
IMS Version 7



The world depends on it

*Rich Lewis and Suzie Wendler
IBM Dallas Systems Center*

© IBM Corporation 2000, 2001



Trademarks

▲ The following are registered trademarks of International Business Machines Corporation

- ▶ ACF/VTAM
- ▶ APPN
- ▶ DB2
- ▶ IBM
- ▶ IMS/ESA
- ▶ MVS
- ▶ OS/390
- ▶ Parallel Sysplex
- ▶ RACF
- ▶ VTAM
- ▶ S/390

▲ The following are trademarks of International Business Machines Corporation

- ▶ CICS
- ▶ DFSMSdss
- ▶ Hiperspace
- ▶ IMS
- ▶ MVS/ESA



Abstract

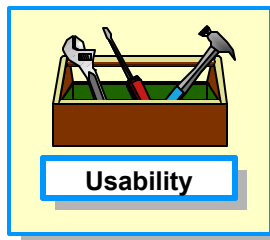
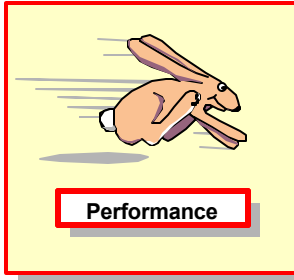
IMS, the recognized leader in database and transaction management, continues to evolve. IMS may be used to power e-business environments while preserving customers' investments in mission-critical applications and data.

IMS Version 7 introduces many enhancements to help its users provide increased availability, improved performance, and a more usable system. This presentation highlights the new system, database, and transaction manager capabilities. IMS V7 includes High Availability Large Database (HALDB) which supports databases with up to 1001 independently managed partitions, Online Recovery Service (ORS) which allows users to recover databases with online facilities, Rapid Network Restart which implements VTAM persistent sessions, Open Database Access (ODBA) which provides for IMS database access from DB2 stored procedures, DBRC usability and performance enhancements, improved DEEDB availability, and many performance, usability, and availability enhancements.

This presentation will be of interest to both IMS TM and DBCTL users.



IMS Version 7



© IBM Corporation 2000, 2001

IMS V7 delivers enhancements to IMS which allow users to increase their systems and data availability, improve their performance, and to enjoy greater usability. This presentation indicates the which of these characteristics each enhancement is designed to address.



IMS Version 7

▲ System Enhancements

- ▶ Install/IVP process
- ▶ System Parameters display
- ▶ CPLOG specification
- ▶ Concurrent upgrade of RECONs
- ▶ Online RECON access preference
- ▶ RECON loss notification
- ▶ Very Large RECON record warnings
- ▶ ACBLIB Online Change enhancement
- ▶ ACBGEN processing and limits
- ▶ IMS Control Region Resource Manager
- ▶ DB2 attach enhancements
- ▶ IMS Monitor and IMS PA enhancements
- ▶ OSAM 64 bit Real Support

The enhancements listed on this page and the next two pages are explained in this order in the rest of this presentation.



IMS Version 7

▲ Database Enhancements

- ▶ High Availability Large DB (HALDB)
- ▶ Online Recovery Service (ORS)
- ▶ VSAM REUSE Support
- ▶ OSAM DEBS above 16M line (ECSA)
- ▶ 255 OSAM Database Buffer Subpools
- ▶ Change Accum enhancements
- ▶ Image Copy 2 compression
- ▶ DBRC GENMAX and RECOVPD
- ▶ DBRC PROCOPT=L|LS support
- ▶ DBRC LIST command enhancements
- ▶ Batch Backout enhancements
- ▶ I/O error handling for DEDBs
- ▶ DEDB Scan segment expansion
- ▶ Open DB Access (ODBA)



IMS Version 7

▲ Transaction Manager Enhancements

- ▶ CQS Enhancements for Shared Queues
- ▶ Asynchronous OTMA/APPC
- ▶ VTAM Generic Resources enhancements
- ▶ Rapid Network Reconnect (RNR)
- ▶ TM and MSC Message Routing and Control User Exit
- ▶ Queue Space Notification Exit enhancement
- ▶ Deferred VTAM ACB open
- ▶ RACF PassTicket support
- ▶ USERID clarification
- ▶ SLUP/Finance Session Cold Termination
- ▶ Spool enhancement
- ▶ SLU2 enhancement
- ▶ ETO Enhancements
- ▶ Callable Interface to OTMA
- ▶ IMS Connect
- ▶ IMS Java



System Enhancements



▲ System Enhancements

- Installation and Definition
- System Execution
- ACBGEN
- Performance Monitoring



Install/IVP Enhancements

▲ Style and format of Install/IVP panels standardized

- Same look and feel as other IBM products
 - ▶ Action Bar Pull-Downs
 - ▶ Pop-up windows
 - ▶ Standard command and message areas



▲ Sample IVP jobs support DFSMS allocation parameters

- ▶ STORCLAS
- ▶ MGMTCLAS

Usability

IMS V7 changes the style of the panels displayed in the Install/IVP so that they have the same look and feel as other IBM products. In general, the changes are reflected in the colors shown and placement of the various panel areas.

The IMS IVP dialogs build JCL to install IMS. DFSMS MGMTCLAS and STORCLAS class names may be specified in the variable gathering phase with IMS V7. They then will be used for the allocation of IVP data sets.



Install/IVP Enhancements

▲ OS/390 Standards

- Data set names
 - ▶ Distribution libraries: ADFS prefix
 - ▶ Target libraries: SDFS prefix
- Macro libraries
 - ▶ One distribution library: ADFSMAC
 - ▶ One target library: SDFSMAC



Usability

▲ SYSGEN reduction

- More than 1000 modules moved from SYSGEN to DFSJCLIN

IMS V7 has changed the standard data set names for its distribution and target libraries. Data set names include ADFS (distribution libraries) or SDFS (target libraries) prefixes. Macro libraries have ADFSMAC or SDFSMAC prefixes. This adheres to the OS/390 naming standards for these types of data sets. These naming standards are also used by other OS/390 products, such as CICS and DB2.

IMS V7 has moved over 1000 modules from the system definition process to DFSJCLIN processing. This eliminates some system definition processing and allows SMP/E to track these modules when maintenance is applied to them. The exposure to installation errors has been lessened.



System Parameters Enhancements

▲ System parameters display

- Written to system console and job log at initialization

▲ IMS V6 CPLOG

- CPLOG (system checkpoint frequency) set by system definition
 - Change requires a new system definition
- Default is 1000

Usability

▲ IMS V7 CPLOG

- CPLOG default is 500,000
 - Reasonable default
- CPLOG is execution parameter in DFSPBxxx
- CPLOG may be displayed with /DIS CPLOG command
- CPLOG may be altered by /CHANGE CPLOG command
 - Easily changed

© IBM Corporation 2000, 2001

A new IMS message displays the system parameter values that are in effect during IMS initialization. The values can come from the system generation (defaults), the overrides in the DFSPBxxx member of PROCLIB, or override specification in the EXEC statement. The information provides an easily-identifiable format to document the values that are actually in effect.

IMS V7 has made the setting of CPLOG, which controls system checkpoint frequency, more dynamic.

In IMS V6 the CPLOG was specified in the system definition and could not be overridden. The default value was 1000. This default specified that a system checkpoint should be taken after writing 1000 log records.

The default for CPLOG in IS V7 is 500,000. This is a much more reasonable default value. The IMS V7 CPLOG specification may be overridden in the DFSPBxxx member. It also may be displayed and changed during IMS executions.

This combination on enhancements eliminates a cause for some installations doing an additional system definition. Incorrect checkpoint frequencies are much less likely to occur and, if they do, will not require new system definitions.



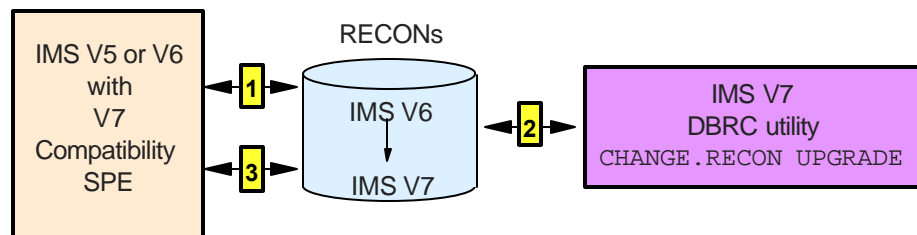
Concurrent Upgrade of RECONS

▲ IMS V6 RECONS are upgraded to V7 without terminating IMS

- Upgraded by DBRC command

Availability

▸ CHANGE .RECON UPGRADE



© IBM Corporation 2000, 2001

RECONS may be upgraded from the IMS V6 format to the IMS V7 format while systems remain active and connected to them. IMS systems do not have to be terminated to make this upgrade. This eliminates the outage that was required when upgrading the RECONS for previous releases.

The RECON compatibility SPE (small programming enhancement) must be on any system which has the RECONS open during the upgrade process. This SPE allows the IMS system to use RECONS with either the IMS V6 or IMS V7 formats. There are SPEs for both IMS V5 and IMS V6 systems.

The upgrade is done by the new CHANGE.RECON UPGRADE command using the DBRC utility. The command causes the utility to read the records in COPY1 and rewrite them in the IMS V7 format. When this completes, the records from COPY1 are written to COPY2. While these copies are being done, the RECONS are reserved by the DBRC utility. In this illustration, at step 1 the IMS system is accessing the RECONS. In step 2 the utility is upgrading the RECONS. In step 3 the IMS system is again accessing the RECONS.

Upgrades from IMS V5 RECONS to IMS V7 RECONS are done by the DBRC RECON Upgrade utility. Concurrent upgrades are not supported for these upgrades.



Online RECON Access Preference

▲ Reserves of RECONS by batch jobs will be serialized

- Only one batch job per MVS will request a RESERVE at any time
 - ▶ Batch jobs will request exclusive enqueue before issuing RESERVE
 - ▶ Utilities are included in "batch" jobs

▲ Avoids batch jobs locking out online systems from RECONS

Performance

© IBM Corporation 2000, 2001

IMS V7 gives preference to online systems over batch jobs and utilities in accessing the RECONS. Only one batch job or utility on an MVS system will ever attempt to RESERVE the RECONS at any time. This is done by forcing the batch jobs and utilities to acquire an enqueue on a resource. The same resource is requested with exclusive access by each batch job and utility. The enqueues use SCOPE=SYSTEM to limit the enqueue to one MVS system.

Online systems do not request this enqueue before issuing RESERVEs for the RECONS.

The result of this new technique is to give access preference to online systems. They will not be forced to wait for the RESERVE requests on many batch jobs to be satisfied before their RESERVE requests are granted.



RECON Loss Notification

▲ Loss of RECON requires all IMS subsystems to reconfigure

- Switch to "spare" RECON
- Required to delete and redefine bad RECON

Availability

▲ Reconfiguration occurs on next RECON access

- RECONs may be infrequently accessed

▲ IMS V7 adds MVS console message on RECON loss

- Message identifies all subsystems using RECONs

▲ Allows automation to force reconfiguration

© IBM Corporation 2000, 2001

The loss of a RECON data set requires that all IMS subsystems using the set of RECONs reconfigure. That is, they must all discontinue using the bad RECON and switch to using the data set that was the spare. This reconfiguration occurs automatically when a subsystem next accesses the RECONs. Each subsystem must go through the reconfiguration process to deallocate the bad RECON. This RECON needs to be deleted and redefined so that it will become the new spare.

IMS V7 adds a new message when the loss of a RECON data set forces a reconfiguration of the RECONs. The message is designed to drive an installation's automation so that the reconfiguration may be done more expeditiously. This message identifies the IMS subsystems which have SUBSYS records in the RECONs and have not been abnormally terminated. This allows automation to know which subsystems need to access the RECONs to invoke the reconfiguration process which will deallocate the bad RECON. Automation could issue a /RMLIST DBRC='RECON STATUS' command to cause these accesses by online subsystems.



Very Large RECON Record Warnings

▲ IMS issues messages when RECON record becomes large

- Additional warning message
- User specified thresholds for messages
- Additional options for reaction by automation

RECON records may grow beyond the maximum size specified for them. If they do, the IMS subsystem creating the record terminates. Obviously, records of these sizes should be avoided. IMS V7 makes it easier to identify that a record may be approaching a very large size. It does this by issuing warning messages based on user defined thresholds.



Very Large RECON Record Warnings

▲ Previous releases of IMS

- Issue DSP0287W when PRILOG nears the maximum record size
 - ▶ Based on room to hold only 3 more data set entries each with 16 volumes

```
DSP0287W *WARNING* PRILOG RECORD LENGTH CRITICAL
```

- Terminates IMS (U0071 abend) when archives would cause PRILOG to exceed maximum record size

```
DSP0287I OLDS SWITCH PROCESSING TERMINATED PRILOG  
RECORD LENGTH CRITICAL
```

Previous releases issue a DSP0287W message at OLDS switch time when a PRILOG or PRISLDS record approaches its maximum capacity. The message indicates that if logging continues, additional OLDS switches could cause the PRILOG or PRISLDS record length to exceed the maximum RECON record length after all OLDS are archived. The message is sent when the record does not have room for at least three more archived data sets of 16 volumes each.

IMS terminates with a U0071 abend at OLDS switch time when the amount of space needed to record the archive data sets after all the OLDS are archived, including the OLDS being switched from and the OLDS being switched to, exceeds the maximum RECON record length. This abend is accompanied by a DSP0287I message.



Very Large RECON Record Warnings

▲ User controls warning messages with new parameters for CHANGE.RECON

LOGALERT(dsnum,volnum)

- ▶ If PRILOG record no longer has space for dsnum data set entries with volnum volumes each, issue message DSP0287W

SIZALERT(dsnum,volnum,percent)

- ▶ If PRILOG record no longer has space for dsnum data set entries with volnum volumes each, issue message DSP0387W
- ▶ If any RECON record's size exceeds percent of RECON RECORDSIZE, issue message DSP0007I

© IBM Corporation 2000, 2001

The CHANGE.RECON command has new parameters to specify thresholds for the sending of warning messages when RECON records become very large.

The LOGALERT parameters are used to control the issuing of the DSP0287W message. The defaults are LOGALERT(3,16). This produces the same actions as previous releases. Installations may cause this message to be issued earlier than it was in previous releases by using higher values.

The SIZALERT parameters control the issuing of DSP0387W and DSP0007I messages. These are new messages in IMS V7. The defaults are SIZALERT(15,16,95).

It is assumed that users will set the parameters such that the DSP0387W message is issued before the DSP0287W message. One would expect to use the same value for volnum in both the LOGALERT and SIZALERT parameters, but use a larger value for dsnum in SIZALERT than in LOGALERT.



Very Large RECON Record Warnings

▲ Messages:

- SIZALERT for dsnum,volnum exceeded:

```
DSP0387W WARNING: rrrrrrrr SIZE ALERT
DSP0387W RECORD LENGTH = lllllll, pp% OF RECORDSIZE SSSSSS
DSP0387W SSID = iiiiiiiii STARTIME = tttttttt
```

- LOGALERT for PRILOG record

```
DSP0287W *WARNING* PRILOG RECORD LENGTH CRITICAL
```

- SIZALERT for percent exceeded:

```
DSP0007I RECORD LENGTH APPROACHING RECON MAXIMUM
DSP0007I RECORD LENGTH = lllllll pp% of RECORDSIZE ssssss
formatted record key
```

© IBM Corporation 2000, 2001

This shows the text of the messages generated when the thresholds defined in SIZALERT and LOGALERT are reached.

rrrrrrr is the log record type.

The DSP0287W and DSP0387W messages are issued at each OLDS switch when the subject log record exceeds the thresholds set by the SIZALERT or LOGALERT dsnum and volnum parameters.

As with previous releases, IMS terminates with a U0071 abend at OLDS switch time when the amount of space needed to record the archive data sets after all the OLDS are archived, including the OLDS being switched from and the OLDS being switched to, exceeds the maximum RECON record length. This abend is accompanied by a DSP02871 message.

RECON records that may grow very large due to event-driven activities are:

- PRILOG - due to archiving many logs during an IMS execution without taking Image Copies of some database data sets.
- DBDS/Area - due to excessive number of EEQEs.

RECON records whose size is determined by the number of objects recorded are:

- CAGRP - 2000 members create a record of 32052 bytes
- CA - 2000 members create a record of 72096 bytes
- DBDSGRP - 2000 members create a record of 32052 bytes
- SSYS - 2000 database authorizations create a record of 64072 bytes
- PRIOLD - 235 OLDS create a record of 32760 bytes



Online Change for ACBLIB Enhancement

▲ Storage for ACBLIB online change moved above the 16M line

- BLDL used temporary storage below the line in previous releases
 - Lack of available storage in control region private could cause online change to fail
- Removes constraint for users with large numbers of databases and PSBs

Availability

Usability

Online change uses temporary storage to hold BLDL lists for the active and inactive ACBLIBs. Previous releases got this storage from below the 16M line. IMS V7 gets the storage from above the line. Some users were constrained by the previous implementation. Those were users with large numbers of databases and PSBs defined in their systems and with limited available space below the line in the private area of the control region. This enhancement removes this constraint.

Each database or PSB defined in the system definition requires 84 bytes of this temporary storage. In all releases of IMS, the storage is released when online change completes.



ACBGEN Enhancements

▲ New PSB Limits

- Maximum SENSEGs increased to 30,000
- Maximum PCBs increased to 2500
- Maximum PSB size increased to 4 Meg

Usability

▲ Generally, available ACBGEN buffer storage will be exceeded before limits are reached

© IBM Corporation 2000, 2001

IMS V7 relaxes the limits for numbers of some statements that may be specified in one PSB. Up to 30,000 SENSEG statements and 2500 PCB statements may be included in a PSB. A PSB requiring up to 4 Meg may be generated. Since a PSB with large numbers of SENSEG or PCB statements will also require large numbers of other statements, the new limitations for SENSEG or PCB statements may not be reached due to the exceeding of buffer storage available for ACBGEN.

The previous limit for SENSEG statements in one PSB was 3000. The previous limit for PCB statements was 500.

The changes will allow some unusual PSB requirements to be met. For example, an application generator which requires access to many, if not all, of the databases in an installation might require such a PSB.



Reduction of ACBGEN PSB Rebuilds

▲ IMS V7 does not rebuild PSBs for many DBD changes

■ Examples:

- ▶ Change of exit routine names
- ▶ Change of exit routine parameters
- ▶ Change in field names

Performance

Availability

▲ Shortens ACBGEN times

▲ PSB rebuilds still required for some DBD changes

■ Examples:

- ▶ Addition of exit routines
- ▶ Changes in segment descriptions

IMS V7 has eliminated some processing done by ACBGEN. When some changes are made to DBDs, IMS V7 does not rebuild the PSBs which reference the databases. For example, one can change the segment/edit compression exit used to compress a segment without rebuilding all of the PSBs which reference the database. Similarly, one can change the randomizing parameters, such as the number of blocks in the root addressable area for an HDAM database without rebuilding PSBs. This will greatly reduce the time required for ACBGENs for these types of changes. Rebuilds of PSBs are still required for some DBD changes, such as the addition of an exit routine type or changes in segment descriptions.



IMS Control Region Resource Manager

▲ **Ensures that all subordinate address spaces are properly shutdown**

- *Even if* the control region terminates without resource cleanup

▲ **Loaded during IMS control region initialization**

- Utilizes the MVS service RESMGR
- Runs in the Master Scheduler address space

▲ **Receives control at IMS control region address space termination**

- Checks to see if DLISAS and DBRC have terminated properly
 - ▶ Abnormally terminates any surviving DLISAS or DBRC region
 - ▶ Issues a message to alert the operator

© IBM Corporation 2000, 2001

IMS utilizes an MVS service called RESMGR to establish an address-space level resource manager that can monitor the control region address space for termination. This new IMS resource manager is added during control region initialization and is passed the address of a parmlist which contains information about each subordinate address space. The resource manager runs in the master scheduler address space and always gets control at IMS control region address space termination. The resource manager checks to see if the DLI and DBRC address spaces have been properly terminated. If not, the surviving regions are abnormally terminated and an alert message is sent to the operator.



IMS Control Region Resource Manager

▲ New message

```
DFS0799E rgntype REGION jobname BEING TERMINATED AT CONTROL  
REGION END OF MEMORY
```

▲ Recommended actions when message is received:

- Check for other problems in the system
 - ▶ Storage shortage or resource-related problems
 - ▶ May require an IPL

© IBM Corporation 2000, 2001

The situation which causes IMS to abend without completing resource cleanup is an unusual condition. This may occur when the control region address space is FORCED down or experiences memory termination due to a storage shortage. If message DFS0799E is received, make sure that the DLISAS and DBRC regions have actually terminated before restarting IMS. Also, be aware that if the problems were initially caused by a storage shortage or some other MVS resource-related problem that some IMS-allocated resources, such as common storage, may not have been freed from the prior IMS instance and may require an IPL to recover or release these resources.



DB2 Attach Enhancements

▲ Enhanced Trace entries

- Additional information on existing entries and new entries
- Improves the ability to analyze problems between an external subsystem, such as DB2, and the IMS control region or dependent region

▲ Generation of X'67FF' Snap trace record when the external subsystem attach facility detects a deadlock

- Deadlock report (DFSERA30)
- Provides a consolidated report for deadlocks (IMS and DB2)

▲ DB2 Group Attach Support

- Allows attachment to any DB2 in a group

© IBM Corporation 2000, 2001

The External Subsystem Attach Facility (ESAF) traces have been improved to provide better information about the activities that occur between IMS and the External Subsystem. Typically an external subsystem is DB2. It is MQSeries when the MQSeries IMS Adapter is used.

Additionally, a X'67FF' SNAP record will now be written whenever the IMS receives a deadlock indication from DB2. A deadlock report can be produced using exit DFSERA30 out of program DFSERA10.

Support for DB2 Group Attach has been added.

The following pages explain these enhancements in greater detail.



External Subsystem Trace

▲ Additional information on trace entries

```
/TRACE SET ON TABLE SUBS OPTION LOG
```

Same command as previous releases

- Size increase
 - ▶ Common identifier (UOW recovery token) in related entries
 - IMS identifier (8 bytes of IMSID plus blanks or batch jobname),
 - IMS PSB schedule number (4 bytes),
 - Number of commits in this scheduled PSB (4 bytes)

▲ New entries

- Documented in the Diagnosis Guide and Reference

▲ Modified points at which trace entries are written

- After (instead of before) an event, such as sign on

© IBM Corporation 2000, 2001

The ESAF trace continues to be invoked by the command: "/TRA SET ON TABLE SUBS OPTION LOG". The Log keyword causes the trace records to be written as X'67FA' records.

Enhancements to the trace include:

- Additional information is in the trace entries. Their size has increased from 4 to 8 words. All entries associated with a particular Unit of Work (UOW) now carry a recovery token. This is the same identifier that DB2 uses. This makes it easier to correlate IMS trace information with DB2 information.
- There are new trace entries. These records provide a more complete view of the events that occur during execution of the UOW. These are documented in the Diagnosis Guide and Reference.
- Trace entries are now always written after the event occurs. In previous releases they were sometimes written before an event occurred. Signon was one of these cases.



External Subsystem Trace

FUNCTION	WORD 0	WORD 1	WORD 2	WORD 3	WORD 4	WORD 5	WORD 6	WORD 7
	WORD 8	WORD 9	WORD 10	WORD 11	WORD 12	WORD 13	WORD 14	WORD 15
ESS3 LOGGING	5700758B	00160014	E2E2D6D7	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
ESI6 XS R-I-D	57007594	04040042	E2E2D6D7	00008C00	00000058	00000000	00000000	00000000
	C9D4E2F1	40404040	00000004	00000000	C1C20000	00000000	00000000	00000000
ESI3 R-I-D	570075A1	04020021	E2E2D6D7	00008C00	00000058	00000000	00000000	00000000
	C9D4E2F1	40404040	00000004	00000000	C1C20000	00000000	00000000	00000000

UOW Recovery token -

- IMS identifier (8 bytes) - IMSID plus blanks or batch jobname
- number of scheds for this PSB (4 bytes)
- number of commits in this sched (4 bytes)

WORD 1 entries (documented in Diagnosis Guide and Reference):

- 00160014 - ESS logging exit
- 04040042 - Control region ESS resolve in-doubt exit
- 04020021 - Control region resolve in-doubt

This page provides a set of sample trace entries from an IMS V7 system. The primary enhancements that can be seen are as follows: trace entries are all 16 words; associated trace entries all carry the same UOW recovery token and there are a few new entries.

All the trace entries are documented in the Diagnosis Guide and Reference.



DB2 Deadlock Information

▲ External subsystem (DB2) detected deadlocks

- Logged in X'67FF' records
- Reported by DFSERA30 Deadlock reports

DEADLOCK

EXTERNAL SUBSYSTEM DB2A DETECTED A DEADLOCK DURING NORMAL CALL

REGION TYPE : MPP

REGION NUMBER : 0001

JOB NAME : SQT03AA

PSB NAME : DCSQL6D

SMB NAME : TXSQL6D

RECOVERY TOKEN: C9D4E2F1404040400000000400000000

PST ...

DLI ACCT ...

PSB DIR ...

When IMS receives a return code from DB2 which indicates a deadlock, IMS writes the X'67FF' SNAP trace record. This record includes both IMS and DB2 deadlock information,

Program DFSERA10 with a specification of exit DFSERA30 prints the deadlock report. In previous releases this only produced information about IMS deadlocks, not DB2 deadlocks. With IMS V7 the report also includes information about DB2 deadlocks. The information includes the recovery token. This may be used to find more information about the situation in IMS and DB2.

DFSERA10 with exit DFSERA70 (parm specification of TOKEN=) can be used to print IMS log records with the specified token.

When DB2 detects deadlocks, DB2 prints out messages that provide the information about the resource. The recovery token is included in this information,



DB2 Group Attach

▲ IMS online system may attach to any DB2 using the Group Name

- Requires DB2 V7 and PQ42180 for IMS V7
- DB2 group attachment name is specified in DB2 IEFSSNxx parmlib member
- DB2 group name is specified on the SSN= parameter in the IMS SSM member in IMS's PROCLIB (instead of DB2 subsystem name)

▲ Multiple DB2 subsystems may use the same Group Name

- IMS may be moved to any system with a DB2 using this Group Name

▲ Restart 'resolve in-doubt' processing automatically uses the specific name

- Emergency restart must have 'original' DB2 available

© IBM Corporation 2000, 2001

DB2 Group Attach is supported by IMS V7. It requires DB2 V7 and PQ42180 for IMS V7. Group Attach allows an IMS system to attach to a DB2 using a generic name. That is, IMS does not use the specific name of the DB2 to which it is connecting. Instead, it uses the Group Name. This allows IMS systems to be more easily moved among MVS systems where different DB2s are running. Of course, the DB2s should be able to do the same work. That is, the DB2s using the same Group Name should be in a DB2 data sharing group.

If there is a failure, emergency restart is required. When the restart is done, IMS must connect to the same instance of DB2. This is required to resolve any in-doubts. In this case, IMS emergency restart will use the specific name of the DB2 with which it was previously connected. This means that that DB2 must be available on the same MVS with the restarted IMS.



IMS Monitor and IMS Performance Analyzer

▲ Tracing of Fast Path added by IMS Monitor

- DEDBs, MSDBs, EMH, and IFP regions

Usability

▲ IMS monitor subsetting added

- Limit tracing to a set of databases or regions
- Limit tracing to a time interval

Performance

▲ IMS Performance Analyzer reports Fast Path activity

- IMS Monitor report program does not report Fast Path activity
 - IMS Monitor report program is not being enhanced

▲ Benefits

- More complete performance information
- More manageable reports
- Less impact by monitor tracing

© IBM Corporation 2000, 2001

IMS monitoring has been enhanced in three ways.

First, tracing of Fast Path activity has been added. This includes calls and I/Os for DEDBs and MSDBs as well as EMH and IFP region activity.

Second, tracing subsetting has been added. Tracing can be limited to only a set of databases or regions. This may greatly reduce the size of reports generated by the monitor. For example, if an installation wants to address a problem with a BMP, it need only trace and report on the BMP. This could reduce the size of a report from 1,000 pages to a handful.

Third, when starting a trace, the requesting command can include a time interval. At the end of the interval, tracing will be stopped.

The IMS Performance Analyzer product reports Fast Path activity. The IMS Monitor report program which comes with IMS will not report this activity. This report program has not been enhanced to process the new Fast Path trace records.



OSAM 64-bit Real Storage Support

- ▲ **OS/390 V2R10 and z/OS support 64-bit real addresses on z900**
 - Addresses above the 31-bit address limit of 2 gigabytes are "above the bar"
- ▲ **OSAM supports real addresses above the 2 gigabyte "bar"**
 - I/Os may be done with buffers above the bar
 - Data in buffers do not have to be moved below the bar before I/Os are done
 - Database buffers may reside in real storage above the bar
 - Log buffers may reside in real storage above the bar
 - Delivered by PQ42127
- ▲ **In previous releases of IMS, data in buffers which are above the bar must be moved below the bar before I/O is done**
 - Performance cost

© IBM Corporation 2000, 2001

The z900 processors have 64 bit real addressing support. This means that real storage may be greater than 2 gigabytes which was the 31 bit limitation. This greatly increases the real storage available to a processor.

We still address virtual storage with 31 bit addresses. The operating system may store this virtual storage with a 64 bit address real address. Virtual storage management handles this translation. On the other hand, I/O is done using real addresses. OSAM in IMS V7 has added support for these 64 bit real addresses. This means that OSAM can do I/O with buffers above the 2 gigabyte "bar". Access methods which do not have this kind of support must do I/O with buffers which are below the 2 gigabyte addressing limit. If they have data above the 2 gigabyte bar for real storage, they must move it to a real storage location below this bar. This movement of data has a performance cost.

IMS uses OSAM for databases and logs. With IMS V7, OSAM may place the buffers for OSAM database buffer pools and for logs above the 2 gigabyte bar.



Enterprise Storage Server (Shark) Support

▲ ESS allows concurrent I/Os to the same volume

- Parallel Access Volumes (PAV) provide support for concurrent I/Os from one OS/390 or z/OS system
- Multiple Allegiance provides support for concurrent I/Os from different systems
- Concurrent reads to any data on a volume
- Concurrent write must be to a different "extent" (except for OSAM)
 - ▶ Extent is set by access method's define extent command

▲ OSAM allows concurrent reads and writes to the same extent!

- OSAM provides its own serialization (database locking)
- PQ37003 for IMS V7, PQ37020 for IMS V6

▲ No need to spread data across volumes to avoid I/O bottlenecks

© IBM Corporation 2000, 2001

The Enterprise Storage Server, sometimes known as Shark, allows concurrent I/Os to the same volume. This has not been supported with previous DASD control units. ESS has two capabilities that support these concurrent I/Os. Parallel Access Volumes (PAV) allows concurrent I/Os from the same OS/390 or z/OS system. PAV also requires support in the operating system so that it will attempt these concurrent I/Os. This is done by creating multiple UCBs for the same device. Multiple Allegiance (MA) allows concurrent I/Os from different systems or LPARs.

When PAV or MA is used, concurrent reads to the same volume are always allowed. There are restrictions on writes. Integrity requires that a write of a block and a read or another write to the same block cannot occur at the same time. For this reason, writes cannot occur to an extent while another I/O is also being done to the extent. In general, an extent is not the same as the extent which is used for allocation of space in a data set. The size of the extent is determined by the access method. It is set by the DEFINE EXTENT command issued by the access method. OSAM is the exception to this processing. OSAM allows writes to occur concurrently with other I/Os to the same extent. Other protocols within IMS, such as database locking, ensure that while an OSAM block is being written, a concurrent I/O will not be attempted. This support is provided by the listed APARs for both IMS V7 and IMS V6.

The support for concurrent I/Os has an important performance implication. When using PAV and MA it is not necessary to spread data across multiple volumes to avoid I/O bottlenecks. Such bottlenecks are produced because of the serialization of I/Os. This serialization is eliminated by PAV and MA.



Database Enhancements



▲ Database Enhancements

- HALDB
- ORS
- Other Database Enhancements



High Availability Large Database (HALDB)



HALDB (High Availability Large Database)

▲ Large Database

Up to 10,010 data sets per database!

Greater than 40 terabytes

- Databases are partitioned
 - ▶ Up to 1001 partitions per database
 - ▶ Partitions have up to 10 data set groups

▲ High Availability Database

- Partition independence
 - ▶ Allocation, authorization, reorganization, and recovery are by partition
- Self healing pointers
 - ▶ Reorganization of partition does not require changes to secondary indexes or logically related databases which point to it

© IBM Corporation 2000, 2001

IMS V7 introduces a new capability for full function databases. This is High Availability Large Database (HALDB). HALDB databases have up to 1001 partitions. Each partition has up to 10 data set groups. This gives HALDB up to 10,010 data sets per database. Each of these data sets may be up to 4 gigabytes. So, the limit is 40 terabytes per database.

HALDB provides two availability benefits. First, partitions are managed independently. Each partition in a database may be allocated, authorized, reorganized, and recovered independently. Second, the reorganization of a partition does not require utilities to update the pointers in secondary indexes and logically related databases which point to the reorganized data. Even though the reorganization moves segments, pointers to those segments are not updated by the reorganization process. Instead, these pointers are updated as needed. This is a "self healing" process. This combination of capabilities can greatly reduce the windows required for database maintenance. Multiple partitions allow users to reorganize and image copy smaller amounts of data. This takes less time. The reorganizations may be done in parallel as can the image copies. Since pointers are self healing, there is no need for utilities such as Prefix Resolution and Prefix Update to correct pointers. This also reduces the time required for reorganizations.

The use of HALDB is optional. The non-HALDB databases of previous versions of IMS remain available with IMS V7.



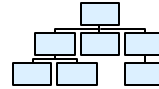
Highlights

▲ New database types

- PHDAM - partitioned HDAM
- PHIDAM - partitioned HIDAM
 - Index is also partitioned
- PSINDEX - partitioned secondary index

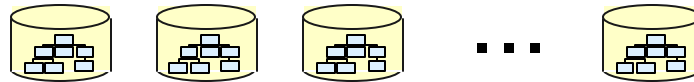
▲ Hierarchic structure is maintained

- A database record resides in one partition



▲ Partition selection

- By key range or by user exit routine



© IBM Corporation 2000, 2001

HALDB introduces three new full function database types, partitioned HDAM (PHDAM), partitioned HIDAM (HIDAM), and partitioned secondary index (PSINDEX). As the names imply, these are partitioned versions of the corresponding database types for non-HALDB databases. PHIDAM includes its index which is also partitioned.

HALDB databases have the same hierarchic structure that is used for other full function databases. With HALDB a database record, which is a root segment and all of its dependents, resides in one partition.

Partitioning may be done either by key range or by a user written exit routine. Either method may be used with each of the three database types, PHDAM, PHIDAM, and PSINDEX.



Highlights

▲ Logical relationships and secondary indexes are supported

- Secondary indexes may be partitioned

▲ OSAM and VSAM (ESDS and KSDS) are used

▲ DBRC is required

- Databases must be registered
- Dynamic allocation from DBRC information, not DFSMDA

▲ Minimal (or no) application changes required

- Initial load cannot insert logical children (must be added by update)
 - New status code for load programs
- 'Data unavailable' conditions apply to partitions
 - Database may be available, but partition unavailable

© IBM Corporation 2000, 2001

HALDB as complete full function database capabilities. This includes support for logical relationships and secondary indexes.

Like non-HALDB databases, HALDB databases use OSAM, VSAM ESDS, and VSAM KSDS data sets.

HALDB databases must be registered with DBRC. In fact, the definition process stores information about partitions in the RECONS. Since the information about HALDB data sets must be stored in the RECONS, this information is used for dynamic allocation. DFSMDA members cannot be used.

Most users will be able to convert non-HALDB databases to HALDB without any application program changes. Initial loads of HALDB databases cannot insert logical children. There are two choices for handling these programs. Either the logical children may be added by update program which is run after the initial load or the "initial load" program may be executed with PROCOPT=I instead of PROCOPT=L. If an attempt to insert a logical child is made by initial load (PROCOPT=L), a status code 'LF' is returned for the call.

If you choose to make some partitions unavailable while others remain available, this could have application considerations. Application programs can know if a database is available or not, but they are not aware of the availability of partitions. Of course, this only applies if you change the way you operate by making some partitions unavailable while continuing to process others.



Direct and Indirect Pointers

▲ HALDB uses both direct and indirect pointers

- Combination of pointers are used for logical relationships and secondary indexes
- Direct pointers are relative byte addresses (RBAs) of target segments
- Indirect pointers "point" to Indirect List Entries (ILEs) in Indirect List Data Set (ILDS)
 - ▶ Indirect pointers are keys of ILEs (Indirect List Keys (ILKs))
 - ▶ An ILK is a token associated with the target segment
- ILEs contain direct pointer to segment
 - ▶ Updated only by reorganizations
- ILDS is a KSDS associated with a Partition

© IBM Corporation 2000, 2001

HALDB uses a combination of direct and indirect pointers for logical relationships and secondary indexes. The direct pointers are relative byte addresses (RBA) which point to segments. Indirect pointers "point" to Indirect List Entries (ILEs). These indirect pointers are actually keys of these ILEs. The ILEs are stored in an Indirect List Data Set. These ILDSs are new with HALDB. Each partition has an ILDS. The ILDS is updated by reorganizations. It contains direct pointers to the new locations of the segments after the reorganization. The ILDS is a VSAM KSDS.



Extended Pointer Set

▲ Extended Pointer Set (EPS) is used for logical relationships and secondary indexes

- Replaces direct or symbolic pointer used in Non-HALDB databases
- Key of root is used to determine partition
- EPS contains direct pointer, indirect pointer, reorganization number, and identification of partition
 - If partition is correct and reorg number is current, direct pointer is used
 - If partition is wrong or reorg number is not current, indirect pointer is used
 - Indirect pointer points to Indirect List DS (ILDS) containing pointers from last reorg
- EPS is not updated by reorganizations!
- Direct pointer in EPS is updated when indirect pointer is used

▲ Self healing pointers!

© IBM Corporation 2000, 2001

An Extended Pointer Set (EPS) is used for logical relationship and secondary index pointers. It replaces the direct or symbolic pointer used with non-HALDB databases. The EPS has two "pointers". One is the direct pointer. The other is the indirect pointer. The EPS also contains information that is used to determine if the direct pointer is accurate. The direct pointer is the RBA of the target segment. It could be out of date. That is, it could point to the location of the target segment before the last reorg. The indirect pointer is the key of the Indirect List Entry (ILE) in for the target segment. The EPS also contains a reorganization number and partition ID. These were the partition ID and the partition's reorganization number when the direct pointer was up to date. If a reorganization has been done since this direct pointer was created or updated, the partition ID and/or the reorganization number will have changed. This information is used to determine which pointer to use.

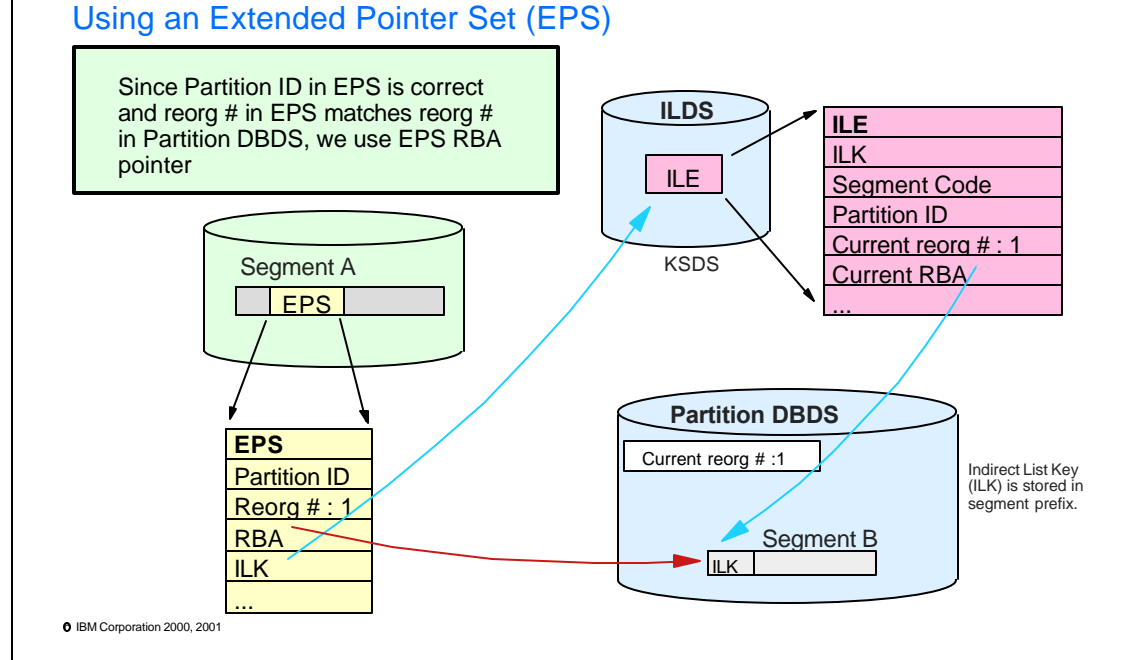
When following a pointer, IMS first determines the partition in which the target segment currently resides. It does this by using the key of the target segment's root. This key is always stored in the segment with the EPS. IMS then compares the target partition's ID and reorg numbers with those stored in the EPS. If they match the direct pointer is used. If they both do not match, the indirect pointer is used to find the target segment's ILE in the ILDS.

The EPS is not updated by reorganizations. Reorganizations update the ILEs in the ILDS. The direct pointer in the EPS is updated when the EPS is used. We will see how this "self healing" process works in the example which follows.



Self-Healing Pointers

Using an Extended Pointer Set (EPS)



In this example segment A has a pointer to segment B. This pointer is either for a logical relationship or a secondary index. That is, segment A is either a logical child or a secondary index segment. Segment B is the target of the pointer. The Extended Pointer Set (EPS) is in the prefix of segment A.

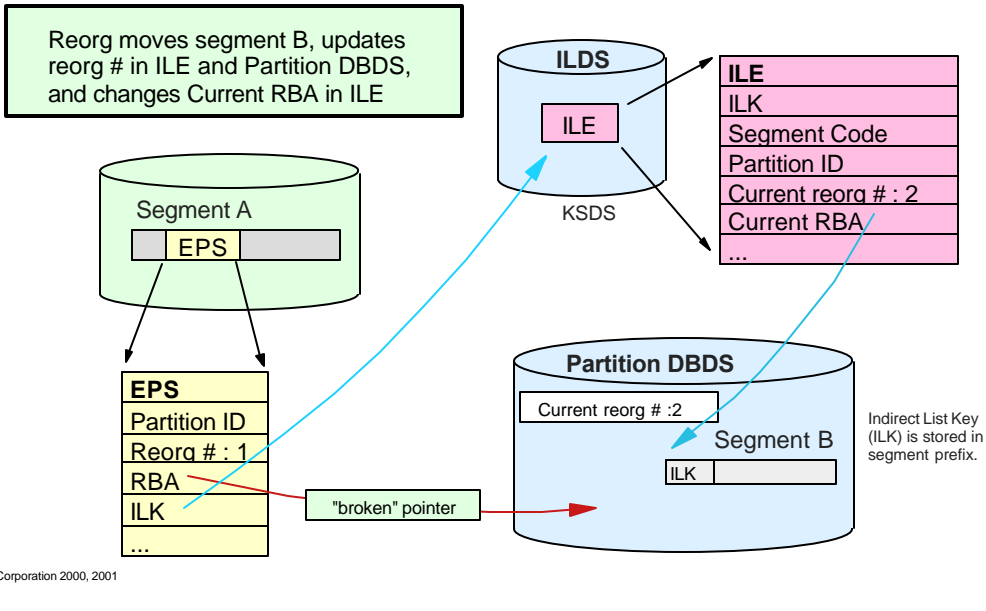
The EPS contains a direct pointer to segment B. The direct pointer is a Relative Byte Address (RBA). The EPS also contains the partition ID of the partition where segment B resides. The EPS contains a reorganization number for the partition where segment B resides. In this case it is 1. The reorganization number is also stored in the partition database data set. When IMS opens the partition, it keeps the reorg number in its control blocks for the partition.

When the pointer from segment A to segment B is used, IMS first determines the partition where the segment currently resides. It does this by using the root key to find the partition. (With HALDB, segments with EPSs also contain the key of the target segment's root.) If this partition matches the partition ID stored in the EPS, the reorg number in the EPS is compared to the reorg number from the partition DBDS. If they are the same, the direct pointer (RBA) in the EPS is used to go directly to segment B and the ILDS is not used. In this case, the reorg numbers are both 1, so the direct pointer is used.



Self-Healing Pointers

After reorganization of Partition



Reorganizing a partition changes the location of its segments. It also updates the ILEs for these segments in the ILDS and updates the reorg number stored in the partition DBDS. It does not update the pointers in EPSs. The ILE in the ILDS is easily found by the reload utility. The ILK, which is the key of the ILE, is stored in the prefix of the segment.

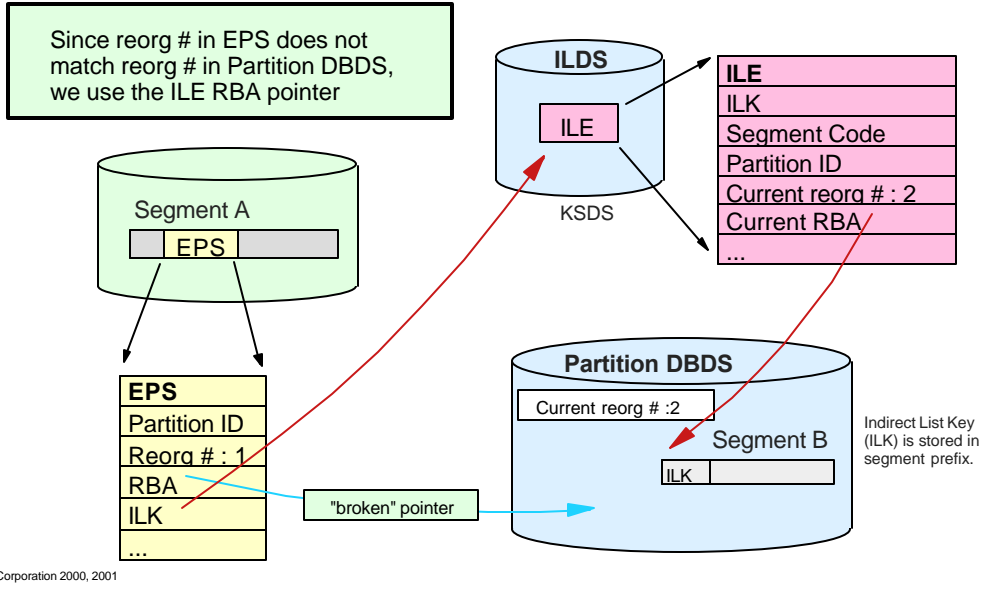
In this example, segment B is moved and its ILE is updated with an RBA which points to the new location of B. The reorg number in the partition DBDS is also updated.

Since the EPS in segment A is not changed, its RBA does not point to the new location of segment B. It is now a "broken" pointer. The reorg number in the EPS remains 1.



Self-Healing Pointers

Using the EPS after the reorganization

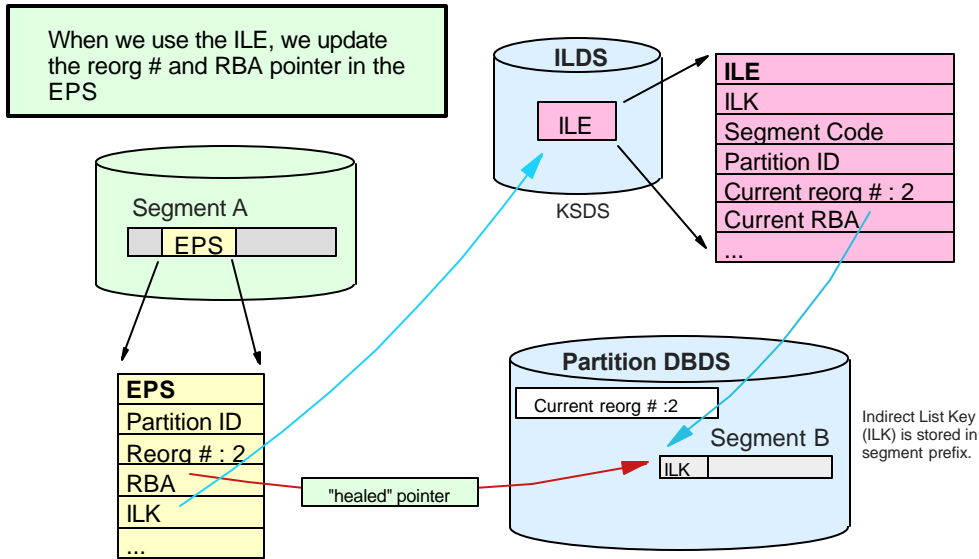


An IMS call which needs to use the pointer from segment A to segment B will discover the "broken" pointer. IMS knows that the RBA pointer should not be used because the reorg number in the EPS does not match the reorg number in the partition DBDS. Instead of using the RBA, IMS will use the ILK in the EPS to find and read the ILE for segment B in the ILDS. The ILE contains a correct RBA pointer. It was updated by the reorganization.



Self-Healing Pointers

"Healing" the EPS



© IBM Corporation 2000, 2001

When an ILE is used, IMS updates the direct pointer (RBA) and the reorg number in the EPS. This allows future uses of the EPS to avoid the overhead of referencing the ILDS. When this "healing" process completes, the EPS in segment A contains reorg number 2 and the RBA of segment B's new location. This update to the EPS is done only by programs which reference segment A with an update PROCOPT.



Reorganizations with HALDB

▲ Reorganizations may be shortened for three reasons:

- Partitions may be reorganized in parallel
- Logical relationship pointers are not updated
- Secondary indexes are not rebuilt

▲ Reorganization times may be reduced to meet your window

- Reorg time for the database is the longest reorg time for a partition
- Reorg time for a partition is shortened by making the partition smaller
- Partitions are made smaller by splitting the partitions into multiple partitions

▲ You can manage to your reorg window time!

- If you only have 15 minutes, make your largest partition small enough to be reorganized in 15 minutes

© IBM Corporation 2000, 2001

One of the greatest benefits of HALDB is capability it gives you to shorten your reorganization times. HALDB shortens reorg times in three ways. First, partitions may be reorganized in parallel. Second, logical relationship pointers do not have to be updated. We do not have to run Prefix Resolution, Database Scan, or Prefix Resolution to update logical relationship pointers. These steps are eliminated from the reorganization process. Third, a reorganization of a database does not require that we rebuild all of the secondary indexes that point to it. With non-HALDB databases these rebuilds are required in order to get the correct direct pointers. With HALDB these steps, such as HISAM Unload and HISAM Reload or an Index Build tool, are eliminated from the reorganization process.

HALDB allows users to reduce their reorganization times to meet the time allowed to do the reorgs. When partitions are reorganized in parallel, the reorg time for the database is the longest time required for reorganizing a partition. Users can shorten the reorg time for a partition by making the partition smaller. This is done by splitting the partition into multiple smaller partitions. So, users can manage to the time available for reorganizations. If you only have a 15 minute window, create your partitions small enough so that each can be reorganized in the available time.

You are in control!



HALDB Support

▲ HALDB is supported with:

- Data sharing
- Remote Site Recovery (RSR)
- Extended Recovery Facility (XRF)
- Online Change
- OSAM Sequential Buffering
- IMS Monitor and IMS Performance Analyzer
- ...

© IBM Corporation 2000, 2001

HALDB has the same support that non-HALDB full function databases have. That is, they may participate in data sharing, RSR, XRF, and online change. HALDB database data sets may use OSAM sequential buffering and VSAM hiperspace buffers. HALDB activity is monitored by the IMS Monitor and reported by the IMS Monitor and the IMS Performance Analyzer product.



HALDB

▲ Database candidates for HALDB

- Very large databases
 - ▶ Approaching 4G (VSAM) or 8G (OSAM) limitations
 - ▶ Theoretical limit is now over 40 terabytes

- Medium and large databases
 - ▶ Parallel processing to meet time deadlines
 - ▶ Less frequent reorganizations due to more free space

- Any size database
 - ▶ More frequent reorganizations
 - ▶ Making only parts of the data unavailable for database maintenance

© IBM Corporation 2000, 2001

The most obvious candidates for HALDB are very large databases. Databases which are approaching the data set size limits of 4GB for VSAM or 8GB for OSAM have their size restrictions removed when they are converted to HALDB.

Medium and large sized databases with time deadlines for batch executions may benefit from the use of parallel processing against multiple partitions. They also may benefit from having more free space. Since there is almost unlimited space available in a HALDB database, greater free space may be used. This could eliminate the need for many reorganizations.

Even relatively small databases are candidates for HALDB. Since reorganizations may require less time, they may be done more frequently. Installations that can benefit from removing only parts of a database for maintenance processing, will take advantage of HALDB.



HALDB

▲ Benefits

- Greater database capacity
 - Without application changes
- Increased database availability
 - Partitions, not databases, are removed from system
 - Shortened reorganization process
 - Batch window is shortened with concurrent processing
- Improved manageability
 - Data sets may be smaller

Performance

Availability

Usability

HALDB benefits users in several ways. It allows databases to become larger without requiring changes to application programs. Availability is increased by multiple means. HALDB allows partitions, not entire databases to be removed from systems for database maintenance purposes. Reorganization times are shortened by providing for concurrent reorganizations of partitions and eliminating utility step requirements to update logical relationship and secondary index pointers. Since partitions may be processed in parallel, HALDB may reduce batch window time requirements. Finally, HALDB eliminates the need to have very large data sets. This improves their manageability.



Online Recovery Service (ORS)



Online Recovery Service (ORS)

▲ Online recovery of databases

- Logs are read once for all database data sets
- Parallel reads of inputs
 - ▶ Image copies, change accumulations, and logs
- Parallel writes of outputs
 - ▶ Databases recovered in parallel
- Change accumulation not required for data sharing
- Time stamp recovery to any time

▲ A separate product (5655-E50)

- IMS V7 is a prerequisite

© IBM Corporation 2000, 2001

Online Recovery Service (ORS) is a new feature in IMS V7. It is used to recover database data sets using an online system. ORS reads its inputs only one time. Logs and Change Accumulation data sets may be needed as inputs for the recovery of each of multiple database data sets. When ORS recovers these data sets, it reads these inputs only once. ORS reads inputs in parallel. That is, multiple read processes, such as the Image Copy and the Change Accumulation data set for a database recovery may be read at the same time. Similarly, the Image Copies for multiple data sets may be read concurrently. Databases data sets are also recovered in parallel. ORS does not require Change Accumulation input for recovering databases that have been updated with block level data sharing. It can merge log records from multiple IMS subsystems. This eliminates the Change Accum requirement. These techniques can greatly shorten the time needed to recover a set of IMS databases.

ORS supports time stamp recovery to any time. It will recover all updates to the database that have been committed up to the specifies time. Quiescing the database with a /DBR command is not required.

ORS is a separate product. It is not part of IMS V7 but requires IMS V7. It cannot be used with previous IMS releases.



Online Recovery Service

▲ A control region function

- Invoked by IMS commands
- Executes in parallel with other online activity
- Uses Remote Site Recovery (RSR) techniques
 - ▶ Merges updates without requiring Change Accumulation

▲ Supports

- HDAM, HIDAM, HISAM, SHISAM, Index
- PHDAM, PHIDAM, PSINDEX
- DEDB

© IBM Corporation 2000, 2001

ORS is a control region function. It uses other address spaces, such as DLI SAS and DBRC, but is controlled by commands processed by the online control region. ORS recoveries execute in parallel with other online activity such as transactions and BMPs. ORS is based on Remote Site Recovery techniques. For example, it reads logs in parallel and merges records from them. Like RSR, it does not require Change Accum, even with input from data sharing subsystems.

ORS recovers the same databases that the batch Database Recovery utility recovers. These are HDAM, HIDAM, HISAM, SHISAM, Index, and DEDB databases, as well as, the HALDB types of PHDAM, PHIDAM, and PSINDEX.



Online Recovery Service

▲ Recovers a list of database data sets

- List is built by /RECOVER ADD xxx commands
 - Command may add a data set, a database, or a group

▲ Recovery of list is started by command

- /RECOVER START command
- Database, partition, or area must be deallocated (/DBR) before recovery

▲ Option to /START database, partition, or area after recovery

- /START GLOBAL may be specified for data sharing

© IBM Corporation 2000, 2001

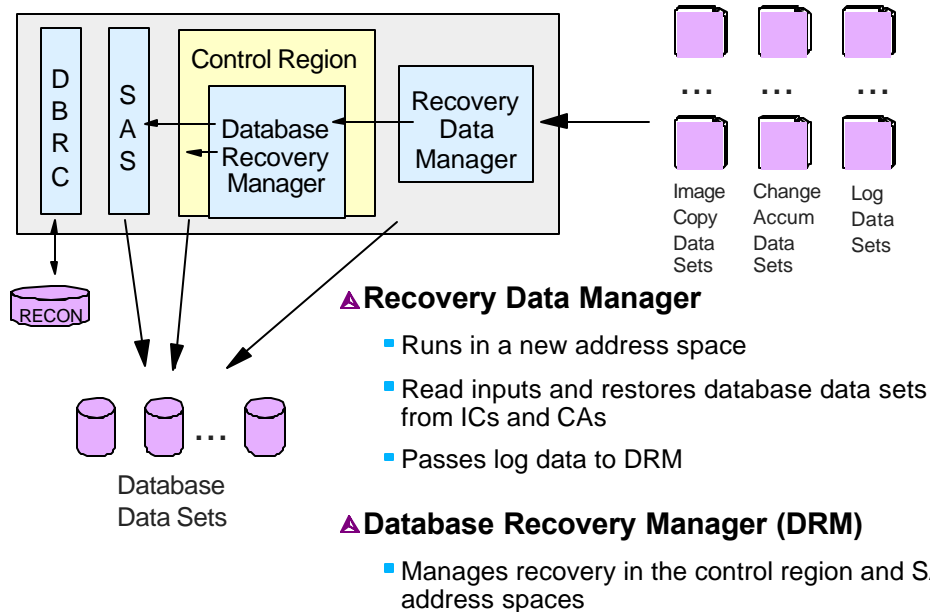
ORS can recover multiple database data sets concurrently. The user builds a list of data sets to recover and starts the recovery of the list. These actions are done with the new /RECOVER command. The list is built with the /REC LIST command. The user may specify a data set, a database, a HALDB partition, or a group in this command. The group may be any of the groups defined to DBRC. These include database data set groups, database groups, Change Accum groups, and recovery groups. Recovery groups are new with IMS V7.

The recovery is started with a /REC START command. The databases, partitions, or areas must be deallocated from all systems before recovery may begin. This includes the system on which the recovery is executing. ORS gets its own special authorization for recovery. It does not use the authorization for its online system.

The user has the option of having ORS automatically issue a /START DB command for a database after the recovery completes. The /START DB may include the GLOBAL parameter which is used with data sharing.



ORS Components and Flow



© IBM Corporation 2000, 2001

This is an illustration of ORS components. When ORS is invoked, it starts the Recovery Data Manager (RDM) address space. RDM reads all inputs to the recoveries. It merges Image Copies with their Change Accum data and restores them to the database data sets.

RDM passes log records to the Database Recovery Manger (DRM). DRM runs as part of the control region. DRM passes the log records to the full function tracker and the Fast Path tracker. The full function tracker runs in the DLI SAS address space. It uses the full function buffer pools to apply updates to the databases being recovered. The Fast Path tracker runs in the control region. It has its own buffers which it uses to apply updates to the areas being recovered.



Faster Recoveries

- ▲ **Multiple database data sets are recovered in parallel**
- ▲ **Logs and CA data sets are read once for all database data sets**
 - Some data is saved in data spaces for later use
- ▲ **Parallel reads of inputs**
 - Image copies, change accumulations, and logs
- ▲ **Parallel writes of outputs**
 - Databases recovered in parallel
- ▲ **Change accumulation not required for data sharing**
 - Database change log records are merged in ORS process

© IBM Corporation 2000, 2001

One of the greatest benefits of ORS is its recovering of multiple database data sets in parallel.

Almost all database recoveries involved multiple databases and multiple database data sets. Recoveries are typically done to recover from operational or programming errors. This means that more than one data set is usually being recovered. ORS processes are designed to be most efficient when recovering multiple data sets.

The same logs and change accumulation data sets are typically used for the recovery of more than one data set. ORS reads these logs and change accumulation data sets only once. If required, it may save records in data spaces until they are needed for later processing.

ORS reads its inputs in parallel. The user may specify the number of concurrent read processes used. This eliminates much of the read delay time in a recovery.

ORS writes its updates in parallel. Multiple database data sets are recovered simultaneously.

ORS does not require the running of Change Accumulation, even when data sharing has been used. ORS can merge data sharing logs as it processes them.

These techniques lead to significantly faster recovery times than would occur with standard database recovery techniques.



Time Stamp Recovery

▲ Two types of time stamp recovery

1. Time stamp recovery (not point-in-time)
 - ▶ Recovery to DBRC ALLOC record boundary
 - ▶ Like non-ORS time stamp recovery
2. **Point-in-time Recovery (PITR)**
 - ▶ Data sets can be recovered to any time
 - ▶ New with ORS!

ORS has improved time stamp recovery capabilities. It supports two types of time stamp recoveries. First, there is the same type of time stamp recovery that may be done with the Database Recovery utility. This is recovery to a DBRC ALLOC record boundary. That is a time at which the database was not authorized to any subsystem. Second, it is a point-in-time recovery (PITR). PITR may be done to any time. Even one when the database was being updated. This is a new capability with ORS.



Point-in-Time Recovery (PITR)

▲ Data sets can be recovered to any point in time

- Not restricted by allocation boundary
- May be a time at which the database was being updated
- **Databases do not have to be /DBRed to create recovery points!**

▲ All updates that are committed as of the specified time stamp are applied

- Uncommitted updates are not applied

© IBM Corporation 2000, 2001

PITR does not limit recoveries to DBRC ALLOC record boundaries. They may be done to any time. This includes times at which the database was being updated. Many installations regularly quiesce their databases by issuing /DBR commands for them. Often, this is done once per day. This creates a recovery point to which the databases may be later recovered. These /DBR command are not required when ORS PITR is used.

PITR may be used to increase database availability by eliminating /DBRs of databases which are used merely to create recovery points.

When a PITR is done, all updates that had been committed at the time stamp are recovered. Those that had not been committed at the time stamp are not recovered.



Online Recovery Service

▲ Benefits

- Faster recoveries
 - ▶ Inputs read once for all database data sets

- Change Accumulation requirement eliminated
 - ▶ Especially beneficial for data sharing users
 - ▶ Change Accum may be used, but is not required

- Time stamp recoveries to any time
 - ▶ Creation of "recovery points" no longer required

Performance

Usability

Availability

ORS has several benefits.

It recovers sets of database data sets faster than multiple executions of the Database Recovery utility would do. When recoveries are done, they usually involve multiple databases. For time stamp recoveries, all related databases and their indexes must be restored to the same time. Since most recoveries involve multiple data sets and the same log and change accumulation inputs, ORS's reading of these inputs only once provides significant time savings.

Change Accumulation may be used with ORS, but it is never required. Without ORS, data sharing users must run Change Accumulation before they can begin the Database Recovery utility. ORS eliminates this time consuming step.

ORS allows databases to be recovered to any time. "Recovery points" do not have to be created to do these time stamp recoveries. A recovery point is created by quiescing all update activity to a database. This is done with either a /DBR or /DBD command. This makes the database unavailable across the recovery point. ORS eliminates this need for deallocating the database from online systems.



Other Database Enhancements



VSAM REUSE Support

▲ VSAM data sets do not have to be deleted and redefined before recoveries when :

- Online Recovery Service (ORS) is used and REUSE is specified
- Recovering HALDB database data sets
 - ▶ REUSE is required for HALDB
- Image Copy was produced by Image Copy 2 utility
 - ▶ Does not require REUSE
 - ▶ True for IMS V6 and IMS V7

Usability

VSAM REUSE may be used in certain instances with IMS V7. REUSE is specified in the IDCAMS DEFINE statement.

With REUSE, VSAM database data sets do not have to be deleted and redefined before they are recovered.

REUSE may be used in three cases:

1. When the Online Recovery Service (ORS) product is used to recover the data set.
2. When the data set is a HALDB database data set. In fact, REUSE is required for HALDB VSAM data sets except those used for ILDSs and PHIDAM indexes. ILDSs and PHIDAM indexes cannot use REUSE. HALDB VSAM databases also may be reorganized without deleting and redefining the database data sets.
3. When the recovery is done using an Image Copy produced by the Image Copy 2 utility. This capability is exists with both IMS V6 and IMS V7.



OSAM DEBS Above 16M Line (ECSA)

▲ OSAM DEBS are moved from CSA to ECSA

- Relief is approximately:
 - ▶ 160 bytes per open OSAM data set
 - ▶ 32 bytes per extent in each open OSAM data set

Usability

▲ Example of savings:

- 400 open OSAM data sets: (400 x 160 = 64,000 bytes)
- 800 extents in open OSAM data sets: (800 x 32 = 25,600 bytes)
- Savings: 89,600 bytes

© IBM Corporation 2000, 2001

IMS V7 has moved the OSAM Data Extent Blocks (DEBs) above the 16M line from CSA to ECSA. This provides some CSA constraint relief. The relief is approximately 160 bytes per open OSAM data set and 32 bytes per extent in the open OSAM data sets.



255 OSAM Database Buffer Subpools

▲ Up to 255 OSAM database buffer subpools may be defined

- Previous limit was 100
- Provides additional buffer pool tuning possibilities
- Requires PQ41126 for IMS V7
- Also available with IMS V6 via PQ41125

Performance

IMS V7 has increased the number of OSAM database buffer pools that may be used by a subsystem. The limit was 100. It is now 255. This provides more flexibility in tuning. It may be especially important to HALDB users who could have many more database data sets. Support in IMS V7 requires PQ41126. The support is also available in IMS V6 with PQ41125.



Change Accumulation Enhancements

▲ Data sharing limits merging of logs in IMS V6

- Cannot merge incomplete log set
 - Merging requires /DBRs or termination of all IMS systems
 - "Spills" unmergable logs

▲ IMS V7 eliminates this restriction

- Merges all records up to end of oldest log
- "Spills" only later timed records

Usability

Availability

IMS V7 improves the Change Accumulation utility. This utility is used to merge logs from different subsystems and to keep only the latest updates for a location in a data set. Data sharing users in previous releases had to use Change Accum before they could recover a database data set. ORS in IMS V7 makes the execution of Change Accum optional; however, it still may be desirable.

Change Accum can either merge records for the same location or spill them. Merge records do not include the times of the updates. Spill records include the times of the updates. These times are required for later change accum processing. Spill records are created when some logs which were produced at the same time as the logs being processed are not available to this Change Accum execution. Since the missing logs will be processed by a later Change Accum execution, Change Accum must understand whether the updates from these logs occurred before or after the updates from the previous Change Accum execution.

IMS V7 Change Accumulation makes its decision on which records to merge or spill based on different information than was used in previous releases. The result is that many more records will be merged for data sharing users. IMS V7 merges all records up to the end of the oldest log it has received. It spills only later timed records. The next pages contain examples which illustrate this change.

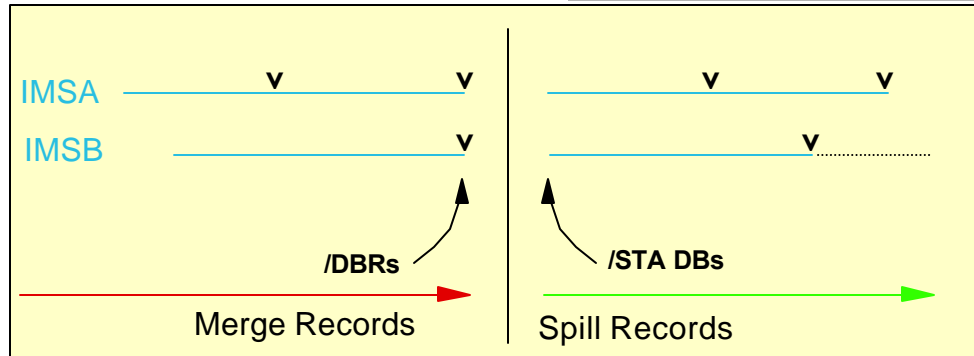
The Database Recovery utility will not accept as input any Change Accum data set which has spill records for the data set it is recovering. ORS will not use spill records for its recoveries.



IMS V6 Change Accum Illustration

▲ /DBRs are used so that CA may merge records

Merge vs. spill decision based on DBRC ALLOC record times



V: end of volume

— : log is input to Change Accumulation

..... : log is not input to Change Accumulation

© IBM Corporation 2000, 2001

This shows a typical use of Change Accumulation with data sharing in IMS V6. IMSA and IMSB are producing log streams. To create a point to which logs may be merged, /DBRs are issued for both IMS systems. If logs from before and after the /DBR times are input to Change Accum, only the changes before the /DBRs are accumulated. Later change records become spill records. The /DBRs are done so that some of the records may be accumulated. IMS V6 bases its decision on whether to merge or spill database change log records on the DBRC ALLOC record times. That's why /DBRs are used. They create ALLOC record close times. Of course, these /DBRs make the database unavailable until the /START DB commands are processed.

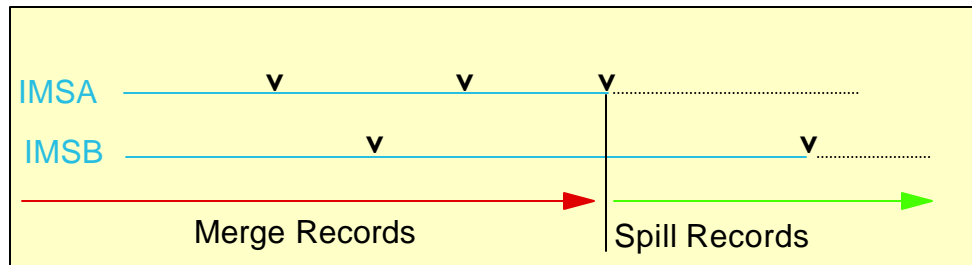


IMS V7 Change Accum Illustration

▲ /DBRs are not needed

- Fewer spill records are created

Merge vs. spill decision based on earliest end time for each input log stream



V: end of volume

— : log is input to Change Accumulation

..... : log is not input to Change Accumulation

IMS V7 eliminates the requirement for /DBRs for Change Accum purposes. Change Accum accumulates all of the change log records up to the time for which it has received all log records. IMS V7 bases its decision on whether to merge or spill log records based on the latest time for which it reads all of the log records. This is the earliest end time among all of the log streams. All records before this time are merged.

In this illustration, the inputs to Change Accum include logs from IMSA and IMSB. The input logs from IMSA end before those from IMSB. All of the change records on the IMSA logs are accumulated. All of the change records from IMSB that were created up to the time of the end of the IMSA input logs are also accumulated. The only log records which become spill records are those from IMSB which were created after the time of the end of the IMSA log records.

There are two benefits with this change. First, /DBRs for Change Accum purposes are no longer needed. Second, the size of Change Accum data sets is reduced. Many of the spill records are no longer needed.



Change Accumulation Enhancements

▲ Sort efficiency automation

- Automatically calculates sort key length
- Simplifies user interface
 - ▶ User does not specify size
 - ▶ Value in "ID" statement is ignored
- Avoids unnecessarily large sizes
 - ▶ Large sizes have negative performance effect

Usability

Performance

Change accumulation sorts database change log records. The sort key for KSDS records is the record key. For other types of data sets, the sort key is the four byte relative block number. In previous releases, Change Accumulation assumed a 10 byte key size unless the user specified a size on the ID control statement. For keys greater than 10 bytes, this specification was required.

If the key size specified (or defaulted) was greater than required, the sort had unnecessary overhead.

IMS V7 eliminates these user considerations. It calculates the minimum required sort key size and uses it. The calculation is done by examining the DBD for KSDS data sets.

These changes remove the user's responsibility for understanding and specifying the sort key size. In the past, some installations have specified the maximum size (255 bytes) for the sort key to avoid the discovering the minimum required size for each change accumulation group. For these installations, the change will result in improved performance.



Image Copy 2 Enhancement

▲ Compression may be used for Image Copy 2 copies

- Invoked by control statement or GENJCL.IC keyword
- Invokes COMPRESS for DFSMSdss DUMP
- DFSMSdss RESTORE automatically expands data
 - RESTORE is invoked by IMS Database Recovery and ORS

▲ Benefits

- Smaller space requirements

Performance

Availability

The Image Copy 2 utility was introduced by IMS V6, It invokes DFSMSdss DUMP to copy an IMS database data set. IMS V7 adds the capability to specify that the copy should be compressed. This may be done either with a control statement for the utility or through a new keyword on the DBRC GENJCL.IC command. The Database Recovery utility and Online Recovery Service recognize that their input is a DFSMSdss dump and invoke DFSMSdss RESTORE for these data sets. Restore automatically expands compressed data sets.

This compression capability creates smaller dump data sets.



DBRC GENMAX and RECOVPD Enhancement

▲ RECOVPD will not cause changes in GENMAX value

- RECOVPD and GENMAX will operate independently
 - ▶ GENMAX is max. number of ICs that DBRC tracks
 - ▶ DBRC keeps all ICs that are not older than RECOVPD
- Previous releases increased GENMAX when RECOVPD required more ICs

▲ Benefit

Usability

- Users will not have to manually reset GENMAX

RECOVPD specifies the minimum number days which Image Copies for a database data set should be kept. GENMAX specifies the minimum number of Image Copies that should be kept. DBRC keeps a record of an Image Copy if either RECOVPD or GENMAX require it.

In previous releases of IMS, the user specified value for GENMAX may be changed due to the processing for RECOVPD. This made it necessary for users to reset the GENMAX value when the requirements of the RECOVPD processing allowed it.

IMS V7 makes the two parameters independent. Users do not have to reset GENMAX due to RECOVPD processing.



DBRC PROCOPT=L and LS Support

▲ DBRC will require Image Copy after initial load

- 'Image Copy Needed' flag set
 - ▶ Prevents updates which cannot be recovered
- REORG record written for each database data set
 - ▶ Prevents recovery using IC taken before initial load

▲ Benefits

- Improved data integrity
- Elimination of user actions
 - ▶ CHANGE.DBDS ICON
 - ▶ NOTIFY.REORG

Usability

Availability

IMS V7 increases data integrity by setting the DBRC 'Image Copy Needed' flag and writing a REORG record for each data set when a database is initially loaded.

The 'IC Needed' flag prevents updates which cannot be recovered by a subsequent database recovery. If an Image Copy is not taken before updates are done, a subsequent recovery may not be possible since it would require a copy of the data as it appeared at the time of the initial load.

Reorg records prevent an invalid recoveries from being executed. The REORG record tells DBRC not to allow a recovery with IC or log inputs before the REORG record time and log inputs after the REORG record time. Databases which are initially loaded (using PROCOPT=L or LS) on a regular basis are subject to an integrity exposure without this REORG record.

With previous releases of IMS, users often added the Image Copy Needed flags and Reorg records to provide increased database integrity. These user actions are not required with IMS V7.



DBRC LIST.DBDSGRP Enhancement

▲ LIST.DBDSGRP enhancement

- Option to specify databases and data sets with ALL

- ▶ A list may be specified

```
LIST.DBDSGRP ALL ((DBA,DBADS2),(DBB))
```

- Lists all DBDSGRPs which contain either data set DBADS2 from database DBA or any data set from database DBB.

- Limits output when using LIST.DBDSGRP ALL

This enhancement makes the output of a LIST.DBDSGRP command more useful by allowing the user to limit what is produced. IMS V7 adds the capability to limit a LIST.DBDSGRP ALL command. Previous releases did not allow any parameters to be specified with ALL. A LIST.DBDSGRP ALL command produces a list of all the DBDS and DB groups identified in the RECONS. IMS V7 allows database and data set names to be specified with ALL. This produces a list of only those groups containing the specified databases and/or data sets.



DBRC LIST.HISTORY Enhancement

▲ LIST.HISTORY enhancement

- Extraneous records for database(s) deleted
 - ▶ Does not list all data from PRILOG records
 - ▶ Does not list data from other data sets in CA group
- Graphic timeline added
- More useful and easier to read information

© IBM Corporation 2000, 2001

There are two enhancements to the LIST.HISTORY command output. First, the output is simplified eliminating unneeded lines. Second, a graphical timeline is added. These make the output easier to use.

Records no longer listed are CA group records and the CA execution record entries for the other DBDSs in the group. PRILOG records are listed only once. Previously, a PRILOG record would be listed multiple times if multiple ALLOC records referred to it.

An example of the graphic timeline is shown on the next page.



Batch Backout

▲ Enhanced Log Verification failure message DFS3294

■ Prior to IMS V7:

- ▶ Indicated input was wrong and DDNAME in error
- ▶ Did not account for concatenated data sets

```
DFS3294A Log Verification Failure nn - rrrrr  
[DDN=...[, VOL=...]]
```

■ IMS V7:

- ▶ Indicates DSN or VOLSER in error
- ▶ Indicates what was expected

```
DFS3294A Log Verification Failure nn - rrrrr  
[DSN=DSNexamined[,VOL=VOLSERexamined]]  
[EXPECTED= DSNexpected | VOLSERexpected]
```

© IBM Corporation 2000, 2001

IMS V7 has expanded the log verification failure message DFS3294A. The message from prior releases only indicated the DDNAME of the DD statement examined and, possibly, the VOLSER at the time of the problem. In IMS V7 the message also includes which data set or volume is in error and what was expected.



Batch Backout

▲ Performance improvements

- Backout of batch jobs
 - ▶ Saves log records on forward pass
 - ▶ Does not read backwards
- Backout of online and batch jobs
 - ▶ Uses data spaces when address space memory is exhausted

© IBM Corporation 2000, 2001

The performance of the Batch Backout utility has been improved.

For batch jobs, the log records are saved on the forward pass of the log data set thereby eliminating the need to read backwards. In previous releases, Batch Backout read forward to the end of the log. It then read the log backward to the last sync point. The backout was done during the backward read. The new method is the same method that Batch Backout has used in previous releases when invoked to backout an online system. With this method log records are saved until a sync point record is reached. Then, the saved records are discarded. The save process then starts again. When the end of the log is reached, the backout is done using the saved log records. This eliminates the read backward processing.

The log records that are saved on the forward pass of the logs are saved in Batch Backout's address space memory. If the address space memory is exhausted then a data space is dynamically allocated. The use of data spaces is new in IMS V7. IMS V7 Batch Backout can allocate as many data spaces as are needed. This assumes that there are no constraints on data space allocation.



Batch Backout ...

▲ Migration consideration:

- The READBACK statement may be used. If so:
 - ▶ Log records are not saved during forward pass
 - ▶ Read backwards is performed
- The READBACK statement should be used
 - ▶ when Batch Backout requires more memory than is available in the address space
 - and
 - ▶ the use of data spaces is limited

© IBM Corporation 2000, 2001

There are some migration considerations for the new Batch Backout processing.

The READBACK control statement is not new. It has been used in the past for back outs of online systems. The READBACK statement indicates that we don't want Batch Backout to save log records on the forward read. Instead, it Batch Backout reads the log to the end and then reads it backwards to do the backout. With READBACK log records are not saved in memory or data spaces on the forward read.

A READBACK should be used if there is not enough space in the address space to save the log records and a data space(s) cannot be allocated. If a data space cannot be allocated when it is needed then Batch Backout terminates. In such a situation, the READBACK statement should be specified.



ALTER Support for Data Sharing Structures

▲ ALTER support for OSAM and VSAM structures

- Structure size may be changed while in use
 - ▶ Could be useful when adding more IMS systems to a sharing group
- Also available in IMS V6 via PQ38946
- OS/390 V2R10 provides autoalter support
 - ▶ Specified in CFRM policy for structure
 - ▶ System can automatically increase structure size to avoid reclaims

▲ IRLM Lock Structure already had ALTER support

© IBM Corporation 2000, 2001

ALTER support of OSAM and VSAM data sharing cache structures has been added. It is also available in IMS V6 with PQ38946. ALTER may be used to change the size of a structure or the allocation of elements within a structure. Typically, we do not need to alter IMS OSAM and VSAM structures. Nevertheless, there are times when it is handy.

OSAM and VSAM structure keep track of the location of OSAM blocks and VSAM CIs in IMS's database buffer pools. They should be large enough to track each block or CI that might be in a buffer pool. Obviously, we cannot have more blocks or CIs than the number of buffers we have. Since we do not change the number of buffers in an IMS system dynamically, the size requirements of these structures tend to be static. On the other hand, when we add an IMS system to our data sharing group, we add more buffers. We could have additional blocks or CIs in our pools. At these times, we probably need to increase the size of our structures. ALTER allows us to do this easily. Without ALTER support we would have to rebuild the structures to increase their sizes.

OS/390 V2R10 added autoalter support. Autoalter allows the system to automatically increase a structure's size to avoid reclaims. That is, if buffers in our IMS systems are being invalidated because we do not have enough directory entries in our OSAM or VSAM structure, the system can automatically alter the size of the structure. With the new support for alter, autoalter can help IMS systems.

The use of autoalter is controlled by the CFRM policy. The policy indicates the initial, minimum, and maximum sizes which a structure may have. To use autoalter for a structure ALLOWAUTOALT(YES) must be specified for it in the CFRM policy,



DEDDB I/O Error Handling

▲ New way of handling write errors for DEDBs

- Write error CIs kept in memory
- Write error CIs may be read from memory
- Area not stopped after write errors for 10 CIs

▲ Benefits

- Increased CI availability
- Increased area availability
- Processing similar to full function

Availability

© IBM Corporation 2000, 2001

IMS V7 introduces a new way of handling write errors for DEDBs. When a write error occurs, a copy of the CI is kept in memory. This copy may be used to satisfy later read requests. Unlike previous releases of IMS, IMS V7 does not stop an area when more than 10 write errors occur for the last area data set.

A CI for which a write error occurs is not lost to later processing. It remains available to the subsystem. More than 10 write errors does not cause an area to be lost to processing. It remains available. These changes make the handling of DEDB write errors similar to the handling for full function databases.



DEDDB I/O Error Handling

▲ IMS V6 handling of write errors

- Write EQE created in CI1 and kept in memory
- Data sharing partners notified
- CI is no longer available
 - ▶ Requests for CI result in 'AO' status code
- If more than 10 write errors for ADS
 - ▶ ADS is stopped
 - ▶ Area stopped if no remaining ADS

© IBM Corporation 2000, 2001

This is a review of the way that write errors are held in releases prior to IMS V7. A write error would create an error queue element (EQE). This EQE was stored in CI1 and kept in memory. When data sharing was in use, a notification of the error was sent to data sharing partners. After the write error, the CI was not available to any subsystem, including the one that experienced the error. Any call which required the CI would receive an 'AO' status code. Since there is only room for 10 EQEs in CI1, more than 10 write errors would result in the area data set (ADS) being stopped. If this was the last ADS, the area was stopped.



DEDDB I/O Error Handling

▲ IMS V7 uses processes similar to full function

▲ Write error causes creation of EEQE and creation of I/O Toleration (IOT) buffer in memory

- CI stored in IOT buffer for subsequent reads and writes
- CI written to log at system checkpoint
 - Buffer restored on subsequent restarts of IMS system
- EEQE written in RECONS
- EEQE notification sent to data sharing systems

© IBM Corporation 2000, 2001

IMS V7 introduces processes similar to those used for full function databases.

In IMS V7 a write error causes the creation of an extended error queue element (EEQE). This EEQE is kept in memory and written in the RECONS. If data sharing is being used for the DEDB, the EEQE is sent to data sharing partners via an IRLM notification. These partners create a copy of the EEQE in their memory. New users of the area get a copy of the EEQE from DBRC when they get authorization for the area.

IMS V7 also creates a IO toleration (IOT) buffer containing a copy of the CI when a write error occurs. Later requests for the CI are honored from this buffer. A copy of the CI is logged at system checkpoints. Restarts of the system restore the CI to an IOT buffer. This means that the CI remains available to the system experiencing the write error. The recovery of the area data set may be delayed until a convenient time.

These actions allow IMS V7 to increase data availability.



SDEP SCAN Expansion of Compressed Data

▲ SDEPs may be compressed with segment edit/compression exit

- Previous releases did not include SCAN utility support for segment edit/compression exit
- Users could expand during scan process by using Scan exit routine
 - ▶ Different exit routine or different coding required for each compression routine

▲ IMS V7 eliminates need for Scan exit for expansion

- Scan utility option to invoke exit
 - ▶ Users may continue to use old techniques for compatibility

Usability

Previous releases of IMS support compression of DEDB sequential dependent segments (SDEPs). The Scan utility for those releases does not include explicit support for expanding these segments. Users could expand these segments during Scan processing by specifying a Scan exit routine. This routine would then invoke the expansion function of the segment's edit compression/expansion routine. Since each segment might have a different edit compression/expansion routine, each invocation of the Scan utility might require a different exit routine. This implementation was unnecessarily complex. IMS V7 eliminates the need for invoking the Scan exit. Instead, the user may simply ask for expansion of the scanned segments. The utility will then invoke the expansion function of the segment edit/compression exit routine specified in the DBD. This simplifies the use of compressed SDEPs.

For compatibility reasons, the old technique may still be used.



Open Database Access (ODBA)

▲ Available in IMS V6 with PQ15784

▲ Callable interface to IMS DBCTL or IMS TM/DB

- Similar to CICS connection to DBCTL (DRA)
- Caller is in MVS address space outside of IMS TM or CICS
 - Such as DB2 Stored Procedure

▲ Caller connects to IMS, schedules PSB, issues DL/I calls, commits work, ...

- Uses AIB interface
- Supports two-phase commits

▲ Prerequisites

- OS/390 Release 3 or later
- Resource Recovery Services (RRS/MVS)

Usability

© IBM Corporation 2000, 2001

Open Database Access (ODBA) was introduced in IMS 6.1 by PQ15784.

ODBA provides a callable interface to IMS from an MVS address space other than CICS or IMS TM. The interface uses the Database Resource Adapter (DRA) as CICS does when using DBCTL services. Like CICS, ODBA may use either a DBCTL subsystem or an IMS TM system with IMS DB.

ODBA is designed for use with DB2 Stored Procedures but is also applicable to other uses.

When using ODBA the caller connects to IMS, schedules PSBs, issues DL/I calls, and commits its work. The interface supports multiple threads. That is, one address space may have multiple concurrent units of recovery in execution. Callers must use an AIB interface. ODBA supports two-phase commit processing. Callers may access other resource managers, such as DB2.

ODBA requires at least release 3 of OS/390 and the use of Resource Recovery Services (RRS/MVS) with the MVS System Logger.

ODBA makes it easier for installations to access IMS database information. Its use is not limited to DB2 Stored Procedures. It may also be used with OS/390 server address spaces, such as Web servers.



Transaction Manager Enhancements



▲ Transaction Manager Enhancements

- Parallel Sysplex Enhancements
- Exit Routines
- System Execution Enhancements
- Connectivity Enhancements
- Application Development



Parallel Sysplex Enhancements



CQS Enhancements

▲ IMS V7 Enhancements

- Support for multiple clients
 - ▶ Achieves better utilization of the CQS address space
 - Storage
 - Problem determination
 - Operations

- Security checking during CQS registration

- Interface enhancements
 - ▶ Diagnostics
 - ▶ CQS requests

Usability

Availability

IMS V7 enhances the Shared Queues support that was introduced in IMS/ESA V6 by allowing a single CQS (Common Queue Server) address space to support multiple clients, e.g., IMS systems, that reside on the same CEC as CQS. CQS continues to be the component that accesses the coupling facility structures. The multiple client support improves storage utilization and simplifies problem determination (e.g., only one CQS address space to analyze) as well as operational procedures (fewer address spaces to monitor).

The ability to use RACF or another security product to control CQS registration is also being added. This prevents an unauthorized client from accessing CQS and issuing commands that could prevent other clients from connecting to a structure.

CQS also provides several interface enhancements in the area of diagnostics and in the CQS requests. IMS does not take advantage of these but they are available for new clients, e.g., vendor monitors, to use.



Shared Queues Enhancements

▲ Common Queue Server (CQS) enhancements

- Support for multiple clients
- Security checking during CQS registration
- Interface enhancements

▲ APPC/OTMA enhancement

- Capability to process asynchronous messages on a back-end IMS system

© IBM Corporation 2000, 2001

IMS V7 enhances the Shared Queues support that was introduced in IMS/ESA V6 by allowing a single CQS (Common Queue Server) address space to support multiple clients, e.g., IMS systems, that reside on the same CEC as CQS. CQS continues to be the component that accesses the coupling facility structures. The multiple client support improves storage utilization and simplifies problem determination (e.g., only one CQS address space to analyze) as well as operational procedures (fewer address spaces to monitor).

The ability to use RACF or another security product to control CQS registration is also being added. This prevents an unauthorized client from accessing CQS and issuing commands that could prevent other clients from connecting to a structure.

Support of the processing of asynchronous messages from APPC and OTMA on a "back-end" shared queues system has been added in IMS V7.



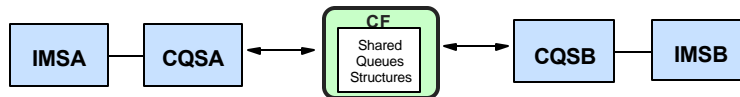
CQS Background

▲ Shared Queues were introduced in IMS/ESA V6

- Message queues are stored in CF structures
- Message queues are available to multiple IMS systems
 - Any IMS system may process a transaction
 - Provides for dynamic workload balancing

▲ Common Queue Server (CQS)

- Component that manages the shared queue structures on the CF
 - Connects to the shared queue structures
- Runs in separate address space connected to IMS
- Acts as server for IMS control region
 - V6 - One client (IMS) per CQS



© IBM Corporation 2000, 2001

To better understand the IMS V7 enhancement, a little background information is provided on this page.

Shared queues are exactly what the name implies. They are a set of IMS message queues and EMH queues which can be shared between multiple IMS systems in a Parallel Sysplex environment. The queues physically reside in list structures in Coupling Facilities (CF) where they are accessible to any IMS in the same "shared queues group". IMS does not directly access the shared queues structures in the CF. To do this, IMS interfaces with a subsystem running in a separate address space called the Common Queue Server, or CQS. All queue manager activity such as putting messages on the queue or reading them off the queue is done through CQS.

In IMS/ESA V6, each IMS in the shared queue group is served by its own dedicated CQS.



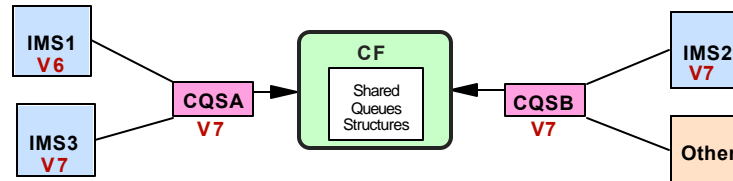
V7 CQS - Multiple Client Support

▲ Support for multiple clients

- Achieves better utilization of the CQS address space
- Allows other clients to access shared queues
 - Others could include monitoring tools

Usability

Availability



▲ Security checking during CQS registration

- Connector (e.g. IMS) to CQS is checked
- Since multiple users are allowed, we need to check who they are

© IBM Corporation 2000, 2001

The multiple client support in IMS V7 opens up the CQS address space to allow up to 32 clients on an MVS system to use the same CQS address space.

The CQS Clients can be a mix of IMS control regions and other clients. Some details include:

- IMS clients can be at different release levels
- IMS V6 and IMS V7 can attach to the same CQS
- IMS starts CQS if it is not active, otherwise, IMS registers with the active CQS
- CQS registration is controlled via RACF or an equivalent product

For migration purposes, both V6 and V7 CQSs can connect to the same CF structures and both releases can run on the same CEC.



Asynchronous APPC/OTMA

▲ Support for Asynchronous input message processing on a Shared Queues back-end system

- APPC Asynchronous inbound requests (Allocate-Send-Deallocate)
- OTMA Commit-then-Send (commit mode 0)
- Note:
 - ▶ IMS/ESA V6 required all APPC/OTMA input messages to process on the Shared Queues front-end IMS system
 - ▶ Synchronous messages still process on the system in which they are received

Performance

Usability

© IBM Corporation 2000, 2001

Another enhancement in IMS V7 affects the Shared Queues environment. Asynchronous APPC and OTMA input messages can now be processed on any IMS system in the shared queues group.

Synchronous transactions will continue to be processed on the front-end system.

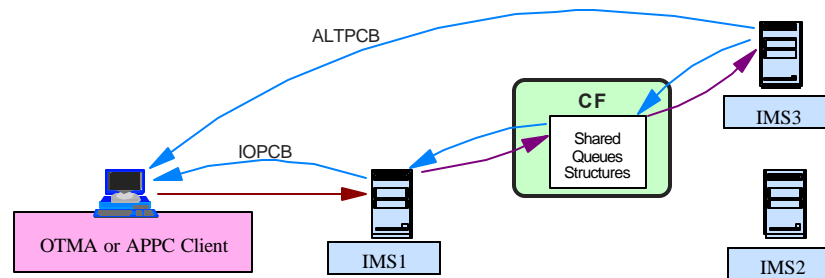
Note that the "asynchronous" nature of a message is still the same as it was in previous releases. APPC asynchronous messages are initiated by an LU 6.2 program issuing an Allocate-Send-Deallocate sequence for a non-conversational, non-Fast Path, non-response mode transaction. For OTMA, the message is sent in with a request of commit-then-send (commit mode 0) for a non-conversational, non-Fast Path transaction.



Asynchronous APPC/OTMA

▲ *Asynchronous* OTMA/APPC input messages are allowed to process on any IMS system in the shared queues group

- ▶ **Assumes** APPC/OTMA are enabled on all back-end systems



- ▶ IOPCB messages are delivered by the system which receives the input message
- ▶ ALTPCB messages are delivered by the system that processes the transaction

© IBM Corporation 2000, 2001

In IMS/ESA V6, an input message arriving from APPC or OTMA can be scheduled only on the IMS where it arrived. In an IMS V7 environment, asynchronous input messages from OTMA/APPC can be processed on any back-end system in the shared queues group that has interest in the transaction.

- IOPCB output messages

IOPCB responses must be delivered by the front-end system that receives the message from the client. A special internal "notify" message has been implemented as part of the enhancement. It informs the front-end that the back-end has placed an OTMA/APPC IOPCB response on the shared queues. The front-end then knows to retrieve the message from the queues and send the response to the appropriate OTMA/APPC destination.

- ALTPCB - program-to-program switches

If the asynchronous input invokes a transaction which subsequently issues a program-to-program switch then IMS copies the input message prefix to the prefix for the new message. This allows the subsequent transaction's IOPCB reply to be correctly delivered by the front-end. IMS queues the spawned message to the shared queue to allow the message to be processed by any back-end IMS.

- ALTPCB output messages - OTMA/APPC destination

Alternate TP PCB output is delivered by the system on which the application inserting the output is processing. This assumes that OTMA and APPC support have been enabled on every back-end system in the shared queues group that is capable of processing the application. If a back-end system does not have the appropriate support enabled, any asynchronous OTMA/APPC output inserted to the ALTPCB is queued and remains queued until such time that the support is enabled (/STA OTMA or /STA APPC). If the transaction abends, any DFS messages, such as DFS5551, are delivered asynchronously by the system on which the transaction processes.



VTAM Generic Resources Enhancements

▲ IMS DFSDCxxx options: GRAFFIN and GRESTAE

Availability

- Delivered via APAR PQ18590 in IMS/ESA V6
- Part of Base IMS V7
- Greater control over access availability to any IMS when failures occur

- **GRAFFIN = IMS | VTAM**
 - ▶ System option that specifies which component is to manage the Generic Resource affinities

- **GRESTAE = Y | N**
 - ▶ System option that defines whether or not IMS should reset affinities (CLSDSTs) during ESTAE processing
 - ▶ Applies to GRAFFIN=IMS

Usability

Note: MSC does not use VGR (sessions specify APPLID)

APAR PQ18590 delivered two new VTAM Generic Resources (VGR) options, GRAFFIN and GRESTAE, in IMS/ESA V6. They are included as part of the IMS V7 base support for VGR. Both options are specified at a system level and have no provision to provide different options at a terminal level. These enhancements address availability issues that were raised as a result of the original implementation of VGR in IMS/ESA V6.

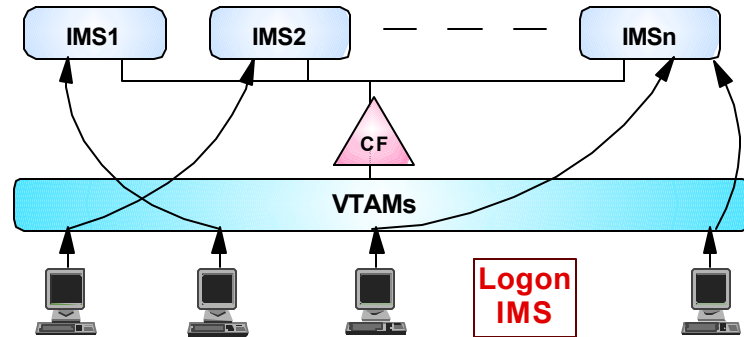
The GRAFFIN option provides the ability to choose whether IMS or VTAM is to manage the generic resource affinities. This is described in more detail in the following pages.

The GRESTAE option which applies to GRAFFIN=IMS environments, further provides the capability to choose whether or not IMS should issue VTAM CLSDSTs during ESTAE processing.

- Specifying GRESTAE=Y tells IMS to issue CLSDSTs which releases affinities if no terminal-related status conditions, such as response mode or conversational mode, exist. The CLSDSTs are issued synchronously and can result in delaying IMS termination. Under normal conditions, the time this action takes is not significant. On the other hand, problems might occur. An example is I/O time-outs that are commonly associated with cross-domain terminals.
- Specifying GRESTAE=N tells IMS to bypass the VGR CLSDSTs during ESTAE processing. This results in a fast termination of IMS but retains the terminal affinities to the failed IMS system.



Background - VGR ...



- All IMSs in the Sysplex join a **Generic Resource Group**
- IMS user logs on to a **Generic Resource Name** for the IMS Sysplex
- VTAM connects the user to one of 'n' IMS **Members** of the Group

© IBM Corporation 2000, 2001

To better understand the enhancements to the VGR support, some background information is provided.

VGR is a service provided by VTAM to minimize the knowledge that an end user needs to logon to one of several like instances of an application, such as IMS, in a Parallel Sysplex. To the end user, there is a common name, called the Generic Resource Name (GRSNAME) which is used by VTAM to refer to any of the members of a Generic Resource Group (GRG). VTAM decides which member to use when processing an end user LU's session request.

This service provides a number of benefits including network load balancing across all the instances of IMS in the GRG. Were it not for VGR, the user, through system definition and special logon instructions, would have to perform the work of trying to balance these sessions. It also allows for easy changes in the number of IMSs in the sysplex. When a new IMS is added to the GRG, logons will be directed there without the end user ever knowing that a new IMS has been added. Likewise, IMSs can be removed with minimum impact on the end user. Obviously, there is some impact since the users would have to log off and then log back on to the GRSNAME to get connected to another IMS.

Note that there is a list structure in the CF called ISTGENERIC. This is where VTAM keeps track of which applications belong to which group, and which member each LU is logged onto. If an LU is logged onto a specific member, it is said to have an "affinity" for that member.



Background - VGR ...

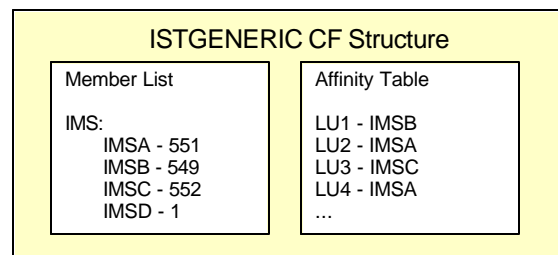
AFFINITY: A mapping of an LU to a specific member (IMS)

▲ Affinity is set when a session is first established

- VTAM maintains the **affinity table** in ISTGENERIC

▲ When an LU requests a session using the GRSNAME

- VTAM checks to see if an affinity exists
 - If one exists, the session request is routed to that member
 - Otherwise, VTAM chooses one of the members
 - Special considerations for ISC



© IBM Corporation 2000, 2001

The concept of affinities is important in VGR, and is particularly important to IMS. When an LU logs on to IMS, VTAM checks to see if an affinity exists. If one exists, the session request is routed to that IMS. Otherwise, VTAM chooses one of the members of the GRG.

There are special considerations for ISC affinities:

The first request for an ISC parallel session establishes the guidelines for subsequent parallel session requests:

- If the first parallel session request specifies the GRSNAME then subsequent requests using the GRSNAME will be routed to the same member IMS.
- If the first parallel session request is for a specific IMS APPLID then a subsequent request using a generic name is interpreted by VTAM as a request for a session different from the specific IMS. VTAM will select an IMS from the Generic Resource Group excluding the specific IMS. This request will fail because all parallel sessions have to be established with the same IMS.

For all parallel sessions to be established with the same IMS, all must request the generic resource name, or all must request the specific APPLID



GRAFFIN=IMS

▲ Affinity deletion may occur at session termination

- Special treatment for ISC, SLUP/Finance, ETO
- Special treatment for LUs that have a VGR-related status
 - ▶ Conversation mode, full function response mode, FP response mode, exclusive mode, test or MFS mode, preset mode
 - ▶ User-written exits (DFSSGFX0 and DFSLGFX0) can reset status

▲ Affinities are not (cannot be) deleted for MVS/CEC/VTAM failures

- IMS ESTAE not driven and affinities still exist when IMS is restarted

▲ If affinity is not deleted at session termination

- Next logon to generic name establishes session with existing affinity
- May need to wait for an IMS restart

© IBM Corporation 2000, 2001

With the GRAFFIN=IMS option, IMS takes the full responsibility of deleting the affinity. At session termination, which may occur for a number of reasons (e.g., user logoff, VTAM LOSTERM / NSEXIT, IMS normal shutdown, IMS failure, MVS, VTAM or CEC failure), IMS decides whether or not to delete the affinity. Some types of sessions, however, get special treatment from IMS:

- ISC parallel session affinities are deleted when all sessions are quiesced or cold. An ISC single session affinity is deleted if the session is quiesced.
- SLUTYPEP and FINANCE terminals require resynchronization using STSN (set and test sequence number). Only the IMS they were in session with has the necessary information to resynchronize a session when it is reconnected. Affinities are not deleted at session termination, except when IMS shuts down with the LEAVEGR option. LEAVEGR indicates that IMS is leaving the GRG.
- All affinities are retained for LUs with significant status. Significant statuses are response mode, conversational mode, exclusive (/EXC), test (/TEST), preset destination (/SET xxxxxxxx), and MFS test (/TEST MFS).
- When an ETO user signs off, significant status is moved from the terminal to the user structure. When logoff occurs, the affinity is deleted because there is no significant status on the terminal. When the ETO user logs on again to the GRSNAME, if VTAM selects the same IMS where the significant status was held and the Signon Exit selects the same User Structure, significant status will be restored. For example, the terminal may be returned to its conversation. If VTAM selects a different IMS with no existing user structure there will be no significant status. The next time the ETO user logs on and VTAM selects the original IMS, the original significant status is restored. This could be confusing and is, therefore, a good reason to use the SIGNOFF exit to reset significant status for ETO terminals.

Note that if MVS or VTAM or the CEC fails, IMS ESTAE is not driven and IMS cannot delete any affinities. If a user were to log on again immediately, he would be directed back to the IMS which just failed and the logon request would fail.

If the affinity does not get deleted, then the next time the user logs on the session request is directed back to the same IMS. If the IMS is still active, the logon succeeds. If it is not, the logon request fails.



GRAFFIN=VTAM

▲ Non-ISC affinities are reset at failure:

- CEC, MVS, IMS/ESA, VTAM, and Network/Session terminations
- VGR-related status conditions are automatically reset by IMS
- Terminal sessions can be reestablished immediately with any surviving IMS in group

▲ ISC affinities continue to be managed by IMS regardless of GRAFFIN

© IBM Corporation 2000, 2001

In a GRAFFIN=VTAM environment the affinities are deleted regardless of the type of failure. IMS has been enhanced to reset any significant terminal-related status conditions. The conditions are reset by IMS at signoff/logoff if possible. Otherwise, they are reset at a terminal's subsequent attempt to logon/signon to the same IMS.

ISC affinities continue to be managed by IMS and not by VTAM.



Rapid Network Reconnect (RNR)

▲ RNR implements VTAM persistent session support

Availability

- Higher availability and reduced overhead
 - ▶ Quickly reestablishes VTAM sessions following system outages (IMS, MVS, CEC or VTAM)
- Eliminates session cleanup/restart following an outage

▲ New IMS option PSTIMER

- Specifies time VTAM waits for recovery of the persistent session before terminating sessions

▲ New IMS option RNR = NRNR | ARNR

- ▶ ARNR - automatic session reconnect
- ▶ NRNR - no reconnect

▲ Prerequisite - OS/390 V2R5, ACF/VTAM V4R4.1 (HVT4411)

© IBM Corporation 2000, 2001

Rapid Network Reconnect (RNR) support in IMS V7 implements the first step towards VTAM persistent session (single node and multinode) support in IMS. This function provides the ability for sessions to be maintained across a failure (e.g., IMS) and automatically reconnected when the failed component has been restarted. Additionally, an option called PSTIMER is provided to specify how long IMS is willing to wait from the time of the failure to the reopening of the ACB. The use of RNR in IMS requires OS/390 V2R5 and the associated VTAM V4R4.1.

Some benefits for this support include:

- higher availability since the sessions are maintained and therefore already available for the application after restart.
- less CPU overhead during restart, since the significant work of cleanup/restart of sessions is eliminated.

RNR=ARNR is required to get the benefits of persistent sessions.



Background

▲ Persistent Sessions

- VTAM Single-Node Persistent Session (SNPS)
 - ▶ Reconnect must be on same CEC as original IMS
 - ▶ Supports only application (IMS) failure/reconnect

- VTAM Multinode Persistent Session (MNPS)
 - ▶ Reconnect may be on another CEC in a sysplex
 - ▶ Supports failures/reconnects, including IMS, VTAM, MVS, and CEC failures

- Applies to VTAM nodes supported except MSC
 - ▶ Persistent session support for APPC is provided by APPC/MVS

© IBM Corporation 2000, 2001

VTAM persistent session support was introduced in VTAM V3R4. This capability allowed sessions to survive application failures and then be restored as soon as the application was restarted. The value of this capability consisted primarily in the elimination of the overhead associated with session termination and re-establishment. The support, however, as initially implemented in VTAM V3R4, was limited to application failures/restarts on the same CEC (single node).

Persistent session support was further enhanced in VTAM V4R4 to provide nondisruptive session recovery from any failure including network, application, VTAM, MVS, or CEC in a parallel sysplex environment. This enhancement allowed the failed application (IMS) to be restarted on a different VTAM system in the sysplex (multinode support). As a result of this added capability, the terminology associated with persistent sessions was expanded to include: single node persistent sessions (SNPS), and multinode persistent sessions (MNPS).

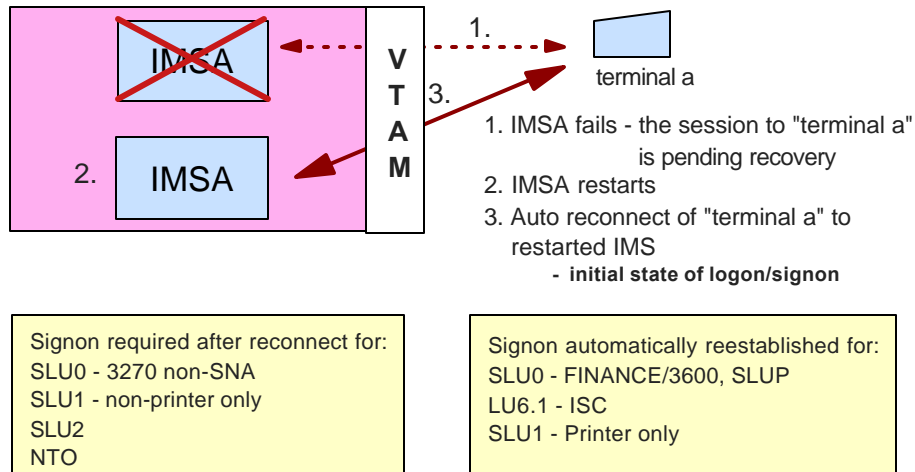
Although RNR support in IMS requires VTAM persistent session capability, the installation can select the use of either single or multinode persistent sessions.

Note that RNR does not apply to APPC. The support for APPC persistent sessions is provided by APPC/MVS.



Background ...

Single Node Persistent Session Scenario



© IBM Corporation 2000, 2001

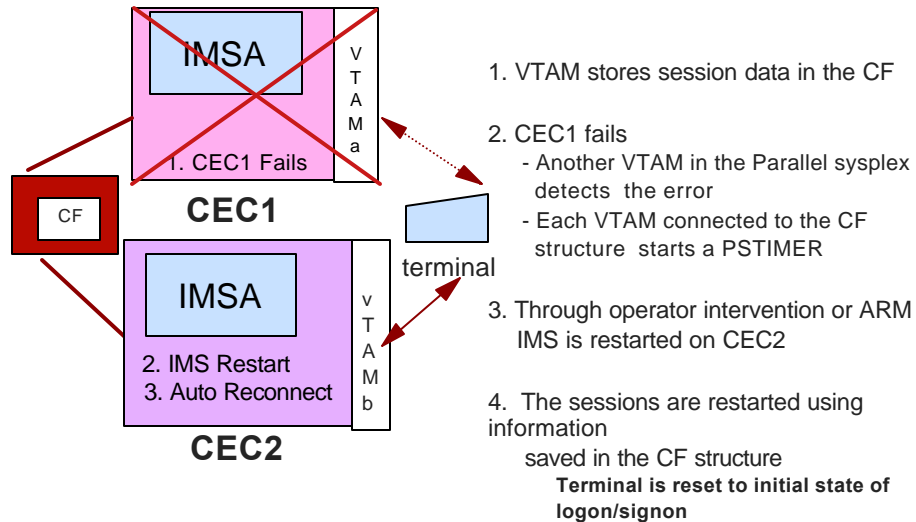
Single Node Persistent Sessions (SNPS):

When a persistence-enabled application program (i.e., IMS with RNR=ARNR specified) fails, VTAM closes the ACB on IMS's behalf but retains the LU-LU sessions, saves the allocated resources and control blocks in a data space, and shields the network from knowledge of the application program failure. VTAM stores the incoming data so that the network views the session as active but not currently responding. When the failed IMS restarts, VTAM reconnects the sessions. If the PSTIMER option is not specified, the session can remain pending recovery for an indefinite period. In this case, a VTAM "VARY NET INACT" command can be issued to close the session.



Background ...

Multinode Persistent Session Scenario



© IBM Corporation 2000, 2001

Multinode Persistent Sessions (MNPS):

In an MNPS environment with IMS and RNR=ARNR specified, session data is stored by VTAM in the MNPS structure of a Coupling Facility.

In this example CEC1 fails. When VTAMa fails, VTAMb in the same sysplex detects the error and IMSA is marked pending recovery. VTAMb which is connected to the same MNPS structure in the CF starts a timer (PSTIMER value). If the PSTIMER expires before IMSA recovery is successful, VTAMb will perform cleanup of the failed IMS's session information in the CF structure. Otherwise, if IMSA is restarted on CEC2 either by operator intervention or by the Automatic Restart Manager (ARM), then IMSA's sessions are restored using the information from the MNPS structure in the CF.



Background - Requirements

▲ VTAM V4R4

- VTAM APPL definition

```
IMSPROD  APPL  ACBNAME=IMSPROD
          ...
          PERSIST= SINGLE | MULTI
          ...
```

▲ If VTAM MNPS is used:

- VTAM end nodes must be running with APPN/HPR
 - ▶ HPR (High Performance Routing) network environment
- All VTAMs must be connected to a coupling facility
 - ▶ Parallel Sysplex environment
 - ▶ Coupling Facility structure - ISTMNPS

MNPS requirements:

VTAM V4R4 - This release provides the VTAM code for MNPS support. Applications that choose to establish persistent sessions, must specify the PERSIST option in the APPL definition.

- PERSIST=SINGLE indicates that only single-node persistence is allowed.
- PERSIST=MULTI indicates multinode persistence is allowed in addition to single-node persistence support. If PERSIST=MULTI is coded and the application program is activated in an environment that does not support multinode persistent sessions then VTAM will issue an error message.

High Performance Routing (HPR) - HPR is an extension to APPN that provides nondisruptive path switching which can switch sessions around failed links or nodes. HPR connections are required to dynamically reroute session paths during failures in an MNPS environment.

Parallel Sysplex environment - VTAM uses the coupling facility in the MVS sysplex to maintain session and connection information for all multinode persistent session application programs.



RNR - Highlights...

▲ IMS Rapid Network Reconnect (RNR=)

- If specified, **always** establishes persistent sessions
- **/START DC** following IMS restart initiates RNR action

Two IMS levels of RNR support on a terminal level:

RNR=ARNR

Automated session reconnect

RNR=NRNR

CLSDST scheduled
Forced for MSC/VTAM

When the RNR option is specified in an IMS system, the associated sessions automatically become persistent. Persistence is maintained across an IMS, VTAM, MVS, or CEC failure.

On a subsequent /START DC process to activate the IMS network, the persistent sessions are either:

- terminated (NRNR), or
- automatically reconnected (ARNR) to a data traffic reset point equivalent to a logon without user signon data. This results in either a DFS3649 signon message if appropriate for the terminal type, or a terminal connected DFS3650 message.

VTAM MNPS support allows terminals to be defined with either session initiation/termination level of persistence or with a full active session send/receive level of persistence. IMS RNR takes advantage of the VTAM capability (introduced with UW50802) which allows IMS to request only session initiation/termination tracking (ARNR, NRNR options) rather than full session tracking (not supported by IMS RNR). This minimizes the number of accesses to the CF structure and the amount of data kept in the structure.



PSTIMER

▲ DFSDCxxx execution option: PSTIMER= 0/86400

- Specifies time VTAM waits for recovery of the persistent session before terminating sessions
 - ▶ 0 = no timer used (24 hours)
 - ▶ 1-84600 seconds (up to 24 hours)
 - ▶ IMS default = 3600 seconds (1 hour)
 - ▶ Null or invalid value = IMS default

VTAM start option (HPRPST) for MNPS will override PSTIMER if HPRPST is less than PSTIMER

The PSTIMER option specifies a 1-5 digit number of seconds. IMS tells VTAM to keep persistent sessions in pending recovery state for this amount of time after a failure. The timer is started by a failure (IMS, VTAM, MVS, or CEC) and continues until either the time limit expires or IMS issues the OPEN ACB. If the time expires prior to the OPEN ACB, VTAM terminates the sessions and releases the held resources.

Note the warning. If the VTAM value of HPRPST is set, then the lower of PSTIMER or HPRPST takes effect. The maximum value for HPRPST is 24 hours.



Terminal User / Remote Program

▲ At IMS, VTAM, CEC, or MVS failure, session persistence takes effect

- ARNR
 - ▶ Session is suspended until /STA DC
 - ▶ Session is reconnected
 - DFS3649 (signon) or DFS3650 (terminal connected) as appropriate
 - ▶ If session cannot be reconnected, an error message is sent to the MTO

- NRNR
 - ▶ Session is suspended until /STA DC
 - ▶ Session is terminated
 - ▶ Affinities are reset as appropriate
 - ▶ Terminal user can log back on

© IBM Corporation 2000, 2001

The end user impact of RNR is based upon the option that has been specified for the device by the DFSDCxxx system default or by any overrides in the ETO logon descriptor or the DFSLGNX0 exit.

Note that with RNR, if a failure occurs (IMS, VTAM, CEC, or MVS), the devices are suspended until the /STA DC can be issued. If IMS, MVS, or the CEC has failed, then the suspension also includes the time for an IMS restart. Depending on the RNR option, the sessions are then terminated (NRNR) or a session reconnect (ARNR) is attempted. If a session reconnect is unsuccessful, IMS sends an error message to the MTO.



APPC Persistent Sessions

▲ Persistent session support is provided by APPC/MVS

- Sessions are persistent, conversations are not
- SYS1.Parmlib (APPCPMxx)

```
LUADD
  ACBNAME (IMSLUA)
  SCHED (IMSA)
  BASE
  TPDATA (SYS1.APPCTP)
  PSTIMER (3600)
```

PSTIMER indicates the length of time that the sessions persist.
The VTAM definition for IMSLUA has PERSIST=SINGLE | MULTI.

APPC sessions are managed by APPC/MVS and VTAM. Support for persistent APPC sessions is, therefore, provided by APPC/MVS and not IMS. This is an example of an LUADD statement that defines an LU named IMSLUA with persistent session support. The specification of PSTIMER= as well as the definition of PERSIST=SINGLE or MULTI in the VTAM definitions enables the support for APPC/MVS.



Exit Routines



New TM and MSC Exit - DFMSCE0

▲ TM and MSC Message Routing and Control User Exit (DFMSCE0)

▲ New exit that combines and replaces:

- Terminal Routing Exit (DFSCMTR0)
- Input Message Routing Exit (DFSNPRT0)
- Link Receive Routing Exit routines (DFSCMLR0/DFSCMLR1)
- Program Routing Exit (DFSCMPR0)

Usability

▲ Eases coding by reducing the number of exit routines

▲ A consistent set of routing capabilities for all types of messages

▲ Provides the ability to attach a user prefix that follows the message and is passed to each exit interface

- Allows the message to be customized for accounting, security, etc.
- Not available to application programs

© IBM Corporation 2000, 2001

The new TM and MSC Message Routing and Control User Exit (DFMSCE0) combines, replaces, and enhances several of the existing message routing exits. As a consolidated exit, it is invoked at different entry points (terminal routing, link receive routing and program routing) and provides a variety of options such as the ability to: change the destination name, route messages to remote IMSs, keep the message on a local shared queues IMS system, reject the message, override directed routing and add a user prefix. The user prefix can be passed to each of the entry points and can be used to contain any data. This data could include accounting, statistical, or auditing information.

DFMSCE0 works with both MSC and non-MSC messages and lifts many of the restrictions of the previous routing exits. For example,

- the terminal routing entry point enhances the capabilities provided by DFSCMTR0/DFSNPRT0 by being given control at the same points as the previous exits as well as now being called for: ISC messages where the destination name is specified in the FMH headers, messages from terminals in preset destination mode, and any Front End Switch (FES) messages.
- the link receive routing entry point enhances the capabilities of DFSCMLR0/DFSCMLR1 by also being called for: intermediate messages, reply messages, and conversational transactions.
- the program routing entry point enhances the capabilities of DFSCMPR0 by receiving control for: ISRTs of both non-modifiable and modifiable ALT PCB messages, IOPCB messages, and conversational transactions.



Queue Space Notification Exit Enhancement

▲ Enhancement passes a stopped status to DFSQSPC0 for conversational destinations

- Previous releases passed stopped status only for non-conversational destinations

▲ Benefit

- Prevent looping applications from impacting the message queue for stopped conversational destinations
 - Application may receive "A7" status code

Availability

IMS V7 provides an enhancement to the existing DFSQSPC0 exit interface to address stopped conversational transaction destinations. Prior to this enhancement when a conversational program inserted a SPA, the insert was always accepted even if the SPA destination was stopped. The exit routine could not prevent it. In V7 the DFSQSPC0 exit routine is provided a mechanism to detect and address the stopped condition for both situations where:

- The SPA is inserted to another conversational transaction.
- The SPA is inserted back to the input device via the IOPCB.

The exit routine may cause an 'A7' status code to be returned to the application program's ISRT call. This indicates that the SPA was not inserted to its destination.



System Execution Enhancements



Deferred VTAM ACB Open

▲ New system option to delay the opening of the VTAM ACB until IMS is ready to accept logons

- /STA DC makes IMS ready to accept logons (with or without new option)

- **VACBOPN = INIT | DELAY**

- ▶ Specified in the DFSDCxxx member of PROCLIB
- ▶ INIT - Open ACB is issued during initialization (as before)
- ▶ DELAY - Open ACB is delayed until /STA DC



- Prevents potential queuing and timing out of logon requests
 - ▶ Impacts devices (e.g., ATMs) that immediately send in logon requests when IMS opens the VTAM ACB (e.g., during /ERE)

Availability

Performance

A new system option is provided that delays the queuing of the VTAM logon requests until IMS is ready to start accepting logons. This change delays the opening of the VTAM ACB until the processing of the /START DC command. With the use of the new option, VTAM logon requests are rejected by VTAM until the processing of the /STA DC command.

This is of value particularly in situations such as emergency restarts where there might exist elongated times between the opening of the VTAM ACB (when IMS comes up and when devices can begin sending in logon requests) and the actual time when the /STA DC command is entered (the time that IMS can begin processing the logon requests). These elongated times can result in network problems because the remote devices could time out, unbind, relogon, possibly time out again, etc. until the logons are actually accepted.

The VACBOPN option is specified in the DFSDCxxx member of PROCLIB. Valid values are INIT (default) and DELAY. INIT specifies that processing should continue as in previous releases. DELAY invokes the new support.



Security Enhancements

▲ Enhanced PassTicket Support (uses RACF or equivalent)

- New APPL keyword parameter on the /SIGN ON command
*/SIGN ON userid passticket **APPL applname***
- Provides greater flexibility for the end-user/program
 - ▶ PassTicket creation can use IMSID (same as before)
 - ▶ PassTicket creation can use the IMS VTAM application name
- Allows the creator of PassTickets to specify the value by which it knows IMS

▲ System-wide default **applname** for use in /SIGN ON with PassTicket

- SAPPLID=applname in DFSDCxxx
- Enables the use of PassTickets for VGR connections to IMS
 - ▶ Specify the same value in each IMS

DFSDCxxx

Usability

© IBM Corporation 2000, 2001

The RACF PassTicket is a one-time-only password that is generated by a requesting product or function. It is an alternative to a password and removes the need to send passwords across the network in clear text. It makes it possible to move the authentication of a mainframe user ID from RACF (or equivalent) to: (a) another authorized function executing on the host system or, (b) to the workstation environment. In prior IMS releases, when a /SIGN ON command was received by IMS that contained a PassTicket instead of a password, the signon process failed unless the PassTicket was created using the IMSID as the application name. Since the IMSID may not be known to other systems that might enter the signon command, more flexibility has been needed in the IMS processing of the PassTicket. The new keyword APPL in the /SIGN ON command allows the end-user or program to specify a name, e.g., the IMS VTAM application name, rather than the IMSID when creating the PassTicket.

In a VGR environment, the remote end user does not know which IMS will be chosen for the session. The user cannot be expected to know what value to use for "applname" in the /SIGN ON command. The SAPPLID parameter in the DFSDCxxx PROCLIB member overcomes this problem. It provides an IMS system-wide default name for applname in the /SIGN ON command. It is used with all /SIGN ON commands that include a passticket unless overridden by specifying APPL and applname. In a VGR environment, each IMS system should specify the VTAM generic name as the value of SAPPLID. This allows the user to /SIGN ON without having to specify APPL and applname.



Security Enhancements

▲ USERID Clarification

- An indicator associated with the *userid* field that defines its content
 - ▶ USERID, PSB name, LTERM name, or other
 - ▶ Added to IOPCB, INQY ENVIRON call, and exit parameter lists
 - For example, DFSBSEX0 (Build Security Environment) exit
- Provides a method that allows IMS application programs and exits to determine whether a user was signed on at the time a transaction was entered

Usability

When signon security is active in an IMS system, the userid field can contain: the user's identification from the source terminal during signon, the LTERM name of the source terminal if signon is not active, or the PSB name of a source BMP or transaction. Many IMS applications depend on knowing whether or not the value reflects a userid in order to perform different application functions. A common method to determine this is to check for the equality of the values in the USERID and LTERM fields. If the USERID is equal to the LTERM, it is assumed that this is not a true USERID. With ETO support, however, the method of equality of these values cannot be used in situations where the LTERM and USER structures are created based on the userid name. In this case, the LTERM is always equal to the USERID.

The USERID clarification enhancement provides a method by which IMS application programs, either by querying the IOPCB or by issuing an INQY call, can determine the nature of the value passed in the USERID field. The information is also provided to several user exits.



SLUP/Finance Session Cold Termination

▲ Extension to the /CHANGE NODE command for SLUP/Finance

/CHANGE NODE *nodename* | *nodename** | **ALL COLDSESS**

- New keyword COLDSESS
 - ▶ Resets status to 'COLD'
 - ▶ Impacts devices that are **not** in session and **are** idle
 - ▶ Applicable to ETO and static terminals

- Allows a forced reset of terminals via command if an unrecoverable STSN sequence number mismatch occurs during system warm start

- Allows applicable ETO control blocks to be cleaned up at next system checkpoint

Availability

© IBM Corporation 2000, 2001

IMS starts a SLUP or Finance session 'COLD' after every IMS cold start. All subsequent sessions for the same device are restarted 'WARM' until IMS is again cold started. Warm restarts synchronize SNA STSN sequence numbers. In the case of an unrecoverable STSN sequence number mismatch during a session 'WARM' restart, the device will be unable to connect.

The /CHA NODE ... COLDSESS command enhancement allows IMS to force a cold session initiation without also requiring an IMS cold start when the terminal and IMS experience a non-recoverable attempt to 'warm' start.

For ETO environments, the enhancement further enables IMS to cleanup control blocks that were dynamically created from an incorrect ETO session initiation request, e.g., incorrect LU name. In the situation where the control blocks are incorrect and the command enhancement is NOT used to reset the state to 'COLD', the control blocks remain in the system and are recovered across IMS checkpoints. In these cases, the associated storage and IMS checkpoint/restart times for the 'dead' blocks could affect IMS performance.



Spool Enhancements

▲ Change to the way EOF markers are written to spool data sets

- Improves spool performance by reducing EOF writes from one per record to one per track

Performance

▲ A new IMSWT = yyyyy parameter in DFSDCxxx

DFSDCxxx

- Identifies the first 5 characters to use when auto scheduling the spool print utility
- Example:
 - IMSWT=IMSA causes IMS to issue: /STA REGION IMSA000 command to print the first spool line data
 - If IMSWT= is not coded, IMSWT is used as a default
- Facilitates the use of cloned IMS SYSGENS and PROCLIBs in a Parallel Sysplex environment
 - Each IMS generates unique spool print JCL

Usability

© IBM Corporation 2000, 2001

The spool data set support allows users to browse a spool data set via TSO as it is being written by IMS. Prior to IMS V7, an end of file (EOF) mark is written after each message to delineate the end of the current data. The EOF marks are overwritten by each write of a new record. This is done by using BSAM and issuing a 'Close Type=T' macro. This can cause extra writes to the VTOC to update the Format 1 DSCB. Up to 7 I/Os may be required to complete writing a single record. The enhancement in V7 improves the performance of the EOF writes by using a more efficient EXCP call to write the EOF mark and by only writing one per track.

The second enhancement implements a new option, IMSWT=, in the DFSDCxxx member of PROCLIB. This parameter is of value in a Parallel Sysplex environment. When auto scheduling of the spool print utility takes place in a cloned IMS using a shared PROCLIB, there is no guarantee that the correct JCL will be generated. The generated JCL depends only on the spool line number and the member of PROCLIB. To overcome this problem in the past, many customers came up with different bypass solutions. Some applied usermods to the spool device dependent module DFSDN155. Others generated as many spool lines in each cloned IMS as existed in the entire sysplex. With IMS V7, the new parameter IMSWT= allows the automatic generation of a unique jobnames (members in IMS.JOBS) for each cloned IMS system.



SLU2 Enhancement

▲ New DFSDCxxx option: **SLU2=EXR/NOEXR**



DFSDCxxx

- Specifies whether or not to suppress the SNA exception response prior to sending a DFS error message during error recovery processing
- Addresses Program check/keyboard lock for SLU2 devices that implement DFT (Distributed Function Terminal) architecture
 - ▶ Applies to static and ETO terminals

Usability

© IBM Corporation 2000, 2001

A Distributed Function Terminal (DFT) is a programmable 3270 terminal that can perform operations previously performed by the control unit. These terminals can directly interpret the 3270 data stream. Devices that implement the DFT architecture include 327x hardware, such as 3192-G and 3194-F/H devices, and emulation products such as CM/2 and PC3270. Due to a SNA deviation that was implemented by this architecture, DFT devices can suffer from PROG726 errors and keyboard locks when IMS sends out SNA exception responses that precede DFS error messages.

The option SLU2= allows an installation to specify whether or not to suppress the SNA exception response prior to sending a DFS error message. When it is used, DFT terminals will not be the PROG726 errors or keyboard locks when DFS error messages are sent.



ETO Enhancements

▲ Descriptor Limitation (50 records/descriptor) has been removed

▲ Associated Print Support Enhancements

- Allows more timely delivery of output messages for Associated Printers regardless of where the transaction is processed in a Shared Queues environment

▲ Autologon

- Enhancements for Associated Printers, Dynamic terminals activated via /OPNDST
- New keywords in the /CHANGE command
 - Update autologon information, e.g., Mode
 - SAVE|NOSAVE across restarts

Usability

▲ LTERM Assignment

- Allows users and LTERMs to be moved between printers more easily
 - Assignments can persist across session and system restarts

© IBM Corporation 2000, 2001

Prior to IMS V7, a descriptor that contained more than 50 records was ignored and an error message issued. This limited, for example, the number of remote LTERMs that could be defined on a single MSC link in an environment that used ETO MSC descriptors. In IMS V7, a new algorithm is used to remove the limit on the number of records that can be defined for a descriptor.

The ETO Associated Printer support is a technique that allows an end-user (at signon) to specify or associate a specific printer or printers for output messages. When an end-user signs on to an IMS system, the Associated Print (ASP) user structures, if specified, are created. For a Shared Queues environment, this IMS is considered the front-end for the end-user. The programs that process the input messages, however, may run in either the front-end IMS or in a back-end IMS system. IMS V7 enhances the associated printer support in a Shared Queues environment to ensure more timely delivery of the printer output. This is done by increasing the number of times when interest is registered in the associated printer LTERMs.

The autologon capability in ETO allows IMS to dynamically acquire a NODE as a result of queued output. It is applicable to all terminal types but primarily used for printers. Autologon processing in IMS V7 has been improved to be more compatible with associated printers as well as with printers that have been started interactively. In prior releases, the /OPNDST command to start a printer disabled the automatic autologon processing for the printer. IMS V7 does not disable this capability. Additionally, the autologon parameters can be modified using the /CHA command and saved across a subsequent restart.

The enhancements for ETO LTERM processing in IMS V7 include:

- the ability to assign LTERMs between users regardless of whether or not the target user structure exists.
- the capability to maintain persistence and remember the assignments across: session failure and restart, system failure and restart, and CLSDST due to printer sharing.



Connectivity Enhancements



OTMA Callable Interface

▲ Introduced in IMS/ESA V6 (PQ17203) - Delivered with IMS V7

▲ A high-level C/C++ API interface for OS/390 applications and subsystems

- Provides access to IMS transactions and commands through OTMA
- Provides a high-level interface for non-authorized and authorized programs to invoke OTMA facilities
 - Facilitates the coding of an OTMA client
 - Hides the complexity of XCF and OTMA

Availability

Usability

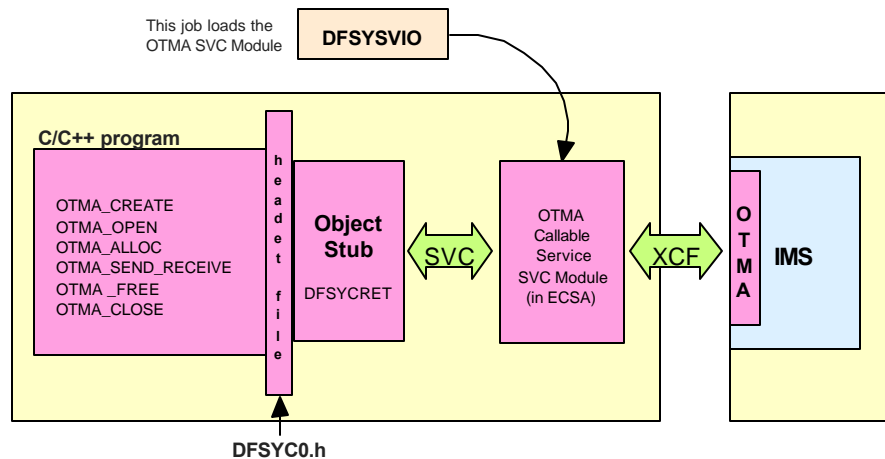
The Open Transaction Manager Access (OTMA) interface was introduced in IMS/ESA V5 to provide a standardized access mechanism (XCF services) from MVS address spaces into the IMS TM component. The applications or subsystems that use this interface invoke IMS services to access transactions and commands and are called OTMA clients.

XCF (Cross System Coupling Facility) is an MVS service which provides a high speed connection capability between two MVS address spaces and requires the calling programs (IMS OTMA and the OTMA client) to be authorized. Additionally, the direct use of XCF services requires portions of the OTMA client to be written in assembler to allow execution in both TCB and SRB mode. This imposes a degree of coding complexity beyond the scope of the average application or system programmer. Most of the OTMA clients that are available today are provided by vendors. Examples of existing clients are the MQSeries Bridge for IMS and the IMS TCP/IP OTMA Connection (ITOC).

The OTMA Callable Interface simplifies this development environment by providing a high-level C/C++ interface to invoke the XCF services. It can be used by non-authorized as well as by authorized programs. The support was initially introduced by APAR in IMS/ESA V6 and is now delivered as part of base IMS V7.



OTMA Callable Interface Flow



- Program is linked (bound) with object stub
- OTMA_xxx calls invoke object stub which issues SVC
- SVC module issues XCF requests to OTMA function in IMS

© IBM Corporation 2000, 2001

The OTMA CI environment consists of header files in the C/C++ program, object stubs and several interface modules.

The header file DFSYC0.h defines the OTMA API for the programs and has to be included with the C/C++ program. This is delivered in the IMS ADFSMAC data set.

The object stub, DFSYCRET, provides the underlying support for the API calls and issues an SVC 146 which passes control to the OTMA SVC interface module. This stub must be provided to the program during the bind process. It is delivered in the IMS ADFSLOAD data set.

Note: IMS provides basic systems services to new address spaces through a component called Base Primitive Environment (BPE). MVS has assigned SVC 146 to BPE. The SVC allows callers to register a name and associate that name with an SVC interface routine.

The OTMA Callable Services SVC interface module is the module that is called in supervisor state when the SVC is invoked with the registered name.

Setup: This module needs to be loaded and registered to the BPE SVC services by an authorized address space. A stand-alone program, DFSYSVIO, is provided to do this. The program must be run after an MVS IPL. Once the OTMA SVC module is registered with BPE, it is available for use by authorized or non-authorized programs.

Flow:

When a C/C++ program issues an OTMA CI request, the object stub DFSYCRET that is linked with the program issues a BPE SVC (146) request which passes the OTMA function name. When the OTMA Callable Services SVC module receives the request, security and validity checking is performed and an input request is built for OTMA/IMS.



IMS Connect

▲ Replacement product for IMS TCP/IP OTMA Connection (IMS TOC)

- Usable with IMS V5, V6, and V7

▲ Enhancements:

- SMP Installability
- Persistent Sockets
- Asynchronous Output Support
- Initialization Exit
- Dump Formatting capability

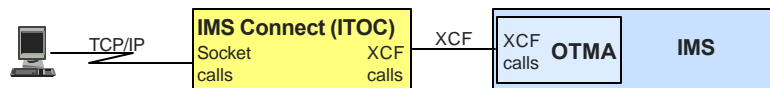
Performance

Usability

Availability

▲ Executes in its own MVS address space

- Functions as a TCP/IP server for communication with external clients
- Uses MVS XCF Services to access IMS OTMA



© IBM Corporation 2000, 2001

IMS Connect is a new product (5655-E51). It incorporates and enhances the TCP/IP socket connection support that was originally called the IMS TCP/IP OTMA Connection (ITOC). IMS Connect is usable with IMS V5, V6, and V7. The asynchronous output message capability requires the use of IMS V7. Other IMS Connect capabilities are usable with IMS V5, IMS V6, and IMS V7.

In addition to the capabilities of ITOC, IMS Connect provides:

- SMP installability and maintenance.
- Persistent sockets support
- Support for asynchronous output messages (requires IMS V7)
- An initialization exit
- Dump formatting capability to assist in problem determination.

Persistent sockets, asynchronous support, and the installation exit are explained on the next page.

IMS Connect runs in its own address space. It functions as a TCP/IP server for communications with external clients. TCP/IP clients communicate with IMS Connect, which passes input messages to IMS. IMS Connect uses MVS XCF services to communicate with the OTMA component of IMS. Output messages from IMS are sent via OTMA to IMS Connect. IMS Connect sends these output messages to the TCP/IP clients.



IMS Connect Enhancements

▲ Persistent Sockets

- Capability that allows the socket connection to remain active for multiple transaction iterations
- Only supported for Send-then-commit (Commit mode 1)

Performance

▲ Asynchronous output support

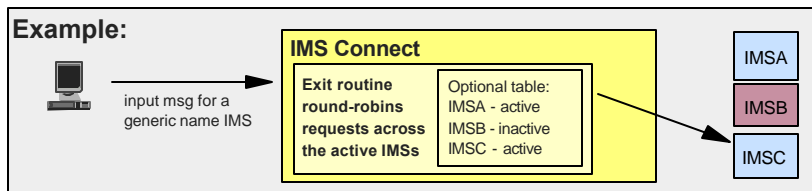
- ISRT's to Alternate TP PCBs (ALTPCB)
- Requires IMS V7

Usability

▲ Initialization exit

- User may write routine to route input to available IMS

Availability



© IBM Corporation 2000, 2001

Persistent sockets support allows a TCP/IP connection to remain active across multiple transaction iterations. It is provided for Send-then-commit (commit mode 1) transactions.

Asynchronous output support allows IMS to send messages which are not response messages a remote TCP/IP client. These are messages inserted using an alternate TP PCB. This requires the use of IMS V7.

IMS Connect includes an initialization exit. This exit is passed the status (active or not active) of the IMS systems to which it can connect. This information may be used to route a message to one of multiple candidate systems. This is especially useful in Parallel Sysplex environments where multiple IMS systems can provide the same services.



Application Development



IMS Java

▲ New function of IMS V7 TM

▲ Capability to write, compile and run IMS Java programs

- Development environment
 - ▶ Provides a set of packages (groups of classes)
 - Allow access to IMS services
 - Support APIs familiar to Java programmers
 - ▶ Uses the S/390 HPJ (high performance java) compiler
 - Compiles bytecode into high-performance executables/DLLs
- Runtime environment supports
 - ▶ Dependent regions (MPP, IFP, and BMP)

▲ Benefit

- Incorporation of the Java programming model into the IMS environment

© IBM Corporation 2000, 2001

IMS Version 7 adds the capability to run Java application programs in dependent regions of IMS TM systems.

A set of Java packages, or groups of classes, are provided. They allow access to IMS services, such as database calls and IMS message calls. This interface is familiar to Java programmers.

IMS Java uses the S/390 High Performance Java (HPJ) compiler. Programs are compiled from Java bytecode into high-performance S/390 executables. These are executed in the IMS dependent regions.

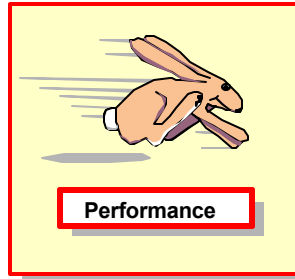
IMS Java allows application programmers familiar with Java to easily develop IMS applications with minimal of IMS training.



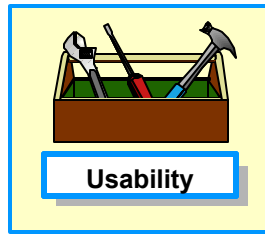
IMS Version 7



- HALDB
- ORS
- RNR
- VTAM Gen. Resources
- DBRC
- Change Accum
- IC2
- DEDB I/O error handling
- SLUP/Finance Session Cold Termination
- ...



- HALDB
- ORS
- DBRC
- Change Accum
- ACBGEN
- Async OTMA/APPC
- Deferred ACB Open
- ...



- HALDB
- ORS
- ODBA
- Install/IVP
- IMS Connect
- DBRC
- ACBGEN
- Change Accum
- RACF PassTicket
- ...

© IBM Corporation 2000, 2001

IMS Version 7 provides a wide variety of enhancements. IMS installations may migrate to IMS Version 7 to increase their system's availability, improve its performance, and enhance its usability.