



# **Concurrent VSAM access for batch and CICS: A white paper series**

**VSAM Record-Level Sharing (RLS)**

**A white paper from:**





---

IBM, CICS, Db2, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. SYSB-II is a registered trademark of H&W Computer Systems, Inc. All other products mentioned herein may be trademarks or registered trademarks of their respective companies.

---

---

## Introduction

CICS® applications and VSAM data still drive a large percentage of the transactional processing at the core of worldwide commerce and information systems. According to current estimates, CICS applications handle more than 30 billion transactions per day and process more than \$1 trillion worth of business every week. When these systems are offline or data is inaccessible due to batch processes, productivity stalls, crucial information is unavailable, and decisions are delayed.

This white paper explores issues relevant to batch/CICS file sharing and record-level sharing (RLS). It offers a more detailed look at VSAM RLS itself and the changes required to use files in VSAM RLS mode for batch and CICS processing.

This white paper is one of four in the H&W series titled *Concurrent VSAM Access for Batch and CICS*. The other white papers in the series are:

- *Transactional VSAM (TVS or DFSMStvs)*. This explores a method for sharing recoverable VSAM data between CICS and batch without compromising data integrity. Considerations with this method are discussed.
- *General Overview of Conventional and Current Alternatives* — This white paper explains the conventional and previous approaches, which include SHAREOPTIONS, file open-and-close products, shadow files, memo posting and EXCI, VSAM RLS and Transactional VSAM.
- *Transparent VSAM File Sharing via CICS and Batch*. This explores a method for sharing VSAM files inside CICS so that batch jobs appear to CICS like any other online transaction.

## VSAM RLS historical context

VSAM, CICS, and parallel sysplex environments grew in popularity throughout the 1970s, 1980s, and 1990s. VSAM was originally shipped by IBM® in 1973 and it became the file access method of choice when much of the CICS application development occurred.

---



By the 1990s, IBM had already been using the coupling facility to share Db2<sup>®</sup> and IMS database data across operating system images. Sharing VSAM data amongst CICS systems or even between CICS and batch in this manner was still an issue. It made sense to use the same technology on the CICS VSAM platform as well. The first step toward greater sharing of CICS VSAM data came with the announcement of VSAM RLS. Changes were made to both CICS and VSAM toward the goal of sharing VSAM files between multiple CICS regions and batch. While a good first step in reaching that goal, VSAM RLS still had significant restrictions that meant the challenge of sharing VSAM files between batch and CICS persisted, and even magnified the need for full continual access to VSAM data.

## How VSAM RLS works

VSAM RLS is a function introduced by DFSMS/MVS V1.3 and exploited by CICS Transaction Server. It is designed to enable VSAM files to be shared, with full update capability, between many applications running in many CICS regions. Prior to the inception of RLS, VSAM data sets that were opened for update were owned and accessed through a single address space, either by stand-alone batch or by a CICS file-owning region (FOR). With RLS, the VSAM files are owned by the RLS server address space (also known as SMSVSAM). Multiple CICS regions can access the data concurrently with full update integrity, thereby eliminating the CICS File Owning Region that had become a bottleneck and a single point of failure for many installations.

## RLS logging, locking, and integrity

### Logging

VSAM RLS files fall under two main categories: recoverable and nonrecoverable. The recoverability of an RLS VSAM file depends entirely on the value specified for the LOG parameter in the VSAM cluster definition. This parameter was introduced to support RLS and has three possible values: NONE, UNDO, and ALL. A recoverable data set is one whose in-flight changes are backed out (by CICS) if a transaction fails. The LOG value used on the RLS recoverable VSAM file definition is either UNDO or ALL.

- **NONE**

A value of NONE identifies the data set as nonrecoverable because CICS does not log any changes for the data set and is unable to provide transactional or data set recovery.

---

- **UNDO**

The data set is backward recoverable. That is, any in-flight changes made by a unit of work that does not succeed (uncommitted) is backed out. Changes added to the CICS logger are used to back out any in-flight unit of work that failed. This is also known as *transactional recovery*.

- **ALL**

The data set is both backward and forward recoverable. Forward recovery is possible only when another product such as CICS VSAM RECOVERY (CICSVR) is used. When ALL is used, CICS records both the before and after images of any change. In addition to transactional recovery, ALL allows the capability for products such as CICSVR to rebuild data if hardware fails or software problems occur against the data set. This type of recovery is known as *data set recovery*.

## Locking

Files opened in RLS mode can be accessed by many CICS regions simultaneously. This means it is impractical for individual CICS regions to attempt to control record locking. Therefore, VSAM maintains a single, central lock structure using the lock-assist mechanism of the coupling facility. This central lock structure provides sysplex-wide locking at a record level. No control interval (CI) locking is used. This is in contrast to the locks for files in non-RLS mode, which are normally limited to a single CICS region and are either CI locks or CICS ENQs.

VSAM supports two types of locking for files accessed in RLS mode:

- **Shared locks**

Shared locks support read integrity. They ensure that a record is not in the process of being updated during a read-only request. Shared locks can be owned by several tasks at the same time.

- **Exclusive locks**

Exclusive locks protect updates to file resources, both recoverable and nonrecoverable. They can be owned by only one transaction at a time. Any transaction that requires an exclusive lock must wait if another task currently owns an exclusive lock or a shared lock against the requested resource.

---

Exclusive locks can be *active* or *retained*, whereas shared locks can only be active. When a lock is first acquired, it is an active lock. Normally this lock would eventually be released, but if a unit of work fails, and this would cause the lock to be held for an abnormally long time, the active lock is converted into a retained lock. This has implications for batch processes that require RLS files to be quiesced prior to batch execution because a quiesced data set can be opened only in non-RLS mode if no retained locks are present.

## Integrity

To request access to a RLS VSAM data set, a batch program must do one of the following:

- Specify the RLS read integrity level on the MACRF parameter of the VSAM ACB.
- Add an RLS parameter to the DD statement for the file in JCL.

If the RLS parameters are omitted then the program will attempt non-RLS access to the file. Depending on the type of OPEN (read or update) and the recovery attribute of the RLS file, this situation could give rise to program abends or, potentially worse, access to data in native VSAM mode with none of the integrity benefits of accessing the shared RLS SMSVSAM buffers.

For files opened by CICS, the read integrity level is specified in the file definition. The RLS recovery option (if not specified) is taken from the VSAM file definition stored in the ICF catalog.

### RLS read integrity

VSAM RLS supports three levels of read integrity:

- **NRI (No Read Integrity)**  
No record locking is performed by VSAM when a GET/POINT request is issued. Although this avoids the overhead of locking, it can allow the requestor to obtain uncommitted data and is sometimes referred to as a *dirty read*.
- **CR (Consistent Read)**  
A shared lock is obtained by VSAM for GET/POINT requests. This ensures that no uncommitted data are ever be returned to the application. GET/POINT requests wait for any pending change to be committed or backed out and the currently held lock to be released.
- **CRE (Consistent Read Explicit)**  
This is similar to CR except that the shared lock is held by VSAM RLS until the unit of recovery or unit of work has been completed. This type of lock is available only to CICS and TVS.

---

## Update integrity

For updates to a recoverable file, data integrity is ensured by SMSVSAM by maintaining locks on data changed in the unit of work until CICS explicitly declares that locks can be released. CICS signals this as a result of one of the following:

- Successful completion of the unit of work
- Processing of a SYNC call
- Successful backout of in-flight changes, should a unit of work fail

RLS provides locking and sysplex-wide parallel shared-data access, while CICS provides the logging and recovery capabilities. It is both of these features used together that make transactional recovery of a VSAM RLS data set from a failed unit of work possible.

For updates to a nonrecoverable file, RLS releases a lock when the buffer containing the modified control interval has been written. Because no transactional recovery is ever performed on a nonrecoverable file, changes are not backed out and there is no need to maintain any locks. In fact, it is reasonable to assume that locks against records in a nonrecoverable data set remain held only for the duration of the requests — that is, they are acquired at the start of a request and released upon its completion.

## What does this mean for batch processing?

### SHAREOPTION

SHAREOPTION is largely ignored under RLS, with the exception of SHAREOPTION (2, x). This means that non-RLS reads of a data set opened in RLS mode are possible. No data integrity is provided for the non-RLS reader. Both CICS and batch can have concurrent read and update access to nonrecoverable data sets. Again, in this instance, no coordination between CICS and batch occurs, so data integrity issues are possible.

---

## Batch and VSAM RLS recoverable files

As previously stated, RLS addresses the limitation of a single CICS address space owning a VSAM file for update and the associated single point of failure. For recoverable VSAM files, CICS read and update integrity is ensured by synchronized SMSVSAM buffers and the coupling facility's system-wide locks on accessed records. With VSAM RLS, you no longer need to restrict VSAM update activity to a single CICS file-owning region. Now you have the possibility of channeling your workload to any number of additional available CICS regions, allowing you to better balance the workload and evenly distribute access to VSAM data.

For batch processes that require inquiry access to RLS-managed recoverable VSAM files, read integrity can be obtained through RLS implementation by utilizing the shared SMSVSAM buffers. In other words, shared access is allowed for read only from a batch program.

For batch processes that require update access to RLS-managed recoverable VSAM files, a batch processing window is still required. This requires deallocation of the VSAM file from RLS (and CICS) while batch updates the file natively. When batch is completed, the VSAM file can be reopened under RLS management. If CICS requires inquiry access to the VSAM file during batch processing, this can be achieved with a SHAREOPTION 2,3. However, the VSAM file must be opened under CICS as non-RLS mode for inquiry only. This is typically achieved by providing a separate FCT entry for CICS inquiry during batch processing.

## Batch and VSAM RLS nonrecoverable files

For nonrecoverable files, batch and CICS can process concurrently for both read and update capabilities. By definition, recovery is not ensured for either a CICS transaction or a batch transaction. With nonrecoverable files, if a transaction updates multiple records and fails before the last record update is successful, the unit of work is partially committed.

For example, a single transaction is made to transfer \$100 from a savings account to a checking account. This single transaction issues two record update requests. The first record update adds \$100 to the checking account and adjusts the account balance accordingly. The second record update subtracts the \$100 from the saving account and adjusts the balance accordingly. If the files are defined as nonrecoverable, the \$100 might be added to checking but the subsequent \$100 subtraction to the savings account is unsuccessful. This increases the checking account by \$100 but the associated equivalent savings account debit does not occur.

---

This potential exposure escalates with batch in a shared environment. Records are committed with every successful write, rewrite, or delete request. If an abend occurs, the status of files might be in a state that is less than ideal when auditing takes place. This might not be an issue for batch jobs that can be rerun at the point of failure or rerun without adverse affects to the data. However, such is not typically the case.

In other words, recovery becomes your responsibility when batch and CICS share update access through RLS for nonrecoverable files. If you choose to restore a file to a point-in-time backup, what happens to the updates that have occurred after the point-in-time backup was established? Implications to sharing nonrecoverable VSAM files for update between batch and CICS need to be carefully considered.

## Conclusion

Companies that need to share VSAM data across multiple z/OS® images can benefit greatly from VSAM RLS. However, if you need to process batch while RLS files are fully available to CICS, a VSAM file-sharing solution is the preferred methodology. The reasoning is fairly simple. If the RLS file is defined as recoverable, technically you cannot process batch unless a file-sharing solution is used. More importantly, if the RLS file is defined as nonrecoverable, the risk for data integrity errors is present because recovery is not ensured for either a CICS or batch transaction.

VSAM file-sharing software, on the other hand, allows you to ensure data integrity while maintaining investments in the mainframe itself, CICS, existing CICS and batch applications, and the people and skills that support everything.

The world now expects data availability and application access on a 24/7 schedule. By sharing VSAM files inside CICS, you can avoid many of the pains associated with alternative approaches. At the same time, you can ensure high levels of performance, data integrity, implementation ease, reliability, and recoverability.

To learn more about existing batch management issues and transparent VSAM file sharing, please review the other white papers in this *Concurrent VSAM Access for Batch and CICS* series (*Transactional VSAM, Overview: Conventional and Current Alternatives*, and *Transparent VSAM File Sharing via CICS and Batch*).

---

## About H&W

Headquartered in Boise, Idaho, H&W has been a leading provider of quality software solutions since 1979. H&W creates reliable, technically sound solutions like SYSB-II® that provide long-term value. Today, corporations worldwide, including many Global 500 companies, trust H&W for their IT software and services needs. Call 1-800-338-6692 or visit [www.hwcs.com](http://www.hwcs.com).

## References

The authors consulted the following sources in creating this white paper.

- International Business Machines Corporation. *z/OS DFSMStvs Planning and Operating Guide* (Publication number SC26-7348-00.) IBM Corporation, 2003
- Lovelace, Dave, Rama Ayyar, Alvaro Sala, Valeria Sokal, and IBM Technical Support Organization. *VSAM Demystified*. 3<sup>rd</sup> Ed. (Publication number SG24-6105-02.) IBM International Technical Support Organization, 2013, <http://www.redbooks.ibm.com/redbooks/pdfs/sg246105.pdf>
- Menendez, Terri. *VSAM Record Level Sharing (RLS Overview)*, Parts 1 and 2. IBM System Storage and Storage Networking Symposium. October 15-18, 2007. Montpellier, France.
- Rogers, Paul, Redlef Janssen, Andre Otto, Rita Pleus, Alvaro Salla, and Valeria Sokal. *ABCs of z/OS System Programming*. Vol. 3. 4<sup>th</sup> Ed. (Publication number SG24-6983-03). IBM International Technical Support Organization, 2010. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246983.pdf>
- Smith, Hugh, Karl-Heinz Marquardt, Luigi Ingresso, Rayno vanZyl, Byron Jones, and Maddelena Tosoni. *CICS and VSAM Record Level Sharing: Planning Guide*. (Publication number SG24-4765-00.) San Jose, CA: IBM International Technical Support Organization, 1996



6154 North Meeker Pl., Ste 100  
Boise, ID 83713  
800-338-6692  
[www.hwcs.com](http://www.hwcs.com)

©2018 by H&W Computer Systems, Inc.  
All rights reserved  
(v.1)