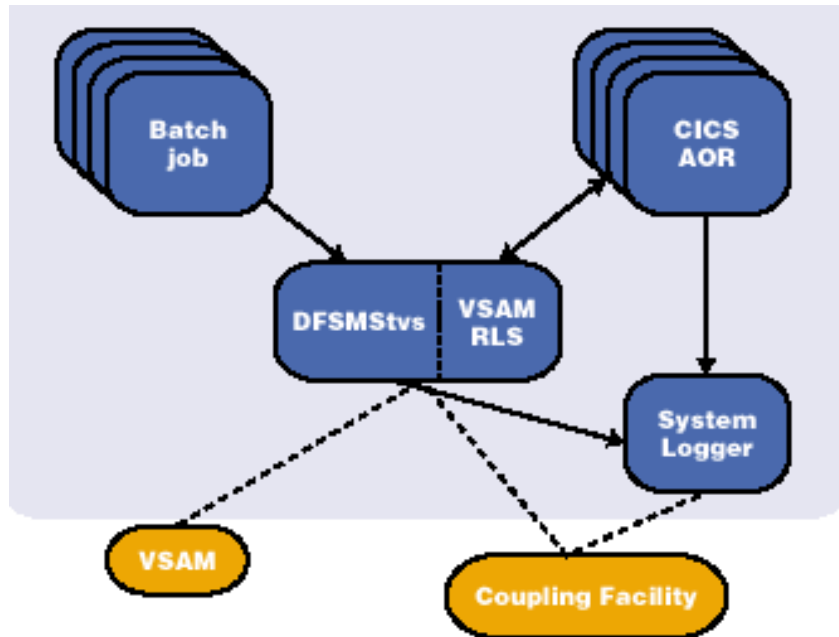


Transactional VSAM: An Application Programmer's Perspective



IBM zSeries Technical Conference
Session TSS05

Ruth Ferziger

IBM Design & Development

ruthf@us.ibm.com

Agenda

- What is Transactional VSAM?
- How do you use Transactional VSAM?
- How Transactional VSAM fits into the system
- Batch job considerations
- Peer Recovery

What is Transactional VSAM?

- **Objective: Provide transactional recovery within VSAM**
- RLS allows batch sharing of recoverable data sets for *reac*
 - ▶ RLS provides locking & buffer coherency
 - ▶ CICS provides logging & two-phase commit protocols
- Transactional VSAM allows batch sharing of recoverable data sets for *update*
 - ▶ Logging provided using the MVS System Logger
 - ▶ Two-phase commit & back out using MVS Recoverable Resource Management Services (RRMS)

Transactional VSAM Overview

- Transactional VSAM uses
 - ▶ Recoverable Resource Management Services (RRMS) manages the unit of recovery (UR)
 - ▶ System Logger manages the log streams
 - ▶ VSAM RLS manages locking & buffer coherency
- Allows atomic commit of changes -- all or nothing
- Transactional VSAM does *not* do forward recovery!

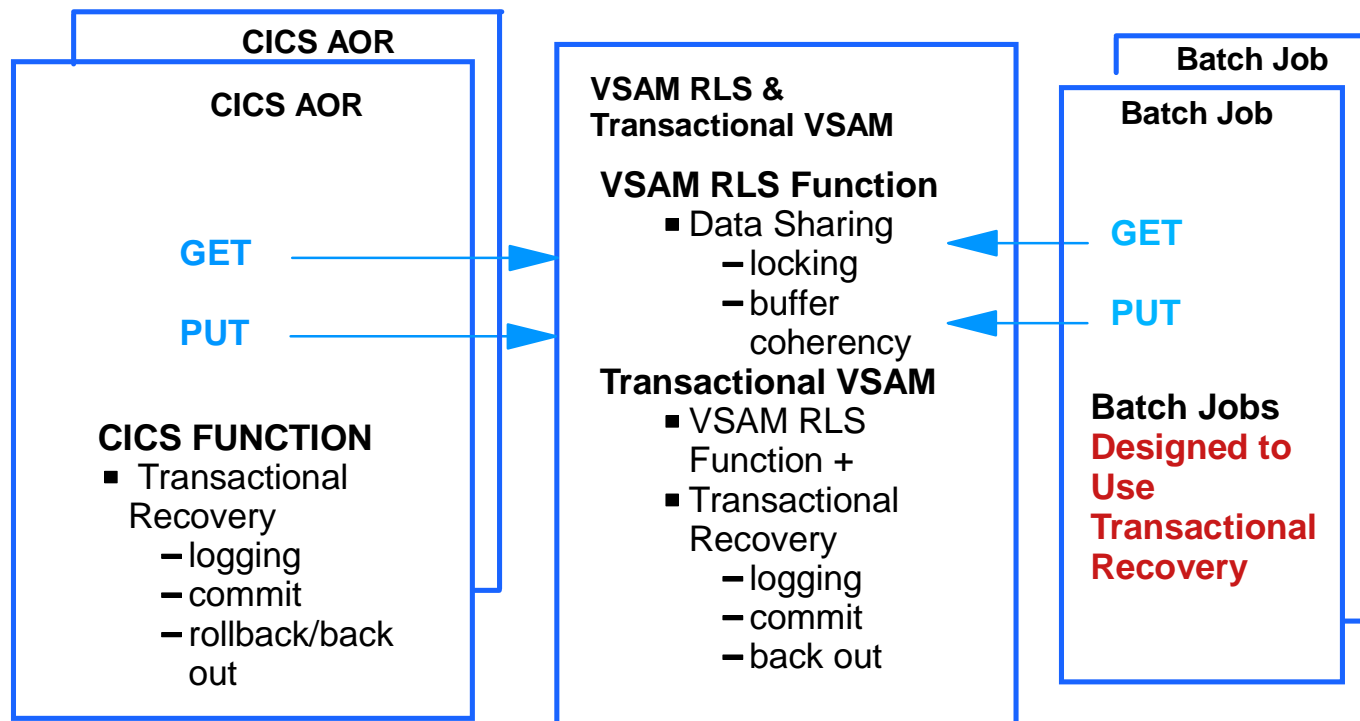
Transactional VSAM Log Streams

- Each MVS image has two System Log Streams
 - ▶ Short-lived transactions (primary system log)
 - ▶ Long-lived transactions (secondary system log, or shunt log)
- Forward Recovery Log Streams
- Log of Logs (can be same as CICS)
- All log streams must be predefined before Transactional VSAM is started

Accessing a Data Set with Transactional VSAM

- Define VSAM data set as recoverable
 - ▶ LOG(UNDO)
 - ▶ LOG(ALL)
- Specify RLS access in the ACB or the JCL
 - ▶ Open for input with read integrity option CRE *or*
 - ▶ Open for output
- Interface with the RRS component of RRMS to define sync points (commit or back out)

Using Transactional VSAM



Concurrent Read/Write Sharing of **RECOVERABLE FILES**
Across CICS and Batch Jobs

What Transactional VSAM Does

Application

RLS & Transactional VSAM

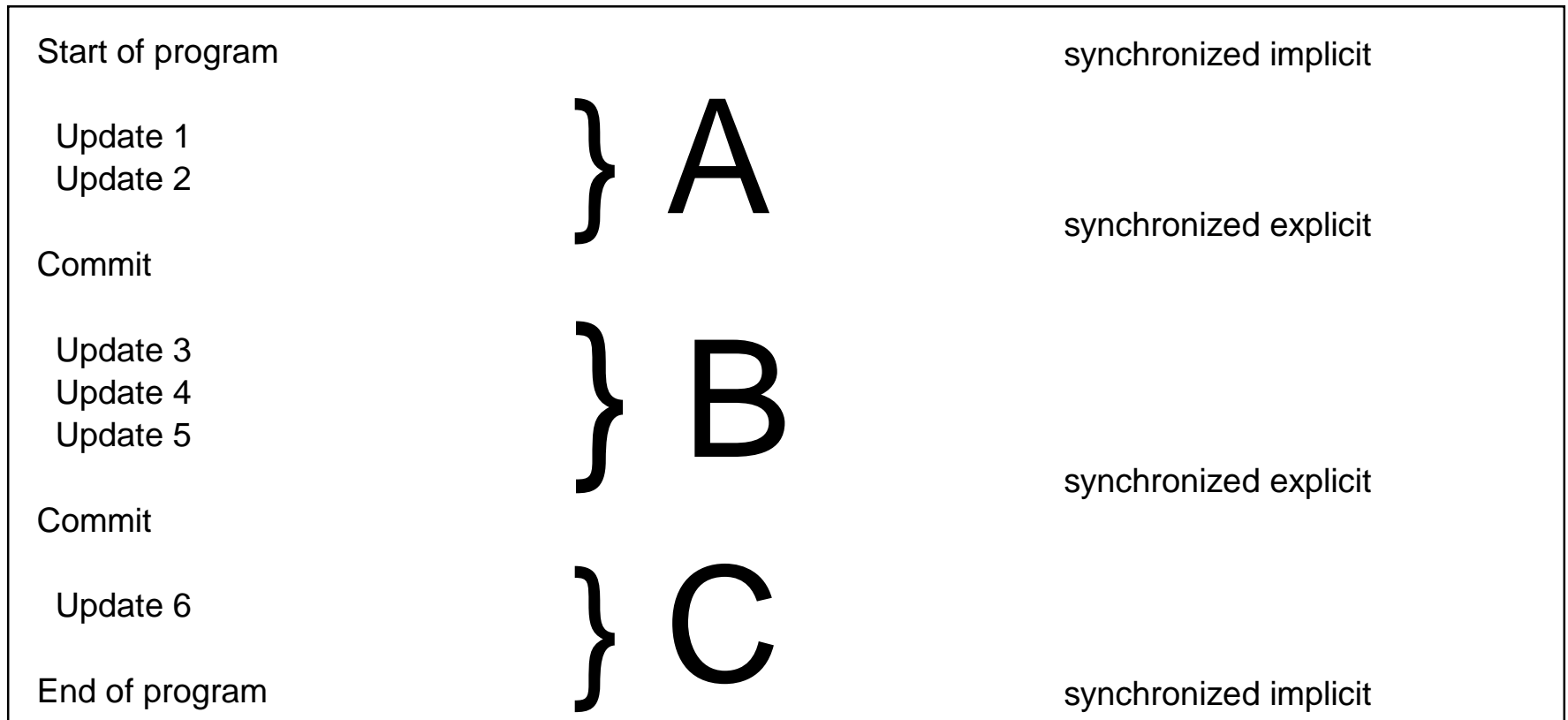
GET UPD -----	Obtain Lock, Log Undo
PUT UPD -----	Log Redo
GET repeatable read -----	Obtain Lock
PUT Add -----	Obtain Lock, Log Undo/Redo
GET UPD -----	Obtain Lock, Log Undo
PUT UPD -----	Log Redo
Call SRRRCMIT -----	Commit Changes, Release Locks

What is a Transaction?

Application	Result
Start of Transaction -----	No locks held
GET UPD record 1-----	Obtain an exclusive lock on record 1
PUT UPD record 1 -----	Lock on record 1 remains held
GET repeatable read record n-----	Obtain a shared lock on record n
PUT ADD record n+1-----	Obtain an exclusive lock on record n+1
GET UPD record 2 -----	Obtain an exclusive lock on record 2
GET UPD record 2 -----	Lock on record 2 remains held
Call SRRCMIT -----	Commit changes, all locks released

Unit of Recovery

- Set of changes processed by a single commit of backout is call a *unit of recovery*



Application Interfaces: VSAM

■ File Attributes

- ▶ LOG (NONE or UNDO or ALL)
- ▶ LOGSTREAMID

■ Transactional VSAM access to a file

- ▶ JCL RLS=CRE and OPEN for Input
- ▶ JCL RLS=NRI or CRE, recoverable file, and OPEN for Output
- ▶ ACB MACRF=RLS, recoverable file, and OPEN for Output
- ▶ ACB MACRF=RLS, ACB RLSREAD=CRE

■ Standard VSAM interfaces

- ▶ OPEN/CLOSE, GET/PUT/ERASE/POINT/ENDREQ
- ▶ New GET repeatable read option (CRE)
- ▶ Some new error return codes

Application Interfaces: VSAM....

The table below shows the valid data set OPEN combinations for SHAREOPTION 2

- Left side shows existing open
- Column headings indicate subsequent open

Existing Open	Non-RLS read	Non-RLS update	RLS read	RLS update	TranVSAM read	TranVSAM update
Non-RLS read	Yes	Yes	Yes	Yes	Yes	Yes
Non-RLS update	Yes	No	No	No	No	No
RLS read	Yes	No	Yes	Yes	Yes	Yes
RLS update	Yes	No	Yes	Yes	Yes	Yes
TranVSAM read	Yes	No	Yes	Yes	Yes	Yes
TranVSAM update	Yes	No	Yes	Yes	Yes	Yes

Application Interfaces: VSAM....

The table below shows the type of OPEN resulting from the parameters specified

- Left side shows the type of data set & type of open
- Column headings indicate the RLS option specified

Data Set Type & Type of OPEN	NRI	CR	CRE
Recoverable Open for Input	RLS	RLS	Transactional VSAM
Recoverable Open for Output	Transactional VSAM	Transactional VSAM	Transactional VSAM
Nonrecoverable Open for Input	RLS	RLS	Transactional VSAM
Nonrecoverable Open for Output	RLS	RLS	Transactional VSAM

New VSAM Error Codes

RC	Rsn	Explanation
RPL 8	40 (28)	Transactional VSAM was unable to expand the pool for its context/UR related control blocks.
	92 (5C)	PUT UPD or ERASE was issued without previous GET UPD in the same unit of recovery.
	205 (CD)	Transactional VSAM restarted while the UR was in-flight. To continue processing, the application must issue a commit or a back out and then begin a new unit of recovery.
	206 (CE)	The data set is quiesced or quiescing for copy. Wait for the data set to be unquiesced and then retry the request.
	207 (CF)	Transactional VSAM is quiescing or disabling. Close all data sets so that the process can complete.
	209 (D1)	Transactional VSAM was unable to complete the request because the forward recovery log is disabling.
	210 (D2)	The record length is greater than the installation maximum supported by the forward recovery log.

New VSAM Error Codes....

RC	Rsn	Explanation
RPL 8	211 (D3)	A permanent I/O error was detected on the forward recovery log. See accompanying Transactional VSAM logger messages for appropriate action.
	213 (D5)	Transactional VSAM was unable to complete the request because the undo log is unavailable.
	214 (D6)	A permanent I/O error was detected on the undo log. See accompanying Transactional VSAM logger messages for appropriate action.
	217 (D9)	RRS restarted while the UR was in-flight. To continue processing, the application must issue a commit or a back out and then begin a new unit of recovery.
	220 (DC)	Transactional VSAM was unable to complete the request because RRS is currently unavailable
	235 (EB)	VSAM RLS or Transactional VSAM internal error
	249 (F9)	The record length is greater than the installation maximum supported by the undo log.
RPL 16	12 (C)	Transactional VSAM processing is currently unavailable because Transactional VSAM is initializing.

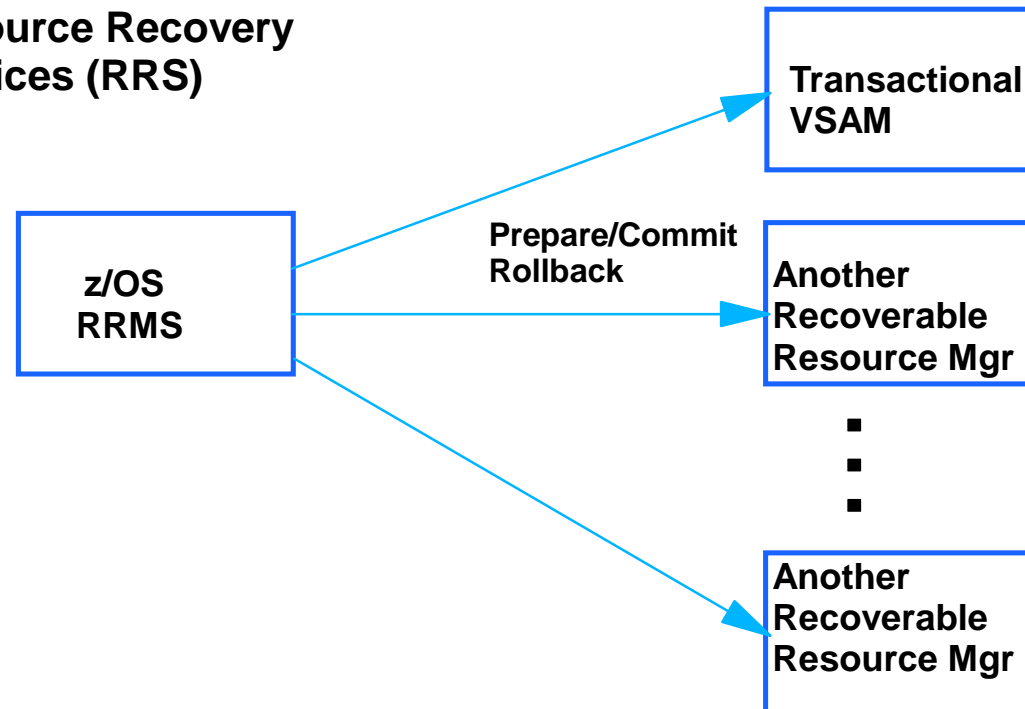
New VSAM Error Codes....

RC	Rsn	Explanation
Open 8	132 (84)	Unable to connect to redo log, unable to write tie up record, data set cannot be opened because forward recovery is required, or Transactional VSAM is not available; see message IEC161I for more information.
	174 (AE)	The LOG parameter is ALL but LOGSTREAMID is not specified
	213 (D5)	The LOG parameter is ALL but LOGSTREAMID specifies a Transactional VSAM system log
	214 (D6)	The maximum logical record length for the data set is greater than the maximum Transactional VSAM supports
Close 4	172 (AC)	Close was successful, but an attempt to disconnect from the redo log was unsuccessful

Transactional VSAM & z/OS RRS

z/OS RRMS:

- Registration Services
- Context Services
- Resource Recovery Services (RRS)



Application Interfaces: RRS

- **Unit of Recovery (UR) identifier**
 - ▶ provided by RRS
 - ▶ associated with the current TCB
- **Application requests Commit/Back out via RRS interfaces**
 - ▶ Call SRRRCMIT or SRRBACK
 - ▶ RRS drives Transactional VSAM
 - ▶ RRS drives other resources managers (DB2,MQ,....)
- **Commit or back out can be either explicit or implicit**
 - ▶ explicit when invoked by application
 - ▶ implicit at end of step/job

Supported Languages

- **PLI**
- **C & C++**
- **COBOL**
- **Assembler**

Application Redesign

- Break processing into a series of transactions
 - ▶ invoke RRS for commit and back out
- Modify program/JCL to request Transaction VSAM access
 - ▶ specify via JCL or ACB
- Prevent multiple RPLs from causing intra-LUWID lock contention
- Handle potential loss of positioning at sync point
- Handle all work that is part of one UR under the same context
- Do not use file backup/restore as job restart technique

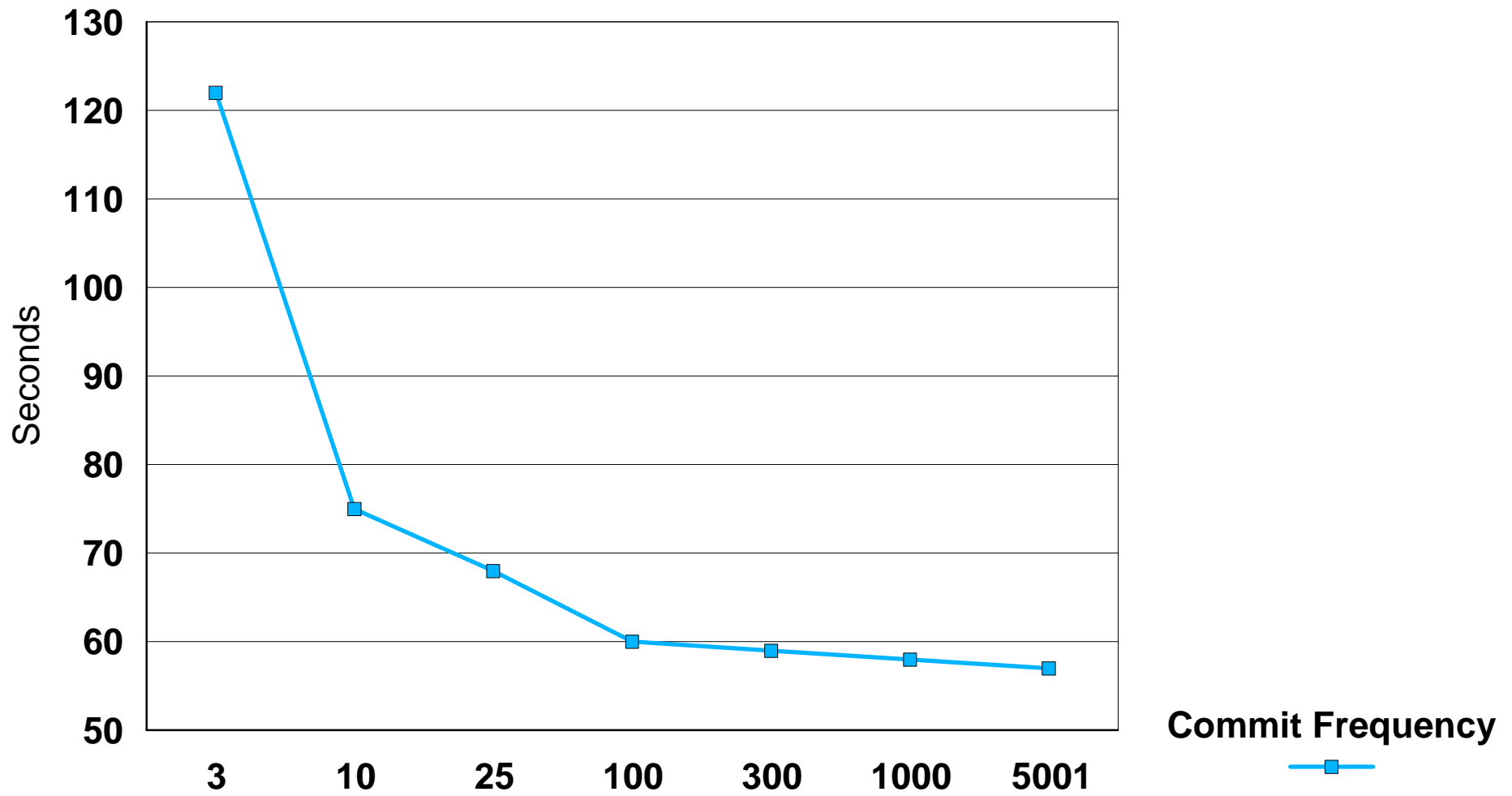
Application Considerations

- **Application must have transaction perspective when accessing shared VSAM files**
 - ▶ RLS & Transactional VSAM provide isolation until commit/back out:
 - share locks on records accessed with "repeatable read"
 - write locks on changed records - held until end of transaction
 - commit applies all changes and releases all locks
 - ▶ Application should issue frequent sync points (commit or back out)
- **Information extracted from shared files must not be used across commit/back out**
 - ▶ need to reaccess the record(s)
 - ▶ cannot get a record before a sync point and update it after
 - ▶ do not position to a record before a sync point and access it after

Application Considerations....

- **Application must hold read or write lock on record to keep it from being changed**
 - ▶ Use CRE (repeatable read) on GET
 - hold lock until end-of-transaction (commit or back out)
 - causes updaters to wait
 - use only when repeatable read is absolutely required
 - ▶ GET UPD holds write lock until end-of-transaction
 - undo log record is written at time of GET UP
 - lock is NOT released by PUT UPD or ERASE
 - ▶ PUT NUP (insert) holds write lock until end-of-transaction
- **All locks are released at commit or back out**

Application Considerations -- Commit Frequency



Exclusive Control of Resources

- GET UPD & PUT NUP obtain an exclusive lock
 - Updating records
 - Erasing (deleting) records
 - Adding records
- Lock is NOT released by PUT/ERASE
- Paired PUT/ERASE should be done in proximity to its GET UPD
- Take sync points (commit or back out) frequently

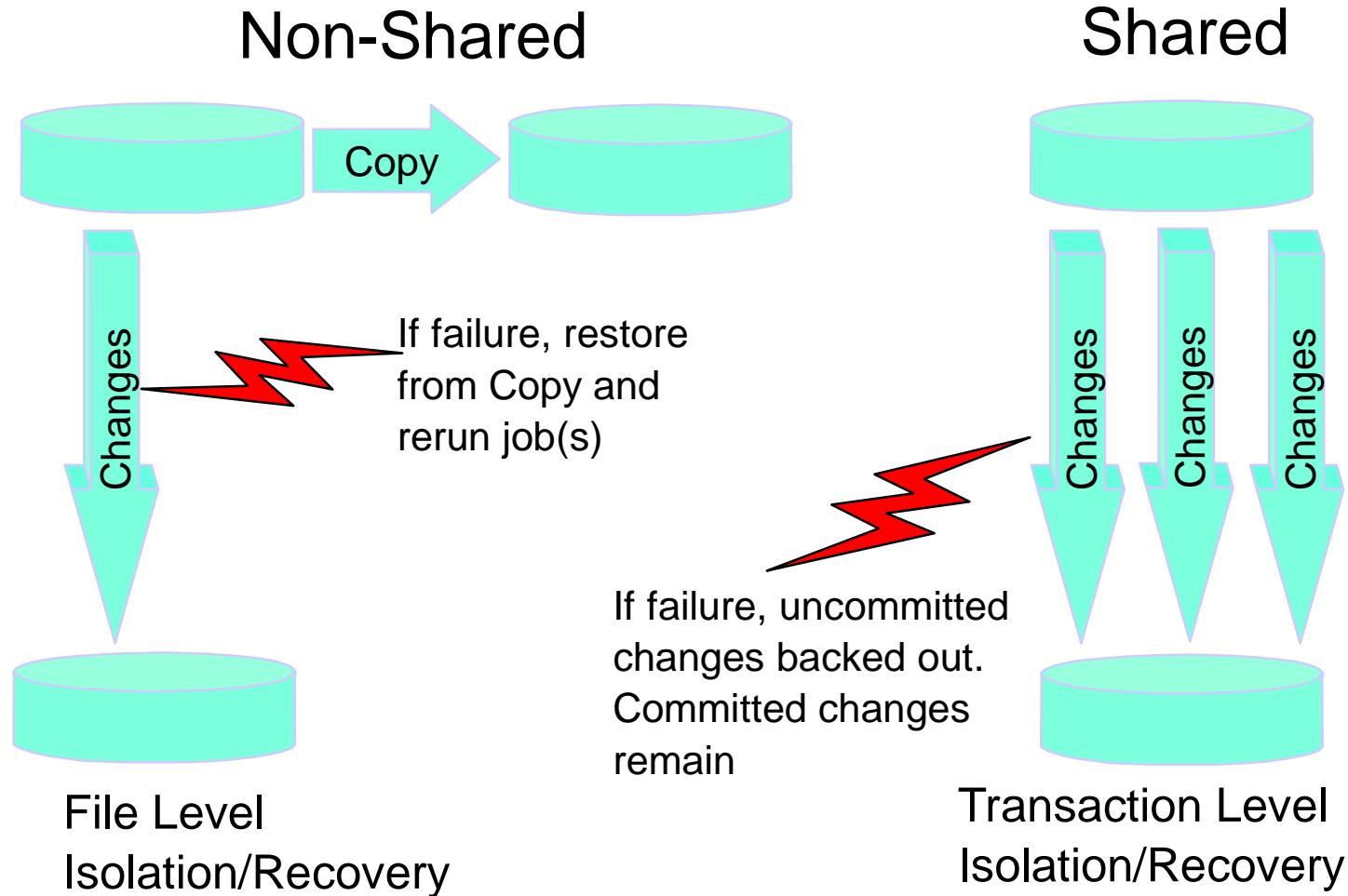
Browsing

- Do NOT use GET UPD
 - Obtains an exclusive locks
 - Writes a record to the undo log
- Use GET NUP with the appropriate read integrity option
 - NRI -- no read integrity
 - CR -- consistent read
 - CRE -- consistent read explicit (repeatable read)

Avoiding Deadlocks

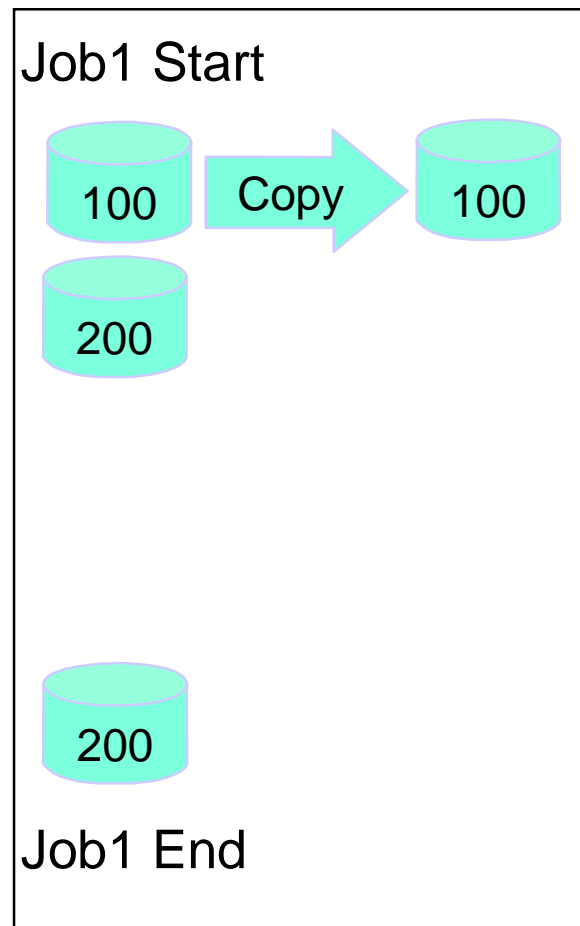
- Access records in a consistent order
 - Application A gets record 1 of data set X, then tries to get record 2
 - Application B gets record 2 of data set X, then tries to get record 1
- Access data sets in a consistent order
 - Application A gets record 1 of data set X then gets record 1 of data set Y
 - Application B gets record 1 of data set Y then gets record 1 of data set X
- Be careful when using alternate indexes

Non-Shared vs. Shared: Job Rerun

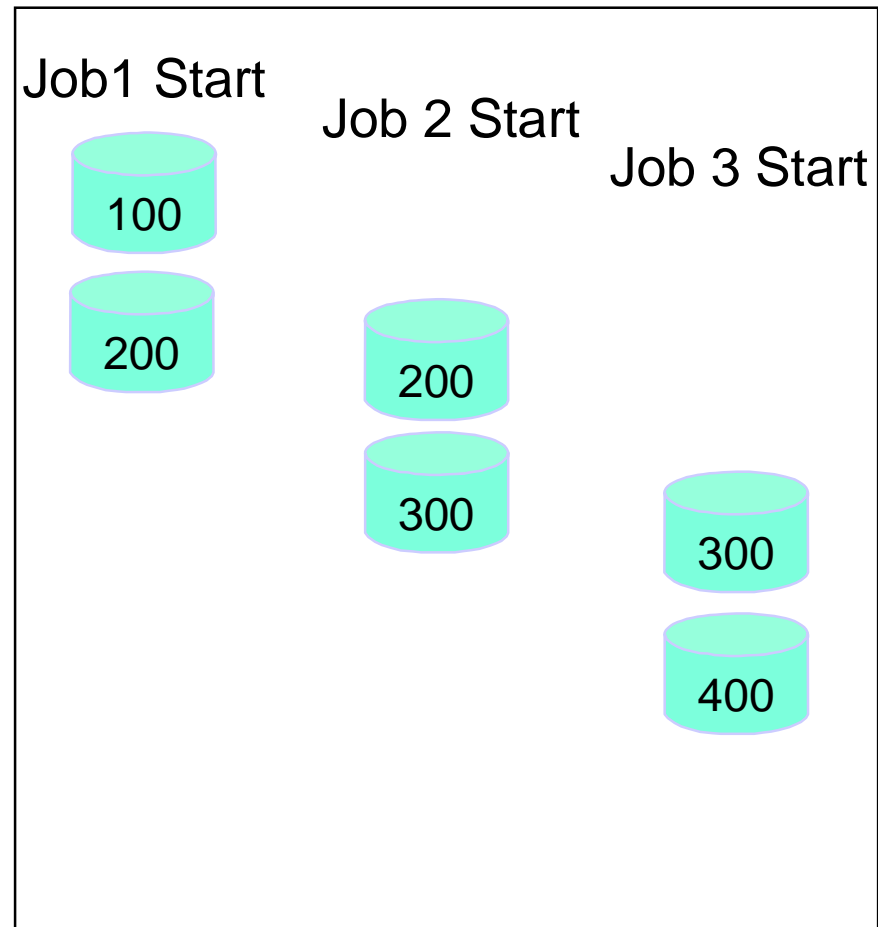


Job Restart: Change Isolation

Non-Shared



Shared



Failed Job Rerun

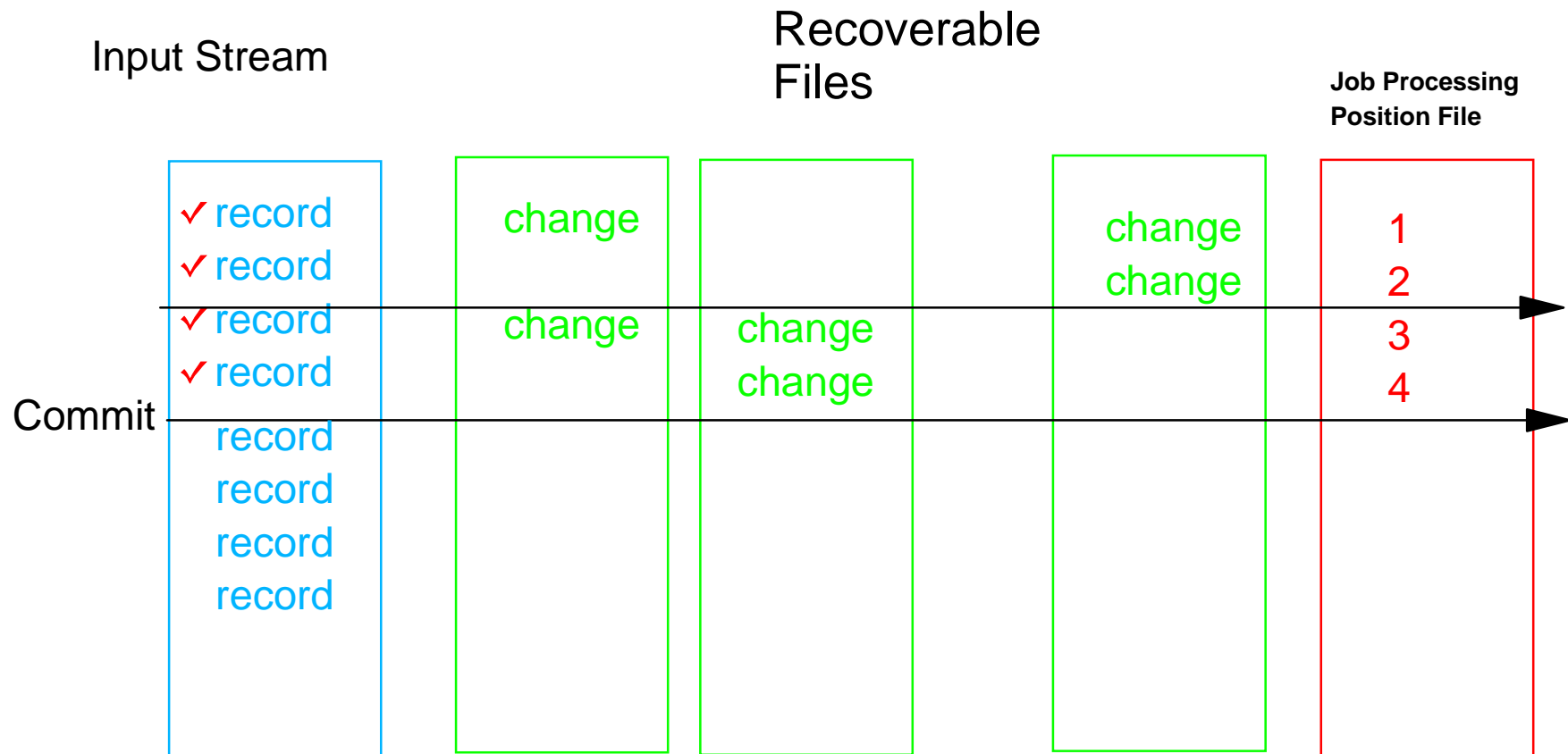
Non-Shared

- ▶ Restore File from Copy
- ▶ Reapply All Changes

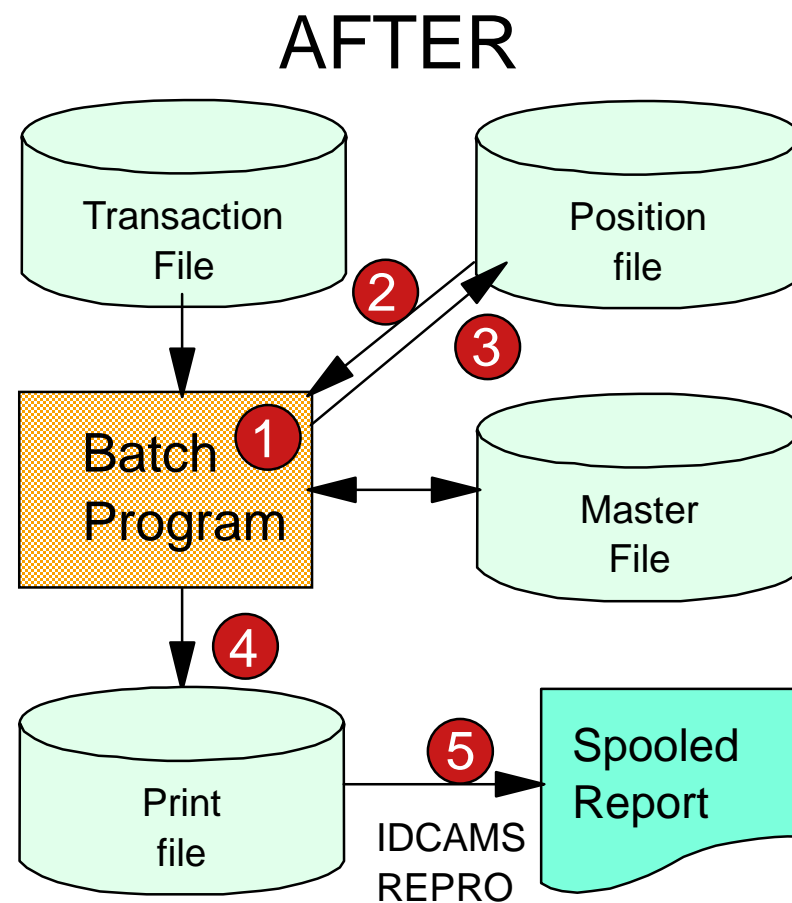
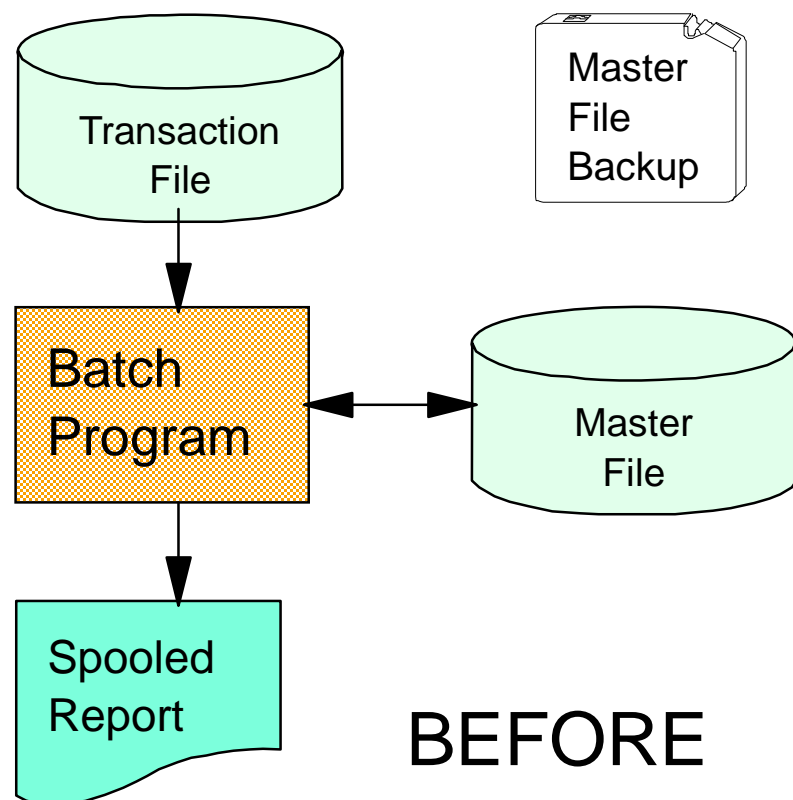
Shared

- ▶ Transactional VSAM restores uncommitted changes
 - back out
- ▶ Job must determine last commit point
- ▶ Reapply current uncommitted transaction at failure time
- ▶ Do not reapply transactions that completed/committed before job failure
- ▶ Continue to end-of-job

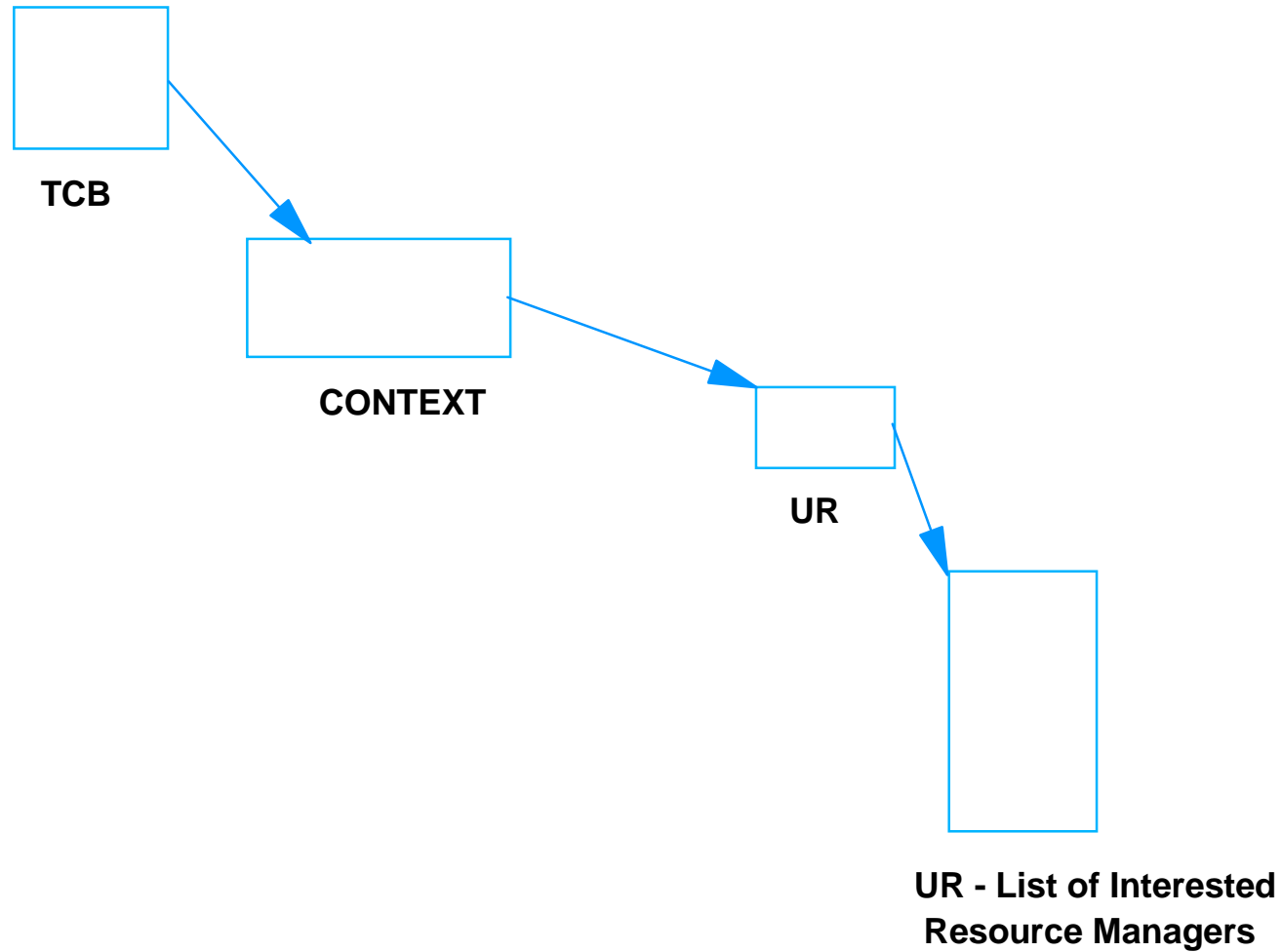
Job Processing Position File



Application Redesign



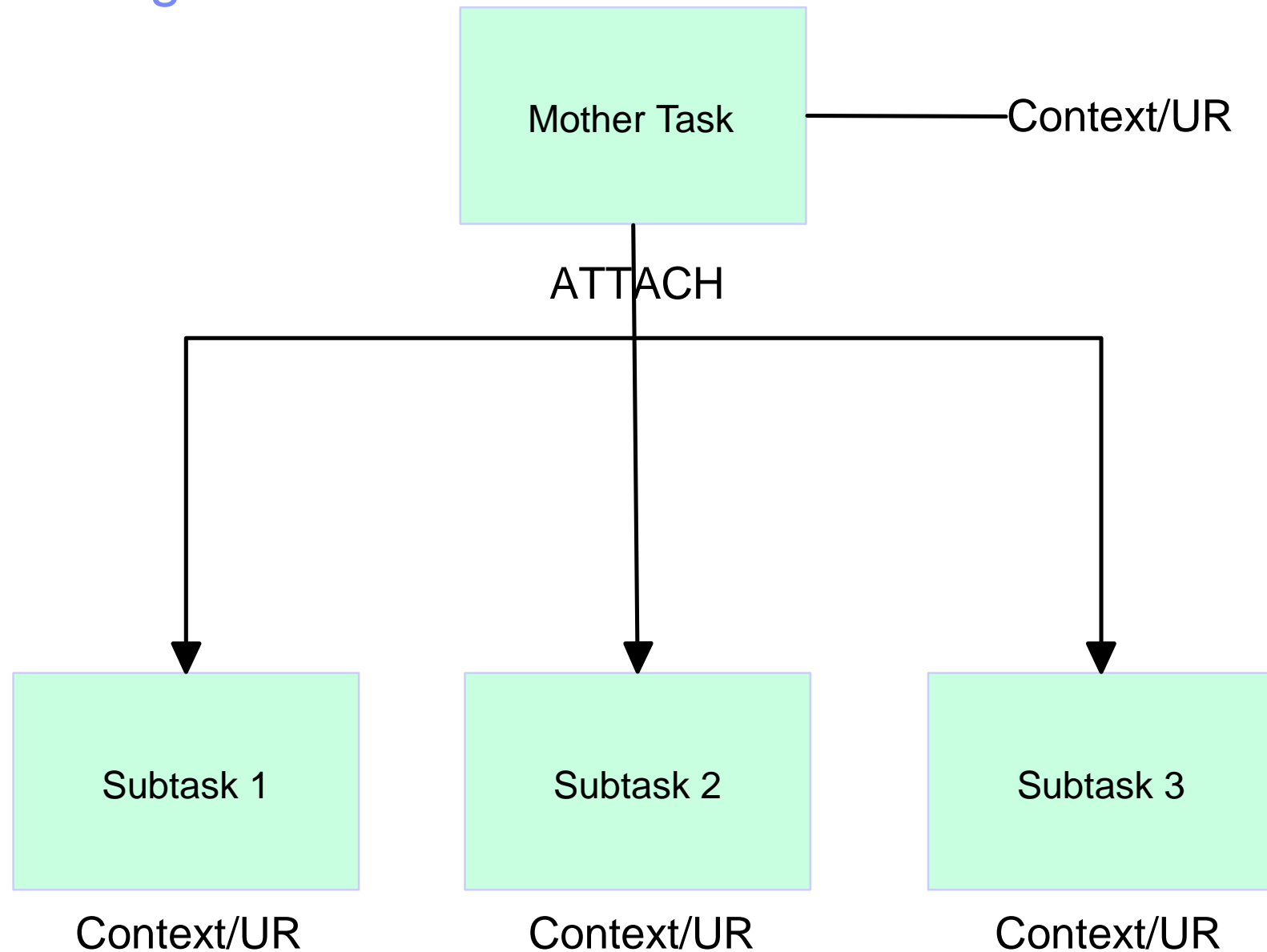
Context & Unit of Recovery (UR)



Context Management

- Default context for each dispatchable unit (TCB/SRB)
 - ▶ Provided by z/OS (Context Services)
- Privately Managed Context
 - ▶ Created/used by work managers/transaction monitors
 - ▶ Associated with the TCB by the work manager
- Current Context/UR - associated with the current TCB
- Transactional VSAM is NOT a work manager
 - ▶ It supports both simple application environment (native context) and work manager environments
 - ▶ Uses current context/UR

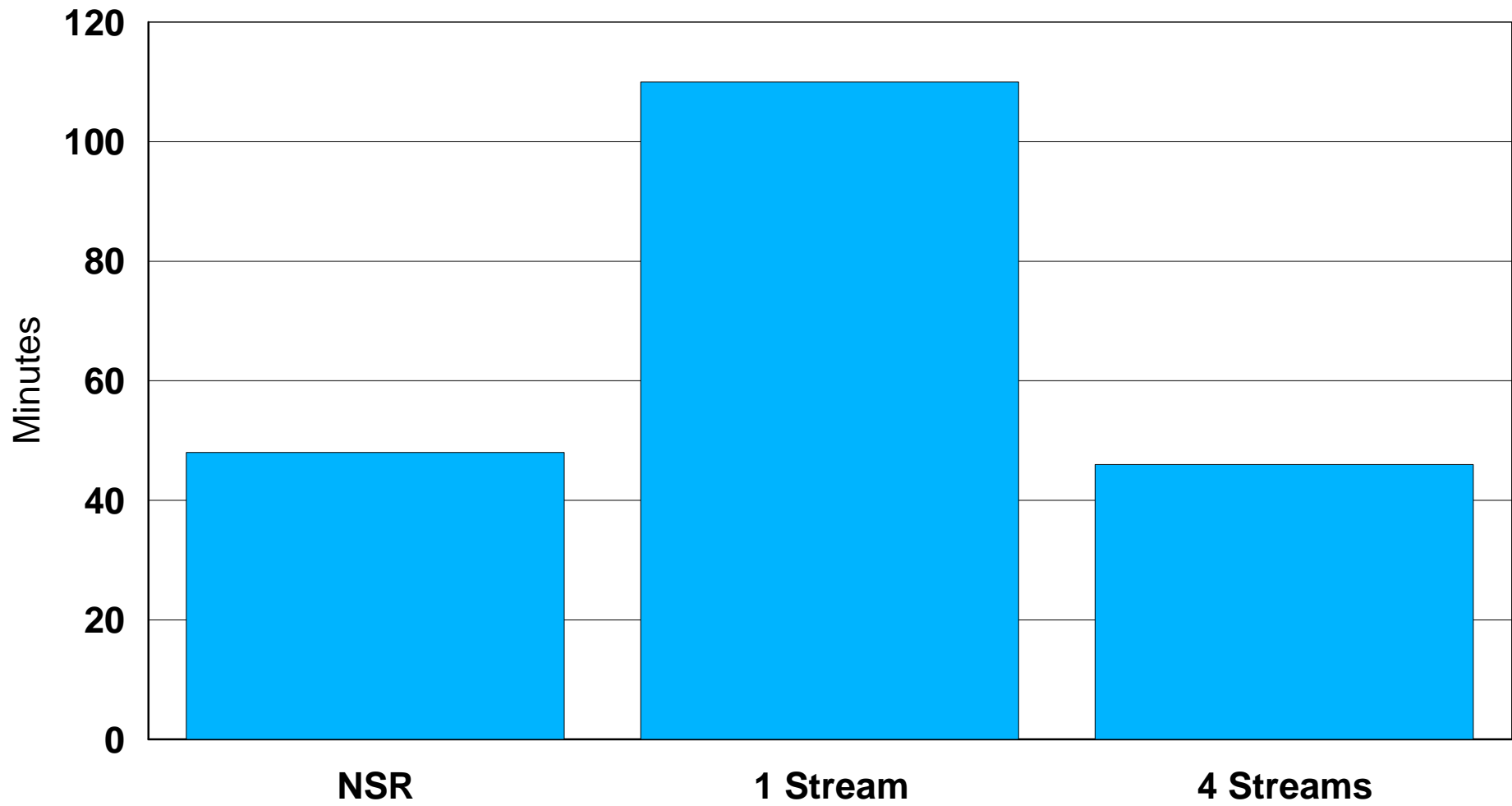
Multitasking Considerations



Performance

- **Extra Overhead TranVSAM vs. NSR or LSR**
 - ▶ Cross-Address Space access to server
 - ▶ Loss of NSR chained sequential I/O
 - ▶ Loss of LSR deferred write
 - ▶ New overhead of record locking
 - ▶ New overhead of CF cache access
 - ▶ New overhead of logging
- **Expect TranVSAM overhead to be similar to VSAM file access via CICS RLS**

Performance -- Parallelizing the Workload



Peer Recovery

- Allows another instance of Transactional VSAM to recover for a failed instance
- Cleans up any work that was left in an incomplete state
- Clears retained locks that resulted from the failure
- Does *not* accept any **new** work!
- Since peer recovery instance must register as failed instance, failed instance cannot restart while peer recovery is in progress

Permit Non-RLS Update

- Data set has retained locks or is in lost locks state
- Non-Transactional VSAM, non-RLS batch update job needs to be run
- Without PermitNonRLSUpdate, batch open would fail
- Specified on IDCAMS SHCDS command
- Data set should also be quiesced

What Happens to the Back outs?

- Transactional VSAM invokes an installation exit, the batch override exit, for each record
- The exit must be named IGW8PNRU
- Modeled on CICS XCFBOVER exit
- Exit must reside in LPALIB or LINKLIB
- Exit must indicate whether or not to back out the record
- The exit should **not** update any recoverable resources

Closing Data Set with In-flight UR

- Closing the data set could allow any of the following to occur:
 - ▶ The data set can be deleted
 - ▶ The data set can be renamed
 - ▶ A PermitNonRLSUpdate can be done
 - ▶ The data set can be quiesced
 - ▶ The locks associated with the data set can become retained

Data Set Delete/Rename Concerns

Do not delete/rename data sets with:

- An outstanding in-flight UR
 - ▶ If back out is required, UR will be shunted
- Shunted log records and retained locks
 - ▶ Can cause loss of association between the data set and its log records and locks

A new data set could be allocated with the old name!

Transactional VSAM Summary

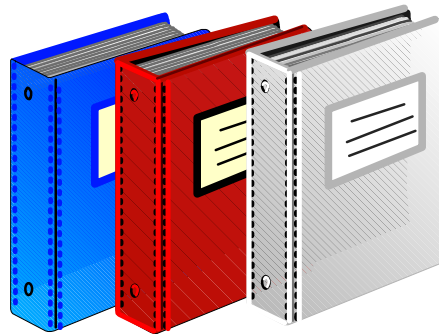
- **Transactional VSAM is a general purpose VSAM recoverable file manager for z/OS**
- **Transactional VSAM addresses a long-standing CICS batch window requirement**
 - ▶ Read/write sharing across CICS and batch jobs
 - ▶ Read/write sharing across multiple batch jobs
 - ▶ Major steps towards 24x7 CICS on-line availability
- **Transactional VSAM requires some application changes:**
 - ▶ Must have a transactional perspective
 - ▶ Must understand that it runs in a shared environment
 - ▶ Must take steps to avoid deadlocks and time outs
 - ▶ Must issue sync points (commit or back out)



IBM TotalStorage®

Appendix A.

References



References

- ***z/OS MVS Setting Up a Sysplex - SA22-7625***
- ***z/OS MVS Programming: Resource Recovery - SA22-7616***
- ***z/OS MVS Diagnosis: Tools and Service Aids - GA22-7589***
- ***z/OS MVS Programming: MVS Assembler Services Guide - SA22-7605***
- ***z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 - SA22-7610***
- ***OS/390 Parallel Sysplex Configuration Cook Book, Volumes 1-3 - SG24-2075, SG24-2076, SG24-2077***
- ***CICS Recovery and Restart Guide, SC33-1698***

Notices & Disclaimers

Copyright © 2003 by International Business Machines Corporation.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Consult your local IBM representative or IBM Business Partner for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Notices & Disclaimers

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

The responsibility for use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's or user's ability to evaluate and integrate them into their operating environment. Customers or users attempting to adapt these techniques to their own environments do so at their own risk. **IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.**

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

DFSMSdftp, DFSMSStvs, DFSMSdss, DFSMSHsm, DFSMSrmm, IBM, IMS, MVS, MVS/DFP, MVS/ESA, MVS/SP, MVS/XA, OS/390, SANergy, and SP are trademarks of International Business Machines Corporation in the United States, other countries, or both.

AIX, CICS, DB2, DFSMS/MVS, Parallel Sysplex, OS/390, S/390, Seascope, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Domino, Lotus, Lotus Notes, Notes, and SmartSuite are trademarks or registered trademarks of Lotus Development Corporation. Tivoli, TME, Tivoli Enterprise are trademarks of Tivoli Systems Inc. in the United States and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

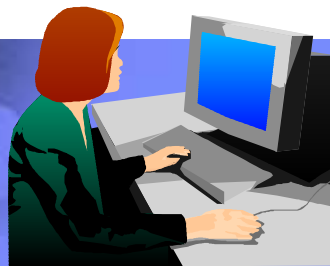
Other company, product, and service names may be trademarks or service marks of others.



IBM TotalStorage®

Appendix B.

IDCAMS SHCDS Commands



RLS Commands

SHCDS	LISTDS(base-cluster)
	LISTSUBSYS(subsystem ALL)
	LISTSUBSYSDDS(subsystem ALL)
	LISTRECOVERY(base-cluster)
	LISTALL
	FRSETRR(base-cluster)
	FRRESETRR(base-cluster)
	FRUNBIND(base-cluster)
	FRBIND(base-cluster)
	FRDELETEUNBOUNDLOCKS(base-cluster)

RLS Commands....

SHCDS	PERMITNONRLSUPDATE(base-cluster)
	DENYNONRLSUPDATE(base-cluster)
	REMOVESUBSYS(subsystem)
	CFREPAIR(INFILE(ddname))
	CFREPAIR(INDATASET(dsname))
	CFRESET(INFILE(ddname))
	CFRESET(INDATASET(dsname))
	CFRESETDS(base-cluster)

Transactional VSAM Commands

SHCDS	LISTDS(base-cluster) {JOBS}
	LISTSHUNTED SPHERE(base-cluster)
	LISTSHUNTED URID(urid ALL)
	RETRY SPHERE(base-cluster)
	RETRY URID(urid)
	PURGE SPHERE(base-cluster)
	PURGE URID(urid)

SHCDS LISTDS Example

SHCDS LISTDS(SYSPLEX.KSDS.RETAINED.CLUS1) JOBS

----- LISTING FROM SHCDS ----- IDC SH02

DATA SET NAME----SYSPLEX.KSDS.RETAINED.CLUS1

CACHE STRUCTURE----CACHE01

RETAINED LOCKS-----NO NON-RLS UPDATE PERMITTED-----NO

LOST LOCKS-----NO PERMIT FIRST TIME-----NO

LOCKS NOT BOUND-----NO FORWARD RECOVERY REQUIRED-----NO

RECOVERABLE-----YES

SHARING SUBSYSTEM STATUS

SUBSYSTEM NAME	SUBSYSTEM STATUS	RETAINED LOCKS	LOST LOCKS	NON-RLS UPDATE PERMITTED
-----	-----	-----	-----	-----
RETLK05A	ONLINE--ACTIVE	YES	NO	NO

JOB NAMES:

TRANV001	TRANV002	TRANV003	TRANV004	TRANV005
TRANJOB1	TRANJOB2	TRANJOB3	TRANJOB4	TRANJOB5

IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

SHCDS LISTSHUNTED Example

```
SHCDS LISTSHUNTED SPHERE(SYSplex.KSDS.CLUSTER.NAME)
```

```
-----  
CLUSTER NAME----SYSplex.KSDS.CLUSTER.NAME
```

URID	DISPOSITION	JOB NAME	STEP NAME	CAUSE
-----	-----	-----	-----	-----
ABCDEFGH00000001	BACKOUT	TRANJOB1	TRANSTP3	B-FAILED
XYZ@#\$0000000000	BACKOUT	TRANJOB2	STPTRAN1	IO-ERROR
0101BF\$\$22222222	COMMIT	TRANV001	TRANSTP1	C-FAILED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0				