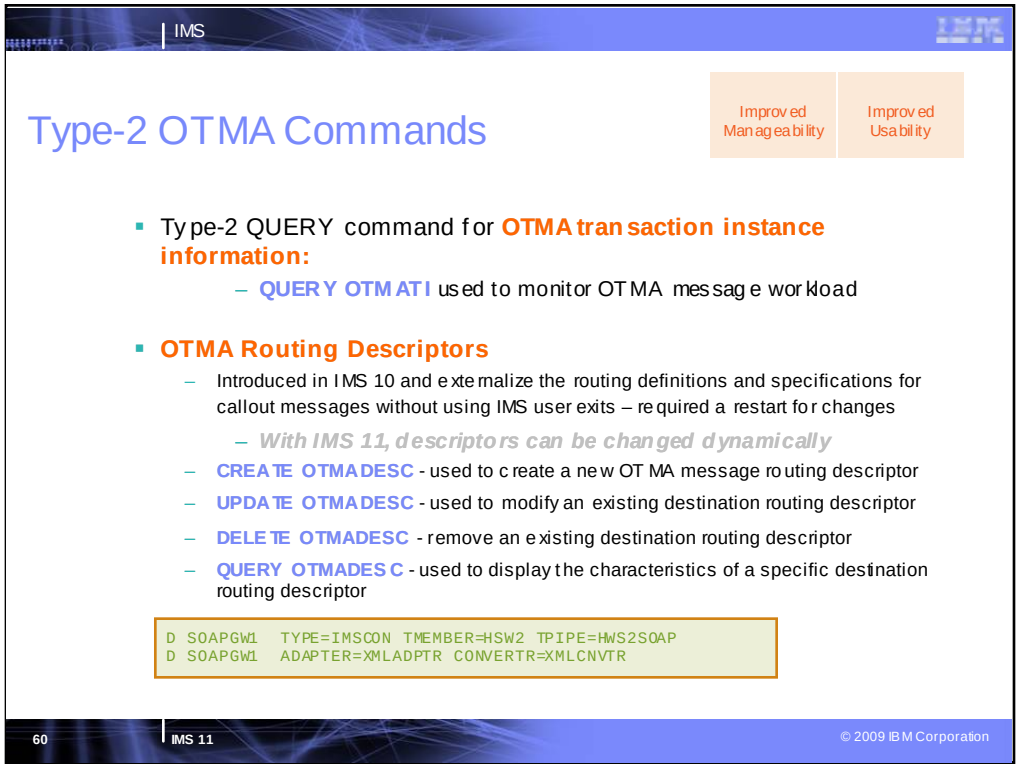
Example 2: Both IMS Connect and MQSeries submit transactions to IMS. If the total number of un-completed send-then-commit (CM1) transactions reaches the warning level, OTMA resource monitor will send out an action message to inform them. Also, a heart-beat message which includes any flood control status will be sent out later.

Example 3: Due to the potential I/O hung or IMS TCB hung conditions, OTMA transactions may not be able to be enqueued and the OTMA transaction message block (YTIB) may not even be created. However, an OTMA client such as IMS Connect may not know this condition and continues to submit transactions to IMS OTMA via the XCF interface. OTMA has been enhanced to detect the number of unprocessed OTMA XCF messages and send out an action message if a high number of XCF messages exist in OTMA.



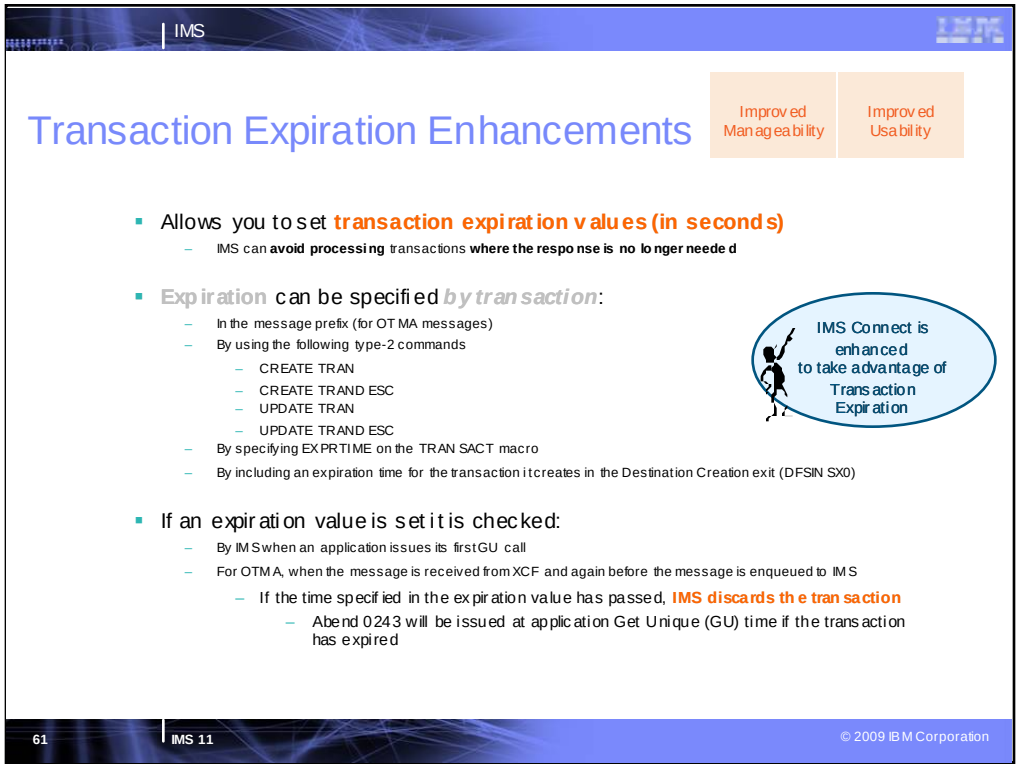
If an ACK or NAK (positive or negative acknowledgement) is not received, commit mode 0 TPIPEs could hang and become unusable. Lots of output messages could then be queued to the TPIPE. The time-out function is needed for CM0 TPIPEs so that the TPIPE can be usable. This is similar to the existing CM1 Timeout function.

If a client application receives commit-then-send (CM0) output via an OTMA transaction pipe (TPIPE) and fails to respond with an acknowledgement (ACK), the TPIPE in IMS will hang and the subsequent output on the TPIPE cannot be delivered. The IMS 11 OTMA CM0 ACK timeout enhancement will detect this hang condition and take the time-out action so that the CM0 output on the hung TPIPE can be moved to a time-out message queue and the rest of the output on the TPIPE can continue to flow.

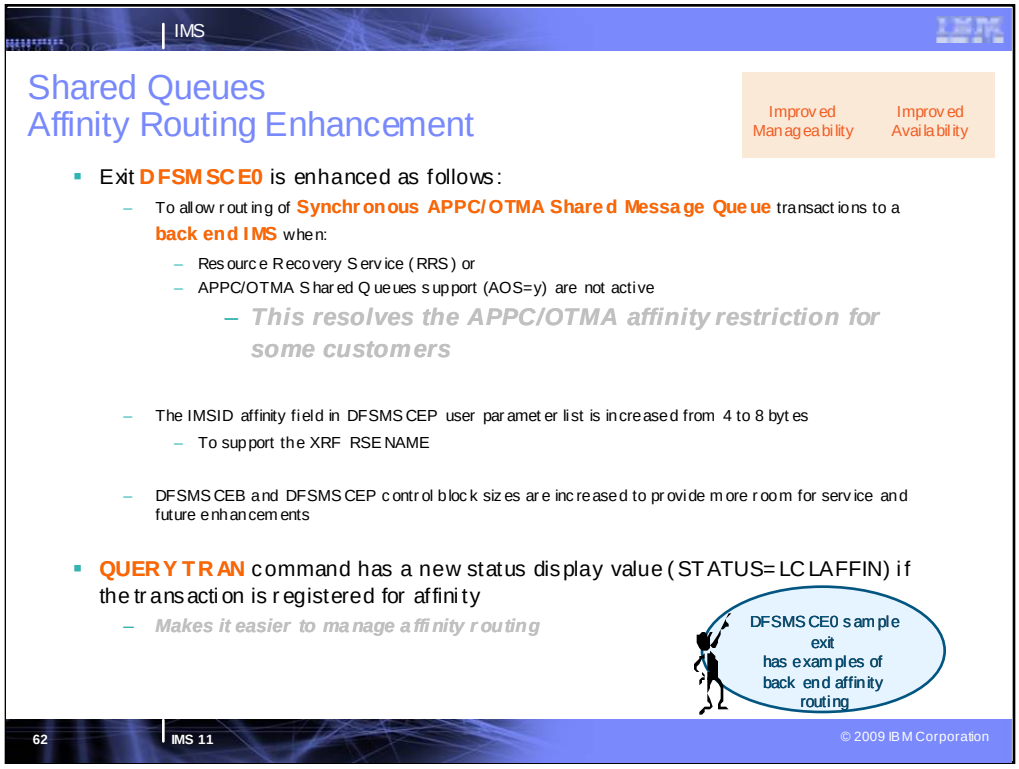


The OTMA type-2 QUERY OTMATI command for monitoring workload enables you to view a summary of the number of messages in the OTMA send-then-commit (CM1) message queue. You can retrieve this information by issuing the command QUERY OTMATI and include parameters to filter the result information. If the SHOW parameter is used, the individual characteristics of each transaction instance block are displayed. You can view the length of time that a transaction instance has been in existence and the correlation ID of the input message. This information can help you determine if there are problems in processing OTMA input messages.

Prior to IMS Version 11, if you wanted to make changes to the destination routing descriptors you had to wait for a scheduled outage or for IMS to be restarted. The OTMA type-2 commands for destination routing descriptors allow you to change the descriptors dynamically without interruption to a running IMS instance. You can dynamically add, update, delete and query destination routing descriptors while IMS is actively running by issuing these commands: CREATE OTMADESC, UPDATE OTMADESC, DELETE OTMADESC, and QUERY OTMADESC. In IMS Version 10, the definition of destination routing descriptors in the DFSYDTx PROCLIB member data set had to be entered in a specific order. In IMS Version 11, you no longer have to be concerned with order. You can input the entries in any order.



IMS Version 11 has the ability to interrogate an expiration time associated with transactions and discard (not process) the transactions if the time specified has passed. By discarding transactions whose expiration time has passed, processing costs and CPU cycles are not spent for the unwanted transactions. The expiration time is specified in the TRANSACT macro, type-2 commands, or the OTMA message prefix. OTMA, IMS Connect, some type-2 commands, and the RDDS Extraction utility are enhanced to support the transaction expiration function. (The Resource Definition Data Set (RDDS) Extraction utility is enhanced to recognize the new transaction expiration attribute so that transactions with the attribute specified can be imported from or exported to the RDDS.)



The capability and usability of the TM and MSC Message Routine and Control User exit routine (DFSMSCE0) is enhanced for IMS Version 11. The affinity support in DFSMSCE0 is enhanced in the following ways:

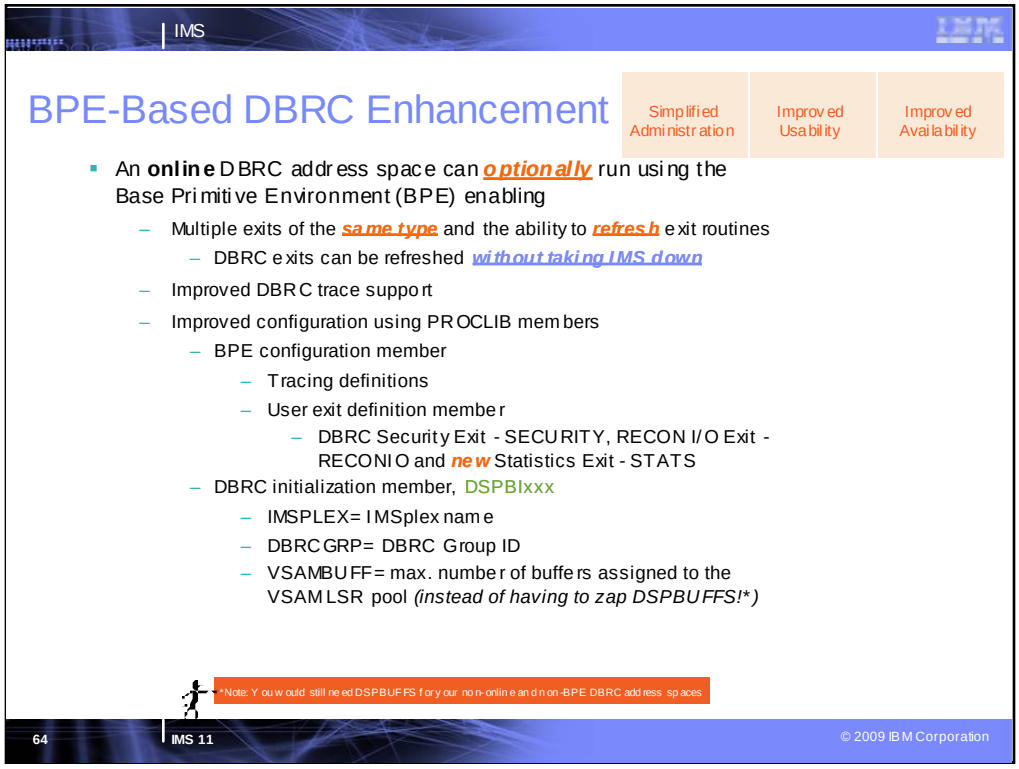
•The DFSMSCE0 user exit can route APPC/OTMA synchronous messages to a back-end IMS when the Resource Recovery Service (RRS) or the APPC/OTMA Shared Queues support (AOS) are not active. This resolves the APPC/OTMA affinity restriction for some customers.

•The IMSID affinity field in DFSMSCEP user parameter list is increased from four to eight bytes to support XRF.

•A new status display value is added to the QUERY TRAN command that displays if the transaction is registered for affinity (STATUS=LCLAFFIN). This makes affinity routing easier to manage.

•The DFSMSCEB and DFSMSCEP control block sizes are increased to provide more room for service and future enhancements to the DFSMSCE0 user exit.

There are also enhancements for DBRC usage



In order to take advantage of several of the services the Base Primitive Environment provides, we are adding an option to start DBRC on a BPE environment. With BPE, we can provide improved user exit management and trace support. This support is totally optional. You either start DBRC with your current procedures, or you can use a new startup JCL member that uses BPE. This allows you to convert your JCL over time.

Unlike the current DBRC start-up JCL, the new startup JCL and procedures will have a PROCLIB DD statement which allows online DBRC to have proclib members to support certain functional capabilities. For example, a BPE configuration member can specify trace options as well as user exit members. The user exit member defines which user exits should be used by this DBRC. BPE provides a much better user exit interface than currently provided through DBRC. For one thing, you have the ability to update user exits without shutting IMS/DBRC down. Also, you can have more than one exit routine called.

SECURITY type exit replaces DSPDCAX0. The parameter list is mapped by DSPDCABK.

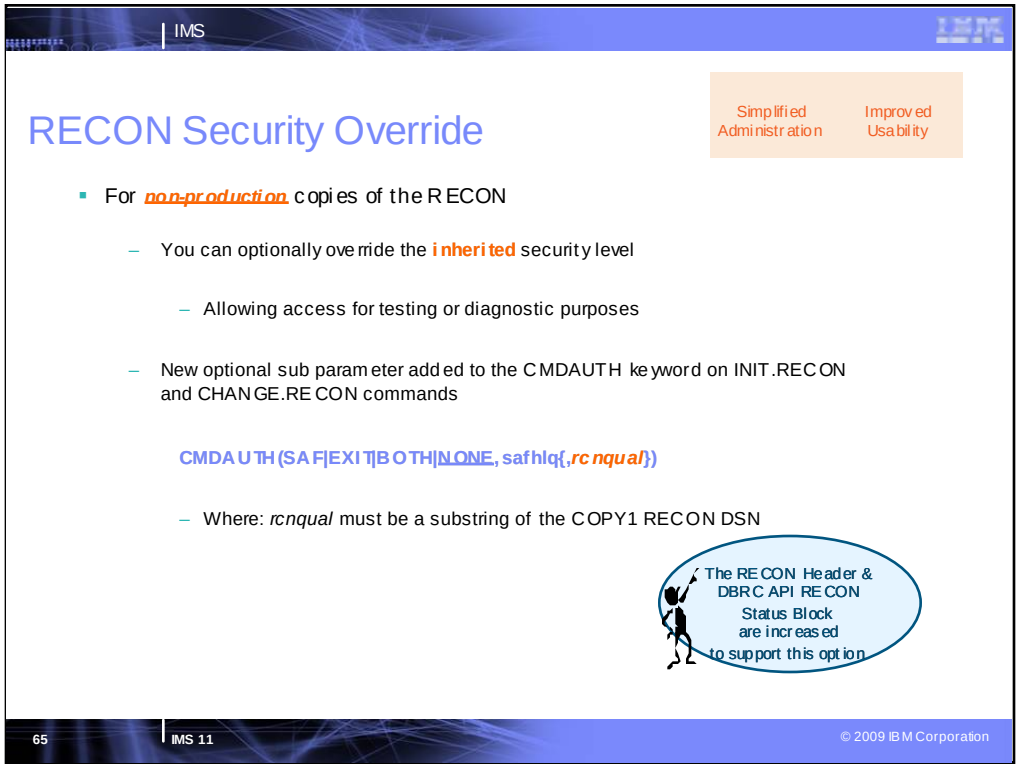
RECONIO type exit replaces DSPCEXT0. The parameter list is mapped by DSPRIOX.

STATS type exit is new. The parameter lists are mapped by DSPBSTX, DSPBST1 and DSPBST2.

The STATS exit includes a lot of information on specific types of work done by DBRC, such as the number of individual DBRC requests that are issued, number of VSAM requests made by DBRC and average time it takes to process a DBRC request.

We now allow you to define a DBRC Security Exit and RECON I/O Exit through the BPE EXITDEF proclib members. If specified, they would be used instead of the current DSPDCAX0 and DSPCEXT0. The new exits would have a slightly different interface as the current exits. Basically, we plan on using an interface that is similar to other CSL user exits, and put the current parameters passed to the exits at the end of the new parameters in the same order. We plan on providing a new Statistics Exit that will provide such things as average request times, average wait times (for reserve, and other things). We also plan on taking advantage of BPE tracing services. For example, BPE external trace could be used for the online region instead of GTF tracing.

A DBRC initialization member can be used to specify things like the DBRCGRP and IMSPLEX name. We are also adding the ability to specify VSAM LSR buffer usage for the online DBRC instead of requiring you to zap DSPBUFFS. Note: you would still need DSPBUFFS for your non-online DBRC image and non-BPE.

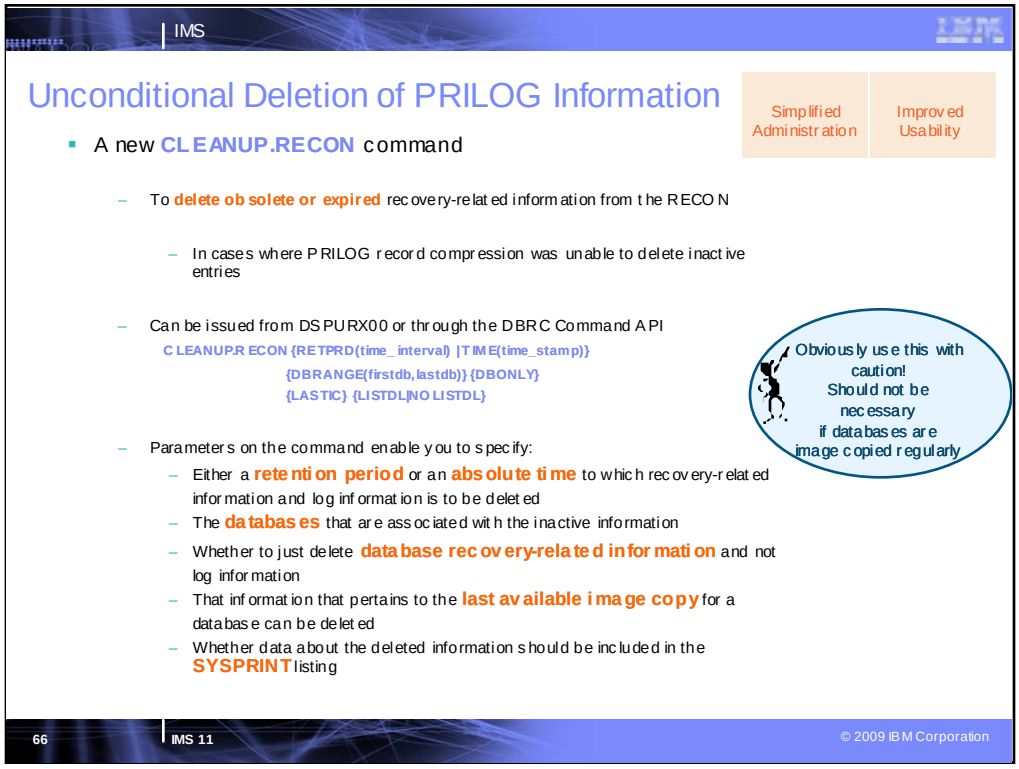


Non-production copies of the RECON data set inherit the security level of the production RECON from which they were copied. This security characteristic poses a problem when someone who does not have the original authority level needs to access the RECON data set, such as for testing or debugging purposes. To solve this access problem:

•The CMDAUTH keyword on the INIT.RECON and CHANGE.RECON commands is enhanced in IMS Version 11 with the option to override DBRC security for non-production copies of the RECON data set.

•The RECON header record (DSPRCNRC) is enhanced with the new RCNQUAL field, RCNCMDRNQ, which tells DBRC if the RECON data set is a copy and whether command authorization should be enforced or not.

•The new apqrc_CmdRNQ field that contains the RCNQUAL value is added to the DBRC API RECON status block (DSPAPQRC).



You can use a new DBRC command, CLEANUP.RECON, to delete old or expired recovery-related information from the RECON data set. The RECON data set contains information that is needed to recover databases. The PRILOG record in the RECON data set can grow to be very large. PRILOG record compression, which is the automatic deletion of inactive data set entries in the PRILOG record, sometimes cannot take place for various reasons. In these cases, the information must be deleted manually.

With the appropriate authorization, you can use the CLEANUP.RECON command to remove obsolete information from the RECON data set. The CLEANUP.RECON command can be issued from within the Database Recovery Control utility (DSPURX00) or embedded in a DBRC Command API request. The parameters on the CLEANUP.RECON command enable you to specify:

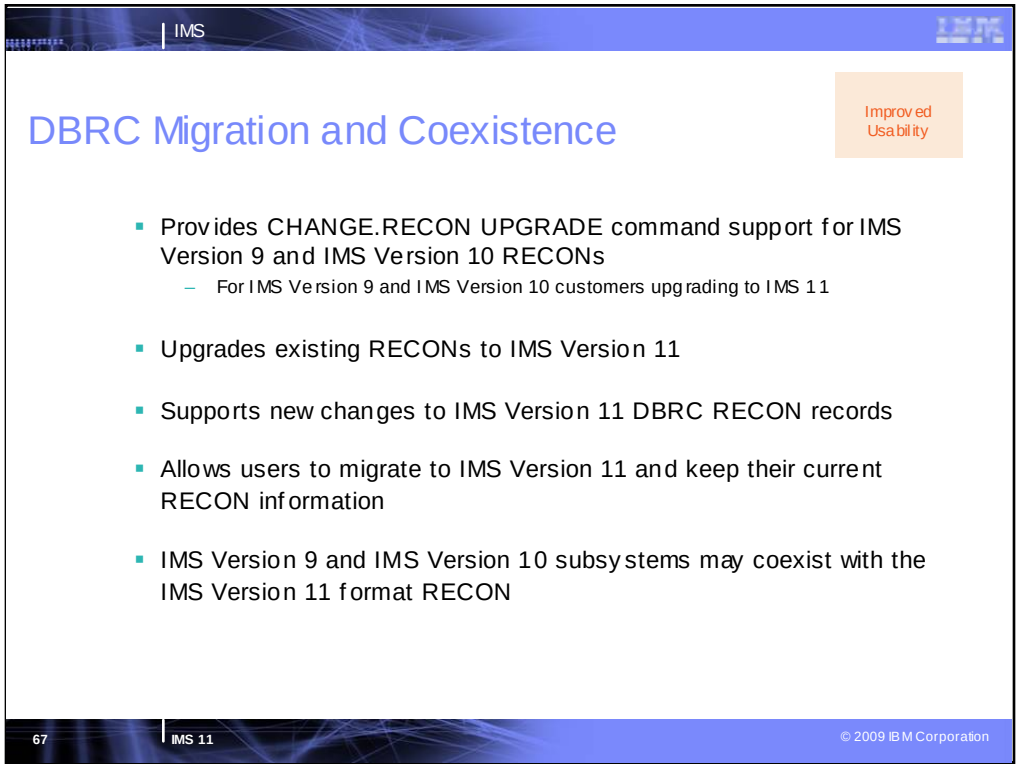
•Either a retention period or an absolute time to which recovery-related information and log information is to be deleted. It is recommended to use small deletion periods when using this cleanup command. The smaller the deletion periods, the less time the RECON will be held.

•The databases that are associated with the inactive information

•Whether to just delete database recovery-related information and not log information

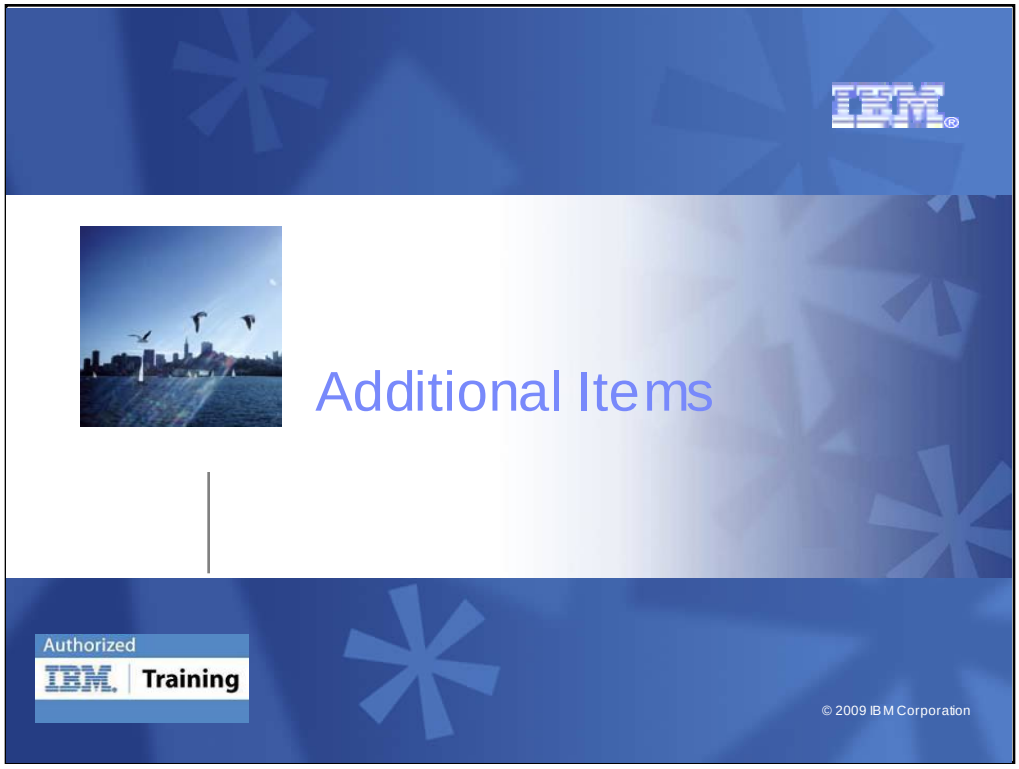
•That information that pertains to the last available image copy for a database can be deleted

•Whether data about the deleted information should be included in the SYSPRINT listing.

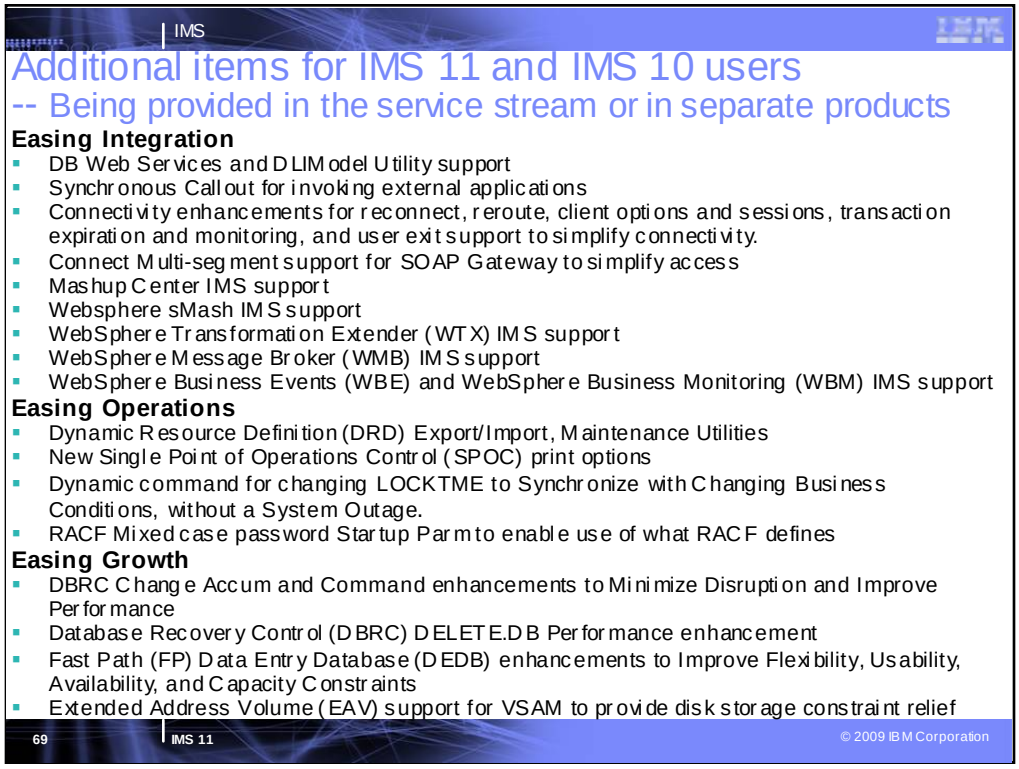


Migration to IMS Version 11 and coexistence support is provided for the DBRC RECONs. An IMS Version 9.1 RECON or an IMS Version 10 RECON may be upgraded to IMS Version 11. RECONs using Parallel RECON Access may also be upgraded. IMS Version 9.1 and IMS Version 10.1 may coexist with an IMS Version 11 RECON provided the respective coexistence Small Programming Enhancements are applied to the lower releases.

The RECON header, Change Accumulation execution record, and database records have been increased in size.



Additional items are also being provided for IMS 11 and IMS 10 users



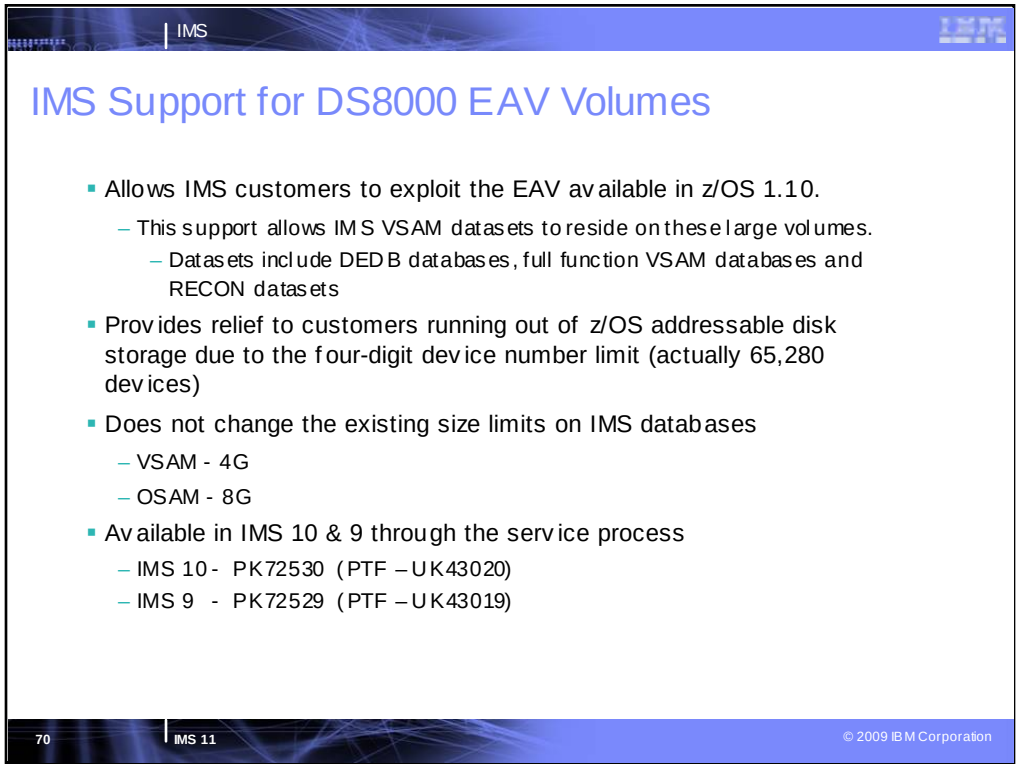
Key Message: These are many additional enhancements being provided for IMS 11 and IMS 10 users.

IMS 10 enhancements are being provided through the IMS 10 service process or as enhancements to separate related products.

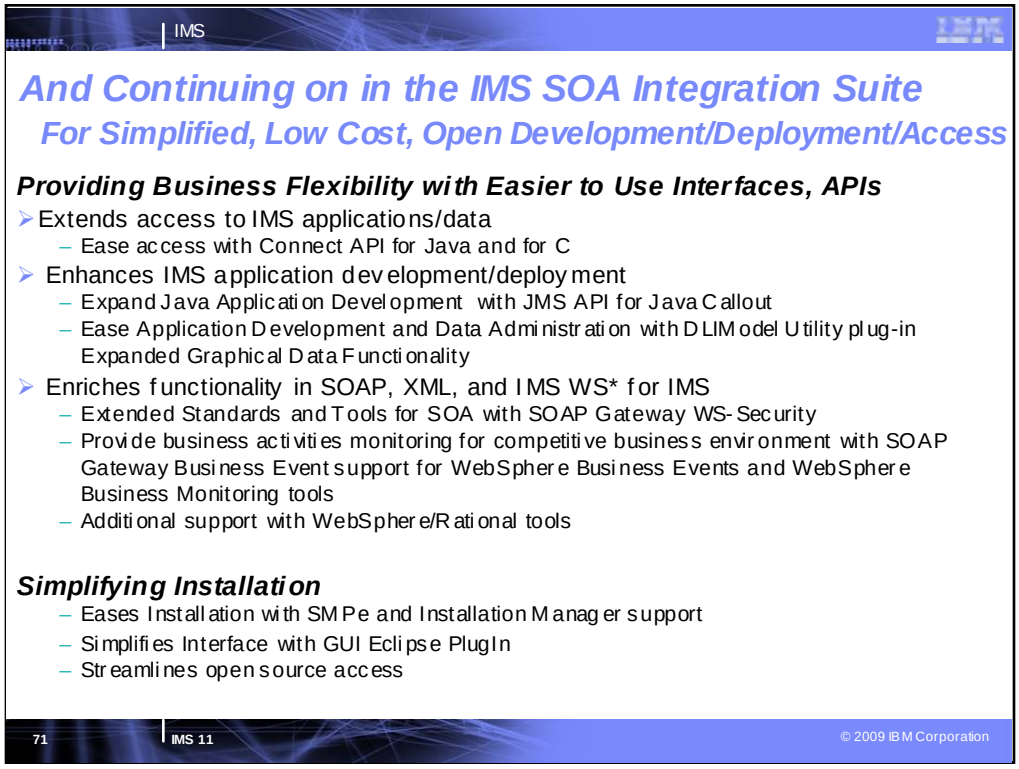
Customers have asked for still more IMS connectivity enhancements that extend access to IMS TM while reducing existing complexity and resource requirements. These enhancements can improve reliability and serviceability and enhance resilience, performance and availability.

Separate related products, like the WebSphere Transformation Extender (WTX), provide support to invoke IMS transactions while leveraging standards-based transaction support on distributed platforms of complex data formats and unique industry requirements. This support provides faster standards compliance and improved data quality with automated data validation using industry and regulatory standards.

I'll expand on a few of these items.



IMS support for the DS8000 Extended Address Volumes is provided for VSAM in conjunction with z/OS 1.10 and allows IMS VSAM Datasets to reside on volumes that contain more than 65K cylindrs. Datasets include DEDB databases, Full Function VSAM databases and RECON datasets. This support provides relief to customers running out of z/OS addressable disk storage due to the 4-digit device number limit.



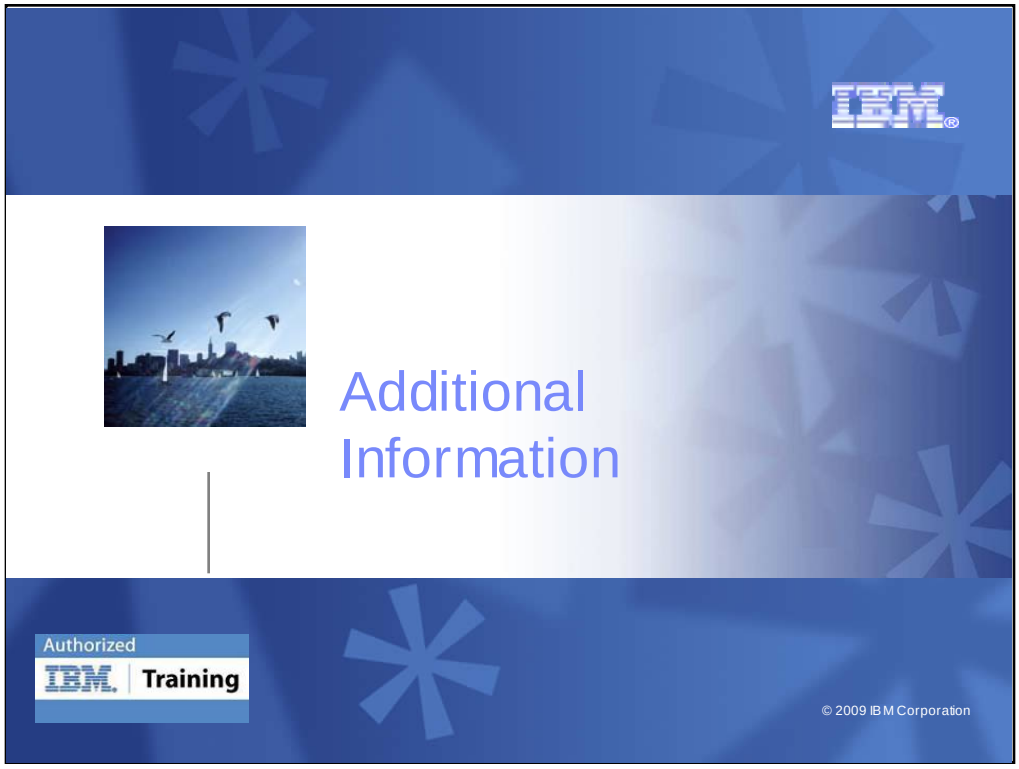
KeyMessage: IBM continues to enhance IMS, addressing customer requirements for simplified, low cost, open development, deployment and access.

IBM is providing IMS solutions that ease integration with new technology for a service oriented architecture -- focusing on open, distributed connectivity, expanded application development access support, extended Web Services and connectivity for SOA

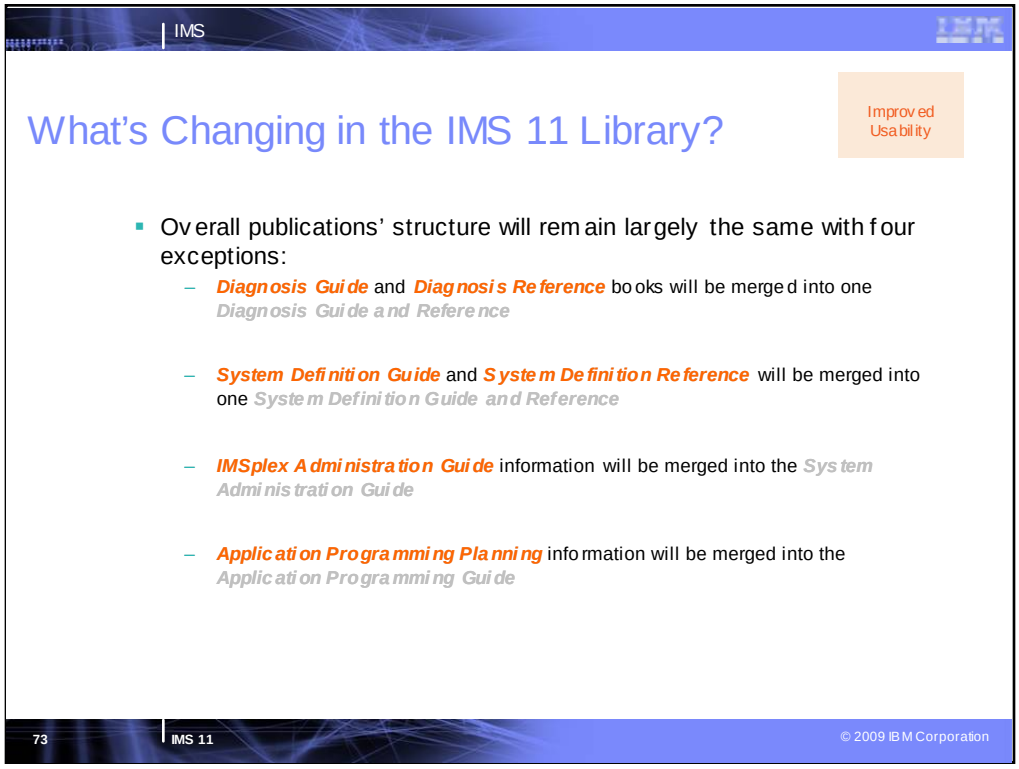
We are also providing solutions that help simplify installation and management.

IBM is adding to the IMS SOA Integration Suite with independent components that extend IMS access and use industry standard tools/interfaces to modernize/speed application development/deployment, enrich functionality and ease installation and use.

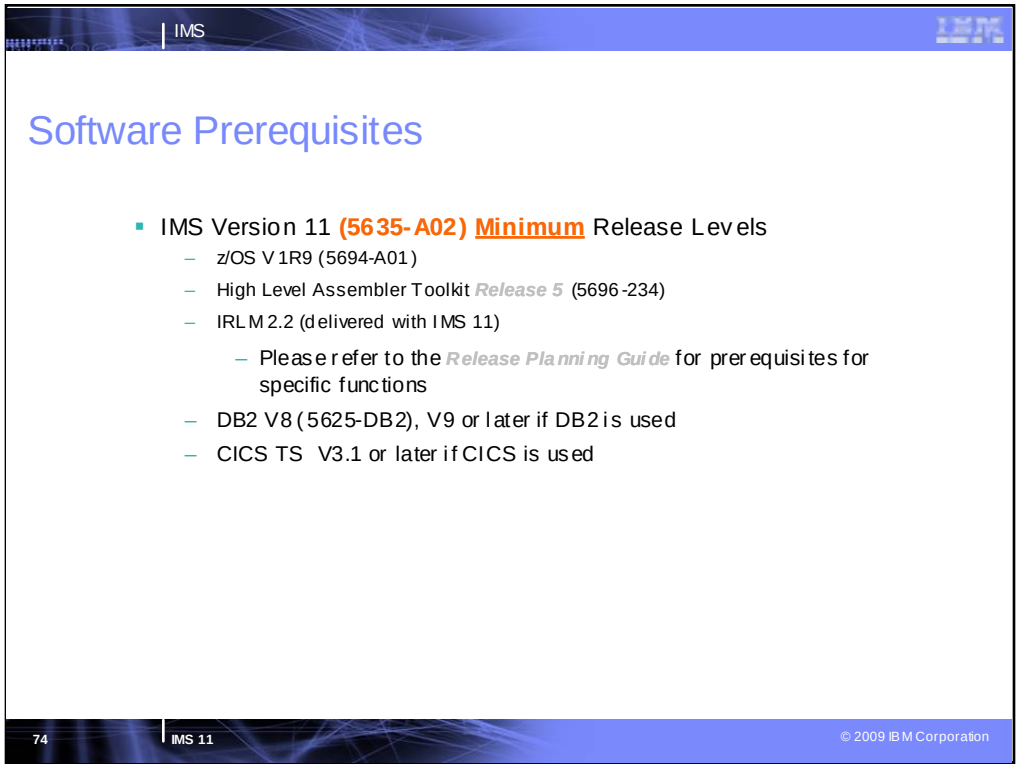
It is a collection of IMS middleware functions and tools that support your IMS on demand systems and your distributed IMS application environment. Components are designed to enhance your use of IMS applications and data. These components deliver innovative new capabilities for your IMS environment that enhance connectivity, expand application development, extend standards and tools for a Service Oriented Architecture (SOA), ease installation, and provide simplified interfaces. It includes the Connect API for Java and for C (though C support is being provided through the service process), the JMS API open source used for IMS Java application callout, the DLI Model Utility plug in, with its enhancements, and required open source, and the IMS SOAP Gateway for enhanced connectivity to/from IMS applications and data along with its open source and enhancements for WS-Security and Business Events. Also provided are the Installation Manager and SMP/E support to ease installation on the distributed and z/OS platforms.



Here is some additional information related to IMS 11 migration.



The structure of the library changed quite a lot with IMS Version 10. In Version 11 we will improve the content of information deliverables, but the overall library structure will remain largely the same, with the exception to the noted areas above.



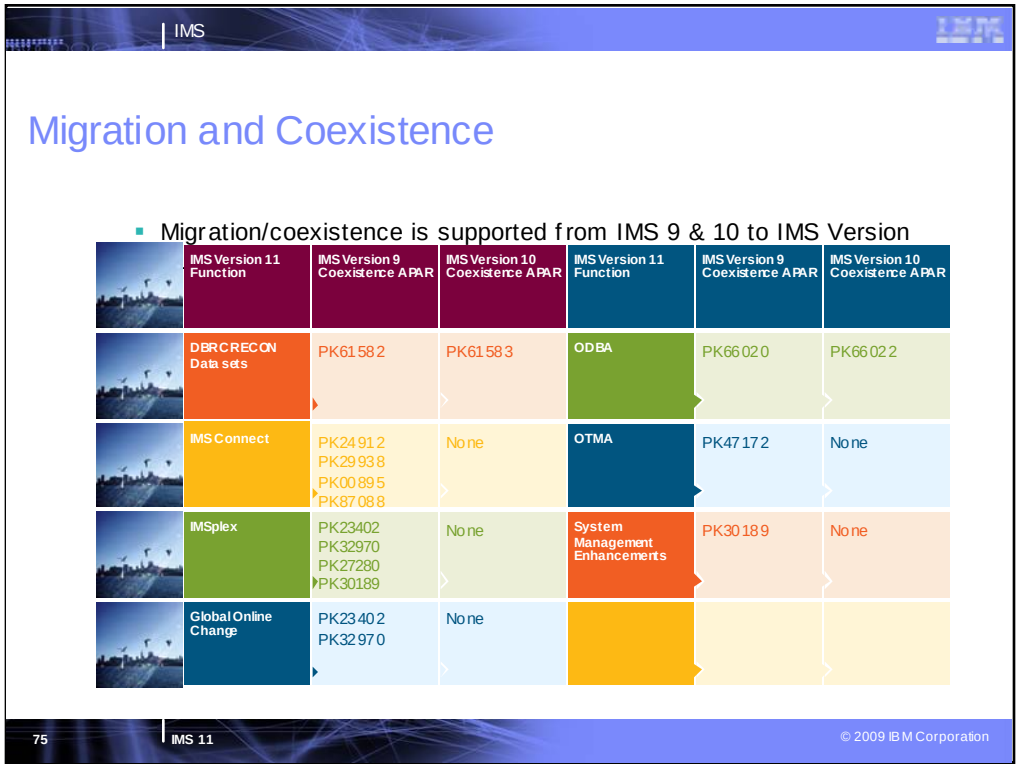
IMS 11 operates under z/OS V1R9 (5694-A01) configurations, or subsequent versions, releases and modification levels unless otherwise stated, and requires the following minimum version, or release or modification levels:

•z/OS V1R9 (5694-A01) with DFSMS (a base element of z/OS V1R9).

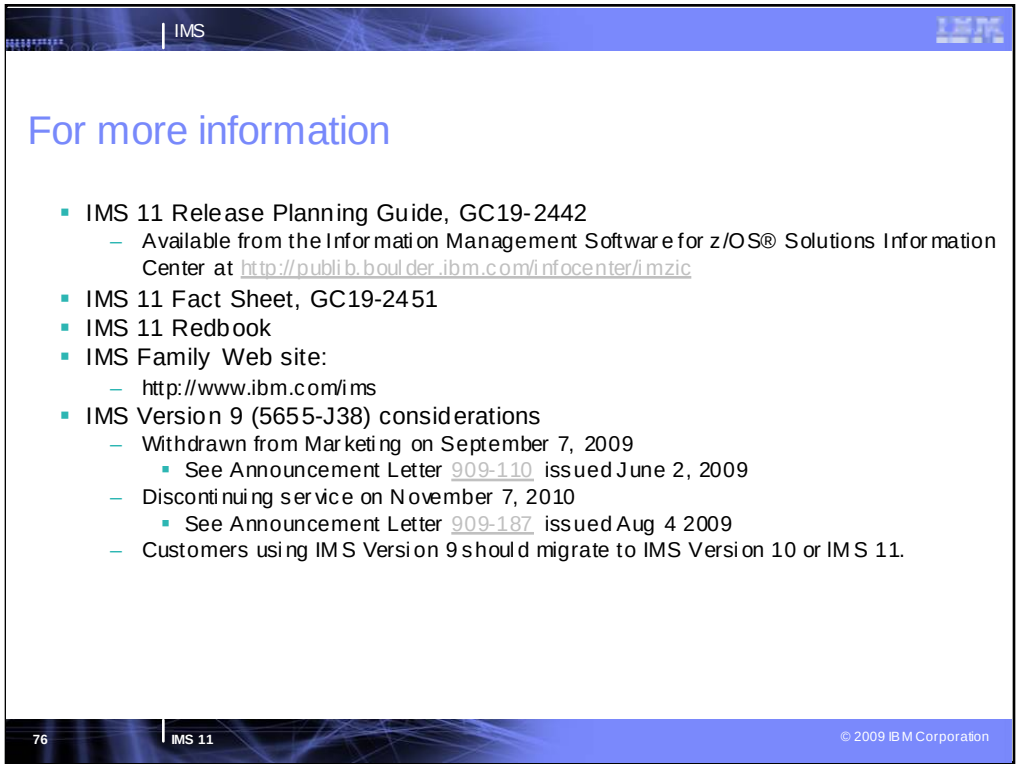
•RACF (included in a separately orderable SecureWay Security Server feature of z/OS V1R9), or equivalent, if security is used.

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

•Note, CICS 3.1 no longer supports VS COBOL (see CICS 3.1 materials). CICS 2.3 is planned to be out of service shortly after IMS 11 GA.



Migration/coexistence is supported to IMS 11 from IMS 9 and IMS 10 environments



There is also additional information being provided in other sources, eg. the Release Planning Guide, Fact Sheet, Redbook. And the IMS web site has links to these and much more.

Since IMS V9 has been withdrawn from marketing and services is being discontinued in November 2010, customers using it should migrate to IMS V10 or IMS 11.

IBM has tremendously enhanced IMS with IMS 11, making it a very integrated, innovative, autonomic, and open solution. IMS 11 is just great really!

Important Disclaimer