

VSHARE™ FOR Z/OS

Installation and Operations Guide

Multiple Region VSAM File Sharing Product

Release 4 Version 0
Manual published October 2018



VSHARE allows multiple programs to access and update the same VSAM file simultaneously by giving batch jobs the ability to update files within CICS.



CSI INTERNATIONAL

“Delivering what the competition can only promise”

www.csi-international.com • info@csi-international.com • 800.795.4914

**Copyright © 2003, 2016, 2018 by CSI International
All Rights Reserved**

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. This material contains confidential and proprietary material of CSI International and may not be used in any way without written authorization from CSI International. This material may not be reproduced, in whole or in part, in any way, without prior written permission from CSI International. Permission is hereby granted to copy and distribute this document as follows:

- Each copy must be a complete and accurate copy.
- All copyright notices must be retained.
- No modifications may be made.
- Use of each copy is restricted to the evaluation and/or promotion of CSI International's VSHARE™ FOR Z/OS product or in accordance with a license agreement.

**VSHARE FOR Z/OS 4.0 *Installation and Operations Guide*
October 2018**

Updates to this edition: Messages and Codes—FSIVSHAR-693

Published by CSI International

Phone: 800-795-4914

Fax: 740-986-6022

Email: *sales@csi-international.com*
support@csi-international.com

Internet: *http://www.csi-international.com*

CSI International Product Technical Support

Product Technical Support

During Business Hours Monday through Friday, 9:00 A.M. through 5:00 P.M. EST/EDT

- Telephone

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

- Email

Questions: support@csi-international.com

- Web

<http://csi-international.com/support.htm>

Emergency Service (24/7) After business hours, and 24 hours on Saturday and Sunday

- Telephone¹

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

¹ Calls are automatically transferred to a 24/7 answering service that forwards all calls based on severity.

Table of Contents

CSI International Product Technical Support	iii
Getting Started	1
Current Environment	2
The Problem.....	3
Past Solutions	3
How Does a DBMS Solve the Problem?.....	4
VSHARE Is the Answer	5
VSHARE's Features	6
Ease of Use.....	7
About This Manual	7
New for Release 4.....	7
Programmer's Guide	8
Overview	9
VSHARE Architecture.....	9
A Sample VSHARE Execution.....	11
Steps to Define a Job to VSHARE.....	12
Cross Memory Services Versus LU 6.2	13
Not All the Files Have to Be Defined to CICS	13
Accessing Files on a Different CPU.....	13
VSAM File Security	14
Running a Program in a Shared Environment	14
Recovery	15
Performance.....	15
Problem Determination.....	15
Assembler Support.....	17
Non-Unique Keys with Alternate Index Files.....	17
Summary.....	17
Recovery	18
Overview	19
Recovery in a Shared Environment	19
Is Delete, Define, Repro Practical for Recovery?	19
Who Else Can Update My Record?	19
How Long Should I Lock a Record?.....	19
What Happens If My Program Abends?.....	19
Locking and Releasing Records Within CICS.....	20
Multiple Programs Updating the Same Record	20
VSHARE's Three Recovery Tiers.....	21
Tier 1: No Syncpointing	22
What Is Tier 1?	22

Table of Contents

What Type of Job Should Use Tier 1?	22
How Is Tier 1 Implemented?	22
How Are the Files Recovered After an Abend?	23
Tier 2: Implied Syncpointing	24
What Is Tier 2?	24
What Type of Job Should Use Tier 2?	24
How Is Tier 2 Implemented?	24
What Is Journaling?	24
Why Use Journaling.....	24
What Is Backward Recovery?	25
Journaling and Recovery with Tier 2	25
Tier 2 Journaling Example	25
Tier 2 Recovery Example.....	26
Define the Job to Use Journaling	27
Define the Syncpoint Level	27
Create a VSHARE Recovery Procedure	27
Creating VSHARE Recovery Parameters	29
What Is Journalled and What Is Not Journalled?	30
Tier 2 Recovery Example with an Executed Procedure.....	30
Recovery Example Without an Executed Procedure	31
Tier 3: Explicit Syncpointing.....	32
What Is Tier 3?	32
What Type of Job Should Use Tier 3?	32
How Is Tier 3 Implemented?	32
Determine Whether VSHARE Journaling Is Required	32
Modify the Program to Issue the Syncpoints or Rollbacks.....	32
Batch-Only Mode	33
Requesting a Syncpoint or Rollback	33
Sample Syncpoint Request.....	34
Issuing Syncpoints and Rollbacks Through a Batch Exit Program	35
Syncpoint and Rollback Journal Records.....	37
Create an Application Recovery Scheme.....	39
Tier 3 Recovery Scenario	39
Recovery and ESDS Datasets	39
Comparison of VSHARE Recovery Tiers	40
Online Control	41
Overview	42
VSHARE Main Menu	43
Dynamic Monitoring of Batch Activity.....	44
Active Job Details.....	47
Obtaining Dataset Statistics on Active Jobs	48
Defining Which Jobs Can Access the Files.....	51
Displaying a List of Jobs Already Defined.....	56
Reviewing the History of Past Job Executions	57
Detailed History Information	59

Table of Contents

Detailed Dataset History Information	62
Defining CICS Regions to VSHARE	64
Listing Defined CICS Regions	68
VSAM Performance	70
Overview	71
Concepts	71
Storage Mediums	72
Synchronous Versus Asynchronous Access	73
Basic I/O Concepts	75
Mechanics of a VSAM KSDS I/O	76
Local Shared Versus Non-Shared Resources	77
How to Improve VSAM Performance	78
Avoid Control Interval and Control Area Splits	79
Reorganize Datasets with Multiple Extents	79
Spread Heavily Accessed Datasets	80
Use Index and Data Buffering	80
Adjust Index and Data Control Interval Sizes	80
Manage One-Cylinder VSAM Datasets	81
Analyze CICS End-of-Day Statistics	81
Use VSAM Local Shared Resources	82
Use CICS VSAM Subtasking	84
Beware of Datasets Using Share Option 3 or 4	84
Reduce Program I/O	84
Place Critical Files on Faster Storage Mediums	85
Installation	86
Overview of VSHARE Installation on z/OS	87
Preparation	87
Installing VSHARE from Tape	88
Installing VSHARE from a Web Download	89
Download the Installation Datasets to a PC	89
Transfer the Installation Datasets	89
RECEIVE and RESTORE the Datasets	90
Install the Software Product Key	92
Installing Product Keys	93
Product Authorization Messages	93
Define a VSHARE Control Dataset	94
Control Dataset Definition	95
Update the CSIVSBDS Parameter Member	95
ISC Installation	96
Place the Load Library in the DFHRPL	96
Environment Definitions	96
VTAM Definitions	100
CICS Definitions	101
Define the Eligible CICS Regions and Jobs	104

Table of Contents

Install the Cross Memory Services Interface.....	105
Installing CMS Batch Component	106
Installing the CICS CMS Component	107
Modify the Tier 2 File Recovery Procedure.....	107
The VSAM Open/Close Routing Facility	108
Introduction.....	108
Installing the VSAM Open/Close Routing Facility Modules	108
Removing the VSAM Open/Close Routing Facility.....	109
Routing Facility Messages	110
VSHARE Activation and Deactivation.....	111
File Deactivation.....	111
File Security	111
Messages and Codes.....	112
FSIVSBAI Messages.....	113
Online Maintenance Messages	113
Batch Messages.....	119
Journal Messages	126
CMS (Cross Memory Services) Messages	141
Presentation Services Messages	146
Presentation Messages	163
CICS Messages	172
Half Session Messages	181
Resource Manager Messages.....	185
LU 6.2 Network Services Messages	188
Batch Abend Codes	193
CICS Abend Codes.....	194

Getting Started

Current Environment

VSAM has been chosen by many companies to be the data access method for their computer applications. Its ease of use, ease of understanding, and efficiency make VSAM appealing over a more complicated database management system. Unfortunately, VSAM provides minimal functionality and flexibility, which imposes constraints on systems that use it. VSAM's major constraint is the inability for both online and batch programs to update the same VSAM file simultaneously with full data integrity. With the current work environment, this constraint is unacceptable. VSHARE lifts this constraint by providing VSAM applications simultaneous access to the *same* VSAM file with full data integrity.

VSAM's single-update constraint and lack of functionality raise concerns over how VSAM systems fit into a longer term data-processing strategy. For example, how can these critical VSAM systems fit into the SAA platform, or how can they be used in a distributed environment?

VSHARE provides an answer to these questions by extending VSAM's functionality. It not only allows simultaneous access, but it breaks down the physical CPU barrier by allowing access to any VSAM file within an SNA network. VSHARE is the best solution for your immediate and long-term needs.

With such a large investment in VSAM, the single-update constraint can no longer be ignored. With today's dynamic business environment, the need for a shared VSAM environment is mandatory. For example,

- MIS departments have invested millions of dollars and many thousands of hours in VSAM CICS applications.
- Computer applications are being required to provide longer online availability because of the current dynamic business environment. Many corporate goals include 24-hour online availability.
- Longer online availability leads to higher volumes to be processed in a shrinking batch-processing time frame.
- Applications are so constricted in batch-cycle processing that any type of error during a cycle means lost online availability to the end users. This can result in lost revenue and poor service.
- The increased complexity of today's processing is forcing companies to question the strategic nature of VSAM systems.

The Problem

Why can't two batch jobs update the same VSAM file simultaneously? The answer is simple: VSAM gives only one program the ability to own a file for update. Therefore, when CICS is up, your batch jobs cannot run, and when your batch cycle is running, CICS programs cannot execute. This basic constraint creates the familiar conflict of online versus batch.

VSAM's single-update limitation has caused many restrictions in your VSAM systems. The online-versus-batch conflict has caused many problems in the past:

- No 24-hour online availability
- Long batch cycles delaying the next online day
- No batch processing during the online day
- Expensive rewrites to use a database management system (DBMS)
- Read-integrity problems for batch reports
- Doubling of DASD costs to support shadow databases
- Poor online response from background CICS transactions
- No access to remote files

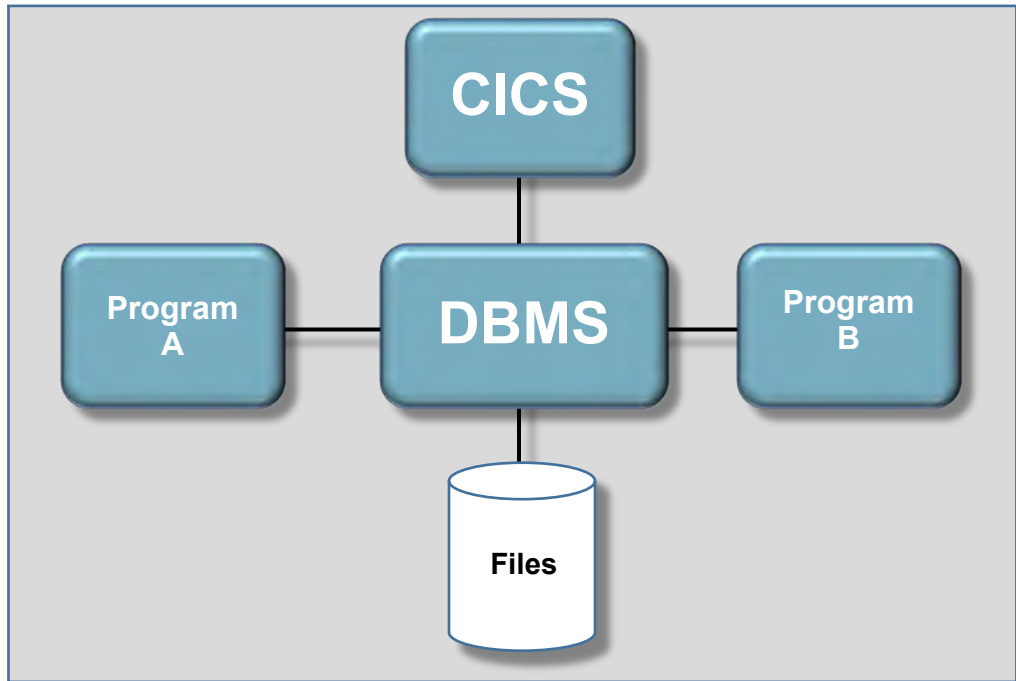
Past Solutions

Few options have been available to handle the pressure of longer online or batch availability. The solutions were either too time consuming, too expensive, or too awkward to maintain. The single-update constraint has forced companies to design VSAM systems based on technical constraints rather than business needs. For example, companies have tried to

- Re-develop the application with a DBMS. This solution is time intensive and expensive to implement.
- Develop CICS transactions to perform batch functions. This results in poor response time for the online system or awkward operational requirements.
- Create shadow databases. This results in high DASD usage and problems keeping the databases in sync.
- Make the batch program open and close the online files for short periods of time. This causes online outages and is not really a long-term solution.
- Develop home-grown software to access files remotely.

How Does a DBMS Solve the Problem?

VSHARE solves the single-update constraint by asking a simple question: How does a DBMS solve the problem of simultaneous access? The database manager is constrained by the same technical limitations as any program, but there is one difference. The DBMS runs continuously in its own address space, receiving I/O requests from batch jobs or online systems. The DBMS, in turn, performs all the I/O on behalf of those programs. In essence, the DBMS becomes the “traffic cop” for database access. It manages requests efficiently with full data integrity. The figure below shows this configuration.



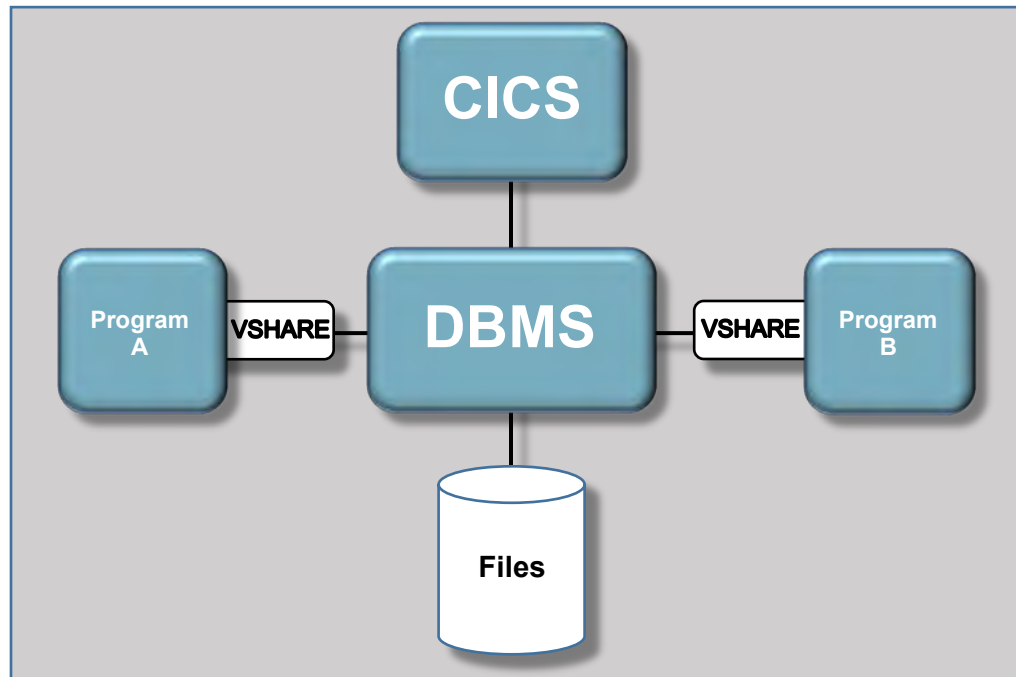
VSHARE Is the Answer

VSHARE was developed with this same concept in mind: what VSAM needs is a traffic cop. Looking at the current VSAM environment, CICS is actually the VSAM traffic cop for online programs. It centralizes all online I/O, guaranteeing full data integrity. VSHARE capitalizes on CICS's multiprocessing environment to give the same file-sharing capability to batch programs. It provides a transparent vehicle that intercepts all VSAM requests, communicates the request to VSHARE software running in CICS, and performs the I/O requests on behalf of the batch job. In this way, any number of batch programs or online programs can update the same set of VSAM files. CICS therefore becomes the traffic cop for both online and batch programs.

VSHARE provides two communication paths between batch programs and CICS. The first is LU 6.2, which allows remote file access. If the batch job and CICS are running on the same CPU, VSHARE's Cross Memory Services (CMS) interface can be used to maximize performance.

VSHARE solves VSAM's single-update constraint in a familiar, efficient, and technically sound manner. Its open architecture easily fits into any data center environment.

CICS has been the VSAM traffic cop for online programs for decades. VSHARE is a vehicle that gives batch programs the same advantages. The figure below shows how programs connect to CICS through VSHARE.



VSHARE's Features

VSHARE is a total solution for a VSAM file-sharing environment. It possesses the four important qualities required to run a VSAM system in a multi-sharing environment.

1. Full Data Integrity

- Online and batch jobs can update the same file simultaneously.
- Full data integrity is provided because all I/O is centralized within CICS.

2. Total Transparency

- No programming changes are necessary.
- There is no new interface to learn.

3. Total Recovery

VSHARE provides a tiered solution for recovery. All changes are journaled. In the event of an abend, all of the updates are backed out at the field level. In this way, no updates made by other batch or online programs are lost.

4. Total Control

- You determine which jobs can use VSHARE.
- VSHARE fits into your current security scheme by interfacing with popular security packages.
- You determine the maximum number of jobs that can access a CICS region at one time for better control.
- VSHARE provides dynamic control that allows you to
 - a) Suspend a job's execution.
 - b) Restart a suspended job.
 - c) Cancel a job.
 - d) Change the number of I/O requests being sent to CICS by a batch job.
- You can view a job's execution history using VSHARE's online system.

Ease of Use

VSHARE provides all the functionality required to execute a VSAM system in a multi-programming environment while being easy to install and use. These advantages include

- Total transparency
- No source code modifications
- No recompiling of current code
- No need for any system software modifications
- No proprietary address space to support
- Support for all programming languages
- No need to learn a new interface—the application programmer codes with the familiar I/O verbs
- Dynamically allocated VSHARE datasets
- Easy installation, activation, and deactivation

VSHARE is a solution, not just a tool.

About This Manual

This manual covers all the information needed to install and use VSHARE. It also addresses issues that arise from executing programs in a shared environment, such as new recovery requirements, the importance of VSAM file tuning, and the other related issues.

New for Release 4

The following feature changes have been made to VSHARE for release 4:

- The Job Maintenance and System Maintenance screens in the online maintenance facility have been redesigned to more logically group the information displayed.
- The selection of batch APPLIDS is now controlled with an option and a selection mask on the System Maintenance screen. These features assist in filtering the available APPLIDS to only those that apply to VSHARE.
- A ROLLBACK control option has been added to more closely manage actions during batch-job abort processing. The ROLLBACK option can be controlled at the system or the job level.

Programmer's Guide

Overview

VSHARE places little burden on an application programmer. There are no new commands to learn or program modifications to perform. In fact, there is no need to re-compile an existing program in order to use VSHARE. VSHARE is behind the scenes during program execution intercepting all I/O requests. Therefore, this chapter does not explain new interfaces or syntax rules. It gives a programmer an understanding of VSHARE, how it works, and what is required to use the software.

This chapter includes a discussion of the relevant issues that arise in a VSAM shared environment. The issues include recovery considerations, resource sharing, performance tuning, and problem determination.

VSHARE Architecture

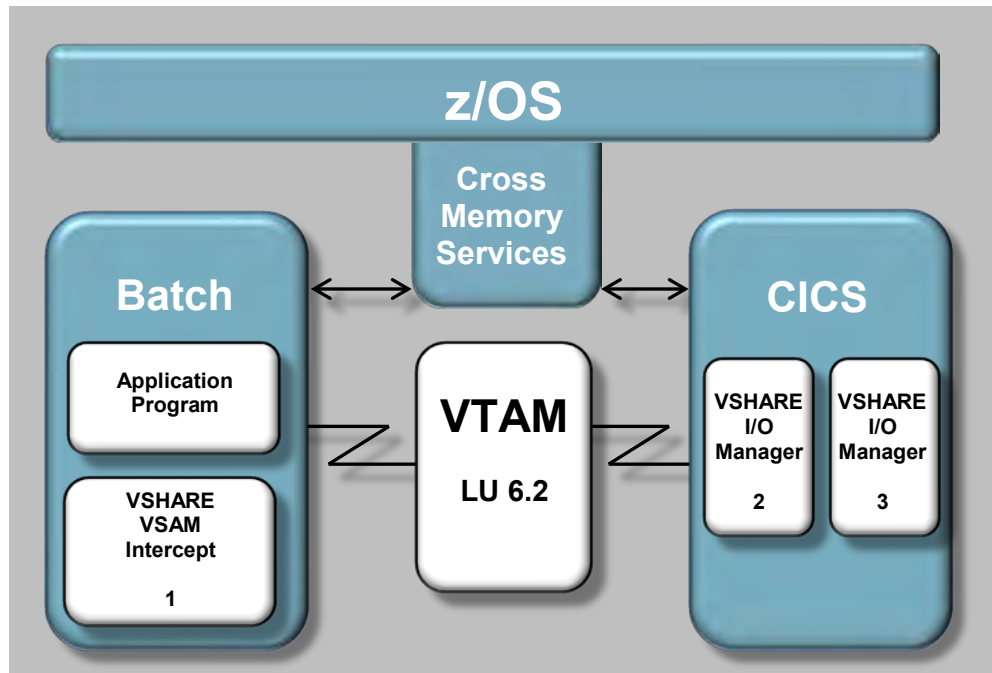
VSHARE allows multiple programs to update the same VSAM file simultaneously. To accomplish this, VSHARE intercepts batch VSAM requests and ships the requests to a CICS region. CICS then updates the file on behalf of the batch jobs. By giving batch jobs the ability to update files within CICS, VSHARE makes CICS the VSAM data manager.

VSHARE consists of three basic components. Refer to the figure below:

1. The first component executes in the batch program's address space. It intercepts the I/O requests and ships them to the second component of VSHARE residing in CICS.
2. The second component of VSHARE resides in the CICS region owning the VSAM files. It performs all of the VSAM I/O on behalf of the batch job. The first and second components communicate through VTAM or Cross Memory Services. VTAM may be used when the batch job and the CICS region reside either on the same CPU or on different CPUs. Cross Memory Services provides a performance boost but the batch job and the CICS region must be on the same CPU.
3. The third component of VSHARE also resides in the CICS region. It is comprised of a series of online screen programs providing the following functions:
 - Define CICS regions to use VSHARE.
 - Define jobs eligible to use VSHARE.
 - Monitor active jobs currently using VSHARE.
 - Suspend a job that is currently active.
 - Restart a job previously suspended.

- Cancel a job.
- Reduce the number of I/O requests being sent to the CICS region in the event of heavy online activity.
- Review prior executions of batch jobs.

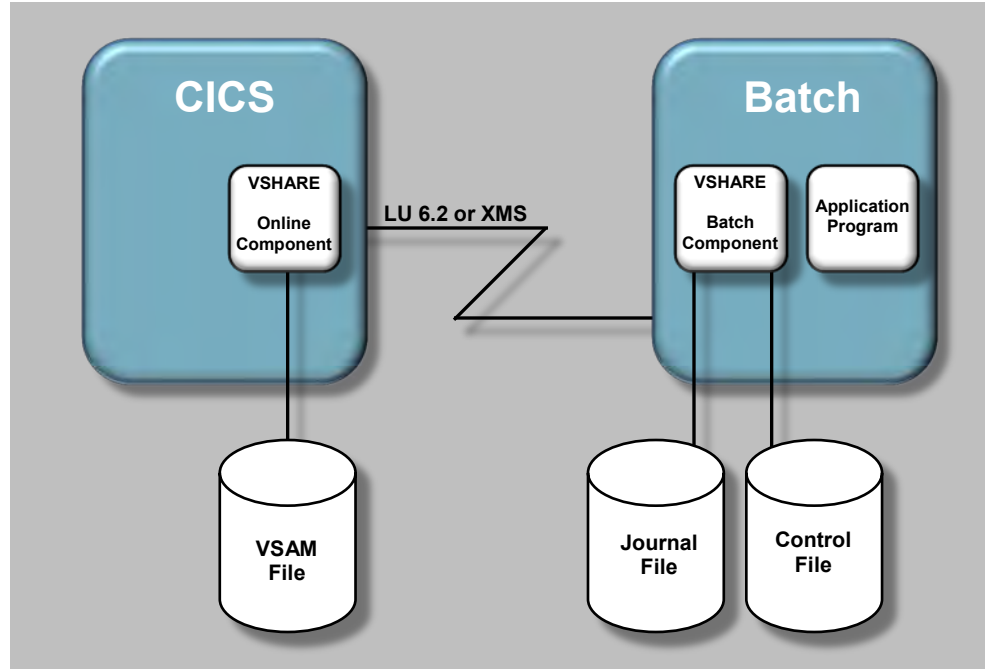
Refer to the chapter “Online Control” on page 41 for details on the online functions.



A Sample VSHARE Execution

The following is an example of program execution using VSHARE. It describes the steps taken and it gives a better understanding of the role that VSHARE plays. Refer to the figure below.

- The program begins execution.
- The program opens a file.
- During the first open of a file, VSHARE gets control and reads the VSHARE control file for a job record.
- After the job record is read, VSHARE retrieves the APPLID of the CICS region owning the files. This CICS APPLID is located on the job record.
- VSHARE reads the system record for the CICS region and retrieves the information necessary to communicate to CICS.
- VSHARE communicates to the CICS region and starts the VSHARE online component. If the job is defined to use Cross Memory Services, VTAM is not used for the rest of the job's execution.
- VSHARE sends the dataset name to the online component to determine whether the file is allocated to CICS.
- The online component finds the file and sends the positive response to the batch component.
- The batch component flags the file as an intercepted file and returns a good-open return code to the batch program.
- Any I/O related to the file throughout the execution of the batch program is intercepted by VSHARE and shipped to the CICS region. The CICS component of VSHARE performs the I/O on behalf of the batch program and returns the records and/or status information.
- When the batch program finishes, VSHARE stops the CICS component and terminates the communication to the CICS region.



Steps to Define a Job to VSHARE

Defining a job or a set of jobs is straight forward. Definition of datasets eligible is performed at the job or job group level rather than at the dataset level. This allows quick activation or deactivation of VSHARE without impacting other jobs. Because VSHARE needs no system software modifications, it is easily deactivated from an application.

Following are the steps to define a job to VSHARE.

- Log on to the CICS region executing VSHARE and execute the VSHARE online system (CICS transaction 'TCC'). Select the Job Maintenance option from the Main Menu. You define a job or a set of jobs from the Job Maintenance screen. Refer to the chapter "Online Control" on page 41 for details on job definition. The name defined can be either the full job name or any number of characters starting from the first. For example, if a programmer wants all jobs starting with PROD to use VSHARE, only the name PROD is defined.
- Concatenate VSHARE batch load library at the beginning of the JOBLIB or STEPLIB.
- Submit the batch job.

Notice how easy it is to implement VSHARE. Deactivation can be accomplished by either removing VSHARE batch load libraries from the concatenation or deleting the job definition through the Job Maintenance screen.

Cross Memory Services Versus LU 6.2

VSHARE provides two ways that the batch job can communicate the VSAM requests to CICS. The first is LU 6.2 or APPC. This is IBM's communications protocol that allows two programs to communicate to each other through VTAM. The advantage of using VTAM is that the two programs can be located anywhere within a SNA network. By using LU 6.2, the batch job can be located on a different CPU than the CICS region. For the first time, a VSAM application can truly run in a distributed environment. You can run a batch job that updates a file in a data center in another state or country. If you use PR/SM, the batch job can be executing on one image while the CICS region is running on another image.

The second communications vehicle that VSHARE can use is Cross Memory Services (CMS). CMS is a faster communications vehicle than LU 6.2, but there is one limitation. The batch job and the CICS region must execute on the same physical z/OS image.

You specify which communications vehicle to use on the Job Maintenance screen. If the job could possibly be submitted on either the same CPU or a different one than the CICS region, define the job to use CMS. If VSHARE determines that the batch job and the CICS region are on different CPUs or PR/SM images, it will automatically use LU 6.2.

Not All the Files Have to Be Defined to CICS

During the opening of a file, VSHARE determines whether the file is eligible to be intercepted. It compares the cluster name contained in the DD card with the CICS VSAM files allocated. If a cluster is located in the CICS region with the same name, the file is intercepted. If a match is not found, the file is not intercepted and VSAM updates the file directly. VSHARE does not require all VSAM files to be allocated to CICS. The mixing of both intercepted and batch-only files allows for greater flexibility.

VSHARE does not require the FCT CICS name to match the DDNAME in the batch program. VSHARE inquires against the CICS FCT on a full cluster name.

Accessing Files on a Different CPU

Accessing a remote file is easy. First, the remote CICS region owning the remote file needs to be defined to the batch owning system. This is accomplished through the online system maintenance screen. Refer to the chapter "Online Control" on page 41 for details on how to define a CICS region to VSHARE.

Second, define the job to connect to the remote CICS system.

Third, define a VSAM dataset to the batch-owning system with the same name and file characteristics. Defining the dataset is necessary so that VSHARE can obtain some dataset characteristics and so that the installation can integrate the remote dataset name into their security system. Now the remote VSAM file can be accessed by any eligible job.

VSAM File Security

There is no need to change file security settings. VSHARE fits into your current security scheme. Before every file is intercepted, VSHARE's batch component interfaces with your security system. If the security system determines that the user ID associated with the batch job does not have authority to update or read the file, VSHARE abends the batch job with a U2000 error. By reading the VSHARE log, you can isolate the dataset causing the problem.

Running a Program in a Shared Environment

Many VSAM batch programs and cycle flows were initially developed with sequential processing in mind because only one program could own a file for update. With this constraint lifted, program design needs to be re-evaluated to consider file sharing. Since VSHARE provides a level of file sharing identical to a database management system (DBMS), a programmer needs to be aware of a new set of issues arising from accessing files in a shared environment. For example,

- What other programs execute while my program is executing? Will those programs change a field that may impact my program? Will I impact them?
- Is my code opening up the potential of a deadly embrace? A deadly embrace is a situation where one program is waiting on a record that another program has locked and the other program is waiting on a record that the first program has locked. Both programs are waiting on each other to release a resource, which never occurs.
- Given the current design of the application, which functions can execute simultaneously and which functions cannot execute simultaneously?
- Will the records I lock with a read for update impact the performance of other programs? How long should a record be locked?

These issues are covered in greater detail in the Recovery chapter on page 18.

Recovery

In previous VSAM batch systems, whenever a program abends, the files are deleted, redefined, and restored to a point in time. This type of recovery is not practical in a shared environment because the files have to be taken off line. VSHARE gives many recovery alternatives to recover from an abend in a shared VSAM environment. The Recovery chapter on page 18 discusses all the recovery options in detail.

Performance

VSHARE was designed with performance in mind. An objective of VSHARE is to give the programming staff total control over performance. First, if the online system needs performance priority, the VSHARE online transaction can be given a lower priority within CICS. In this way, the online programs get serviced by CICS before any batch jobs. Second, the system programmers can define the maximum number of concurrent batch jobs that can access a CICS region's files. By setting the maximum, the system programmers can anticipate the worst case scenario. In fact, the maximum number of concurrent jobs can be varied dynamically during the day. Third, the number of requests sent by a batch job can be reduced before execution or dynamically during execution for better control. Fourth, the VSHARE online transaction uses as little CICS resources as possible. The majority of the time it is suspended waiting for the I/O to complete. The impact to the online system will be minimal. Fifth, when a batch job is not accessing files owned by the CICS region, VSHARE has no impact to your CICS region. When VSHARE is not executing you can expect the same online throughput achieved today.

Now that CICS programs and batch programs can access the same files simultaneously, the need to tune a VSAM file under CICS becomes important. Refer to the chapter "VSAM Performance" on page 70 for hints on how to improve VSAM performance within your CICS region and within your application system. Now that CICS is VSAM's database manager, it should be tuned to run as efficiently as possible. Contact the internal performance group or the system programming staff for suggestions on better VSAM performance. The better a CICS region is tuned for VSAM, the faster VSHARE will perform.

Problem Determination

What happens when a program abends? How can I determine that the problem was an application problem or a problem with the connection between the batch job and the CICS region? These questions are important and easy to answer. VSHARE provides much information about the batch-CICS connection. In the event of an abend, use the following steps:

- Read the job log in the batch job to determine the abend code. If it is a system abend, determine where the abend occurred. If it was in the application, perform normal problem resolution techniques. Also, read the CICS log. VSHARE writes messages to this log when jobs start and complete as well as other messages dealing with Cross Memory Services.

Programmer's Guide

- If the abend was a user abend 2000, 1000, or 999, VSHARE abended for a specific reason. Read the VSHARE log and the CICS log for possible error messages. All of the messages are documented in the Messages and Codes chapter of this manual (page 125). The messages provide all the necessary information to solve the problem.
- If the message points to a VTAM error, refer to the CICS log to determine whether the problem was in CICS or the batch region. Also, look up the sense codes displayed in the VTAM Messages and Codes chapter for more information. You may need to consult a systems programmer since many of the VTAM message explanations are cryptic.
- VSHARE contains an internal trace and external auxiliary trace facilities. In the event of a problem, a dump will contain an internal trace table. Also, an auxiliary trace can be turned on through the online screens to aid CSI technical support in solving any problem.

Assembler Support

Since Assembler allows direct access to certain VSAM control areas, some Assembler capabilities are not supported by VSHARE. The VSAM control areas reside in the CICS rather than batch address space. The Assembler functions that are *not supported* are as follows:

- Control interval access
- Addressed access for KSDS Files
- Chained RPLs
- Multiple RPLs for one file
- AIXPUT operations
- AIXGET operations
- Multiple TCBs

Non-Unique Keys with Alternate Index Files

Because of restrictions imposed by CICS, updating non-unique keyed records of an alternate file has some limitations. For an alternate index built over a KSDS file, a batch program can update or delete only one of the duplicate records. After the operation is performed, the record position is set to the next keyed record not the next duplicate record. For an alternate index built over an ESDS file, a batch program can update only the first duplicate record. If an attempt is made to update any other duplicate record, a return code of 92 (X'5C') is issued that indicates that a previous get for update was not issued before a rewrite. When updating non-unique keyed alternate index files, try to update or delete records against the base cluster.

Summary

VSHARE places little burden on the applications programmer. VSHARE is truly transparent. The programmer needs only to understand how VSHARE uses CICS as the VSAM database manager and the implications of executing a program in a shared environment. VSHARE allows developers to approach a VSAM system exactly like a system that uses a DBMS. The result is greater flexibility and functionality.

VSHARE also provides system programmers with a desirable and familiar architecture. VSHARE gives them total control, a familiar tuning environment, and debugging facilities to solve any problem.

Recovery

Overview

This chapter explains how recovery changes in a shared VSAM environment, and it explains how to recover after an abend with VSHARE.

VSHARE offers a tiered solution to recovery, providing three possible recovery alternatives. The one you choose depends on the characteristics of the batch program.

Recovery in a Shared Environment

Without a VSAM shared environment, a batch program updates VSAM files in a vacuum. No other programs update the files owned by the batch program because of the share options. In a shared environment, this is not the case. Now more than one program can update and read the same VSAM file. How does this affect recovery? By answering the following questions, the difference of executing and recovering in a shared environment can be seen.

Is Delete, Define, Repro Practical for Recovery?

In the past, the typical way of recovering after an abend was to delete the updated files, redefine them, and reload them from a previously taken backup. Is this practical in a shared VSAM environment? The answer is no. If more than one program can update a VSAM file, deleting it requires the file to be taken off-line. This results in the inability of any other program to access the files during file recreation. This is impractical in a shared environment and it defeats the purpose of VSAM sharing. A new scheme other than delete-define-reproduce is required.

Who Else Can Update My Record?

In a shared environment, programmers have to take into consideration the possibility of other programs updating the same files, updating the same records, or updating the same fields. The possibility of another program updating the same file, record, or field at the same time has a direct impact on how to recover after an abend.

How Long Should I Lock a Record?

Not only is there a need to understand who may update the same VSAM file, but programmers also have to be sensitive to how long their programs lock a record. If a program updates every record in a critical file and never releases control, the entire file will be locked by the end of the program. Locking an entire file or a portion of a file defeats the whole purpose of a shared environment. It also has performance implications because other programs wait for the locked records.

What Happens If My Program Abends?

Since Delete, Define, and Reproduction cannot be used, and programs have to be sensitive to external programs updating the same VSAM file, what happens in the event of an abend? How can a file be recovered in a shared VSAM environment? VSHARE has the ability to recover a file in a shared VSAM environment. To explain VSHARE's recovery options, a basic understanding of how CICS handles record locking and data integrity is needed.

Locking and Releasing Records Within CICS

To provide data integrity, CICS locks any record updated, deleted, added, or read for update by a program. By locking the records, CICS guarantees data integrity by not allowing any other program to access the locked records. The records are locked until the program notifies CICS that it is safe to unlock the records for others to access. The time from when the first record is locked until the time the program informs CICS to release the records is called a Logical Unit of Work, or LUW.

Records are unlocked by issuing a syncpoint to CICS. When CICS receives a syncpoint, CICS releases exclusive control of all the records locked by that program. A syncpoint can be issued in two ways. The first is through a direct syncpoint call to CICS from the program. The second way is after the successful completion of the program. When a program ends normally, CICS assumes that the program's Logical Unit of Work is complete, and CICS releases all the records updated.

The opposite operation to a syncpoint is a rollback. Since CICS locks all the records modified during a Logical Unit of Work, a program has the ability to back out the changes by issuing a rollback to CICS. Like the syncpoint, a rollback can be issued in two ways. The first is through a direct rollback call to CICS from the program. The second way is after the program abends. When CICS senses that a program has abended, it calls a facility called Dynamic Transaction Backout, or DTB. DTB backs out or reverses all the changes made to all VSAM files during a Logical Unit of Work.

Multiple Programs Updating the Same Record

What happens when two programs ask CICS to update the same record? CICS allows the first program to access and update the record. Since the record is updated, CICS locks the record until the Logical Unit of Work for the program is complete. The second program has to wait until the first program gives CICS permission to release and commit the change. At that point, the second program can access the record and perform its operation.

If the first program does not issue a syncpoint until it completes execution, there would be no integrity exposure because all of the program's work is completed before the updates are committed. If the program abends during execution, CICS would back out the changes through DTB with 100-percent confidence that no other changes were made to the locked records.

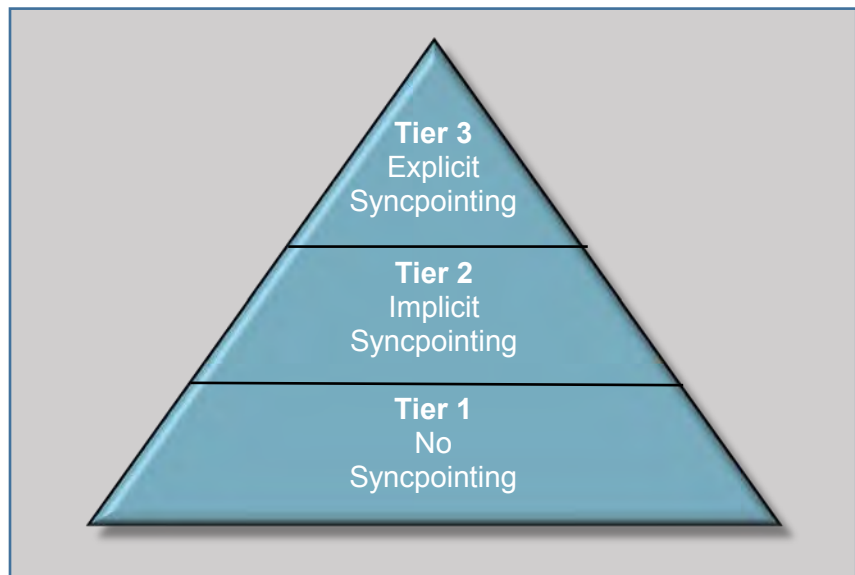
This seems like the ideal solution. Why worry about recovery, then? The side effect of not releasing records for a long period of time is the impact on performance. Other programs are forced to wait for locked records until they are released. If a program updates every record in a file over a period of one hour, any program wanting to update a record would be forced to wait up to one hour. It would be unacceptable to have someone wait for a response from the terminal for even two minutes. For this reason, there is a need in long-running

programs to release records on a regular basis. But when records are released, other programs can update the released records before the program finishes execution. What happens now if the program abends? How will the changes be backed out without affecting other programs that may have updated the records?

VSHARE's Three Recovery Tiers

The conflict between the ease of implementing recovery by locking all records updated by a program until it completes normally, and the practical need to release records on a regular interval for performance reasons, does not make recovery in a shared environment straight forward. VSHARE provides three alternatives to recovery. They give you the flexibility to choose the one right for your situation. The method you choose depends on how long the program runs, how many records are updated, and who is impacted by locked records. The first, Tier 1, provides the ability to lock all the records for the entire job, while the second two, Tiers 2 and 3, provide a way to issue syncpoints on a regular basis.

Tier 1 is the simplest to implement, while Tier 3 requires some work but may be required based on your current work load. With this tiered approach, you can implement the majority of your system quickly while providing full recoverability.



The remainder of this chapter reviews each recovery method. Refer also to “Comparison of VSHARE Recovery Tiers” on page 44.

Tier 1: No Syncpointing

What Is Tier 1? Tier 1 recovery allows CICS to lock all records updated throughout an entire step of a job. This is accomplished by not issuing a syncpoint until the step completes normally. By not releasing any updated records until the job completes normally, there is no danger of any other program changing the records updated. 100-percent data integrity is provided.

What Type of Job Should Use Tier 1? Any job can use Tier 1 theoretically, but there is a practical limit to which jobs should use it. Since no syncpoints are given, CICS will lock all the records updated and will not allow any other program to access the locked records until the program ends. Therefore, if a job runs for one hour updating many records in a critical file, other programs accessing any of the records will wait for up to one hour. In a production online environment, this may not be desirable for performance reasons. Tier 1 is used by either short-running programs or long-running programs that do not lock critical records.

How Is Tier 1 Implemented? Tier 1 implementation is easy. When a job is defined through the Job Maintenance Screen, the field called SYNC LEVEL is set to zero. This field is highlighted in the figure below. Refer to the chapter “Online Control” on page 41 for details. A value of zero in this field means that VSHARE does not issue any syncpoints until the job completes normally. Other than defining the job, there is nothing else to do.

```

=====
APPLID:  UAPPLID      FSI-VSHARE - THE CICS CONNECTION      DATE:  02/01/2015
PROGID:  CSIVSC15    J O B   M A I N T E N A N C E          12:47:36
=====
FUNCTION:  ( A = ADD, C = CHANGE, D = DELETE )

JOB NAME:

JOB INFORMATION      | HISTORY INFORMATION | PROCESSING INFORMATION
CICS REGION: P390CICS | RECORD?: Y          | ABEND IF NO CICS?:
TRANID: TCCB         | GENERATIONS: 003    | DO INPUT FILES?:
SYNC LEVEL: 000      |                     | ABEND IF DISABLED?:
REQS/MINUTE: 00000   | JOURNAL INFORMATION | TURN TRACE ON?:
ACCESS TYPE: CMS     | JOURNAL UPDATES?: N | FORCE ROLLBACK?: N
=====
DYNAMIC JOURNAL NAME:
//ITIJRNL DD DSN= PROD.FSIVSHAR.R30 . .ITIJRNL,DISP=MOD,
DCB=(RECFM=VB,BLKSIZE= 27004 ,LRECL= 09000 ,BUFNO= 010 ),
SPACE=(CYL,( 0010 , 0050 )),UNIT= SYSDA
=====

```

Recovery

How Are the Files Recovered After an Abend?

Since Tier 1 relies on CICS to lock the records entirely, it also relies on CICS to back out the changes in the event of an abend. When an abend is detected by VSHARE, it issues a rollback to CICS. This rollback informs CICS to back out all the changes before releasing the records to others. In this way, there is no need for any recovery. It is totally automatic. After the reason for the abend is isolated and resolved, rerun the job at the beginning of the step in error.

Tier 2: Implied Syncpointing

What Is Tier 2? Tier 2 recovery provides the capability of implied syncpointing. This means that VSHARE issues syncpoints on behalf of the batch job at predefined intervals without any program modification. By issuing syncpoints, the records updated by the job are released, allowing other programs to access them. To keep track of all the changes made by the program for recovery purposes, Tier 2 provides a journaling and recovery system. In the event of an abend, a VSHARE recovery job is executed that backs out all the changes made by the batch program. The recovery job executes while other programs are still accessing the files. VSHARE recovery backs out only the bytes or fields on the records changed by the program. This means that any other fields modified on these records by other programs are not affected.

What Type of Job Should Use Tier 2? Tier 2 is suited for long-running jobs that update critical records and files. VSHARE issues syncpoints periodically, so there is no danger of locking out programs for a long period of time. Since VSHARE keeps track of all the changes, there is a way to back out the changes in the event of an abend. When two programs update the same record, there are two possible scenarios. The first is when both update the same record but different fields, and the second is when both update the same record and the same fields. Tier 2 is only suited for the first scenario, where two programs update the same record but different fields. Since VSHARE recovery backs out the bytes on the record that change, other fields updated on the record are not affected. Like Tier 1, there are no program modifications required. But since the recovery backs out the changes to a field, other modifications made to the same field between the releasing of the record and the program abend would be lost. In this case, Tier 3 recovery should be used.

How Is Tier 2 Implemented? Tier 2 implementation is straightforward since there is no need to make any program modifications. Since Tier 2 uses VSHARE journaling facility, there is a need to quickly review basic journaling and recovery principles.

What Is Journaling? Journaling is the process of recording all changes made to files by a program. A journal must contain enough information to reverse all changes. This includes a list of all records added, all records deleted, and all records updated.

Why Use Journaling Since Tier 2 issues syncpoints to CICS, CICS releases the records and allows other programs to update the same records. For this reason, there is a need to keep track of all the modifications performed by a batch program. Journaling is the way to keep track of the changes.

What Is Backward Recovery? Backward recovery is the process of reversing all changes made to files. The journal dataset provides all of the information needed to achieve this. Records added can be deleted, records deleted can be added, and records updated can have their original contents restored.

Journaling and Recovery with Tier 2 VSHARE contains its own journaling system for Tier 2 recovery. VSHARE records all the file changes made by VSHARE to a journal file. The journaling is automatic and transparent to the application program.

Change requests are journaled in the same way CICS keeps track of changes in its journals, with one important exception: for update requests, both the pre-update record (before image) and the post-update record (after image) are journaled. With both the before and the after images saved, the specific bytes changed are identifiable.

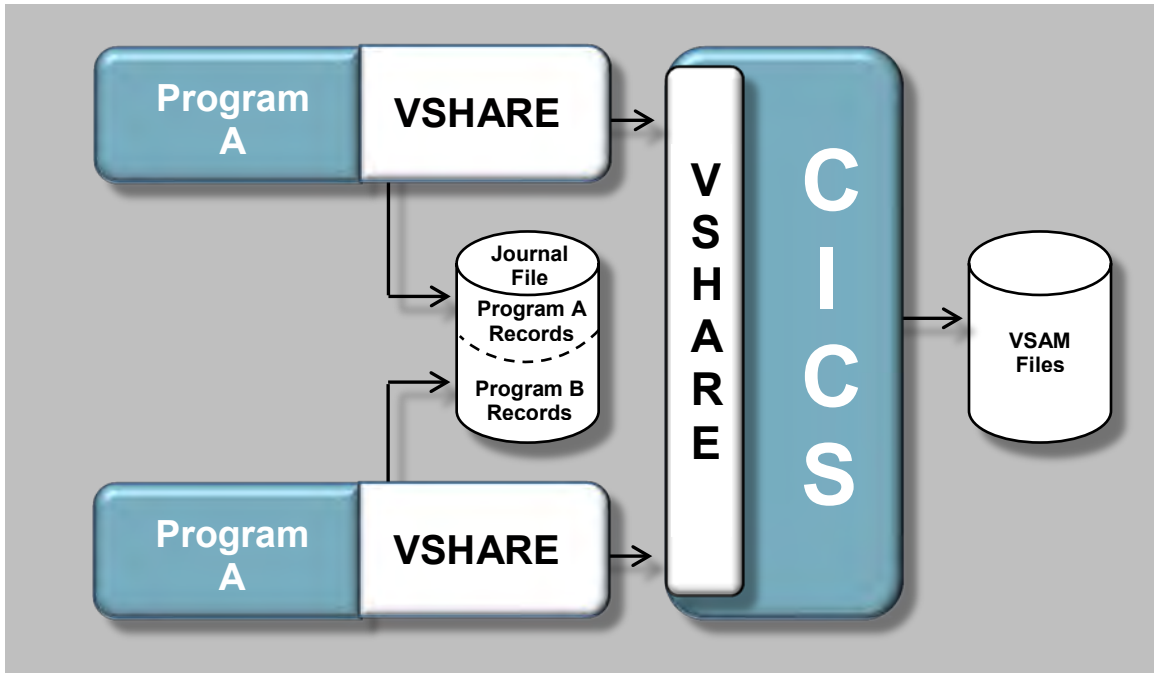
VSHARE also contains its own recovery system. Added records are deleted, and deleted records are added. Updates are recovered at the field level instead of at the record level. The VSHARE recovery system compares the before and after image from the journal file and restores only the bytes that have changed. The file should be taken offline to CICS for recovery purposes so that if the recovery job fails, you do not have to try and recover over top of another recovery job.

Tier 2 Journaling Example The following example shows the steps taken during the journaling process. Refer to the figure below.

- A job stream starts to execute that contains application programs A and B.
- Program A starts to run, thus activating the VSHARE interface.
- VSHARE activates journal processing. Since program A is the first program in the job stream, the journaling system creates, dynamically allocates, and catalogs a new journal dataset.
- As program A makes I/O requests, VSHARE calls upon the journaling system to record each add, change, or delete. The journaling system writes journal records for each of these requests.
- Program A finishes processing. VSHARE terminates journaling activity, and the journal file is closed.
- Program B starts to process, thus activating VSHARE's interface.
- Once again, journal processing is activated by VSHARE. Since a journal dataset is already cataloged for this job stream, the journaling system dynamically allocates the existing journal dataset with MOD for disposition.

Recovery

- As with program A, I/O requests are recorded in the journal dataset as program B makes them.
- Program B finishes processing, and the journal file is closed once again. At this point, the journal file contains all of the changes made by programs A and B, and the job stream is finished.



Tier 2 Recovery Example The following example describes the steps in the recovery process.

- Assume program B from the above example job is halfway through its processing and abends. At this point, the journal file contains a record of all changes made by program A plus all changes by program B prior to the abend. It is determined that those changes must be removed from the files before the job can be restarted.
- The VSHARE recovery job stream is started using the journal dataset.
- The recovery system processes the journal file backwards using VSHARE to remove the changes. The entire journal file is processed, thus removing all changes made by programs A and B.
- The files are restored to their original state, allowing the re-execution of the job.

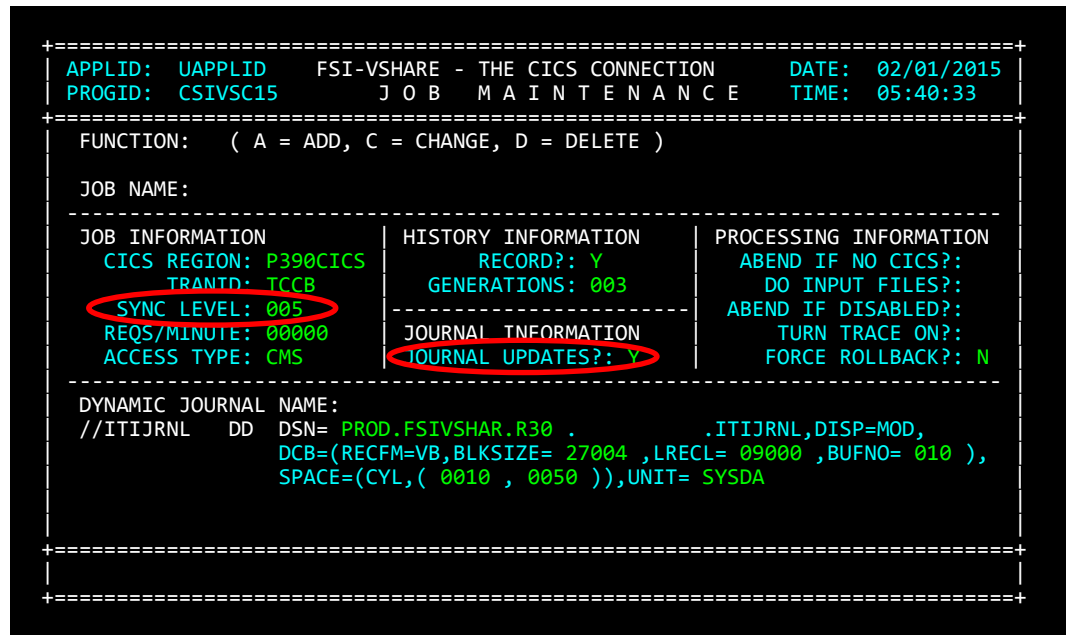
It would also be possible to back out only program B's changes and then restart the job at program B's beginning. This option is discussed later in this chapter.

Define the Job to Use Journaling

Journaling is optional with VSHARE. You can enable journaling at the job or the job group level. Place a Y in the JOURNAL UPDATES? field on the Job Maintenance screen. Refer to the figure below. The dataset characteristics of the journal file can be modified on this screen as well. Refer to the section “Defining Which Jobs Can Access the Files” on page 56 for details on how to define a job to VSHARE.

Define the Syncpoint Level

So that VSHARE can know when to issue an implied syncpoint, a syncpoint level needs to be defined in the Job Maintenance screen. The SYNC LEVEL value tells VSHARE how many locks are permitted before a syncpoint is taken. Refer to the figure below.



Notice that VSHARE can issue a syncpoint only when there is no outstanding read for update. If it did issue a syncpoint when a read for update was outstanding, the record read for update would be released. As a result, it would not be under the control of the batch program. Therefore, if a record in a file is read for update at the beginning of a job and is not updated until the end of the job, VSHARE cannot issue a syncpoint. If messages in the VSHARE log inform you that extensive locks are being held and you defined the job with implicit syncpointing, this situation may be the reason why the locks are being held after the syncpoint threshold has been met.

Create a VSHARE Recovery Procedure

A sample recovery member called RECOVER is in the VSHARE installation library. This JCL is customized during the installation process. Contact the person responsible for installing VSHARE for the name of the JCL member or the procedure created for recovery. If a customized member has not been created, copy the member RECOVER from the VSHARE installation library.

Recovery

The recovery procedure consists of three job steps. Refer to the table below.

Job Step	Function
1	This step deletes and re-defines an ESDS VSAM recovery dataset.
2	<p>This step loads the sequential journal created during the batch job into the VSAM dataset created in job step 1. The file name of the sequential journal is based on three items:</p> <ul style="list-style-type: none">• The high level qualifier is the name defined on the System Maintenance screen for the CICS region.• The second qualifier of the name is the name of the job executed.• The last qualifier is the literal 'ITIJRNL'. For example, if a high level qualifier for all jobs communicating to the production CICS region was defined as PROD.CICS, and a job named PRODUPDT was executed, the journal dataset for this job would be 'PROD.CICS.PRODUPDT.ITIJRNL' <p>This journal is loaded into an ESDS dataset so that the journal records can be read backwards. All changes have to be backed out from most recent to least recent.</p>
3	The third step executes the VSHARE recovery program to back out the changes based on the recovery commands.

The recovery program accepts parameters that control the recovery process. These parameters are passed to the recovery program as control cards in the SYSIN dataset. The next section explains the format and meaning of these recovery parameters.

Creating VSHARE Recovery Parameters

The recovery process always starts with journal records from the last job step executed and works backward through the records for prior steps that used VSHARE. The control cards for the recovery program are used to specify when the recovery process should stop. The parameters identify the last job step for which recovery will be performed. Changes can be backed out for the last step journaled or for any number of prior steps in the job that used VSHARE, depending upon the application's requirements.

The control cards are 80 characters each and are read by the recovery program from the SYSIN dataset. Each control card can contain one recovery parameter.

The format of each parameter is described in the table below.

Parameter	Description
JOBNAME=XXXXXXXX	This is the job name from the job card. It is not required if the JOBNUM= parameter is used. It is useful if journal files are combined as jobs are run.
JOBNUM=NNNN	This is the JES job number assigned by the operating system. It is not required if the JOBNAME= parameter is used. It is useful in cases where a journal contains records for multiple job executions.
PGMSTEP=XXXXXXXX	This is the label from the EXEC PGM=XXXXXXXX JCL card. For instance, STEP100 would be used as the PGMSTEP parameter for the following: //STEP100 EXEC PGM=YOURPGM, COND=(0,NE)
PROCSTEP=XXXXXXXX	This is the label from the EXEC PROC=XXXXXXXX JCL card. For instance, PROC100 would be used as the PROCSTEP parameter for the following: //PROC100 EXEC PROC=YOURPROC, UNIT1=SYSDA
RECOVERALL	This parameter causes the recovery program to process the entire journal file. Other parameter cards are not accepted with this one because the combination would be inconsistent.
* <i>comment</i>	If the first character of a control card is an asterisk (*), the line is treated as a comment.

Whenever a job step is executed using VSHARE, a message log is included in the SYSOUT for the step. All of the recovery parameter values for the step are printed as a message in the log. This provides an easy means for identifying what values to use and for verifying that they are correct. Note that if the parameters specified are not encountered during the recovery process, the entire journal file will be processed.

What Is Journalled and What Is Not Journalled?

VSHARE journals changes only for the files owned by the CICS region and controlled by VSHARE. For example, if a job updates one file owned by CICS and another not owned by CICS (a “batch only” file), VSHARE journals only the updates to the CICS-owned file. The “batch only” file is updated directly by the batch job and is not controlled by VSHARE. Therefore, no journaling is performed for these files, and recovery must be done by alternative methods.

Tier 2 Recovery Example with an Executed Procedure

The job PRODJOB1 contains two procs. The first proc, labelled JOB1001, executes the procedure PROC1. The second proc, JOB1002, executes the procedure PROC2. PROC1 (JOB1001) is comprised of three steps—STEP001 through STEP003—and PROC2 is comprised of four steps—STEP001 through STEP004. Assume that PRODJOB1 is submitted and it abends in STEP003 of PROC2. After analyzing the abend, it is determined to restart the job at STEP002 of PROC1.

To back out all of the changes made through VSHARE except those before the restart point, the following steps are necessary:

1. Add the DD statements to the RECOVER JCL for all of the files being recovered.
2. Add the following control cards to the SYSIN dataset:

```
//SYSIN DD *  
* Back out changes through STEP002 of PROC1  
JOBNAME=PRODJOB1  
PROCSTEP=JOB1001  
PGMSTEP=STEP002  
/*
```

3. Run the recovery job to back out the changes.
4. Recover any files not processed by VSHARE.
5. Back up the sequential journal file if desired.
6. Delete the sequential journal file. If you don't, the next execution of the job will append changes to the same sequential journal. Accumulating journal records for multiple job executions can make recovery confusing, and it also increases the likelihood of running out of space on the journal file. A new file is recommended for each execution of a job.
7. Rerun the job with a restart of STEP002 in PROC1.

The recovery job can be executed with the CICS region either up or down.

Recovery

Recovery Example Without an Executed Procedure

In this example, there is a job ADDCUST that contains three steps. The first step executes PROG1, the second step executes PROG2, and the third executes PROG3. Assume that ADDCUST is submitted and abends in STEP3. After some analysis it is determined to restart the job at STEP3. To restart the job at this step, the following steps are taken:

1. Add the VSAM datasets updated in STEP3 to the RECOVER JCL.
2. Add the following control cards to the SYSIN dataset:

```
//SYSIN DD *  
*  
* Back out changes through STEP3 of ADDCUST  
*  
JOBNAME=ADDCUST  
PGMSTEP=STEP3  
/*
```

3. Run the RECOVER job.
4. Back up the ADDCUST sequential journal if needed.
5. Delete the sequential journal.
6. Restart ADDCUST at STEP3.

Tier 3: Explicit Syncpointing

What Is Tier 3? Tier 3 recovery provides the capability for a program to directly call VSHARE to issue a syncpoint or a rollback. Unlike Tier 2, Tier 3 jobs update fields that other programs may update after a syncpoint is taken. To provide full data integrity, the program has to have total control over its Logical Unit of Work, and it must directly call CICS to issue syncpoints. The implied syncpoints of Tier 2 do not provide for exact syncpointing at specific points in a program's execution. In other shared environments like a DBMS, the DBMS provides the capability of issuing a syncpoint from within an application program. Tier 3 recovery provides this same capability. An interface is provided that allows a program to call VSHARE directly.

When two programs update the same field in the same record, it is important never to back out a change after a syncpoint is given because of the possibility of backing out another program's changes. This is why total control of syncpointing is important. With Tier 3, a program must be modified to issue syncpoints, and an application recovery procedure is needed to start processing after the last good syncpoint. Then there is no danger of backing out updates made by other programs.

What Type of Job Should Use Tier 3? Tier 3 is suited for long-running jobs that update critical records and files. Also, these jobs update the same files other programs may update. Unlike Tiers 1 and 2, program modifications are required. Like any shared environment, Tier 3 programs require some work to provide full data integrity.

How Is Tier 3 Implemented? The first requirement is to make sure that the SYNC LEVEL field on the Job Definitions screen is **set to zero**; otherwise, implied syncpointing is invoked. Implied syncpointing is not desirable since the program would lose control over syncpoint activity.

Determine Whether VSHARE Journaling Is Required If there is no way to determine where the last good syncpoint was taken during recovery, turn on VSHARE journaling. VSHARE journals any syncpoint that is taken. The application's Tier 3 recovery procedure can read the journal backwards to find the last syncpoint taken, which can determine where to continue processing. If the application has another mechanism—like a sequential file—that can determine the last good Logical Unit of Work, VSHARE journaling is not required and should be turned off for Tier 3 jobs.

Modify the Program to Issue the Syncpoints or Rollbacks VSHARE provides an application program interface module that can be called from a user program to request either a syncpoint or a syncpoint rollback. This interface is supplied for programmers that want to define their own logical units of work. Requesting either a syncpoint or a rollback using the interface causes the VSHARE transaction associated with the batch program to execute the

respective request under CICS. The request will be executed regardless of whether file locks are held. For example, if the program reads a record for update and subsequently requests a syncpoint, the read update lock on the record will be lost. It is the programmer's responsibility to ensure that files are in the expected state before issuing syncpoint or rollback requests.

Batch-Only Mode Programs that use syncpoint services can still be run in batch-only mode (that is, without the VSHARE interface). In this case, the syncpoint or rollback requests have no effect on program execution.

For programs that run in batch-only mode by removing the VSHARE load library from either the JOBLIB or STEPLIB concatenation, the module BIMVSBAI will have to be moved to one of the other libraries in the concatenation. This is required for programs that dynamically link to the application interface (that is, they use the COBOL DYNAM option).

Requesting a Syncpoint or Rollback To request a syncpoint or rollback, the program sets up a request parameter list and then calls the VSHARE application program interface module BIMVSBAI. On return from BIMVSBAI, the application program should then interrogate the return code area to ensure that the request was performed successfully. If a critical error is encountered during processing of the syncpoint or rollback request, an abend will be issued and control will not be returned to the application program. The COBOL parameter list and request code literals are included as copybooks CCONPRM and CCONVAL in the installation library, CICS.CONN.INSTALL.

Sample Syncpoint Request

```
WORKING-STORAGE SECTION.  
  
(A)    COPY CCONPRM.  
       COPY CCONVAL.  
  
PROCEDURE DIVISION.  
  
(B)    MOVE CCON-PV-SYNCPOINT TO CCON-REQUEST.  
  
(C)    CALL CCON-API-PGM USING CICS-CONN-PARM-LIST.  
(D)    IF CCON-MSG-RETURNED  
       DISPLAY CCON-MSG-TEXT.  
  
(E)    IF CCON-STATUS-OK  
       DISPLAY 'SYNCPOINT SUCCESSFULLY EXECUTED'  
    ELSE  
       DISPLAY 'SYNCPOINT FAILED'  
       GO TO ABEND.
```

The lettered lines are explained in the table below.

Line	Function
A	Copy the parameter list and parameter list values into program working storage. See the JCL below.
B	Set the request type to syncpoint.
C	Call the program BIMVSBAI, passing the parameter list.
D	Display any messages returned.
E	Check whether the request completed successfully. For critical errors, the VSHARE interface abends the process immediately without returning control to the application program.

Recovery

```
*****
*   FSI-VSHARE - THE CICS CONNECTION                               *
*                                                                 *
*   CCONPRM - APPLICATION INTERFACE COBOL PARAMETER LIST *
*****
01  CICS-CONN-PARM-LIST.
    05  CCON-VERSION                PIC X(04)  VALUE 'R112'.
    05  CCON-REQUEST                PIC X(04).
        88  CCON-REQ-SYNCPOINT      VALUE 'SYNC'.
        88  CCON-REQ-ROLLBACK      VALUE 'ROLL'.
    05  CCON-STATUS                PIC X(02).
        88  CCON-STATUS-OK         VALUE '00'.
        88  CCON-STATUS-FAILED     VALUE '08'.
    05  CCON-OPTION                PIC X(02).
    05  CCON-MESSAGE-AREA.
        10  CCON-MSG-IND           PIC X(01).
            88  CCON-MSG-RETURNED   VALUE 'Y'.
            88  CCON-MSG-NOT-RETURNED VALUE 'N'.
        10  CCON-MSG-TEXT          PIC X(132).
    05  CCON-PARM-END.
        10  CCONN-PARM-END CON     PIC X(04)  VALUE 'PEND'.
*
*   END OF CCONPRM

*****
*   FSI-VSHARE - THE CICS CONNECTION                               *
*                                                                 *
*   CCONVAL - APPLICATION INTERFACE PAARMETER VALUES *
*****
01  CICS-CONN-API-VALUES.
    05  CCON-API-PGM                PIC X(08)  VALUE 'BIMVSBAI'.
    05  CCON-API-PARM-VALUES.
        10  CCON-PV-SYNCPOINT      PIC X(04)  VALUE 'SYNC'.
        10  CCON-PV-ROLLBACK      PIC X(04)  VALUE 'ROLL'.
*
*   END OF CCONVAL
```

Issuing Syncpoints and Rollbacks Through a Batch Exit Program

In the event that the batch program's source code cannot be modified to directly issue syncpoints and rollbacks, VSHARE can be notified to issue a syncpoint or rollback by a user-written batch exit program. After each VSAM I/O, VSHARE links to an exit program called BIMVSBEX. A parameter list like the one shown in the following figure is passed to the user exit program, BIMVSBEX. When the exit program returns control to VSHARE, VSHARE interrogates the parameter list and issues a syncpoint or rollback depending on the request setting.

EXITPRMC Copybook For The Batch Exit Program

```
01 CICS-CONN-EXIT-LIST.
   05 CCON-IDENT          PIC X(04).
   05 CCON-REQUEST       PIC X(04).
       88 CCON-REQ-SYNCPPOINT VALUE 'SYNC'.
       88 CCON-REQ-ROLLBACK VALUE 'ROLL'.
   05 CCON-LAST-IO-REQUEST PIC S9(4) COMP.
       88 CCON-GET          VALUE +0.
       88 CCON-PUT          VALUE +1.
       88 CCON-CHECK        VALUE +2.
       88 CCON-POINT        VALUE +3.
       88 CCON-END          VALUE +4.
       88 CCON-ERASE        VALUE +5.
   05 CCON-DDNAME        PIC X(08).
   05 CCON-RPL-FEEDBACK-CODE PIC X(03).
   05 CCON-PROGRAM-NAME PIC X(08).
   05 CCON-JOB-NAME      PIC X(08).
   05 CCON-STEP-NAME     PIC X(08).
   05 CCON-JOB-NUMBER    PIC X(05).
   05 CCON-DATASET-NAME  PIC X(44).
```

The copybook EXITPRMC provides the layout for the VSHARE batch exit parameter list. This copybook is located in the installation library. You can copy it into your application's programs. The Assembler version is called EXITPRMA.

The user exit can be written in COBOL 2.4 or Assembler. A program example is shown in the following figure. On entry, the exit program is passed a parameter list with the jobname, the VSAM request just executed, the VSAM feedback code, the name of the program requesting the VSAM request, the DD name being processed, the step name, the cluster name, and the job number. The exit program can then interrogate the information and determine whether a syncpoint or rollback is needed. If a syncpoint is needed, move the literal "SYNC" into the request field. If a rollback is needed, move the literal "ROLL" into the request field. If no action is required, move spaces to the request field. After the type of request is set, exit the program. VSHARE will then get control and perform the appropriate request.

The exit program can perform any type of processing except issuing a VSAM operation. Issuing a VSAM request will result in unpredictable results. Also, the exit program is dynamically loaded at execution time by VSHARE; therefore, each application can have its own exit routine if the program named BIMVSBEX is placed at the beginning of the concatenation of the load libraries. If an exit program does not exist, VSHARE will ignore exit processing. Finally, do not call the BIMVSBAI program used to issue syncpoints and rollbacks from BIMVSBEX because the results will be

Recovery

unpredictable. The batch exit program was designed to be given control after each VSAM I/O and before control is passed back to the application program. If you call BIMVSBAL, VSHARE will be invoked at the beginning, resulting in a recursive call.

Example of the Batch Exit Program TCCBTEXT

```
*****
*           L I N K A G E   S E C T I O N           *
*****
LINKAGE SECTION.
  COPY EXITPRMC.
  PROCEDURE DIVISION USING CICS-CONN-EXIT-LIST.
*****
*           M A I N   L I N E           *
*****
* 1) CHECK RETURN CODE. IF IT IS NOT GOOD, EXIT.      *
* 2) CHECK THE JOB NAME, IF IT IS PROD001 CONTINUE ELSE *
*   EXIT.                                             *
* 3) CHECK THE DDNAME, IF IT IS FROM THE DATASET DD01, *
*   CONTINUE SEARCHING.                             *
* 4) CHECK THE VSAM REQUEST, IF IT IS A ERASE THEN REQ A *
*   SYNCPOINT.                                       *
*****

  IF CCON-RPL-FEEDBACK-CODE = LOW-VALUES
    IF CCON-JOB-NAME = WS-PROD001
      IF CCON-DDNAME = WS-DD01
        IF CCON-ERASE
          MOVE WS-SYNCPOINT TO CCON-REQUEST
        ELSE
          NEXT SENTENCE
      ELSE
        NEXT SENTENCE
    ELSE
      NEXT SENTENCE
  ELSE
    NEXT SENTENCE.
GOBACK.
```

The JCL above is an example of an exit program. The program is passed a parameter list on entry. On entry, it checks to make sure that the VSAM operation was successful. It then requests a syncpoint if the dataset DD01 issued an erase in STEP001 of job PROD001.

Syncpoint and Rollback Journal Records

Syncpoint and rollback requests are recorded in the VSHARE journal file associated with the batch job. Implicit or automatic syncpoints are also recorded in the journal. These are supplied for users who write customized recovery programs. See the JCL below.

Recovery

```

*****
*
*   JOURNAL CONTROL - JOURNAL RECORD LAYOUT
*
*****
JCRL  DSECT
*****
*   H E A D E R
*****
JCRLHDR  EQU  *           START OF HEADER
JCRLHDRL DS   H           HEADER LENGTH
*
*   FIXED AREA (RESERVED)
*
*...FIXED PREFIX AREA
          DS   CL49         RESERVED
JCRLDSN  DS   CL44         BASE CLUSTER NAME
          DS   CL45         RESERVED
*
*   CONTROL INFORMATION
*
JCRLCNTL EQU  *           START OF CONTROL INFO
*...STOP KEY INFO
JCRLJNUM DS   CL5          JOBNUMBER
JCRLJOB  DS   CL8          JOBNAME
JCRLPROC DS   CL8          EXEC PROC STEPNAME
JCRLPGM  DS   CL8          EXEC PGM= STEPNAME
          DS   CL3          RESERVED
*...RECORD IDENTIFICATION
JCRLDDN  DS   CL8          FILE DDNAME
JCRLDATE DS   PL4          00YYDDD+ (JULIAN SIGNED)
JCRLTIME DS   CL4          HHMMSSTH (TENTHS,HUNDRETTHS)
JCRLSEQ  DS   PL4          SEQUENCE NUMBER
*
JCRLTYPE          DS   C           DATA TYPE
JCRLADD           EQU  C'1'        1=ADD
JCRLDEL           EQU  C'2'        2=DEL
JCRLREAD          EQU  C'3'        3=READ
JCRLREW           EQU  C'4'        4=REWRITE
JCRLINIT          EQU  C'5'        5=STEP INITIATION RECORD
JCRLTERM          EQU  C'6'        6=STEP TERMINATION RECORD
JCRLSYNU          EQU  C'7'        7=USER SYNCPOINT RECORD
JCRLSYNA          EQU  C'8'        8=USER SYNCPOINT RECORD
JCRLROLU          EQU  C'9'        9=USER ROLLBACK RECORD
JCRLROLA          EQU  C'A'        A=AUTO ROLLBACK RECORD
JCRLSTAT          DS   X           STATUS BYTE
JCRLABND          EQU  X'80'       1... .... JOURNALED DURING ABEND
JCRLRKP           DS   H           FCT RELATIVE KEY POSITION
JCRLKEYL          DS   H           FCT KEYLENGTH
JCRLRECL          DS   H           FCT RECLENGTH
JCRLDATL          DS   H           DATA LENGTH
*****
*   DATA
*****
JCRLDATA  EQU  *           DATA PORTION

```

Create an Application Recovery Scheme

Now that the programs issue the syncpoints directly, there is a need to determine how to start after the last good Logical Unit of Work in the event of an abend. The key to doing this is to first find a way to determine the last good LUW. VSHARE's journal can be used by reading it backwards for the last good LUW, or a sequential file created from the program can be used. Finally, a program has to be able to be started where it left off through a facility like check-point restart or through a customized method. This part of Tier 3 recovery is the most time consuming to design.

Tier 3 Recovery Scenario

The following events occur during a Tier 3 program's execution, during an abend, and during the recovery after the abend.

- Program A is an order entry system. It is used to update an inventory file. Each order may require between 5 and 15 updates to the VSAM database. It was determined that a Logical Unit of Work would be one order; therefore, syncpoints were placed in the code after an order is processed. Program A begins execution. After each completed order, the program writes information to a sequential file with all the information needed to start processing after the last good order processed.
- Program A issues syncpoints throughout its execution after each order. Program A starts processing the 245th order, and it updates 5 of the 11 records required for that ordering. Unfortunately, a quantity field in a record of the sequential input file contains non-numeric data. The program abends with an 0C7 abend code.
- VSHARE detects the abend and issues a rollback in CICS. CICS in turn backs out the five updates performed in the current Logical Unit of Work. At this point, there are no unfinished orders left. All the orders from 1 through 244 have been completed and are committed to the files. VSHARE ends.
- The application programmer performs some error analysis and finds the bad data in the input file. The data is fixed.
- An order-entry restart job is started. Since the sequential recovery file is found, the order-entry program reads the file, repositions all the files to continue processing, and begins processing order 245. The job completes successfully.

Recovery and ESDS Datasets

ESDS files are sequential VSAM files that are accessed by relative byte address or accessed sequentially from beginning to end. Records that are added are placed at the end of the dataset, similar to QSAM datasets. One restriction with ESDS datasets is that records added cannot be deleted. This restriction has implications with respect to recovering ESDS files in a shared environment.

Recovery

In Tier 1, CICS DTB does not back out records added to an ESDS dataset. In Tier 2, the VSHARE recovery program cannot delete added records to an ESDS dataset. An error message indicating that a delete was attempted on an ESDS dataset is written for each delete attempt. In Tier 3, an added record is committed by CICS even before a syncpoint is written.

This limitation in ESDS must be kept in mind when reviewing recovery procedures. One alternative is to make the ESDS dataset a KSDS dataset if backing out additions may be necessary.

Comparison of VSHARE Recovery Tiers

The following table summarizes the key differences between VSHARE's recovery tiers and the definitions required for each.

Tier	Best For.	Sync Level	VSHARE Journaling Required?	Application Coding Required?	Syncpoints Issued? ¹	Abend Recovery Method
1	<ul style="list-style-type: none"> • Short jobs • Non-critical data 	0	No	No	No	Automatic—VSHARE requests a rollback
2	<ul style="list-style-type: none"> • Long jobs • Critical data • Control over target fields not shared 	1–999	Yes ²	No	Yes ³	User runs a VSHARE recovery job
3	<ul style="list-style-type: none"> • Long jobs • Critical data • Control over target fields shared with other apps 	0 ⁴	No ⁵	Yes ⁶	Yes ⁷	User runs a VSHARE recovery job if VSHARE journaling is used

¹ VSHARE always issues a syncpoint request at EOJ if a job completes normally. If an abend occurs, the appropriate recovery method is indicated in the last column.

² Setting VSHARE journaling to NO is *not* recommended because it limits VSHARE's recovery methods to using only the most recent LUW.

³ VSHARE issues an auto syncpoint request each time a sync level is reached.

⁴ For Tier 3, the Sync Level must be set to zero.

⁵ VSHARE journaling is recommended but not required if an alternative recovery method is used.

⁶ The application requires logic to call a VSHARE function to execute a syncpoint and/or rollback request.

⁷ A syncpoint request is issued when program logic calls CSIVSBAI with a SYNC request.

Online Control

Overview

VSHARE is designed to be easy to use while providing all the functionality required to have critical applications execute in a shared VSAM environment. One of the important characteristics required in a shared environment is the ability to have total control. This chapter covers the online programs provided with VSHARE. It is through these programs that much of the control is provided. The control is implemented through static definitions set before a job's execution or through dynamic means while the job is executing. Even with all the functionality that VSHARE provides, the screens are easy to use and easy to understand. There are no cryptic parameters to learn nor too many parameters making implementation complicated.

VSHARE online programs provide four basic functions:

1. Monitor active jobs using VSHARE.
2. Define jobs that can use VSHARE.
3. Browse statistics on previously executed jobs.
4. Define all the CICS regions running VSHARE.

Each function is explained in detail throughout this section. Included in each description is a functional overview and a description of all fields and attention keys.

To execute VSHARE's online system, sign on to the CICS region where VSHARE is executing and enter the transaction TCC. The first screen displayed is VSHARE Main Menu.

All VSHARE screens have the same format. The heading of each screen contains the CICS APPLID, the VSHARE program generating the screen, the date, the time, and the title of the screen. The 23rd line is the message line where all error messages and informational messages are displayed. For an explanation of the valid attention keys, read the message line for details.

3. JOB HISTORY

- a) View previously run job statistics at the step or dataset level.

4. SYSTEM MAINTENANCE

- a) Add the definition of a new CICS region running VSHARE.
- b) Change default processing options of a CICS region previously defined to use VSHARE.
- c) Remove a region from using VSHARE.

Dynamic Monitoring of Batch Activity

```

=====
APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE:  02/01/2015
PROGID:  CSIVSC13   A C T I V E   J O B S                TIME:  05:49:01
=====

```

CMD	JOB NAME	STEP NAME	REQS/ MINUTE	REGION	START TIME	VSAM REQUESTS	STATUS	ACCESS TYPE
-	PRODJOB1	EXECXXX	00000	PRODCICS	08:51:22	00057080	ACTIVE	LU62
-	PRODJOB4	STEP0200	04001	PRODCICS	08:55:10	00002301	ACTIVE	CMS
-	USER001C	STEP1000	01101	PRODCICS	09:01:12	00030704	ACTIVE	CMS
S	MASTERUP	UPDATE09	00000	PRODCICS	09:01:56	00021000	ACTIVE	LU62
-	TESTJOB1	STEP0038	00000	PRODCICS	09:11:21	00002306	ACTIVE	CMS
-	PRODFIX2	FIXIT001	00000	PRODCICS	09:15:34	00001106	ACTIVE	LU62
-	CC045RMA	EXEC0010	00000	PRODCICS	09:24:29	00001002	ACTIVE	CMS
-	SYS1221A	STEP1004	00000	PRODCICS	09:32:17	00000454	ACTIVE	CMS
-	ITIDKVMA	STEP0030	00010	PRODCICS	09:40:34	00000059	ACTIVE	CMS

```

=====

```

Online Control

The Active Jobs summary screen displays all the currently running jobs using VSHARE in chronological order. This screen allows certain processing options to be changed dynamically, and it allows you to monitor the VSAM shared environment. Each detail line on the screen displays information about each job.

If you need more detailed information on a job, type an S in the command field next to the job and press Enter. A job-details screen is displayed. From that screen, a job can be stopped, restarted, canceled, or modified with a new I/O rate. These operations are accomplished by entering the proper command next to the job name. To refresh the screen, press Enter. As each step completes, the detail line for that job disappears. Each field is explained below.

Field	Description
CMD	<p>Each detail line has a command field. The following values are possible, and more than one command can be entered on one screen. For example, placing an X in all the CMD fields stops all the jobs displayed on the screen.</p> <p>S Displays a new job details screen. This screen displays more detailed information about the selected job step.</p> <p>X Sends a request to VSHARE to stop or restart a job based on its current status.</p> <p>R Allows the operator to change the batch I/O request rate. By typing an R and pressing Enter, the request rate field becomes unprotected, which allows the value to be changed.</p> <p>C Send a request to VSHARE to cancel a batch job.</p>
JOB NAME	The name of the job as it appears on the JOB card.
STEP NAME	The name of the step as it appears on the EXEC card in the execution JCL.
REQS/MIN	The maximum number of I/O requests sent per minute to the CICS region. When you place an R in the command field press Enter, the field is unprotected. Enter any number representing the new rate. If zero is entered, the batch job sends requests as quickly as possible.
REGION	The APPLID of the CICS region executing VSHARE. This is not necessarily the same as the CICS job name.

Field	Description
START TIME	The start time of the displayed step.
VSAM REQUESTS	The total number of VSAM requests processed in the job step.
STATUS	<p>The status of the job executing. This message changes as commands are entered from this screen. Possible values are</p> <p>ACTIVE The batch job is active with no pending commands.</p> <p>STOPPED The batch job has been suspended. The job keeps this status until another command is entered.</p> <p>START PEND The starting of the job is pending.</p> <p>C PENDING A cancel was entered, but the batch job has not received the request.</p> <p>UPDATE REQ This message is displayed after an R is placed in the command line. At this point the request rate field is unprotected.</p> <p>REQ UPDATE The new request rate has been entered, but the batch job has not processed the new rate.</p>
ACCESS TYPE	<p>The type of communications used to communicate between batch jobs and the CICS region. CMS is the quickest communications type, but it can only be used when both the batch job and the CICS region are executing on the same CPU. Valid values are</p> <p>LU62 LU 6.2 is being used</p> <p>CMS Cross Memory Services interface is being used</p>

Attention Keys

- Enter Process the entered command(s) or refresh the screen with up-to-date information.
- Clear Return to the Main Menu.
- PF7 Scroll backward through the active job list.
- PF8 Scroll forward through the active job list.
- PF9 Display the beginning of the active job list.

Active Job Details

```

=====
APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE:  02/01/2015
PROGID:  CSIVSC14   A C T I V E  J O B S                TIME:  05:49:01
=====
JOB NAME: MASTERUP      STEP NAME: UPDATE      JES NUMBER: 6906
CICS NAME: PRODCICS     START TIME: 09:11:21
USER SYNCPOINTS: 0002000   AUTO SYNCPOINTS: 000000
DATASET LIST
_ CUSTOMER  _ SHARE    S MASTER
=====
    
```

Based on the selection you made on the Active Jobs summary screen, this screen displays more details, including a list of DDNAMEs currently opened. By selecting one of the DDNAMEs, a detail screen is displayed that gives dataset-specific information. The information displayed is explained below.

Field	Description
Job Name	The job name as it appears in the JOB card of the batch JCL.
Step Name	The step name as it appears in the EXEC card of the batch JCL.
CICS Name	The CICS APPLID of the region where VSHARE is executing.
JES Number	The job number assigned to the batch job by the operating system.
# OF	The number of records deleted in the current

Field	Description
DELETES	step. Pressing Enter refreshes the number.
# OF STARTS	The number of starts requested by the current step. Pressing Enter refreshes the number.
# OF ENDS	The number of end requests. Pressing Enter refreshes the number.

Attention Keys

- Enter Refresh the VSAM I/O statistics.
- Clear Return to the detail active job screen. If more than one DDNAME was selected on the previous detail active job screen, display the chosen DDNAME's statistics.
- PF7 Scroll backward through the DDNAME list.
- PF8 Scroll forward through the DDNAME list.
- PF9 Display the beginning of the DDNAME list.

Obtaining Dataset Statistics on Active Jobs

```

=====
APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE:  02/01/2015
PROGID:  CSIVSC18  D A T A S E T   S T A T I S T I C S   TIME:  09:47:42
=====
          DDNAME      ( MASTER      )
          FCTNAME     ( FILEMSTR   )
          CLUSTER NAME( PROD.MASTER.CLUSTER )
          DATASET TYPE( KSDS      )
          ACCESS      ( RUBAD     )
          LSR POOL    ( 01       )
          STRINGS     ( 004      )
          # OF READS  ( 00023120 )
          # OF ADDS   ( 00000213 )
          # OF UPDATES( 00002305 )
          # OF DELETES( 00000002 )
          # OF STARTS ( 00000001 )
          # OF ENDS   ( 00000000 )
=====

```

The Dataset Statistics screen displays information about the dataset chosen from the detail Active Jobs screen. Along with the number of VSAM operations performed on the dataset, dataset characteristics are also displayed. If this or any other active job screen is entered and the displayed job step finishes, the information pertaining to that step disappears. The information can be retrieved if the job was defined to collect history. Refer to the section “Detailed History Information” on page 59 or the section “Defining Which Jobs Can Access the Files” on page 51 for details. The information displayed is explained below.

Field	Description
DDNAME	The DDNAME of the dataset as it appears in the batch JCL.
FCTNAME	The CICS FCT name of the dataset. The DDNAME and the FCT name do not have to match. VSHARE matches the two by the VSAM cluster name.
CLUSTER NAME	The cluster name of the dataset. This name matches the name in the JCL as well as the name in the CICS FCT table. If a match is not found, VSHARE considers the file a batch-only file and gives control to VSAM.
ACCESS	The access allowed by CICS against this file. There are five possible letters displayed. If any of the letters (below) are not displayed, the corresponding operation is not allowed against the dataset. R—Read access is allowed. U—Update access is allowed. B—Browsing is allowed. A—Adding new records is allowed. D—Deleting records is allowed.
LSR POOL	The number of the LSR pool where the dataset is defined. Refer to the chapter “VSAM Performance” on page 70 for details on LSR in CICS. If NSR is used, the value is zero.
STRINGS	The number of strings defined to this dataset. A string is equivalent to one I/O operation at a given time. For example, if a dataset is defined to have five strings, five different operations are allowed on the file at one time.

Field	Description
# OF READS	The number of reads performed against the dataset in the current step.
# OF ADDS	The number of adds performed against the dataset in the current step.
# OF UPDATES	The number of updates performed against the dataset in the current step.
# OF DELETES	The number of records deleted in the current step.
# OF STARTS	The number of starts requested by the current step.
# OF ENDS	The number of end requests.

Attention Keys

- Enter Refresh the VSAM I/O statistics for the file.
- Clear Return to the detail Active Jobs screen. If more than one DDNAME was selected on the previous detail Active Jobs screen, display the chosen DDNAME's statistics.

Defining Which Jobs Can Access the Files

```

=====
APPLID: C21TS520  FSI-VSHARE - THE CICS CONNECTION      DATE: 07/31/2017
PROGID: CSIVSC15  J O B   M A I N T E N A N C E      TIME: 13:56:39
=====
FUNCTION: C ( A = ADD, C = CHANGE, D = DELETE )

JOB NAME: OMNI01
=====
JOB INFORMATION          | HISTORY INFORMATION   | PROCESSING INFORMATION
CICS REGION: C21TS520   | RECORD?: N           | ABEND IF NO CICS?: N
TRANID: TCCB           | GENERATIONS: 003     | DO INPUT FILES?: N
SYNC LEVEL: 005       |                       | ABEND IF DISABLED?: N
REQS/MINUTE: 00000    | JOURNAL INFORMATION  | TURN TRACE ON?: N
ACCESS TYPE: CMS       | JOURNAL UPDATES?: N  | FORCE ROLLBACK?: N
                       |                       | SRCH OTHER REGIONS?: N
=====
DYNAMIC JOURNAL NAME:
//ITIJRNL DD DSN= RCRAVEN          . OMNI01 .ITIJRNL,DISP=MOD,
              DCB=(RECFM=VB,BLKSIZE= 27998 ,LRECL= 27994 ,BUFNO= 032 ),
              SPACE=(CYL,( 0004 , 0002 )),UNIT= SYSDA
=====
VSHARM15-008 PRESS PF3 TO PROCESS THE MODIFICATION.
=====

```

The Job Maintenance screen (option 2 on the Main Menu) allows you to define the jobs or job groups that can access the CICS system owning the shared VSAM files. The screen allows you to control which jobs can access the VSAM files. This definition process allows you to tailor job-processing options like the recovery type or the communications type to use.

Each field in the Job Information and the Journal Information groups has CICS region defaults. The default values are defined in the System Definition screen and displayed when the screen is first displayed. If a default field is overridden, the field is highlighted. Erasing the override value causes the system default to be displayed.

When the default values are changed on the system definition screen, all jobs defined to that CICS region obtain the changed default value. In this way, you do not have to change individual jobs when a global change is needed.

If the job name is forgotten, enter a command along with the starting character of the job name. VSHARE displays all the jobs starting with the entered character. By selecting a job from the list, the Job Maintenance screen is re-displayed. The fields on the Job Maintenance screen are explained below.

Field/Group	Description
FUNCTION:	<p>Determine which function to perform. The possible values are</p> <p>A—Add a new job to VSHARE.</p> <p>C—Change the values of a previously defined job.</p> <p>D—Delete a job from the VSHARE control file.</p>
JOB NAME:	<p>The partial or full name of the job(s) eligible to use VSHARE. The job name defined to VSHARE does not have to be the full name. Any number of characters starting with the first character can be defined. For example, if an installation wants all jobs starting with “BT” to use VSHARE, the job name is defined as BT. In this way, only one entry needs to be defined rather than every job name starting with BT.</p>
JOB INFORMATION Group	
CICS REGION	<p>The CICS APPLID where VSHARE is defined and where all the files used by the job are defined.</p>
TRANID	<p>This field tells VSHARE which transaction ID to execute when a connection is made to CICS. This is an override to the system default value. Changing the TRANID allows different applications within the same CICS region to have their own VSHARE transaction IDs for charge-back purposes. If each application has its own TRANID, a CICS charge back for batch I/Os can be charged using current online charge-back systems. Make sure that each TRANID executes VSHARE program CSIVSC00. Refer to the section “PCT Definitions” on page 102 for details.</p>
SYNC LEVEL	<p>This field controls the implicit (automatic) syncpoint feature. It is the number of CICS file enqueues allowed before VSHARE attempts a syncpoint. This is only a threshold level. Before VSHARE can issue a syncpoint, none of the files owned by the batch program can have a read-for-update outstanding. It is important not to hold a resource for a long time. A value of zero disables the automatic syncpoint feature. Enter zero if the application program uses the implicit syncpoint feature (Tier 3 recovery) or if no syncpointing is used (Tier 1 recovery).</p>

Field/Group	Description
REQS/MINUTE	The maximum number of I/O requests that the batch job can send to the CICS region per minute. If zero is entered, the batch job sends the requests as quickly as possible. VSHARE gives CICS programs priority. Therefore, if a CICS program needs to be serviced, CICS services the online program before the batch job. For this reason, this field needs a non-zero value only when the CICS region is very active.
ACCESS TYPE	This is the communications type used to send the VSAM requests between the batch job and VSHARE transaction. The possible values are LU62 (for LU 6.2 communications) and CMS (for Cross Memory Services). You can use CMS only when the batch job and the CICS region are executing under the same z/OS system. You can use LU 6.2 when the batch job and CICS are on either the same z/OS system or different z/OS systems. If you choose CMS and VSHARE determines that the batch job and the CICS region are on different z/OS systems, VSHARE automatically uses LU 6.2. CMS is the faster access type, so if a batch job is executing on the same CPU as the CICS region, choose CMS.
HISTORY INFORMATION Group	
RECORD?	The possible values are Y or N. This field tells VSHARE whether to keep history information about the job's execution.
GENERATIONS	The number of generations of history to keep. There is one history record in the VSHARE Control File per step. For example, if five generations are entered and there are six steps in the job, there will be 30 records in the control file.
JOURNAL INFORMATION Group	
JOURNAL UPDATES?	A Y tells VSHARE to journal all changes made during the job's execution. By specifying Y, all changes can be backed out if an abend occurs. An N tells VSHARE not to journal file changes. Refer to the chapter "Recovery" on page 18 for details on VSHARE journaling.

Field/Group	Description
PROCESSING INFORMATION Group	
ABEND IF NO CICS?	If a Y is entered and the job is executed while CICS is not running, VSHARE abends with a user abend 2000. If this option is defined with an N and CICS is not running, VSHARE passes control directly to VSAM.
DO INPUT FILES?	If a Y is entered, VSHARE reads all files open for input through the CICS region. If an N is entered, the batch job reads the VSAM file directly.
ABEND IF DISABLED?	If a Y is entered, VSHARE abends the batch job if any of the files opened by the batch job are disabled in the CICS region. If an N is entered, VSHARE will not abend the job. Instead, the job will allocate the file directly.
TURN TRACE ON?	Always set this field to N unless CSI Technical Support requests a trace for problem resolution. This option tells VSHARE in the batch region to trace certain events during a program's execution.
FORCE ROLLBACK?	If a Y is entered, a rollback will be forced if the batch job abends during processing.
SRCH OTHER REGIONS?	<p>The SRCH OTHER REGIONS option is designed to enable or disable the searching of VSAM files in remote regions. VSHARE permits operations on VSAM files across remote CICS regions that have file definitions in the local FCT. If the target VSAM file cannot be found in the local region in which VSHARE is executing, a Y response to this option will cause the FCT in remote regions to be searched for the dataset name. The transaction TCCR must be defined in the remote regions being searched.</p> <p>If the region in which VSHARE is operating is known to own the VSAM file, an N response to this question may save considerable time in searching other regions. This is the default option.</p>

Field/Group	Description
DYNAMIC JOURNAL NAME Group	
DSN=	The high level qualifier of the journal dataset name created by VSHARE. Refer to the chapter “Recovery” on page 18 for more information on the journal dataset.
BLKSIZE=	The block size of the journal dataset.
LRECL=	The record size of the journal dataset.
BUFNO=	The number of buffers to be allocated for the journal dataset.
SPACE=	The primary and secondary size of the journal dataset. The size is the number of cylinders.
UNIT=	The unit value for the dataset.

Attention Keys

- Enter Perform editing of the data entered.
- Clear Return to the Main Menu.
- PF3 Process the desired function.

Displaying a List of Jobs Already Defined

```

=====
APPLID:  UAPPLID      FSI-VSHARE - THE CICS CONNECTION      DATE:  02/01/2015
PROGID:  CSIVSC11    J O B   M A I N T E N A N C E      TIME:  06:05:13
=====

JOB NAME:  A

      CMD              JOB NAME              DATE              USER ID
-----
      -                MEAVHSRA             02/13/2014       CICSUSER
      -                MEAVSHRA             02/18/2014       CICSUSER
      -                MEAVSHR              03/30/2014       CICSUSER
      -                SJJVSEXT             03/26/2014       CICSUSER
      -                SJJVSHRT             03/31/2014       CICSUSER
=====

| PLACE S FOR SELECTION AND PRESS ENTER.   PF7=BACK           PF9=HOME |
=====
    
```

If an incorrect job name is entered in the detail Job Summary screen, VSHARE displays the screen above with a list of jobs that start with the characters entered. You can scroll this list to find the job you want. By typing an S next to a job name, the detail job screen is displayed with the chosen job-name's definition. The job search name can be changed to display other defined jobs.

To display a list of the jobs defined, go to the Job Maintenance screen, select option C, place any letter in the JOB NAME field, and press the Enter key.

Field	Description
JOB NAME:	The undefined job name entered in the Job Maintenance screen. Modifying this name displays jobs starting with the name entered.
CMD	By placing an S in this field, the detailed definition for the job is displayed.

Attention Keys

- Enter Process the newly entered job name, or return to the Job Maintenance screen with the chosen job definition.
- Clear Return to the detailed Job Maintenance screen.
- PF7 Scroll backward in the list.
- PF8 Scroll forward in the list.
- PF9 Display the list starting with the name entered in the JOB NAME field.

Reviewing the History of Past Job Executions

```

=====
APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE:  02/01/2015
PROGID:  CSIVSC16  JOB HISTORY SUMMARY   TIME:  06:06:34
=====

JOB NAME:  A

  CMD   JOB NAME   STEP NAME   JES NUMBER   DATE       TIME
-----
  -     MEAVSHR    STEP02     00969        04/09/2014 11:12:41
  -     MEAVSHR    STEP02     01001        04/09/2014 12:40:02
  -     MEAVSHR    STEP02     01014        04/09/2014 13:17:32
  -     MEAVSHRA   STEP02     01101        02/18/2014 12:56:36
  -     MEAVSHRA   STEP02     01103        02/18/2014 12:58:05
  -     MEAVSHRA   STEP02     01106        02/18/2014 13:04:16
  -     SJJVSHRT   JRNLST     00684        03/31/2014 08:07:53
  -     SJJVSHRT   JRNLST     00686        03/31/2014 08:11:43
  -     SJJVSHRT   JRNLST     00688        03/31/2014 08:35:50
  -     SJJVSHRT   JRNLST     00691        03/31/2014 08:43:11
=====
S + ENTER = DETAIL  S + PF5 = DELETE.   PF7=BACK  PF8=FORWARD  PF9=HOME
=====
  
```

If a job is defined to save history, you can use the Job History Summary screen to browse all the generations saved by VSHARE. To obtain the desired history information, choose option 3 on VSHARE Main Menu. The Job History Summary screen displays a list of job history information. The desired job run can be found by either scrolling through the history list or by changing the JOB NAME field at the top of the screen. To purge the history records, set the generation number to zero. The next execution of the job purges all the history records for the job.

Refer to the section “Defining Which Jobs Can Access the Files” on page 51 for details on how to save history information. The fields on the Job History Summary screen are explained below.

Online Control

Field	Description
JOB NAME:	The name of the job used to form the list. This value is either spaces or the job name processed by a previous screen. It can be changed at any time.
CMD	Each detail line has a command field. By placing any character in this field, the detail history screen is displayed. More than one selection can be entered at a time. After exiting from the first detail history screen, the second selection's detail information is displayed. This continues until all the selections are displayed.
JOB NAME	The name of the job as it appeared on the JCL JOB card.
STEP NAME	The name of the step as it appeared on the JCL EXEC card.
JES NUMBER	The job number assigned to the job by the operating system.
DATE	The date the job step was executed.
TIME	The start time of the step.

Attention Keys

- Enter Refresh the screen, display a new list when the JOB NAME field is changed, or process the selected step names.
- Clear Return to the Main Menu.
- PF7 Scroll backward through the list.
- PF8 Scroll forward through the list.
- PF9 Display the list starting with the job contained in the JOB NAME field.

Detailed History Information

```

=====
| APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION      DATE:  02/01/2015 |
| PROGID:  CSIVSC17  J O B   H I S T O R Y   D E T A I L    TIME:  06:07:51  |
|-----|
| JOB NAME:  SJJVSHRT      JES NUMBER:  00684      DATE:  03/31/2014 |
| STEP NAME:  JRNLST      TIME:  08:07:53          |
|-----|
| JOURNAL INFORMATION:  UPDATES JOURNALED?          |
|-----|
| //ITIJRNL  DD  DSN=                . SJJVSHRT .ITIJRNL,DISP=MOD, |
|              DCB=(RECFM=VB,BLKSIZE= 02566 ,LRECL= 20480 ,BUFNO= 656 ), |
|              SPACE=(CYL,( 7678 , 1410 )), UNIT= { |
|-----|
| SYSTEM INFORMATION:  CICS REGION: P390CICS          REQ/MIN:  00000 |
|                    FILES OPENED:  001             ACCESS TYPE:  CMS |
|                    USER SYNCPOINTS:  00000000     AUTO SYNCPOINTS:  0000000 |
|                    TOTAL VSAM REQUESTS:  00000010 |
|-----|
| DDNAMES:  _ AKSDSIN |
|-----|
| SELECT A DDNAME AND PRESS ENTER.          PF7=BACK  PF8=FORWARD  PF9=HOME |
|-----|

```

Based on the selection you make in the Job History Summary screen, the Job History Detail screen displays more details, including a list of DDNAMEs opened during execution. By selecting one of the DDNAMEs, a detail screen is displayed that gives dataset-specific information. The information displayed on this screen is explained below.

Field	Description
JOB NAME:	The name of the job as it appeared on the JCL JOB card.
JES NUMBER:	The job number assigned to this job.
DATE:	The date of the job execution.
STEP NAME:	The name of the step as it appeared on the JCL EXEC card.
TIME:	The start time of the step.

Field	Description
UPDATES JOURNALED?	The journal processing option. A Y indicates that journaling was performed. An N indicates that journaling was not performed. This value and other journaling fields listed below are determined by the system defaults and the job definition overrides. The ITIJRNL DD fields are blank if journaling was not active. Refer to the section “Defining Which Jobs Can Access the Files” on page 51 and the section “Defining CICS Regions to VSHARE” on page 64 for details.
DSN=	The high level qualifier of the journal dataset name created by VSHARE. Refer to the chapter “Recovery” on page 18 for more information on the dataset.
BLKSIZE=	The block size of the journal dataset.
LRECL=	The record size of the journal dataset.
BUFNO=	The number of buffers allocated to the journal dataset.
SPACE=	The primary and secondary cylinder size of the journal dataset.
UNIT=	The unit of the dataset.
CICS REGION:	The VTAM APPLID of the CICS where VSHARE was executed.
REQS/MIN:	The initial number of I/O requests per minute defined to the job. This is the maximum number of VSAM I/O requests VSHARE is allowed to process per minute.
FILES OPENED:	The number of files opened in the step. Note that this is the number of files under the control of VSHARE. Batch-only files are not listed.
ACCESS TYPE:	The type of communications used during this job’s execution. The values can be LU62 if LU 6.2 was used, or CMS if the Cross Memory Services interface was used.

Field	Description
USER SYNCPOINTS:	The number of syncpoints issued by the user program by calling VSHARE directly for that purpose. A non-zero value indicates Tier 3 recovery is being used.
AUTO SYNCPOINTS:	The number of syncpoints issued by VSHARE automatically on behalf of the user's program. A value of 0 indicates Tier 1 recovery is being used. A value greater than 0 indicates Tier 2 recovery is being used, per the SYNC LEVEL field on the Job Maintenance Screen.
TOTAL VSAM REQUESTS:	The total number of VSAM I/Os that VSHARE intercepted and issued on behalf of the batch program.
DDNAMES:	A list of DDNAMES opened during the step execution and under the control of VSHARE. For more detailed information on a dataset, type an S next to the DDNAME. A detailed DDNAME screen is displayed.

Attention Keys

- Enter Process the selected DDNAMEs or refresh the screen.
- Clear Return to the Job History Summary screen, or display the next step selected on the previous Job History Summary screen
- PF7 Scroll backward through the DDNAME list.
- PF8 Scroll forward through the DDNAME list.
- PF9 Display the beginning of the DDNAME list.

Detailed Dataset History Information

```

=====
APPLID:  UAPPLID      FSI-VSHARE - THE CICS CONNECTION      DATE:  02/01/2015
PROGID:  CSIVSC18    D A T A S E T  S T A T I S T I C S      TIME:  09:47:42
=====

          DDNAME      ( MASTER      )
          FCTNAME     ( FILEMSTR   )
          CLUSTER NAME( PROD.MASTER.CLUSTER )
          DATASET TYPE( KSDS      )
          ACCESS      ( RUBAD     )
          LSR POOL    ( 01       )
          STRINGS     ( 004      )
          # OF READS  ( 00023120 )
          # OF ADDS   ( 00000213 )
          # OF UPDATES( 00002305 )
          # OF DELETES( 00000002 )
          # OF STARTS( 00000001 )
          # OF ENDS   ( 00000000 )
=====
    
```

The Dataset Statistics screen displays information about the dataset you chose. Along with the number of VSAM operations performed on the dataset, dataset characteristics are also displayed.

Field	Description
DDNAME	The DDNAME of the dataset as it appeared in the batch JCL.
FCTNAME	The CICS FCT name of the dataset at execution time.
CLUSTER NAME	The cluster name of the dataset.
ACCESS	The access allowed by CICS against the file. There are five possible letters displayed. If any of the letters (below) are not displayed, the corresponding operation is not allowed against the dataset. R—Read access is allowed. U—Update access is allowed. B—Browsing is allowed. A—Adding new records is allowed. D—Deleting records is allowed.

Field	Description
LSR POOL	The number of the LSR pool where the dataset was defined. Refer to the chapter “VSAM Performance” on page 70 for details on LSR in CICS. If NSR is used, the number is zero.
STRINGS	The number of strings defined to this dataset. A string allows one I/O operation at a given time. For example, if a dataset is defined to have five strings, five different operations are allowed on the file at one point in time.
# OF READS	The number of reads performed against the dataset in the step.
# OF ADDS	The number of adds performed against the dataset in the step.
# OF UPDATES	The number of updates performed against the dataset in the step.
# OF DELETES	The number of records deleted in the step.
# OF STARTS	The number of starts requested by the step.
# OF ENDS	The number of end requests processed.

Attention Key

Clear Return to the Job History Detail screen. If more than one DDNAME was selected on the previous detail history screen, display the next Dataset Statistics screen.

Defining CICS Regions to VSHARE

```

=====
APPLID:  UAPPLID      FSI-VSHARE - THE CICS CONNECTION      DATE:  02/01/2015
PROGID:  CSIVSC19    S Y S T E M   M A I N T E N A N C E    TIME:  06:09:23
=====
FUNCTION:      ( A = ADD, C = CHANGE, D = DELETE )
CICS APPLID:
=====
                SYSTEM INFORMATION                      HISTORY INFORMATION
TRANID:        ACCESS TYPE:                            RECORD HISTORY:
MODE NAME:     SYNC LEVEL:  000                        GENERATIONS:  000
ROLLBACK:  N   REQ/MIN:    00000                       JOURNAL INFORMATION
APPLID SEL:  N   SELECT MSK:  ****                       JOURNAL UPDATES:
=====
DYNAMIC JOURNAL NAME:
//ITIJRNL DD DSN= .JOBNAME.ITIJRNL,DISP=MOD,
              DCB=(RECFM=VB,BLKSIZE= 00000 ,LRECL= 00000 ,BUFNO= 000 ),
              SPACE=(CYL,( 0000 , 0000 )),UNIT=
=====
BATCH APPLIDS:  FSIAPPC9  P3CICS31  P3CICS33  TR20A   TR80A
                 TR80B   P390CICT  CSICICSP  CSICICST  CSILU62A
                 TRE0A   TR30A    TR90A    P390VSH1  P390VSH2
=====
ENTER=EDIT  CLEAR=EXIT  PF3=PROCESS  PF7=BACK  PF8=FORWARD  PF9=HOME
=====

```

The System Maintenance screen defines which CICS regions can use VSHARE. When VSHARE finds a job definition record in the control file, it retrieves the CICS region name from the job record and reads the CICS system record. The system record is defined by the System Maintenance screen. This screen contains all the information required by VSHARE to communicate to the CICS region. Along with this control information, all of the default journaling and history fields are defined for the CICS region. These defaults are used during a job's definition when no override values are entered.

If you forget the CICS APPLID, enter any command along with the starting character of the CICS APPLID. VSHARE displays all the CICS systems starting with the characters you entered. By selecting a system from the list, the System Maintenance screen is re-displayed with the system's definition.

Field/Group	Description
FUNCTION:	<p>This function to execute. The possible values are</p> <p>A—Add a new system definition to VSHARE.</p> <p>C—Change the values of a previously defined system.</p> <p>D—Delete a system definition from the VSHARE file.</p>
CICS APPLID:	<p>The full CICS APPLID eligible to use VSHARE.</p>
SYSTEM INFORMATION Group	
TRANID:	<p>The name of the transaction defined to execute VSHARE program CSIVSC00. Refer to the section “PCT Definitions” on page 102 for details on defining this transaction. Each application in a CICS region can have its own VSHARE TRANID by defining multiple transactions that execute CSIVSC00 and defining the unique TRANID in the job definition screen. This allows charge back to the appropriate application for work performed by the batch job in CICS.</p>
MODE NAME:	<p>The mode-entry name as defined in CICS and VTAM. Refer to the section “Environment Definitions” on page 107 for details on the mode entry definition. If CICS auto-install is not used in CICS, the mode entry SMSNA100 can be used as a mode name with VSHARE. SMSNA100 is a definition in the default mode-entry table.</p>
ROLLBACK:	<p>If Y is entered, a ROLLBACK will be forced if batch jobs abend during processing.</p>
APPLID SEL:	<p>If Y is entered in this field, and an APPLID selection mask is entered in the SELECT MSK: field, only those APPLIDS matching the mask will be displayed in the BATCH APPLIDS fields.</p>
ACCESS TYPE:	<p>The system default communications vehicle. This field tells VSHARE how to communicate between the batch job and the CICS region. The possible values are LU62 or CMS (Cross Memory Services). Even if you choose CMS, LU 6.2 is used to start VSHARE transaction; therefore, the batch APPLIDS are still required if CMS is solely used. CMS can only be used when the batch job and the CICS region are executing on the same z/OS system. LU62 should be used if the batch job and the CICS region are on different CPUs. CMS provides a faster communications access method between the batch job and CICS.</p>

Field/Group	Description
SYNC LEVEL:	This field controls the implicit (automatic) syncpoint feature. It is the number of CICS file enqueues allowed before VSHARE attempts a syncpoint. Note that this is only a threshold level. Before VSHARE can issue a syncpoint, none of the files owned by the batch program can have a read for update outstanding. It is important not to hold on to a resource for a long period of time. A value of zero disables the automatic syncpoint feature. Zero should be used if Tier 1 recovery (no syncpointing) or Tier 3 recovery (user explicit syncpointing) is used.
REQ/MIN:	The maximum number of requests that the batch job can send to the CICS region per minute. If 0 is entered, the batch job sends the requests as quickly as possible. VSHARE gives the CICS programs priority. Therefore, if a CICS program needs to be serviced, CICS services the online program before the batch job. For this reason, this field needs a non-zero value only when the CICS region is very active.
SELECT MSK:	Specifies the APPLID selection mask for the BATCH APPLID fields. This field can start and/or end in an '*' to indicate that a generic specification is desired.
HISTORY INFORMATION Group	
RECORD HISTORY:	The possible values are Y or N. Entering a Y tells VSHARE to keep history information about the job's execution.
GENERATIONS:	Enter the default number of generations of history. There is one history record added to the VSHARE control file per step execution. For example, if five generations are entered and there are six steps in the job, there will be a maximum of 30 records in the control file per job.
JOURNAL INFORMATION	
JOURNAL UPDATES:	The possible values are Y or N. Entering a Y tells VSHARE to journal all changes made during the job's execution. By specifying Y, all changes can be backed out in the event of a program abend. Refer to the chapter "Recovery" on page 18 for details on VSHARE journaling.
DYNAMIC JOURNAL NAME: Group	
DSN=	The default high level qualifier of the journal dataset name created by VSHARE. Refer to the chapter "Recovery" on page 18 for more information on the journal dataset.
BLKSIZE=	The default block size of the journal dataset.

Field/Group	Description
LRECL=	The default record size of the journal dataset.
BUFNO=	The default number of buffers to be allocated to the journal dataset.
SPACE=	The default primary and secondary size of the journal dataset in cylinders.
UNIT=	The default unit value for the dataset.
BATCH APPLIDS:	<p>The VTAM APPLIDS defined during installation. These are the APPLIDS that VSHARE opens to communicate with the CICS region through a LU 6.2 link. Refer to the section “Environment Definitions” on page 107 for details on the batch APPLID definitions.</p> <p>If more than a page of APPLIDS are desired, the list can be scrolled forward and backward. VSHARE allows a maximum of 90 APPLIDS per CICS system. Batch APPLIDS are required even if CMS is chosen as the access type. LU 6.2 is always used to start a VSHARE transaction in CICS; it is not until a connection is made to CICS through LU 6.2 that the CMS access is used.</p> <p>You can limit the number of APPLIDS displayed here during the initial add of a new CICS region by</p> <ul style="list-style-type: none"> • Specifying a Y in the APPLID SEL: field, and • Entering an APPLID selection mask in the SELECT MSK: field.

Attention Keys

- | | |
|-------|---|
| Enter | Perform edits on the fields entered. |
| Clear | Return to the main menu. |
| PF3 | Process the desired function. |
| PF7 | Scroll backward through the APPLID list. |
| PF8 | Scroll forward through the APPLID list. |
| PF9 | Display the beginning of the APPLID list. |

Listing Defined CICS Regions

```

=====
| APPLID: UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE: 02/01/2015 |
| PROGID: CSIVSC11 SYSTEM MAINTENANCE                 TIME: 06:10:34   |
|=====
|
| APPLID: A
|
| CMD          CICS APPLID          DATE          USER ID
|-----
|              CSIDOIT2            02/03/2014    CICSUSER
|              CSIDOIT3            02/03/2014    CICSUSER
|              CSIDOIT4            02/03/2014    CICSUSER
|              P390CICS            03/31/2014    CICSUSER
|-----
|
|=====
| PLACE S FOR SELECTION AND PRESS ENTER.   PF7=BACK   PF9=HOME |
|=====

```

If an incorrect system name is entered in the detail System Maintenance screen, VSHARE displays the screen above. The screen lists the names of defined CICS regions starting with the characters you entered. You can scroll the list to find the appropriate CICS name, or you can change the system search name to re-position the list. By placing an S next to a system name, the detail system screen is displayed with the chosen definition. You can also use the System Maintenance screen to define the CICS regions that are associated with a particular VSHARE control file. By choosing the Change function and placing a letter in the CICS APPLID field, a list of all the defined systems is displayed in this summary screen.

Field	Description
APPLID:	The initial name entered is displayed. The name can be changed to re-position the list.
CMD	By placing an S next to a CICS APPLID, the detail definition for the system is displayed.

Attention Keys

- | | |
|-------|--|
| Enter | Process the new enter APPLID or return to System Maintenance. |
| Clear | Return to the detail System Maintenance screen. |
| PF7 | Scroll backward through the list. |
| PF8 | Scroll forward through the list. |
| PF9 | Display the list starting with the name entered in the APPLID field. |

VSAM Performance

Overview

Using VSHARE focuses importance on VSAM I/O performance because more programs can access the same file simultaneously. This chapter addresses VSAM performance from both an application and a system point of view so that an installation can tune their VSAM databases as efficiently as possible. This chapter initially discusses basic I/O and VSAM concepts. Lastly, it describes specific tuning recommendations to improve performance. These recommendations are focused on KSDS files because KSDS is the most commonly utilized VSAM file structure.

Concepts

The number of I/O operations is usually the single most important item in program performance. The major percentage of program elapsed time is due to I/O wait time. Eliminating or reducing this time through I/O tuning, or by removing unnecessary I/O, has a major effect on program performance.

To communicate the importance of I/O tuning and I/O elimination, consider this example. An IBM 3090-300S processor is capable of executing approximately 54 million instructions per second. On the other hand, an average disk I/O takes about 25 milliseconds to complete. This means that if a program does not have to wait on one I/O, it can complete 1.35 million additional instructions during the time saved. The concept of reducing program I/O to a minimum is critical for achieving good program performance and maximizing the CPU resource.

Before describing specific performance recommendations, this chapter covers basic I/O concepts so that those unfamiliar with the concepts can better appreciate the impact of I/O performance tuning. Following is a list of the discussed topics. Even those familiar with these concepts should review this section as a refresher.

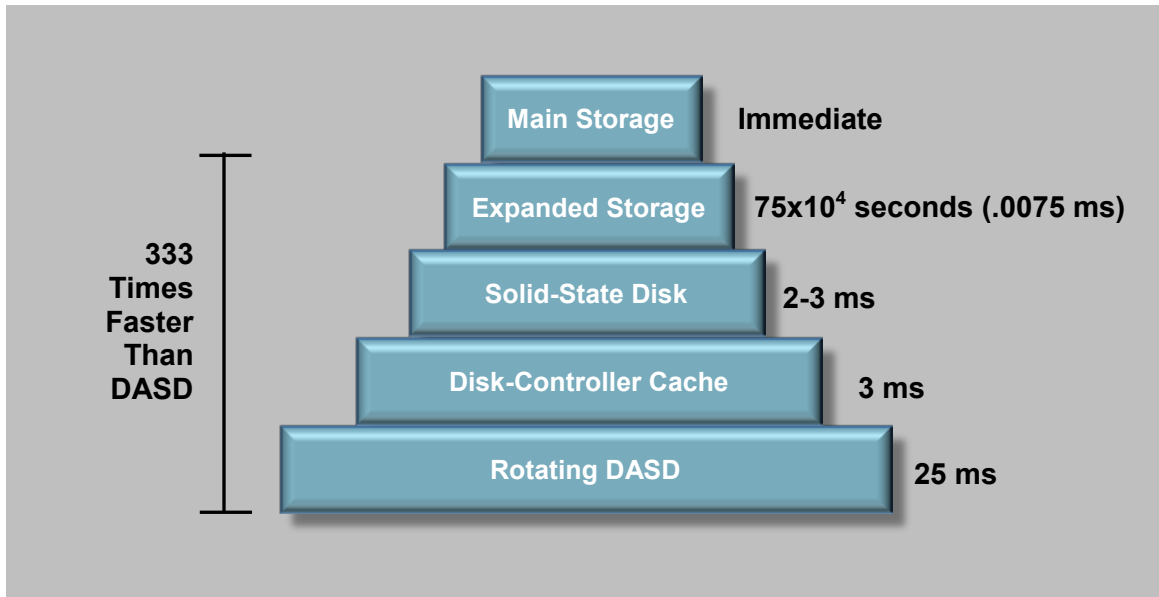
1. Storage Mediums
2. Concept of Asynchronous Versus Synchronous I/O
3. The Mechanics of a Physical I/O
4. The Mechanics of a VSAM KSDS I/O Request
5. Non-Shared Resources versus Local Shared Resources

Storage Mediums

In the past, data could be located either in main storage or on an external disk. Data can now be found on many intermediate locations between main storage and DASD:

- *Buffers in Main Storage.* Data recently requested by a program remains in storage buffers until it is modified or it is flushed out because the operating system requires the buffer space. These buffers are searched by the operating system before the requested data is read from disk. This process is referred to as lookaside.
- *Expanded Storage.* Many processors have expanded storage, which is an extension of main storage. For a program to access the data in expanded storage, the data must first be moved into main storage by the operating system. Expanded storage provides quick data access.
- *Cache.* Many disk controllers contain cache storage, which is similar to main storage except that it is located within the disk controller. This storage is managed by the controller's microcode and searched before the controller attempts to read a record from the disk.
- *Solid State Disks (SSD).* An SSD appears to the CPU as a conventional disk device but it contains no mechanical components of a rotating disk. As a result, an SSD delivers a constant low response time for reading and writing data. Disk read/write operations to an SSD are as fast as cache. The SSD is a good candidate for small, critical datasets.
- *Conventional Rotating Disk.* Reading and writing data from rotating disks is the slowest operation because it relies on mechanical movement.

Data retrieval times are dramatically different depending on the storage medium used. This is shown in the figure below.



Synchronous Versus Asynchronous Access

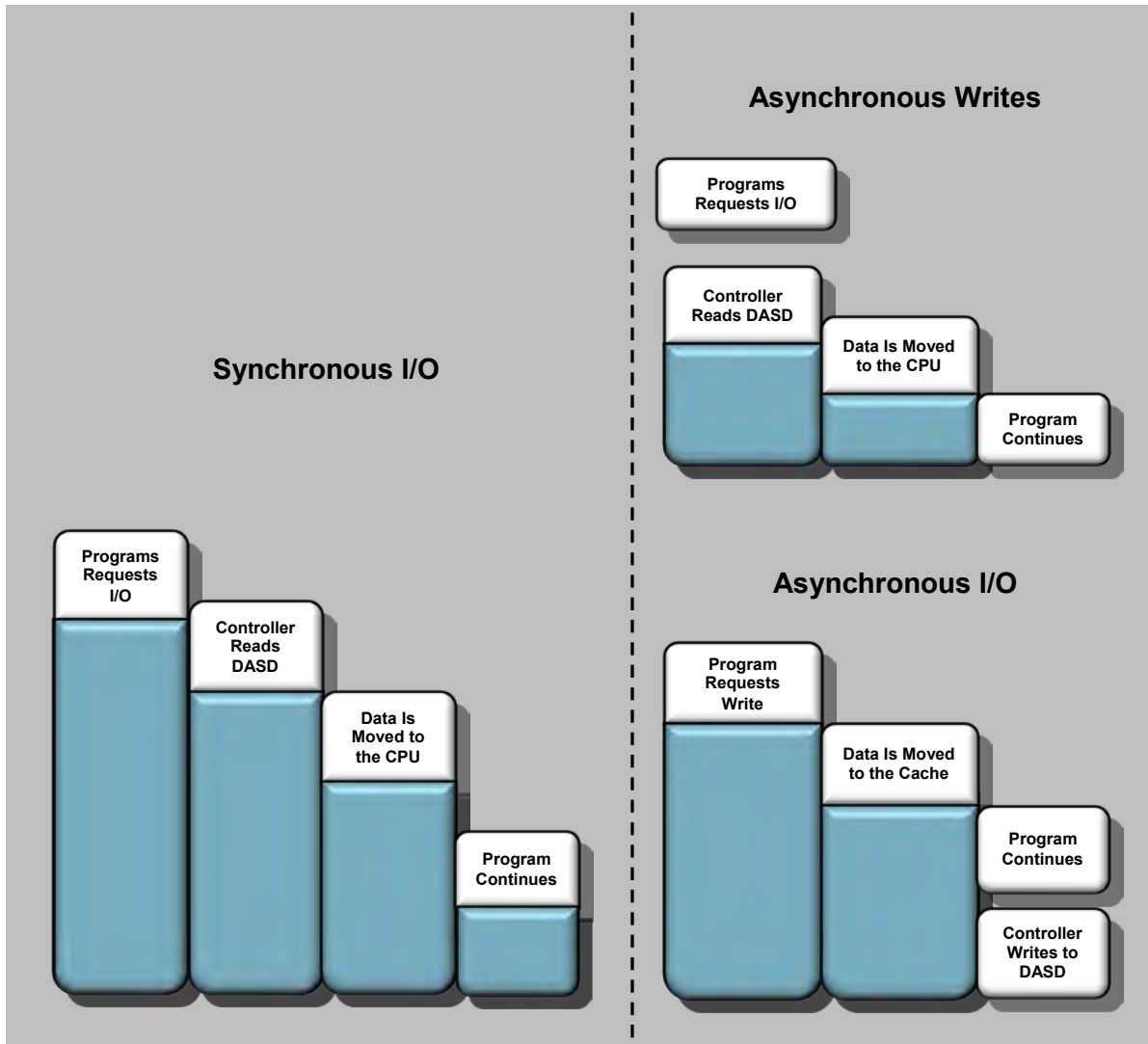
Data can be retrieved from storage mediums either synchronously or asynchronously. Here is a description of how data is read with each method:

Synchronous I/O: The data to be read is physically retrieved from the disk only after a program requests it.

Asynchronous I/O: The operating system or disk controller anticipates the next data request and reads the data from disk before it receives the request from the program. This occurs during sequential processing where read-ahead is performed. Therefore, when the program requests the data, it is moved from the operating system's main storage buffer to the user program, which results in considerable time savings.

In the past, the writing of data to disk was always synchronous because of the importance of data integrity. The disk controllers did not have battery backup, and the controller had to wait for the physical I/O to finish before signaling to the CPU that the I/O was completed. This was important to ensure data integrity. The situation changed with the advent of the IBM 3990 disk controller. The 3990 disk controller has non-volatile (battery backup) cache. Data written to disk is stored in cache temporarily, and the CPU is informed of normal I/O completion. This operation takes 3 ms instead of the 25 ms for a normal (physical) disk I/O. As the program continues, the controller completes the physical I/O.

The figure below compares the asynchronous and synchronous processes.



Basic I/O Concepts

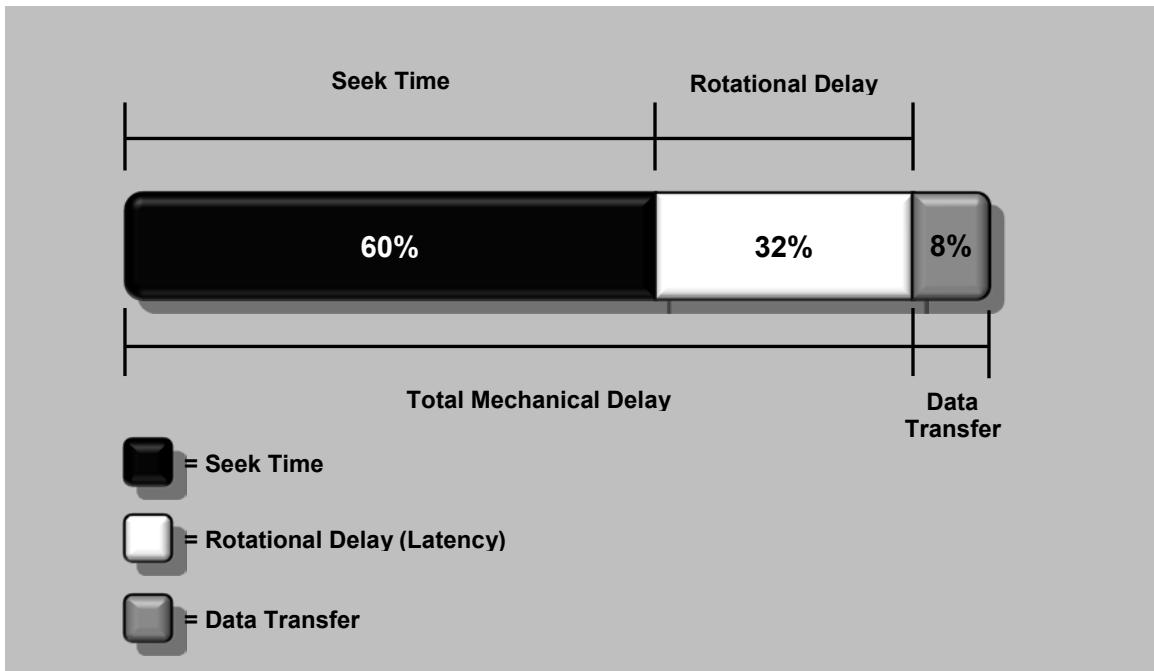
Normal disk I/O can be divided into three operations:

1. The first operation is to move the disk head over the surface of the disk. The head is placed over the appropriate cylinder where the data resides. The average time to complete this seek operation is about 15 ms.
2. Once the seek operation is complete, the channel disconnects the link to the drive and services other requests until the requested record passes under the head. This is referred to as Rotational Positional Sensing (RPS). When the requested record passes under the head, the channel reconnects the disk, and the record is transferred to the CPU.

If the channel is busy servicing another request and is unable to reconnect at the time the record passes under the head, the channel has to wait an additional 16 ms of disk rotation before attempting to reconnect and transfer the record. The missing of the record is referred to as an RPS miss. An RPS miss has a significant impact on disk performance.

3. Once the channel connects to the disk, the data is transferred up the channel to the host. The amount of time this takes to complete depends on the channel's data transfer rate (currently 3 to 4.5 MB per second).

As shown in the figure below, data transfer represents only a small percentage of total I/O time. This is why it is important to reduce the number of physical reads to DASD while accessing a VSAM file.



Mechanics of a VSAM KSDS I/O

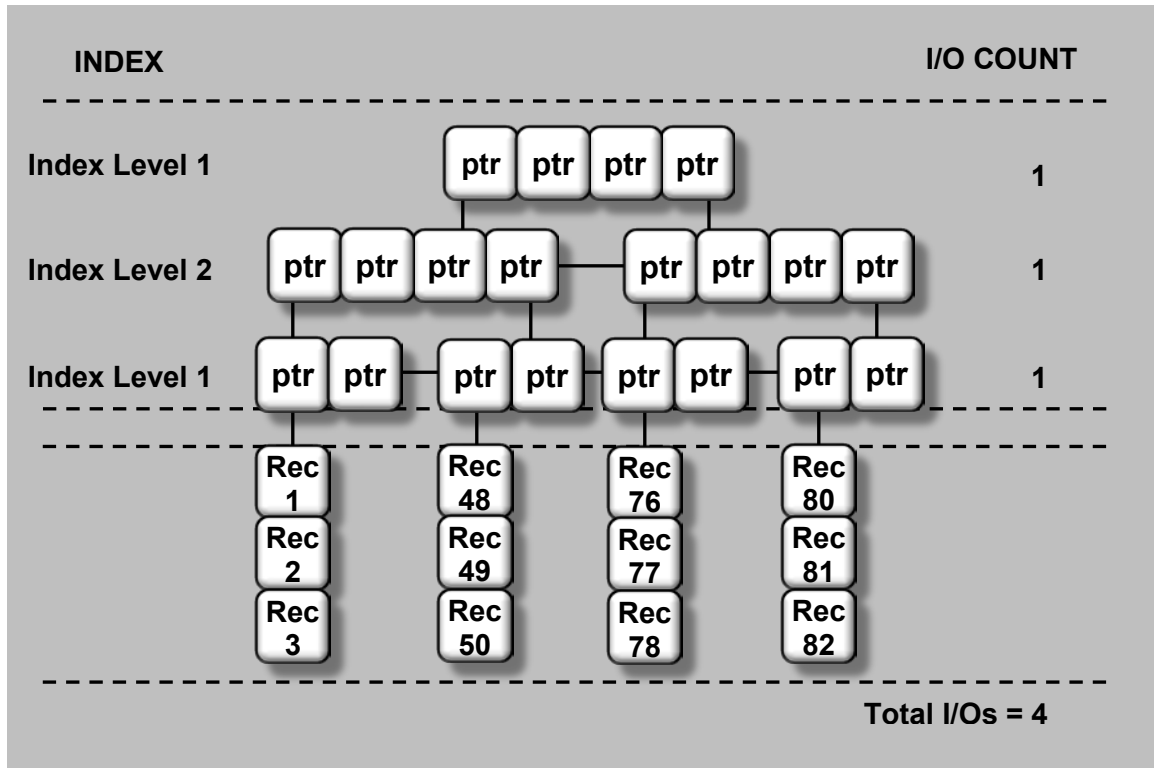
It is important to understand the steps VSAM takes when a program requests an I/O. The first point to understand is the difference between a logical I/O and a physical I/O. When a program issues a read, it is referred to as a logical I/O. A physical I/O is a request issued by the operating system to a controller to retrieve data from DASD. Since VSAM accesses keyed data using an index, there is a need to perform several physical I/Os to obtain the appropriate record. The figure below shows an example of a VSAM KSDS file with three index levels. To satisfy one logical I/O, VSAM may have to perform four physical I/Os to retrieve the requested record. Therefore, the program's request for one logical I/O translates into several physical I/O operations.

During a direct-read operation, VSAM first reads the dataset index to identify which control interval contains the requested record. A VSAM dataset normally has several levels of index, and VSAM starts by reading the highest level of the index set called the root. VSAM then reads each index set level until it reaches the lowest level of the index, called the index sequence set. The index sequence set points to the appropriate data control interval where the record is located. The data control interval is finally read into storage, and the program is passed the record.

A VSAM index often consists of two or three levels. Assuming three levels, 75 percent of the physical I/O related to a logical I/O is for the index, and only 25 percent is for data. This assumes that no levels of the index are buffered in storage. The key to excellent I/O performance is reducing the number the physical I/Os through efficient buffering.

During sequential or skip-sequential processing, the index sequence set is used to locate the next logical control interval. Therefore, browsing requires reading the index as well.

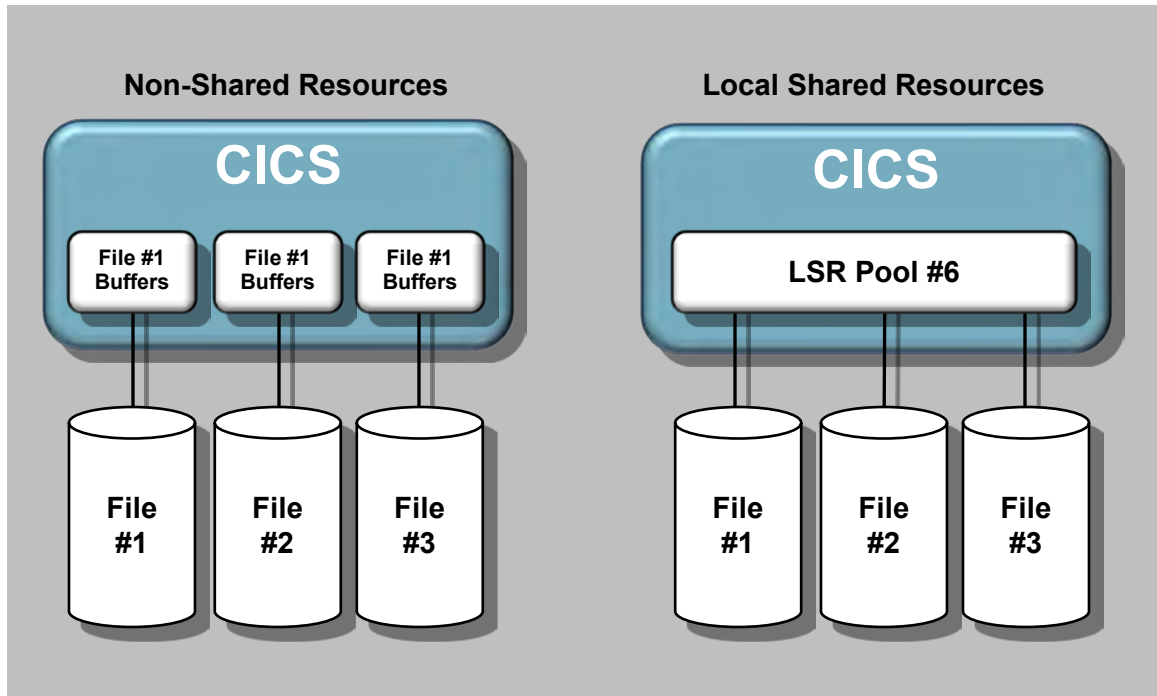
The VSAM KSDS file in the figure below has an index level of three. To perform one logical I/O, VSAM may have to perform four physical I/Os to retrieve a record.



Local Shared Versus Non-Shared Resources

VSAM provides two methods of reserving main storage buffers for VSAM datasets. The first method is called Non-Shared Resources (NSR). As the name implies, a set of index and data buffers are reserved for the exclusive use of one VSAM dataset. No other dataset can share those buffers. The second method is called Local Shared Resources (LSR). This method creates a series of buffer pools that can be shared by multiple VSAM files. The figure below compares the two methods as they relate to CICS.

In the past, batch COBOL programs could only use NSR buffering, and only assembler programs could use LSR. CICS, on the other hand, uses both techniques, and it allows the CICS system programmer to tune each technique based on VSAM file activity and utilization. With IBM's DFP product, programmers can add new JCL parameters to give batch COBOL programs the ability to exploit LSR buffering.



How to Improve VSAM Performance

The remainder of this chapter explains guidelines that, if applied correctly, can improve VSAM dataset performance. These guidelines are as follows.

1. Avoid Control Interval and Control Area Splits (page 87)
2. Reorganize Datasets With Multiple Extents (page 87)
3. Spread Heavily Accessed Datasets Over Several Devices (page 88)
4. Use Index and Data Buffering (page 88)
5. Adjust Index and Data Control Interval Sizes (page 88)
6. Manage One-Cylinder VSAM Datasets (page 89)
7. Analyze CICS End-of-Day Statistics (page 89)
8. Use VSAM Local Shared Resources (page 90)
9. Use CICS VSAM Subtasking (page 92)
10. Beware of Datasets Using Share Options 3 or 4 (page 92)
11. Reduce Program I/O (page 92)
12. Place Critical Files on Faster Storage Mediums (page 93)

VSAM Performance

Avoid Control Interval and Control Area Splits

Control area (CA) and interval (CI) splits are easily detectable and can result in poor performance. The number of CI and CA splits are labeled SPLITS-CI and SPLITS-CA, respectively, on an IDCAMS LISTCAT report (see the example below). When a split occurs, the various index levels are updated to reflect the existence of a new control interval. VSAM locks part of the index to ensure that no other requests are processed while the index and data are being split. This affects not only the program causing the split but also other programs accessing the same VSAM file. Splits cause large amounts of data to be moved around on DASD and can cause considerable delays.

Possible remedies for high split activity are (1) allocating sufficient free space and (2) regularly reorganizing the dataset.

```
CLUSTER--WE.DD.UIOP.PROD.EEDH
ATTRIBUTES
KEYLEN-----3          AVGLRECL-----150
RKP-----0           MAXLRECL-----150
SHROPTNS(2,3)  RECOVERY  UNIQUE           NOERASE
NONSPANNED
STATISTICS (* - VALUE MAY BE INCORRECT)
REC-TOTAL-----21*     SPLITS-CI-----1365*
REC-DELETED-----0*    SPLITS-CA-----457*
REC-INSERTED-----0*   FREESPACE-%CI-----0
REC-UPDATED-----2963*  FREESPACE-%CA-----0
REC-RETRIEVED---1026811*  FREESPC-----202752*
ALLOCATION
SPACE-TYPE-----TRACK   HI-A-RBA-----221184
```

Reorganize Datasets with Multiple Extents

If a dataset is in multiple extents, the logically adjacent records are not physically adjacent on the disk. This requires additional seek time to retrieve the next record, resulting in higher access time. The minimum seek time on a 3380 device is 2–3 ms, and the average is 12–15 ms. Seek time is the largest component of I/O and can be minimized by keeping datasets in as few extents as possible.

If you identify multiple extents by analyzing the IDCAMS LISTCAT report (see the example below), reorganize the VSAM file. If the file has grown out of the primary extent, change the IDCAMS define card to allocate a primary size large enough for the existing data and future growth. The amount needed for additional growth depends on the characteristics of the VSAM file activity.

```
EXCPS-----239*
EXTENTS-----1
SYSTEM-TIMESTAMP:
X' CDCD5CA7538DAC80 '
INDEX:
LEVELS-----3
ENTRIES/SECT-----11
SEQ-SET-RBA-----2048
HI-LEVEL-RBA-----2048
```

Spread Heavily Accessed Datasets

If more than one heavily accessed dataset is present on a volume, program performance suffers. On a volume with one heavily used dataset, the disk heads tend to remain close to the most heavily accessed data. If there are two heavily accessed datasets, the heads spend much time traversing the width of the disk surface servicing I/O requests for both datasets. The heads are seldom near the requested data. This extended seek time results in extended response time.

How much is *extended* seek time, and how can it be monitored? Since an average disk I/O completes in 25 ms, I/O times in excess of this value should be investigated. These times are obtained from IBM's Resource Measurement Facility or similar products. Consistent extended seek times produce a critical performance bottleneck. When extended seek times occur for an I/O request, all other requests for data on the same disk are queued until the first request is complete. Therefore, all programs are affected, and performance suffers.

Use Index and Data Buffering

The most efficient logical I/O is obviously no physical I/O. The use of index and data buffering in main storage can result in physical I/O reduction. The specification of BUFNI (number of index buffers) and BUFND (number of data buffers) in the CICS file control table should be analyzed when using the NSR buffering technique. BUFNI should be at least equal to the number of levels in the dataset's index plus 1. Never enter a smaller value—performance will suffer otherwise.

The use of buffering should be moderated because over-buffering causes paging to increase, thus diminishing buffering's effect. In a CICS environment, paging a record in from a page dataset is efficient but never preferable over a normal I/O. The paging operation places the whole CICS region in a wait, affecting all programs, while a dataset I/O affects only the program requesting the I/O.

Adjust Index and Data Control Interval Sizes

Datasets with high browse activity should have a larger data CISIZE. Datasets accessed randomly should have a smaller CISIZE. If a file is used in a mixed mode, determine the size based on the percentage between direct and sequential access.

The index CISIZE is rarely larger than the data CISIZE. The definition of any dataset for which the index CISIZE is larger than the data CISIZE should be investigated. A general rule of thumb is to allow VSAM to choose the index CISIZE. VSAM generally determines the best size based on the data CISIZE, key length, and record length.

VSAM Performance

Manage One-Cylinder VSAM Datasets

Small VSAM data sets of one cylinder or less should have only one level of index. VSAM requires one index record per data CA. The maximum size of a data CA is one cylinder; therefore, if the file is defined correctly, only one level of index is needed. The minimum size for a data CA is one track. VSAM decides the size of the data CA by the values contained in the data primary or secondary allocation parameters. VSAM uses the smaller of the two values as the data CA size.

If the primary or secondary data allocation sizes are coded incorrectly, VSAM creates another index record. VSAM allows only one index level at the root; therefore, the creation of the new index record forces the creation of a new level. If the dataset has more than one level of index, it is defined incorrectly, and 50 percent more I/O than necessary is being performed.

To determine the number of levels contained in a VSAM dataset, refer to the IDCAMS LISTCAT. See the example below.

```

EXCPS-----239*      INDEX:
EXTENTS-----1      LEVELS-----3
SYSTEM-TIMESTAMP:    ENTRIES/SECT-----11
X'CDC5CA7538DAC80'  SEQ-SET-RBA-----2048
                    HI-LEVEL-RBA-----2048
    
```

Analyze CICS End-of-Day Statistics

When CICS is shut down, it creates a series of end-of-day statistics. The CICS end-of-day statistics are useful for detecting bottlenecks or resource shortage situations. The numbers of index and data EXCPS per dataset are listed as well as string waits and other valuable information. These statistics should be checked to detect any out-of-line situations. See the example below.

***** FILE STATISTICS *****							
FILE NAME	GET REQSTS	GET UPD REQSTS	BROWSE REQSTS	ADD REQSTS	UPDATE REQSTS	DELETE REQSTS	VSAM EXCP DATA
MAFMC007	0	8	0	1	1	0	5
BATCNTL	852	2	0	0	2	0	854
ITIVSCTL	7	1	0	0	1	0	2
PROTHLF	0	0	0	0	0	0	0
PROTSYM	22	0	0	0	0	0	11
ABNDFILE	0	0	0	11	0	0	12
DFHCSD	75	0	1946	0	0	0	35
FILEFILE	0	0	0	0	0	0	0
TOTALS	965	17	1954	16	10	2	946

Use VSAM Local Shared Resources

VSAM LSR under CICS allows multiple files to share a common pool of buffers and strings. Before issuing a read-to-disk, VSAM scans the LSR buffers to check whether the control interval is already in storage. If it is available, CICS does not issue the read. This buffer-lookaside capability reduces I/O time significantly. Also, only one copy of the CI is kept in storage (as opposed to multiple copies using NSR). This reduces virtual storage usage and allows better read integrity.

While LSR can greatly improve performance, performance degradation can occur if it is incorrectly specified. Allocating too many buffers causes excessive paging, while allocating too few buffers reduces lookaside ratios, resulting in increased I/O activity. As stated earlier, the greatest reduction of a program's elapsed processing time is in I/O reduction. VSAM LSR can help achieve this.

LSR use must be analyzed because not all files are good candidates for it. Large files would yield a low buffer "hit rate" and would not benefit from LSR. Datasets browsed heavily should be separated from other datasets by using NSR. This is because the browsing causes other dataset control intervals to be flushed from the buffers and upsets the LSR lookaside capability.

Datasets that experience a large number of CA splits would be good LSR candidates since the CICS subtask or main task is not placed in a wait state while processing the split. With NSR, the CICS subtask or main task is put in a wait state.

The discussion below is a suggested technique for organizing and tuning the LSR pool. Assume that for every VSAM request there are typically three physical I/Os performed. Two of these are for the two levels of index, and the third is for the data itself. The objective in tuning the LSR pool should be to remove or limit the index I/O. This equates to 67 percent of the I/O.

By ensuring that there is sufficient buffer space to hold the index in storage and ensure that the index is not flushed out by data I/O, the goal can be achieved. Unfortunately, by current LSR design, the index and data components must reside in the same LSR pool. If they were allowed to be in different pools, it would be a fairly simple matter to ensure that they did not interfere with one another. This is not the case, however, and another solution must be used. Here is an alternative approach:

VSAM Performance

1. Create several LSR pools. Each pool is used by datasets with a particular index CISIZE.
2. Define the datasets with different index and data CISIZES.
3. Place datasets with the same index CISIZE in the same pool. This allows the index control intervals exclusive use of a set of buffers in the pool with no impact from the data portion of the dataset. This solves the problem efficiently.

Eight LSR pools are currently allowed under CICS 1.7 and 2.1. If DFP 2.3 or higher is installed, the buffers reside above the 16-megabyte line, thus relieving the virtual storage constraint within CICS.

Using LSR not only saves virtual storage within CICS, it also requires less storage within other areas of z/OS, as follows:

- Each NSR file requires 5k to 8k of OSCOR, while each LSR file requires only 3k.
- Each NSR file requires 1200 to 1500 bytes of CSA, while each LSR file requires no storage in CSA.
- Each NSR file requires 240 bytes of SQA per string, while each LSR file requires no SQA storage per string.
- The number of buffers required for NSR are the number of strings plus one. This extra buffer is used for splits and is generally wasted. The number of buffers required for LSR can be equal to the string number, resulting in no waste.

Use care when defining the SHRCTL macro (the LSR pools). Pre-DFP 2.3 systems should not have more than 55 buffers per pool because the scanning overhead for a record in the buffer becomes excessive. DFP 2.3 and above has a new buffer-hashing algorithm that it uses to scan the buffers. Tests using more than 800 buffers indicate no performance degradation.

When specifying the number of 512, 1024 and 2048 buffers, make sure they are a multiple of the z/OS page size. Otherwise, the remaining space on the page is wasted.

To monitor the results of the LSR tuning, refer to the CICS shutdown statistics as shown in the example below. These statistics show the buffer lookaside hit rates achieved. The target rates should be 90 percent or more for index and 80 percent or more for data. This is only a general rule, and the percentages may vary greatly depending on data-access patterns and frequency.

VSAM Performance

```

LSR POOL CREATED: 20.16.14

MAXIMUM KEY LENGTH                100
TOTAL NUMBER OF STRINGS            10
HIGHEST NO. OF REQUESTS WAITED FOR STRING 0
TOTAL NUMBER OF REQUESTS WAITED FOR STRING 0
MAX. NO. CONCUR. ACTIVE FILE CONT. STRINGS 2
    
```

BUFFER SIZE	NUMBER OF BUFFERS	NUMBER OF SUCCESSFUL LOOK-ASIDES	USER-		NON-USER-	
			NUMBER OF BUFFER READS	INITIATED BUFFER WRITES	INITIATED BUFFER WRITES	INITIATED BUFFER WRITES
2,048	70	61	8	1	0	0
4,096	21	0	0	0	0	0
8,192	21	27	6	4	0	0
12,288	21	0	0	0	0	0
16,384	21	0	0	0	0	0
TOTALS		88	14	5	0	0

Use CICS VSAM Subtasking Using VSAM subtasking directs CICS to attach a z/OS subtask to perform VSAM and other dataset I/O. Since this I/O is not performed within the CICS main task, CICS is free to service other CICS transaction activity. This provides for multi-processing within the same region. Although the major benefit of using VSAM subtasking is increased throughput, it also reduces response time.

You specify the VSAM subtasking option in the CICS System Initialization Table (SIT) by using the VSP=YES parameter. VSP=NO is the default.

For CICS 3.3, set SUBTSKS=1 for subtasking. SUBTSKS=0 is the default.

Beware of Datasets Using Share Option 3 or 4 Avoid using VSAM datasets defined using Share Option 3 or 4. Apart from the integrity exposures that Share Option 3 imposes, using datasets defined with Share Option 3 results in degraded performance within CICS. It forces the execution of z/OS ENQ logic, placing additional delays on the CICS system. Also, using Share Option 4 forces a buffer refresh for every read request. This increases the physical I/O rate and negates the benefit of buffer look-aside. In general, using these Share Options degrades performance in all circumstances.

Reduce Program I/O Up to this point, all discussion has centered on improving performance by tuning the environment—buffer utilization and dataset placement, for example. When all of these options are exhausted, the only remaining way to reduce I/O is within the application. Analyzing the application to remove unnecessary I/O is required. By reducing application I/O requests, you achieve two savings. First, there is the substantial savings of the I/O wait time. Second, there is a savings of 7000 instructions per I/O.

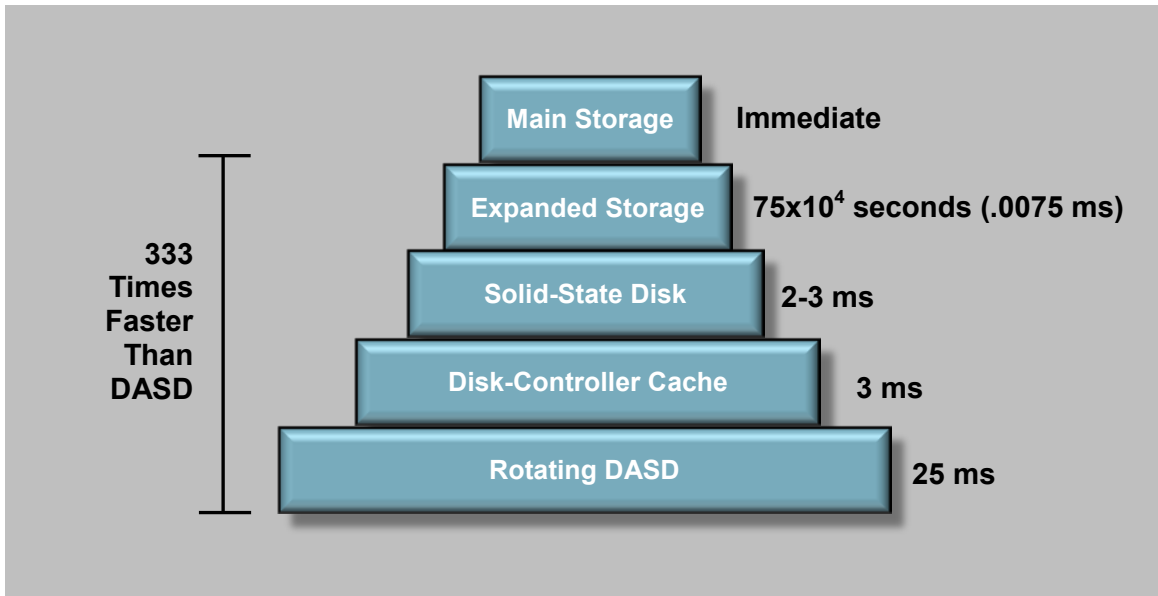
VSAM Performance

The effort needed to analyze the application and remove the I/O is time consuming and expensive. You should perform this work only after all environment-tuning options are exhausted and if further savings are needed.

Place Critical Files on Faster Storage Mediums

Place critical datasets on a quicker storage medium. As described earlier in this section, there are various storage mediums that provide better access speed than a traditional rotating DASD. By using these devices, performance can be greatly improved.

The different storage mediums and their speeds are represented in the figure below.



Installation

Overview of VSHARE Installation on z/OS

This section describes the installation process for VSHARE™ for z/OS. The process consists of the following steps, which are described in more detail below.

- Preparation
 - Ensure that the VTAM and CICS levels are appropriate.
 - Obtain an account number and a product key.
- Review the module types and load the contents to DASD. To do this, you can
 - Load the VSHARE libraries from tape.

OR

- Download the distribution from the Web to your PC, transfer the modules to your z/OS system, and install them.
- Install the software product key.
- Create the VSHARE control dataset.
- If not already implemented, define the CICS region to use ISC.
- Add the VSHARE library to DFHRPL in the CICS startup JCL.
- Create the VTAM and CICS definitions for VSHARE.
- Define eligible CICS regions and batch jobs to VSHARE.
- Install Cross Memory Services.
- Modify Tier 2 recovery procedures.
- Implement the VSAM Router Facility.
- Use VSHARE activation and deactivation.
- Review security needs, and modify the CSIVSBSC module.

Preparation

- Ensure that the system release levels are appropriate:
 - VSHARE executes under CICS/VSE 2.1.2 or greater.
 - VSHARE can run on VTAM level 3.2 or greater.
- Contact your sales representative to obtain an account number and a product key. You can reach a CSI International sales representative at 800-795-4914, or email sales@csi-international.com.

Installing VSHARE from Tape

If you received VSHARE on an IEBCOPY tape, the tape contains the following libraries:

Library	Description
VSHARE.INSTALL	This PDS contains the JCL required to perform this installation process.
VSHARE.LOAD	This PDS contains the load modules used by batch programs.
VSHARE.CICSLOAD	This PDS contains the load modules used by CICS.
VSHARE.APF.LOAD	This PDS contains the authorized load modules; it must be defined as APF authorized. It is required for Cross Memory Services and the low level open interface installations.

Use the following steps to load these libraries from tape.

1. Modify and run the JCL below to load the installation library (“INSTALL”).

```
//USERJOB JOB JOBINFO
//*****
//*          FSIVSHAR.INSTALL
//*
//*****
//STEP001 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,60)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,60)
//OUTPUT   DD DSN=USER.FSIVSHAR.INSTALL,DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3280),
//          SPACE=(CYL,(1,1,50),RLSE),UNIT=SYSDA
//INPUT     DD DSN=FSIVSHAR.INSTALL,DISP=OLD,
//          LABEL=(1,SL),UNIT=TAPE,
//          VOL=SER=FSISYS
//SYSIN     DD *
COPY INDD=((INPUT,R)),OUTDD=OUTPUT
//*
```

2. Use the member LOADTAPE, located in the installation library, to load the remaining libraries (“LOAD,” “CICSLOAD,” and “APF.LOAD”).

Installing VSHARE from a Web Download

Download the Installation Datasets to a PC Using your account number, download the CSI VSAM Router and VSHARE installation ZIP file from the CSI International website to your PC. Use this address:

<http://www.csi-international.com/products/zOS/VSHARE/VSHARE.htm>

Unzip the installation file you downloaded. The folder contains the datasets listed in the table below.

Library	Description
VSHARE.INSTALL	Contains JCL required for the installation process.
VSHARE.LOAD	Contains the load modules used by batch programs.
VSHARE.CICS.LOAD	Contains the load modules used by CICS.
VSHARE.APF.LOAD	Contains the authorized load modules; it must be defined as APF authorized. It is required for Cross Memory Services and low level open interface installations.
ROUTER.APF.LOAD	Contains the load modules for the router.

Transfer the Installation Datasets The z/OS movable files in this distribution set were created with the IBM NJE function 'XMIT' to ensure that the unloaded file is a sequential file with fixed-length records (LRECL=80, BLKSIZE=3120). This ensures that the file can be uploaded to the z/OS platform using any file transfer utility that can transfer data in *binary format*. The z/OS site conditions and procedures dictate what files, names, and security conditions apply to such transfers.

Setting Up the IND\$ Transfer Profile

If you use the z/OS function IND\$ in any of the 3270 emulation programs as the transfer method, ensure that a binary transfer type is defined in the emulation software profiles. If a profile has not previously been defined in the 3270 emulation software, follow your software's instructions to set the parameter values in the table below. Your software may label the parameters differently.

IND\$ Transfer Parameters	
Parameter	Value
Transfer name	As appropriate

IND\$ Transfer Parameters	
Parameter	Value
Record format	Fixed
Record length	80
Block size	3120
ASCII	Off
CLRF	Off
Primary allocation amount	As appropriate
Secondary allocation amount	As appropriate
Allocation units	Cylinders or tracks

Be sure to save this definition.

Using IND\$ to Transfer the File

To transfer an unloaded sequential binary dataset from the PC to the mainframe z/OS system using IND\$, navigate in the 3270 emulator to ISPF Option 6 panel or to the TSO READY prompt. Specify the following information according to your emulator's instructions to send a sequential binary file:

- The source drive, directory and file name (on the PC)
- The destination file name (on the mainframe)
- The transfer profile (configured above)

Then transmit the file.

RECEIVE and RESTORE the Datasets

After you transfer the datasets to the z/OS mainframe, they remain sequential datasets. Use the following steps to install each dataset.

Step	Action
1	<p>Navigate to the ISPF Option 6 panel or the TSO READY prompt and enter the following command:</p> <pre>RECEIVE INDSN(' <i>sequential.dataset.name</i>')</pre> <p>During the TSO setup of the command, the following prompt is displayed:</p> <pre>Restore parameter or DELETE or END +</pre>

Installation

Step	Action
2	<p>Enter RESTORE with any of the following options: DSNAME, SPACE, UNIT, VOL, OLD/NEW. The recommended usage is as follows:</p> <pre data-bbox="771 394 1404 457">RESTORE DSNAME(' your.dsname ') UNIT(sysda) NEW</pre> <p>In the example above, a new partitioned dataset is created on unit <i>sysda</i>. If the transfer succeeds, the following message is displayed:</p> <pre data-bbox="803 590 1339 621">Restore successful to data set ' your.dsname '</pre>
3	<p>Repeat from step 1 for each unloaded file in the downloaded distribution set. See the example job below.</p>

Installation

Example Job

The following example batch job RECEIVES the uploaded sequential dataset ROUTER.TEMP.LOADLIB and RESTORES it to the dataset ROUTER.P390.LOADLIB

```
//P390A JOB (MEA,WBY000000,1500,15), 'ED MOLNAR', NOTIFY=&SYSUID,
//          CLASS=A,MSGCLASS=K,MSGLEVEL=(1,1)
//*
//STEP01 EXEC PGM=FTP,REGION=8192K
//STEPLIB DD DSN=TCPIP.SEZALINK,DISP=SHR
//SYSTCPD DD DSN=PROD.TCPIP.PARMLIB(TCPDATA),DISP=SHR
//SYSFTPD DD DSN=PROD.TCPIP.PARMLIB(FTCDISTO),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUTPUT DD SYSOUT=*,DCB=(LRECL=160,RECFM=FB,BLKSIZE=1600)
//INPUT DD *
172.17.2.28
ANONYMOUS ME
SENDSITE
CD /DOWNLOAD/
BINARY
GET ROUTER.BIN 'ROUTER.TEMP.LOADLIB'
QUIT
//*
//STEP02 EXEC PGM=IKJEFT01,REGION=1024K
//INDD1 DD DSN=ROUTER.V2R0A.LOADLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('ROUTER.TEMP.LOADLIB')
RESTORE DSNAME('ROUTER.P390.LOADLIB') UNIT(SYSDA) NEW
//*
```

Install the Software Product Key

All software products from CSI International are protected by a Product Key. These Product Keys must be installed in your system using the process described below.

A separate Product Key is issued to you for each CSI International product you have licensed. Each key is for a specific product, and it authorizes the use of that product on a specific CPU until a specific date, depending on the type of license agreement you have signed.

If you are licensed to run a product on more than one physical CPU, then you are issued a separate Product Key for each CPU serial number. Note that LPARs are considered to be the same physical CPU. If you did not receive a Product Key for each of your products, please contact your CSI International sales representative.

Installation

Installing Product Keys Each CSI International product looks for its product key in a load module named FSIPKEYS. You generate this load module at your site using the JCL below.

```
//C          EXEC PGM=ASMA90
//SYSLIB    DD   DSN=product.source,DISP=SHR
//SYSUT1    DD   DSN=&&SYSUT1,SPACE=(4096,(120,120),,,ROUND),
//          UNIT=SYSDA,DCB=BUFNO=1
//SYSPRINT  DD   SYSOUT=*
//SYSLIN    DD   DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//          UNIT=SYSDA,DISP=(MOD,PASS),
//          DCB=(BLKSIZE=3040,LRECL=80,RECFM=FB,BUFNO=1)
//SYSIN     DD   *
            PRODKEY MAZE-SLIP-ZOOM-KNOW-GLOW (VSHARE, exp 12/31/20xx)
            PRODKEY ADDS-TOIL-GUN-KNOW-GASP (RELIANT, exp 12/31/20xx)
            END
/*
//L          EXEC PGM=HEWL,PARM='MAP,LET,LIST,NCAL',COND=(8,LT,C)
//SYSLIN    DD   DSN=&&OBJ,DISP=(OLD,DELETE)
//SYSLMOD   DD   DSN=your.auth.LinkLib(FSIPKEYS),DISP=SHR
//SYSUT1    DD   DSN=&&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
//          UNIT=SYSDA,DCB=BUFNO=1
//SYSPRINT  DD   SYSOUT=*
//
```

Notes:

- The *product.source* shown in the SYSLIB DD statement is the location used to find the PRODKEY macro. The PRODKEY macro is supplied with each CSI International product.
- Change the PRODKEY statement to use the product key value you obtained from your CSI International sales representative.
- The *your.auth.LinkLib* shown in the SYSLMOD DD statement is where you want the FSIPKEYS load module to be linked. To be properly available to all CSI International products, it must be APF authorized and in the LNKLLST.
- If you do use a LNKLLST load library, you will need to issue an LLA, REFRESH command each time you change the table.

Product Authorization Messages When you attempt to use a CSI International product, you may receive one of the following messages:

```
prodname-9998 WARNING - PRODUCT AUTHORIZATION EXPIRES IN nn DAYS
```

This message is generated when authorization is set to expire in 30 days or less.

prodname-9999 PRODUCT AUTHORIZATION HAS FAILED, REASON=nn, CPU=xxxxxxxx

This message is generated if the product is unable to validate a product key. The REASON code can be one of the following:

- 12 GETMAIN failed
- 20 Load for FSIPKEYS table has failed
- 24 No product key found for the product and/or CPU serial number.
- 28 Product key has expired.

Depending on the product, these messages could appear on the system console, in the SYSPRINT output of the job, or on the screen if you are using a CICS product.

If you see one of these messages, contact your CSI International sales representative for a new product key.

Define a VSHARE Control Dataset

The control dataset contains all the system, job, and history information. The number of control datasets you choose to create depends on the user. There can be one per CICS region, or one for all CICS regions. If one dataset is used for many regions, all the CICS regions using the control dataset must have access to the dataset through MRO or ISC. The installation library contains a member called DEFCTL that defines this VSAM KSDS dataset. Change the dataset names to conform to installation standards.

Control Dataset Definition

```
DELETE -
    FSIVSHAR.TCCVCTL -
    PURGE
IF LASTCC = 8 THEN SET MAXCC = 0
DEFINE CLUSTER -
    (NAME (FSIVSHAR.TCCVCTL) -
    RECSZ (40 8192) -
    KEY (31 0) -
    INDEXED -
    IMBED -
    SPEED -
    REPLICATE -
    FREESPACE(2 10) )-
DATA -
    (NAME (FSIVSHAR.TCCVCTL.DATA) -
    CISZ(10240) -
    SHR(2 3) -
    REC(20 5) -
    VOL(XXXXXX)) -
INDEX -
    (NAME (FSIVSHAR.TCCVCTL.INDEX) -
    CISZ(1024) -
    SHR(2 3) -
    REC(20 5) -
    VOL(XXXXXX))
```

Update the CSIVSBDS Parameter Member

The CSIVSBDS member is used to specify the control dataset name for VSHARE. Since the name of the control dataset defined can vary, a module is created for each control dataset defined. Edit the member CSIVSBDS. Assemble this module, and link-edit it into the VSHARE library.

The load module CSIVSBDS is loaded by VSHARE at initialization time to determine which control dataset to open. If multiple control files are used, a CSIVSBDS module needs to be assembled and link-edited into different libraries. You will also need a separate activation job for each control dataset. Refer to section “VSHARE Activation and Deactivation” on page 111 for more information.

Below is an example of the coded CSIVSBDS module. Modify the dataset name as required by your installation standards. Make sure this is the same name used in the definition of the control dataset previously discussed.

Installation

```
CSIVSBDS  CSECT
          DC    CL44 'FSIVSHAR.TCCVCTL'
          END
```

ISC Installation

VSHARE uses LU 6.2 to communicate with the CICS region. The CICS region must be defined to use CICS ISC. Make sure your CICS System Initialization Table contains the operand ISC=YES. Details on ISC can be found in IBM's "CICS Inter-Communication Guide." If ISC is already in place, this step may be skipped.

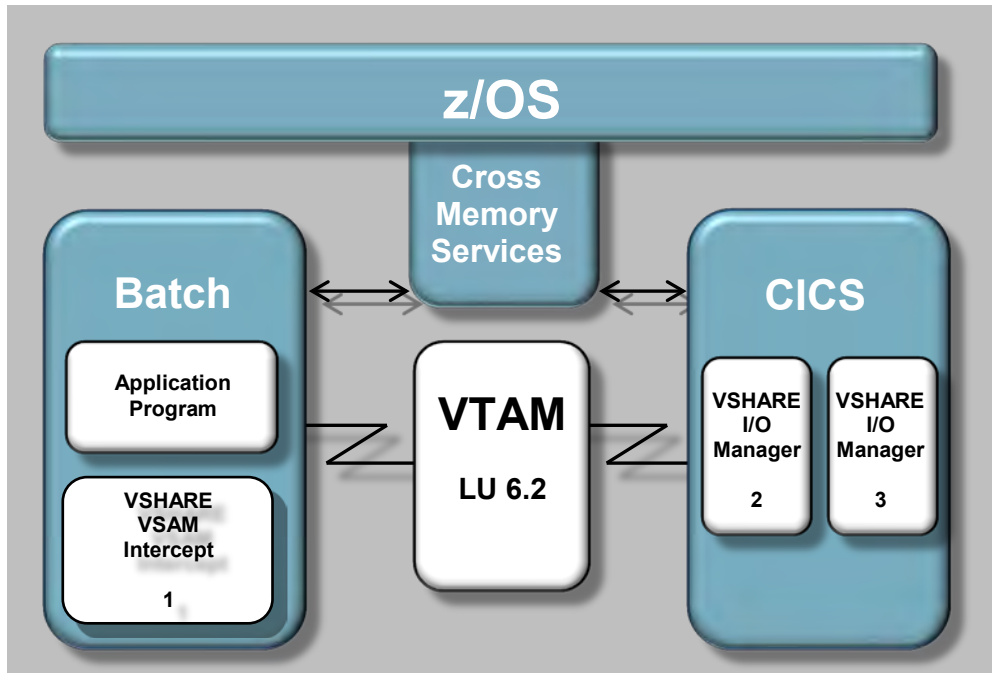
Place the Load Library in the DFHRPL

Add VSHARE CICS load library to the DFHRPL of all CICS regions using VSHARE.

Environment Definitions

The figure below shows VSHARE's general design. VSHARE is located in both the CICS region owning the VSAM files and the batch region. The CICS and batch programs can communicate the VSAM request through LU 6.2 or Cross Memory Services (CMS). LU 6.2 allows the batch region and the CICS region to be running on different CPUs, while CMS provides faster data transfer when the batch region and the CICS region reside on the same CPU.

VSHARE always uses LU 6.2 to start its conversation between the batch region and CICS. After the communications link is established between the batch region and CICS, the CMS interface can be used if it has been requested and both parties are on the same CPU.



Installation

With this in mind, the following definitions are needed:

1. VTAM:

Define an APPLID for the batch programs. This APPLID is opened by VSHARE batch programs to establish an LU 6.2 session with the CICS region. The number of APPLIDs determines the number of concurrent jobs that can execute.

2. CICS:

Define CICS entries for VSHARE'S transactions, programs, and the LU 6.2 sessions.

3. VSHARE online updates:

Define the CICS region and default processing options to VSHARE through the VSHARE online panels. This is discussed in greater detail later in this chapter as well as in the Online Control chapter (page 41).

The definitions needed to allow VSHARE to communicate between the batch region and CICS are interrelated. If certain parameters are not properly linked together, the connection between the two regions will be rejected by CICS or VTAM.

Before describing each definition in detail, the following figures show the relationships between the definitions. The first figure shows the required definitions when session and connection definitions are used within CICS. The second figure shows the required definitions when a single-session terminal definition is used within CICS.

Definitions Required When Terminal Definitions Are Used in CICS

VSHARE System Maintenance Screen

APPLID: CICSQA13 FSI-VSHARE - THE CICS INTERFACE DATE: 10/16/2014
 PROGID: CSIVSC19 S Y S T E M M A I N T E N A N C E TIME: 06:12:11

FUNCTION: (A = ADD, B = BROWSE, C = CHANGE, D = DEL)
 CICS APPLID: PRODCICS

SYSTEM INFORMATION
 TRANID: TCCB ACCESS TYPE: CMS
 MODE NAME: LU62MODE SYNC LEVEL: 005
 ROLLBACK: N REQ/MIN: 00000
 APPLID SEL: N SELECT MSK: *****

CICS APPLID Definition
 PRODCICS APPL AUTH=(ACQ,PASS,VPACE),
 VPACING=20,EAS=20,
 ACBNAME=PRODCICS,
 PARSESS=YES,
 SONSCIP=YES
 MODETAB=MODETAB1

DYNAMIC JOURNAL NAME:
 //ITIJRNL DD DSN= PROD.FSIVSHAR.R30 .JOBNAME.ITIJRNL,
 DCB=(RECFM=VB,BLKSIZE= 27004 ,LRECL= 0900,
 SPACE=(CYL,(0010 , 0050)),UNIT= SYSDA

BATCH APPLIDS: BATCH03 BATCH04 C13TS420 QA21TS

ENTER=EDIT CLEAR=EXIT PF3=PROCESS PF7=BACK PF8=FO

Batch APPLID Definition
 BATCH03 APPL AUTH=(ACQ,VPACE),
 VPACING=20,EAS=1,
 PARSESS=NO,
 SONSCIP=YES
 MODETAB=MODETAB1
 DLOGMODE=LU62MODE

CICS Typeterm Definition
 OBJECT CHARACTERISTICS
 CEDA View
 Typeterm : DFHLU62T
 Group : FSIVSHAR
 RESOURCE TYPE
 DEvice : APPC
 TERmodel :
 SEssiontype :
 LDclist :
 SHippable : No No|Yes
 MAPPING PROPERTIES
 PAGesize : 001,040 0-999
 ALTPage : 000,000 0-999
 ALTSuffix :
 FMhparm : No No|Yes
 OBOperid : No No|Yes
 PAGING PROPERTIES
 AUTOPage : Yes No|Yes
 DEVICE PROPERTIES
 DEFscreen : 000,000 0-999
 ALTScreen : , 0-999
 ALPKybd : No No|Yes
 .
 .
 BAcKtrans : No No|Yes
 CGcsgid : 0000,0000 0-65535

Mode Entry Definition When SMSNA100 Is Not Used
 MODETAB1 MODETAB
 .
 .
 LU62MODE LOGMODE=LU62MODE
 MODEEND

CICS Terminal Definition
 OBJECT CHARACTERISTICS
 CEDA View
 Terminal : LU6T
 Group : FSIVSHAR
 AUTINSTModel : No
 AUTINSTName :
 TERMINAL IDENTIFIERS
 Typeterm : DFHLU62T
 Netname : BATCH03
 Console : No
 REMOTESystem :
 REMOTENAME :
 Modename : LU62MODE
 ASSOCIATE PRINTERS
 PRINTER :
 PRINTERCopy : No
 ALTPRINTER :
 ALTERPRINTCopy : No

Installation

VTAM Definitions There are two definitions required for VSHARE. The first is the definitions of new VTAM APPLIDs. Since LU 6.2 is used, the batch job needs a batch APPLID to open in order to communicate with VTAM.

An example of an APPLID definition for VSHARE is as follows:

```
name    APPL    ACBNAME=applid1,  
          AUTH=(ACQ,VPACE),  
          EAS=1,  
          MODETAB=tabname,  
          DLOGMOD=entname,  
          PARSESS=NO,  
          VPACING=n
```

ACQ: The acquire authority option must be specified.

VPACING: Set the pacing number based on installation specifications.

APPLID: Use installation naming conventions.

MODETAB: Only code when SMSNA100 is not in default mode table.

DLOGMOD: Only code when SMSNA100 is not in default mode table.

One APPLID definition is needed for each simultaneous batch job. If four batch jobs are required to run simultaneously, then four unique APPLIDs need to be defined. There is no need to define a unique APPLID for every batch job. Define only the maximum number of concurrent jobs that can run at a time. Currently, there is a logical limit of 90 simultaneous jobs per CICS region. You can also share the same batch APPLIDs between CICS regions, but a batch APPLID can be used by only one batch job at a time.

In the APPLID there is a reference to MODETAB. This is the name of the mode entry table in VTAM. In the VTAM-supplied default mode entry table, there is a member named SMSNA100. Use this entry for the mode entry value. If it is not part of the default mode entry table, create a dummy entry like the definition located on the opposite page. It is important that the mode entry used is available to both the CICS APPLID and the batch APPLID, otherwise CICS rejects the connection attempt.

At the end of this chapter there is a list of VTAM return codes that are common when the definitions of the sessions and connections are not correct. Check this list to isolate possible problems with the connection between CICS and the batch job. Remember to look not only at the VSHARE batch log, but the CICS log as well.

CICS Definitions

TCT Definitions

The CICS TCT definitions required to define the LU 6.2 link between the batch job and the CICS region are called sessions and connections. These definitions define each batch APPLID to the CICS region.

The connection defines the attributes of the remote LU 6.2 system.

```
CEDA  DEFINE
      CONNECTION(connection name)
      GROUP(FSIVSHAR)
      NETNAME(applid)
      ACCESSMETHOD(VTAM)
      PROTOCOL(APPC)
      SINGLESESS(YES)
      DATASTREAM(USER)
      RECORDFORMAT(U)
      AUTOCONNECT(NO)
      INSERVICE(YES)
      ATTACHSEC(LOCAL)
```

Remember: There is one connection definition per batch APPLID.

The second TCT definition needed for each connection definition is a session. Notice in the figure below that the session is defined as being a single session. Also define the SENDSIZE and RECEIVESIZE to be 30720. Note that this is the allowable maximum; therefore, if the largest record is 1024, the actual amount of data send is 1024 plus a few hundred bytes of control information, not 30720 bytes.

```
CEDA  DEFINESSESSIONS(session name)
      GROUP(FSIVSHAR)
      CONNECTION(connection name)
      MODENAME(modeent)
      PROTOCOL(APPC)
      MAXIMUM(1,1)
      RECEIVECOUNT(NO)
      SENDCOUNT(NO)
      SENDSIZE(30720)
      RECEIVESIZE(30720)
```

Remember: There is one session definition per batch APPLID.

PCT Definitions

The definition for all VSHARE transactions are listed below.

```
CEDA DEFINE TRANSACTION(TCC)
      GROUP(FSIVSHAR)
      PROGRAM(CSIVSC10)
      TASKDATAKEY(CICS) for CICS 3.3 and above

CEDA DEFINE TRANSACTION(TCCB)
      GROUP(FSIVSHAR)
      PROGRAM(CSIVSC00)
      TASKDATAKEY(CICS) for CICS 3.3 and above

CEDA DEFINE TRANSACTION(TCCR)
      GROUP(FSIVSHAR)
      PROGRAM(CSIVSC06)
      TASKDATAKEY(CICS) for CICS 3.3 and above

CEDA DEFINE TRANSACTION(TCXI)
      GROUP(FSIVSHAR)
      PROGRAM(CSIVSCXI)
      TASKDATAKEY(CICS) for CICS 3.3 and above
```

The first PCT definition, TCC, is for the VSHARE online system.

The second definition, TCCB, is for the LU 6.2 program that communicates to the batch region. This transaction may be given any name desired. Any number of these batch tranids may be defined. By having each application within CICS region use a unique batch tranid, charge back systems can be used to charge the appropriate group for CICS resources used. Do not give security to this transaction or CICS will reject the connection request. VSHARE provides dataset security by providing the user the security exit module, CSIVSBSC. Refer to the section "File Security" on page 111 for more details.

By defining the batch tranid TCCB with a priority level of 0, CICS services the online programs before the batch programs, and the impact to the online programs is kept to a minimum.

The third definition, TCCR, is for accessing information about a remotely defined VSAM file within CICS. This transaction must be defined to all CICS regions owning remote files. For example, a batch job is connected to CICSPRD1 and it is accessing a VSAM file owned by a CICS region called CICSPRD2. The transaction TCCR must be defined to CICSPRD2 so that VSHARE can obtain information about the file. The remote file can be defined to use either ISC or MRO.

Installation

The fourth definition is for initializing the CMS interface in the CICS region. The transaction TCXI has to be executed in all the CICS regions using the CMS interface. Refer to the section “Installing the CICS CMS Component” on page 107.

PPT Definition

VSHARE has a series of programs and maps needing PPT definitions. The figure below shows both an Assembler and a COBOL definition. There are three columns listed below with the headings Assembler, COBOL, and BMS Maps. Use the first DEFINE format for the columns with the headings “Assembler”, and “BMS Maps.” Substitute each name from these columns for ‘name1’ in the PROGRAM parameter. For each name in the column with the heading “COBOL”, use the second DEFINE format. Substitute each name from the COBOL column for ‘name2’ in the PROGRAM parameter. In addition to defining these programs in the CICS region that is running VSHARE, the program CSIVSC06 needs to be defined in all regions owning remotely defined VSAM files that will be accessed using VSHARE.

CEDA	DEFINE PROGRAM(name1)	CEDA	DEFINE PROGRAM(name2)
	GROUP(FSIVSHAR)		GROUP(FSIVSHAR)
	LANGUAGE(ASSEMBLER)		LANGUAGE(COBOL)
	RELOAD(NO)		RELOAD(NO)
	RESIDENT(NO)		RESIDENT(NO)
	STATUS(ENABLE)		STATUS(ENABLE)
	EXECKEY(CICS) ^(See Note 1)		EXECKEY(CICS) ^(See Note 1)
	Assembler		COBOL
	CSIVSC00		CSIVSC10
	CSIVSC05		CSIVSC11
	CSIVSC06		CSIVSC12
	CSIVSCXI		CSIVSC13
	CSIVSCMO (letter ‘O’)		CSIVSC14
	CSIVSCDM		CSIVSC15
			CSIVSC16
			CSIVSC17
			CSIVSC18
			CSIVSC19
			BMS Maps
			CSIVSM1
			CSIVSM2
			CSIVSM3
			CSIVSM4
			CSIVSM5
			CSIVSM6
			CSIVSM7
			CSIVSM8
			CSIVSM9

Note 1: For CICS 3.3 and above

FCT Definition

VSHARE saves all of the system, job, and history information in a KSDS control dataset. This dataset needs to be defined with the FCT name CSIVSFL. The figure below contains the necessary definition.

Installation

```
CSIVSFL      DFHFCT TYPE=DATASET,          *
              DATASET=CSIVSFL,          *
              DSNAME=FSIVSHAR.CLUSTER    *
              ACCMETH=VSAM,              *
              SERVREQ=(READ,ADD,BROWSE,UPDATE,DELETE), *
              LSRPOOL=1,                  *
              RECFORM=(VARIABLE,BLOCKED), *
              STRNO=3,                     *
              FILSTAT=(ENABLE,CLOSED)
```

Define the Eligible CICS Regions and Jobs

The next step is to define all the CICS regions and job names eligible to execute VSHARE. This is accomplished through the VSHARE online system. Refer to the chapter “Online Control” on page 41.

The System Maintenance screen is used to define all the information needed by a batch job to communicate to the CICS region. Remember to define the CICS APPLID, not the CICS job name.

```
=====
APPLID:  UAPPLID   FSI-VSHARE - THE CICS CONNECTION   DATE:  02/01/2015
PROGID:  CSIVSC19  S Y S T E M   M A I N T E N A N C E   TIME:  06:13:50
=====
FUNCTION:  C ( A = ADD, C = CHANGE, D = DELETE )
CICS APPLID: P390CICS
-----
              SYSTEM INFORMATION                HISTORY INFORMATION
TRANID:  TCCB   ACCESS TYPE:  CMS              RECORD HISTORY:  Y
MODE NAME: EDITLU62  SYNC LEVEL:  005          GENERATIONS:    003
ROLLBACK: N      REQ/MIN:  000000             JOURNAL INFORMATION
APPLID SEL: N    SELECT MSK:  *****        JOURNAL UPDATES:  N
-----
DYNAMIC JOURNAL NAME:
//ITIJRNL  DD  DSN= PROD.FSIVSHAR.R30 .JOBNAME.ITIJRNL,DISP=MOD,
              DCB=(RECFM=VB,BLKSIZE= 27004 ,LRECL= 09000 ,BUFNO= 010 ),
              SPACE=(CYL,( 0010 , 0050 )),UNIT= SYSDA
-----
BATCH APPLIDS:  P390VSH1  P390VSH2
=====
ENTER=EDIT  CLEAR=EXIT  PF3=PROCESS  PF7=BACK  PF8=FORWARD  PF9=HOME
=====
```

Installation

The Job Maintenance screen, below, is used to define the eligible batch job names. Note that not all of the individual job names need defining. For example, if all job names starting with “FSI” are eligible to use VSHARE, only define a job name in the Job Maintenance screen with the name “FSI.” VSHARE then intercepts any job name starting with “FSI.”

```
=====
APPLID: P390CICS   FSI-VSHARE - THE CICS CONNECTION   DATE: 02/01/2015
PROGID: CSIVSC15   J O B   M A I N T E N A N C E     TIME: 06:37:18
=====
FUNCTION:  ( A = ADD, C = CHANGE, D = DELETE )

JOB NAME:

-----
JOB INFORMATION          | HISTORY INFORMATION    | PROCESSING INFORMATION
CICS REGION: P390CICS   | RECORD?: Y             | ABEND IF NO CICS?:
TRANID: TCCB           | GENERATIONS: 003       | DO INPUT FILES?:
SYNC LEVEL: 005        |                         | ABEND IF DISABLED?:
REQS/MINUTE: 00000     | JOURNAL INFORMATION    | TURN TRACE ON?:
ACCESS TYPE: CMS       | JOURNAL UPDATES?: N    | FORCE ROLLBACK?: N
-----

DYNAMIC JOURNAL NAME:
//ITIJRNL DD DSN= PROD.FSIVSHAR.R30 .             .ITIJRNL,DISP=MOD,
DCB=(RECFM=VB,BLKSIZE= 27004 ,LRECL= 09000 ,BUFNO= 010 ),
SPACE=(CYL,( 0010 , 0050 )),UNIT= SYSDA

=====
VSHARM15-002: ENTER THE DESIRED FUNCTION AND PRESS ENTER.
=====
```

Install the Cross Memory Services Interface

The Cross Memory Services (CMS) interface allows for the moving of data between the CICS region and the batch job directly via computer memory. This provides a faster communications vehicle than LU 6.2. The restriction with Cross Memory Services is that the batch job and the CICS region must be located in the same CPU. If the job and CICS are in different CPUs or different PR/SM images, LU 6.2 must be used as a communications device. Note that even if CMS is being used all the time, LU 6.2 is still used to start the VSHARE transaction in CICS. For this reason, the VTAM definitions, and the CICS connection and session definitions are still required.

There are two components to the CMS interface. Use the following steps to install each component.

Installation

Installing CMS Batch Component

1. Library Authorization

The installation process must be run from an APF authorized load library. All VSHARE modules that require APF authorization are in the product load library VSHARE.APF.LOAD. The modules from this dataset may be copied to an existing APF load library, or the product library itself may be given authorization.

2. SVC Selection

The CMS SVC routine can function either as the only module associated with an SVC or as a “front end” to an existing SVC routine. Thus, if there are few or no free SVC entries, one can be chosen that is already in use on the system, without impact to the existing routine. However, do not front-end a Type 1 or Type 6 SVC.

3. Dynamic SVC Installation

To install the CMS front end program, a job located in the installation library called INSTCMS needs to be run after each IPL. Execution JCL for the program and its parameters are as follows:

```
// < jobcard >
//*
//* PARMs: I  INSTALL,CMS,SVC#
//*         R  REMOVE
//*         E  ENABLE
//*         D  DISABLE
//*         S  STATUS
//*
//RUNUTINS EXEC PGM=BIMXMSUT,PARM=' I,CMS,219 '
//STEPLIB   DD DSN=VSHARE.APF.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
```

In the example above, CMS is being installed as SVC 219. The installation program and the other programs that it calls are in VSHARE.APF.LOAD.

The parameters are positional and must not include any extraneous blanks or unnecessary characters. The possible values for the parameters and their meanings are as follows:

Installation

Parm1: Request Type

- I— Install the CMS specified in parm2 using the SVC number in parm3
- R—Remove the CMS specified in parm2; the SVC number is not required
- E—Enable the CMS
- D—Disable the CMS
- S—Print the status of all systems including the installation history

Parm2: System Component

CMS—Cross Memory Services Communication

Parm3: SVC number

SVC#—The SVC number used for Cross Memory Services.

Installing the CICS CMS Component

Any region that uses the VSHARE CMS interface must be initialized. The CMS initialization is performed by the CICS program CSIVSCXI. CSIVSCXI can be executed in two ways. First, it may be executed through the CICS transaction TCXI. Refer to the PCT definitions for the TCXI requirements. The second way is to put the program CSIVSCXI in the startup PLT for execution. Following is an example of the PLT entry.

```
DFHPLT  TYPE=INITIAL, SUFFIX=XX
        .
DFHPLT  TYPE=ENTRY, PROGRAM=CSIVSCXI
        .
DFHPLT  TYPE=FINAL
END
```

Modify the Tier 2 File Recovery Procedure

In the installation library there is a member called RECOVER. This is the JCL necessary to back out all the file changes during Tier 2 recovery. If you decide not to implement Tier 2 recovery, skip this step. This JCL deck needs to be customized to the installation requirements. The JCL can also be converted into a procedure. Following is an explanation of the three step job.

- Step 1: This step deletes and defines an ESDS VSAM dataset. This ESDS dataset is used temporarily within this job. Change the name of the file to adhere to installation standards.
- Step 2: This step loads the sequential journal created during the batch job execution into the newly defined ESDS file.
- Step 3: This step backs out all updates based on the parameters given. Refer to the chapter “Recovery” on page 18 for details.

The VSAM Open/Close Routing Facility

Introduction The BIMVSR/MVS is a VSAM Open/Close routing facility that intercepts VSAM OPEN and CLOSE processing. This facility allows several CSI International products to intercept OPEN and/or CLOSE processing in a prescribed order. The routing facility is automatically started and stopped when the individual CSI International products are started or stopped.

At present, the following CSI International products require this routing facility:

- OVERRIDE-BUFF Dynamic VSAM Buffering
- RELIANT-JOURNAL VSAM Batch Journaling
- M-PACT-PACK Automatic VSAM File Compression
- VSHARE Multiple Region VSAM File Sharing

Installing the VSAM Open/Close Routing Facility Modules

The programs for VSHARE must be executed from an **APF Authorized Load Library**. This library must also be placed in the Linklist. This means that the library must be listed in *both* the members IEAAPFxx and LNKLISTxx in SYS1.PARMLIB.

You can download the routing facility from the Web (see “Installing VSHARE from a Web Download” on page 89) or receive it on a separate tape labeled “BIMVSR/MVS.” A cover letter with the tape provides more information.

To install the routing facility from tape, create and submit the following JCL:

```
//JOBID      JOB (accounting),'LOAD FSIVSR/MVS',
//           CLASS=A,MSGCLASS=*,MSGLEVEL=(1,1)
//IEBCOPY EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=*
//TAPEF1    DD DSN='BIMVSR.LOADLIB',UNIT=TAPE,DISP=OLD,
//           LABEL=(1,SL),VOL=(,RETAIN,SER=FSISYS)
//LOADLIB   DD DSN='authorized.linklist.library',DISP=SHR
//SYSIN     DD *
            COPY I=((TAPEF1,R)),O=LOADLIB
/*
//
```

Do not proceed to the next installation step until you confirm that the Routing Facility is installed by running the program BIMVSRMCM with parm='INIT'.

Installation

Removing the VSAM Open/Close Routing Facility

You only need to remove the router if you are experiencing problems with your system that are not corrected by de-activating the individual products. You can activate and deactivate the products that use the router, such as VSHARE, by using the interfaces provided with each product.

The following example contains the JCL required to remove the router from your system:

```
//BIMVSR          remove VSAM
//STEP1  EXEC  PGM=BIMVSRM, PARM='DELT01'
//FSILPA  DD  DSN=SYS1.LPALIB(IFG0192A), DISP=SHR
//SNAPDUMP DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
/*
//
```

The router command module (BIMVSRM) can perform several functions depending on the PARM value specified. Each value is described in the table below.

PARM=Value	Function
SHUT	Stops the router from dispatching any CSI products, but it leaves the router in your system.
INIT	Starts the router and resumes dispatching the CSI products. This function can be performed to reverse the SHUT function.
DELT01	Removes the router from your system and restores the original address pointers. This process “unhooks” any vender products that were started after the first CSI product that uses this router.
DELT02	Removes the router from your system and restores the original IBM address pointers. This process “unhooks” all vender products that “hook” the VSAM Open/Close process.

Installation

Routing Facility Messages The following messages are issued by the VSAM Open/Close Routing Facility.

VSAM Open/Close Routing Facility Messages	
FSIVSR-001	xxxxxxx MODULE NOT FOUND Explanation: The module "xxxxxxx" could not be found by the Router. User Action: Make sure that the library containing the program "xxxxxxx" is a linklist library.
FSIVSR-003	NO INIT PARMS (error description) User Action: Request technical assistance for this message.
FSIVSR-005	TABLE xxxxxxx NOT FOUND Explanation: The table "xxxxxxx" could not be found by the Router. User Action: Make sure that the library containing the table "xxxxxxx" is a Linklist library.
FSIVSR-006	FSIVSR NOT ACTIVE User Action: Request technical assistance for this message.
FSIVSR-008	PAGE ERROR (error description) User Action: Request technical assistance for this message.
FSIVSR-009	MODULE xxxxxxx FOR MOD yyyyyyy SUBROUTINE ERROR User Action: Request technical assistance for this message.
FSIVSR-010	GETMAIN ERROR (error description) User Action: Request technical assistance for this message.
FSIVSR-011	ROUTER ALREADY ACTIVE Explanation: The router has already been activated.
FSIVSR-013	ERROR RETURNED FROM SWAREQ (error description) User Action: Request technical assistance for this message.
FSIVSR-014	NO ACB AND TIOT MATCH User Action: Request technical assistance for this message.
FSIVSR-015	FREEMAIN ERROR (error description) User Action: Request technical assistance for this message.
FSIVSR-016	NO ENTRY IN TABLE FSIVSRTB FOR APPL User Action: Request technical assistance for this message.

VSHARE Activation and Deactivation

The installation library contains the members STRTVSB, STOPVSB, and STOPRTR. The member STRTVSB activates VSHARE, and the member STOPVSB stops VSHARE.

You must run the job STRTVSB for each Control Dataset created, with the DD CSIVSFL pointing to the Control Dataset. Refer to section “Update the CSIVSBDS Parameter Member” on page 95 for more information.

If you make changes to the Control Dataset after VSHARE has been started, you will need to run job REFRVSB to update an in-core table.

To activate VSHARE for a particular job, a STEPLIB must be included in the JCL pointing to the library VSHARE.LOAD.

You can deactivate VSHARE in three ways:

- The first way: remove the STEPLIB from the JCL.
- The second way: remove the job name from the VSHARE system. Refer to the chapter “Online Control” on page 41 for details on defining and deleting job names from VSHARE.
- The third way: run the STOPVSB job. This job simply stops VSHARE from being invoked.

Running the STOPRTR job is *not* recommended. This job resets pointers and may stop other vendor products from running if they were started after VSHARE.

File Deactivation

VSHARE processing may be excluded for specific files within a job by coding an expiration date operand of EXPDT=1993/333 on their MVS DD statements.

File deactivation should be used only if there is certainty that the file is not being updated by CICS during the excluded batch run.

File Security

During the opening of a dataset, VSHARE will call module CSIVSBSC. CSIVSBSC issues a RACROUTE macro to determine whether the job has proper authority to access the dataset. This module will have the dataset name, the type of access, and the volume the dataset resides on. There is an example of this source in the install library that may be modified by the installation to adhere to their security needs. When modified, the link-edited module must be placed in your VSHARE execution library. CSIVSBSC needs to be modified only if there is a problem with the existing security package.

Messages and Codes

FSIVSBAI Messages

FSIVSBAI01 FSI-VSHARE NOT ACTIVE, DIRECT CALLS WILL BE IGNORED.

Explanation: During the batch run, an explicit syncpoint or rollback was requested but VSHARE determined that the job was running in stand-alone mode. This message informs the user that any syncpoint or rollback request will be ignored because the I/O is not being managed by CICS.

System Action: Informational message only and processing continues.

User Response: None.

Online Maintenance Messages

VSHARM1n-001 COMMAREA LENGTH IS INVALID.

Explanation: VSHARE online programs received a COMMAREA length error.

System Action: Processing stops.

User Response: This is an internal VSHARE message. Contact CSI technical support.

VSHARM1n-002 ENTER THE DESIRED SEARCH NAME AND PRESS ENTER.

Explanation: The search name for the screen being executed was not entered.

System Action: Screen is re-displayed and the program waits for a name to be entered.

User Response: Enter desired search criteria. Refer to the chapter “Online Control” on page 41 for details.

VSHARM1n-003 INVALID FUNCTION WAS ENTERED. ENTER AN A, D, OR C ONLY.

Explanation: During system maintenance or job maintenance, an invalid function code was entered.

System Action: Message is display and field is highlighted.

User Response: Enter a correct function letter. Refer to the chapter “Online Control” on page 41 for details on the field.

VSHARM1n-004 INVALID PF KEY. MAKE SELECTION AND PRESS ENTER.

Explanation: A PF key was entered that is not used by the online program.

System Action: Re-display the screen with the message.

User Response: Enter proper PF key. Refer to the chapter “Online Control” on page 41 for details on the proper attention keys.

VSHARM1n-005 PROGRAM <progid> WAS NOT FOUND. CONTACT SYSTEMS PROGRAMMER.

Explanation: VSHARE program PROGID was not available when another VSHARE program attempted a link.

System Action: Processing stops.

User Response: Contact CICS system programmer. The CICS definition for this program may not be completed, or VSHARE library may not be available to CICS.

VSHARM1n-006 MAP FAILURE IN PROGRAM <progid>.

Explanation: VSHARE online program PROGID received a map failure response from CICS.

System Action: Processing stops.

User Response: Contact a CICS systems programmer. The CICS definition for the map may not be completed, or the VSHARE library may not be available to CICS.

VSHARM1n-007 PRESS PF3 TO PROCESS THE ADDITION.

Explanation: Editing was successful for the addition of the job or system. By pressing PF3 the job or system will be added to the control dataset.

System Action: Informational only.

User Response: If you want to add the job or system, press PF3. Refer to the chapter “Online Control” on page 41 for details.

VSHARM1n-008 PRESS PF3 TO PROCESS THE MODIFICATION.

Explanation: Editing was successful for a modification to a job or system definition. By pressing PF3 the job or system will be modified with the changes entered.

System Action: Informational only.

User Response: If you want to modify the job or system definition, press PF3. Refer to the chapter “Online Control” on page 41 for details.

VSHARM1n-009 PRESS PF3 TO PROCESS THE DELETION.

Explanation: Editing was successful for the deletion of the job or system entered. By pressing PF3 the job or system will be deleted from the control dataset.

System Action: Informational only.

User Response: If you want to delete the job or system, press PF3. Refer to the chapter “Online Control” on page 41 for details.

VSHARM1n-010 ENTER REQUIRED FIELD.

Explanation: A required field was not entered during a system or job definition.

System Action: Information only. The definition will not be processed until the value for the highlighted field is entered.

User Response: Enter the required field. Refer to the chapter “Online Control” on page 41 for details on the field.

VSHARM1n-011 FIELD MUST BE NUMERIC.

Explanation: The highlighted field must contain a numeric value.

System Action: Informational only. The definition will not be processed until the value is numeric.

User Response: Enter a numeric value. Refer to the chapter “Online Control” on page 41 for details on the field in question.

VSHARM1n-012 THE INDICATOR MUST BE A Y OR N ONLY.

Explanation: The field highlighted must be either a Y or an N.

System Action: Informational only. The definition will not be processed until the value is correct.

User Response: Enter a Y or N. Refer to the chapter “Online Control” on page 41 for details on the field in question.

VSHARM1n-013 THIS FUNCTION KEY IS NOT ACTIVE.

Explanation: A PF key was entered that is not used by the online program.

System Action: Re-display the screen with the message.

User Response: Enter proper PF key. Refer to the chapter “Online Control” on page 41 for details on proper attention keys.

Messages and Codes

- VSHARM1n-014 THE CONTROL DATASET <fctname> IS NOT DEFINED TO CICS.**
Explanation: A VSHARE online program received a DSIDERR from CICS while accessing VSHARE control dataset.
System Action: Processing stops.
User Response: Contact CICS systems programmer. The CICS definition for the dataset may not be completed.
- VSHARM1n-015 THE CONTROL DATASET <fctname> IS DISABLED.**
Explanation: A VSHARE online program received a DISABLED response from CICS while accessing VSHARE control dataset.
System Action: Processing stops.
User Response: Determine why the dataset is disabled and enable the VSHARE control dataset.
- VSHARM1n-016 I/O ERROR WAS DETECTED WHILE READING CONTROL FILE.**
Explanation: An I/O error was detected by VSHARE online screens while accessing the control dataset.
System Action: Processing stops.
User Response: Determine the cause of the control dataset I/O error.
- VSHARM1n-017 NOT AUTHORIZED TO ACCESS THE CONTROL DATASET.**
Explanation: The user executing VSHARE online programs does not have proper authority to access the control dataset.
System Action: Processing stops.
User Response: Contact security administration in order to get proper authority.
- VSHARM1n-018 THE CONTROL DATASET <fctname> IS NOT OPENED.**
Explanation: VSHARE online programs received a not open response from CICS when accessing the control dataset.
System Action: Processing stops.
User Response: Determine the reason why the control dataset is closed and open it.
- VSHARM1n-019 THE <JOB>|<SYSTEM>|<HISTORY> RECORD FOR THIS <JOB>|<SYSTEM>|<HISTORY> WAS NOT FOUND.**
Explanation: If defining a job, a CICS region name was entered that was not previously defined. If reviewing history, a job name was entered that had no corresponding history.
System Action: Screen is re-displayed and the program waits for new data.
User Response: When defining a job, make sure the CICS name was entered properly or that the region was previously defined. If reviewing history, there is nothing required.
- VSHARM1n-020 THIS <JOB>|<SYSTEM> IS ALREADY DEFINED IN THE CONTROL FILE.**
Explanation: An attempt was made to add a job or CICS region previously defined to VSHARE.
System Action: Re-display the screen with the message.
User Response: Make sure the job or CICS region name is correct or look at the current definition for the job/CICS region.

Messages and Codes

- VSHARM1n-021 A VSAM LENGERR HAS BEEN DETECTED.**
Explanation: A VSHARE program received a LENGERR from CICS while accessing the control dataset.
System Action: Processing stops.
User Response: This is an internal error. Contact CSI technical support.
- VSHARM1n-022 THE CONTROL FILE HAS RUN OUT OF SPACE.**
Explanation: A VSHARE program received an out of space condition from CICS when writing to the control file.
System Action: Processing stops.
User Response: The control file needs to be defined larger.
- VSHARM1n-023 THE <JOB>|<CICS REGION> WAS SUCCESSFULLY ADDED TO THE CONTROL FILE.**
Explanation: The job or CICS region entered was successfully added to VSHARE.
System Action: Informational only.
User Response: None
- VSHARM1n-024 THE <JOB>|<CICS REGION> DATA WAS SUCCESSFULLY MODIFIED.**
Explanation: The job or CICS region being modified was successfully processed.
System Action: Informational only.
User Response: None
- VSHARM1n-025 THE <JOB>|<CICS REGION> WAS SUCCESSFULLY DELETED.**
Explanation: While performing system maintenance, the job or region definition selected was successfully deleted.
System Action: Informational only.
User Response: None.
- VSHARM1n-026 THE CICS REGION NAME ENTERED IS NOT DEFINED.**
Explanation: The region name entered while defining a job was not previously defined to VSHARE.
System Action: Informational only. The job will not be added until a valid region is entered.
User Response: Make sure the name is spelled correctly or add the system to VSHARE through the System Maintenance Screen.
- VSHARM1n-027 THE BLOCK SIZE IS INCORRECT BASED ON THE LRECL.**
Explanation: The block size chosen for the journal file is not correct.
System Action: Informational only. The new definition of the journal file will not be updated without a valid block size.
User Response: Enter a value that is a multiple of the record size entered plus 4 bytes.
- VSHARM1n-029 INVALID ACCESS TYPE. ENTER CMS OR LU62 ONLY.**
Explanation: The access type entered is not valid.
System Action: Informational only. A valid type must be entered before the definition is applied.
User Response: Enter LU62 or CMS. Refer to the chapter “Online Control” on page 41 for details on these types.

Messages and Codes

- VSHARM1n-031 I/O ERROR WAS DETECTED WHILE READING TEMP STORAGE.**
Explanation: VSHARE received an I/O error response from CICS after accessing temporary storage.
System Action: Processing stops.
User Response: Contact system programmers. There is something wrong with the CICS temporary storage.
- VSHARM1n-032 LENGTH ERROR WAS DETECTED WHILE READING TEMP STORAGE.**
Explanation: VSHARE received a length error response from CICS after accessing temporary storage.
System Action: Processing stops.
User Response: This is an internal VSHARE error. Contact CSI technical support.
- VSHARM1n-033 QID ERROR WAS DETECTED WHILE READING TEMP STORAGE.**
Explanation: VSHARE received a QID error from CICS after accessing temporary storage.
System Action: Processing stops.
User Response: This is an internal VSHARE error. Contact CSI technical support.
- VSHARM1n-034 NOT AUTHORIZED TO ACCESS TEMPORARY STORAGE.**
Explanation: The person executing VSHARE online programs does not have proper authorization to read temporary storage.
System Action: Processing stops.
User Response: Contact the CICS systems programmer to find out why this is the case.
- VSHARM1n-035 THE REQUEST TO STOP THIS JOB HAS BEEN ISSUED.**
Explanation: A request was given in the active job screen to stop a job.
System Action: Informational only.
User Response: None. Refer to the chapter “Online Control” on page 41 for details.
- VSHARM1n-036 THE REQUEST TO CANCEL THIS JOB HAS BEEN ISSUED.**
Explanation: A request was given in the active job screen to cancel a job.
System Action: Informational only.
User Response: None. Refer to the chapter “Online Control” on page 41 for details.
- VSHARM1n-037 THE REQUEST TO RESTART THIS JOB HAS BEEN ISSUED.**
Explanation: A request was given in the active job screen to restart a previously stopped job.
System Action: Informational only.
User Response: None. Refer to the chapter “Online Control” on page 41 for details.
- VSHARM1n-038 THIS JOB IS NO LONGER ACTIVE.**
Explanation: A job was being analyzed in VSHARE active job process and the job completed.
System Action: The information about the job is no longer available. A screen with no information is displayed.
User Response: None. Refer to the chapter “Online Control” on page 41 for details.

Messages and Codes

- VSHARM1n-039 THE REQUEST RATE HAS BEEN SUCCESSFULLY CHANGED.**
Explanation: A request to change the request rate for a job was processed successfully.
System Action: Information only.
User Response: None. Refer to the chapter “Online Control” on page 41 for details.
- VSHARM1n-040 THE REQUEST RATE MUST BE NUMERIC.**
Explanation: A non-numeric request rate was entered.
System Action: Highlight the field and re-display the screen.
User Response: Enter a numeric value.
- VSHARM1n-042 NOT AUTHORIZED TO INQUIRE ON FILE CONTROL TABLE.**
Explanation: While trying to display a dataset statistics screen, VSHARE received a not authorized response from CICS while inquiring against the FCT.
System Action: Processing stops.
User Response: Contact the system programmers and find out why your user ID does not have authorization to inquire against the CICS FCT.
- VSHARM1n-043 THIS FCT NAME IS NOT IN THE CICS TABLES.**
Explanation: While trying to display a dataset statistics screen in history, VSHARE could not find the FCT of the file written in history.
System Action: Processing stops.
User Response: None. The FCT entry was removed since the jobs execution as indicated in VSHARE history record.
- VSHARM1n-044 AN INVALID REQUEST RETURNED AFTER A TS READ.**
Explanation: VSHARE received an invalid response from CICS when reading its temporary storage record.
System Action: Processing stops.
User Response: This may be caused by a storage violation. Contact system programmers. If the problem persists, contact CSI technical support.
- VSHARM1n-045 FSI-VSHARE HAS COMPLETED.**
Explanation: The VSHARE online system completed successfully.
System Action: This is an informational message. No error occurred.
User Response: None.

Batch Messages

FSIVSHAR-100 NOT ENOUGH PCBS DEFINED.

Explanation: A module requested a PCB and found they were all used.

System Action: Program aborts.

User Response: Contact CSI technical support.

FSIVSHAR-101 MODULE XXXXXXXX NOT FOUND.

Explanation: Module xxxxxxxx was requested and could not be found.

System Action: Program aborts.

User Response: Make sure proper STEPLIB DD cards are in place, and rerun the job. When the module not found is BIMVSBTB, make sure that the dataset name for the TCCVSCCTL control file, as assembled in BIMVSBDS and linked into the batch VSHARE load library, matches the DD card dataset name used by CICS to define the control dataset.

FSIVSHAR-102 DDNAME NOT FOUND ON DST CHAIN.

Explanation: A control block for a DD card was lost.

System Action: Program aborts.

User Response: Try rerunning the job. If the error occurs again, gather dumps and contact CSI technical support.

FSIVSHAR-103 ERROR RETURNED FROM XXXXXXXX.

Explanation: A program xxxxxxxx returned control back to the caller with a nonzero return code.

System Action: Dependent on circumstance.

User Response: Look for other pertaining messages and follow their instructions if any. If program abends, try running again. If the error occurs again call CSI technical support with dumps.

FSIVSHAR-104 ERROR INITIALIZING JOURNALING.

Explanation: Journal initialization encountered a problem.

System Action: Dependent on problem.

User Response: Refer to other related messages.

FSIVSHAR-105 ESTAE ERROR.

Explanation: The ESTAE program encountered an error.

System Action: Program aborts.

User Response: Contact CSI technical support.

FSIVSHAR-106 NO PARMS SPECIFIED ON EXECUTE CARD.

Explanation: A program was executed that required parms passed, but none were.

System Action: Program aborts.

User Response: Review the section in the manual that deals with this function.

Messages and Codes

FSIVSHAR-107 GETMAIN ERROR.

Explanation: A program issued a storage GETMAIN and MVS return an error.

System Action: Program aborts.

User Response: Refer to the IBM message relating to this. If none, contact CSI technical support.

FSIVSHAR-108 INVALID PARM SPECIFIED.

Explanation: A program was passed a parm that it did not recognize.

System Action: Program aborts.

User Response: Review the section in the manual that deals with this function.

FSIVSHAR-109 FREEMAIN FAILED.

Explanation: A program issued a storage FREEMAIN and MVS returned an error.

System Action: Dependent.

User Response: If program aborts, call CSI technical support. If program ends normal, ignore.

FSIVSHAR-110 TABLE NOT REFRESHED, NOT LOADED.

Explanation: A request was issued to refresh the control table and it was not yet loaded.

System Action: Program returns a bad return code.

User Response: Run the job to load the table.

FSIVSHAR-111 TABLE ALREADY LOADED.

Explanation: A request was issued to load the control table when it was already loaded.

System Action: Program returns a bad return code.

User Response: Ignore if table was already loaded, or run Refresh job.

FSIVSHAR-112 CDE LOAD ERROR.

Explanation: The program encountered an error trying to add a entry to the CDE chain.

System Action: Program returns a bad return code.

User Response: Try rerunning the program. If the error occurs again call CSI technical support.

FSIVSHAR-113 CDE UNLOAD ERROR.

Explanation: The program encountered an error trying to remove an entry to the CDE chain.

System Action: Program returns a bad return code.

User Response: Try rerunning the program. If the error occurs again call CSI technical support.

FSIVSHAR-114 SUBPOOL NOT FOUND.

Explanation: A program requested a storage function of a particular subpool, and the storage was not found there.

System Action: Programs returns a bad return code.

User Response: Try rerunning the program. If the error occurs again call CSI technical support.

FSIVSHAR-115 DEQ INVOKED RETURN=XXXXXXXX.

Explanation: A program called the DEQ module, and received a nonzero return code.

System Action: Program aborts.

User Response: Try rerunning the program. If the error occurs again call CSI technical support.

Messages and Codes

- FSIVSHAR-116 INTERNAL ERROR: (DESCRIPTION).**
Explanation: An internal error occurred.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-117 UPDATE HAS/HAS NOT OCCURED - RECOVERY REQUIRED/NOT REQUIRED ON ABEND.**
Explanation: Describes whether an update has occurred, and as a result whether recovery will be required if an abend occurs.
System Action: None.
User Response: None.
- FSIVSHAR-118 MODULE CSIVSBSG FOUND CMCB ADDRESS 0.**
Explanation: The storage module was found but the CMCB address was 0.
System Action: None.
User Response: None.
- FSIVSHAR-119 ENQ FAILED RETURN=XXXXXXXX.**
Explanation: An enqueue function failed.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-120 CSIVSBER FORCING ABEND.**
Explanation: The error handling module forced an abend.
System Action: Program aborts.
User Response: Refer to other related messages.
- FSIVSHAR-121 LOAD ERROR IN MODULE CSIVSBLD FOR MOD XXXXXXXX.**
Explanation: CSIVSBLD program encountered an error loading module xxxxxxxx.
System Action: Program aborts.
User Response: Refer to other related messages.
- FSIVSHAR-122 ZERO LENGTH SPECIFIED ON MSG REQUEST.**
Explanation: A request was made to the message handler with a zero length message.
System Action: None.
User Action: Contact CSI technical support.
- FSIVSHAR-123 ERROR RETURN FOR APPLID XXXXXXXX PROCESSING CONTINUES.**
Explanation: A error was returned for the APPLID xxxxxxxx, but processing continued.
System Action: None.
User Response: None.
- FSIVSHAR-124 NO JOURNAL REC AFTER JOB REC.**
Explanation: Journaling was requested for this job but no record was found in the control storage area.
System Action: Program aborts.
User Response: Run the job to refresh the control storage area, then rerun the job.

Messages and Codes

- FSIVSHAR-125 NO JOB/CICS RECORD FOUND IN CSIVSBTB.**
Explanation: Either a Job or CICS record was missing in the control storage area.
System Action: Program aborts.
User Response: Run the job to refresh the control storage area, then rerun the job.
- FSIVSHAR-126 NOT ABLE TO SET EOJ JOB HOOK, PROCESSING IS ABENDED.**
Explanation: The program was not able to set the end of job routine address.
System Action: Program aborts.
User Response: Rerun the job. If the error occurs again call CSI technical support.
- FSIVSHAR-127 NOT ABLE TO ACTIV/ALLOC/SEND BATCH APPL, PROCESSING IS ABENDED.**
Explanation: The program encountered an error with the batch APPLID.
System Action: Program aborts.
User Response: Refer to other related messages.
- FSIVSHAR-128 NOT ABLE TO ACTIV/ALLOC/SEND BATCH APPL.**
Explanation: The program encountered an error with the batch APPLID.
System Action: None.
User Response: None.
- FSIVSHAR-129 ERROR FOR MOD XXXXXXXX RTC WAS XXXXXXXX RSN XXXXXXXX.**
Explanation: A module encountered an error, and the return and reason codes were returned.
System Action: Dependent.
User Response: Refer to other related messages.
- FSIVSHAR-130 DSN INPUT ONLY.**
Explanation: The open was intercepted, and the data set was open for input only.
System Action: None.
User Response: None.
- FSIVSHAR-131 XXXXXXXX DDNAME DISABLED.**
Explanation: xxxxxxxx DDNAME was found to be disabled in CICS.
System Action: Program aborts.
User Response: Enable the file in CICS and rerun the job.
- FSIVSHAR-132 XXXXXXXX NOT INTERCEPTED.**
Explanation: xxxxxxxx DDNAME was not intercepted.
System Action: None.
User Response: None.
- FSIVSHAR-133 BLDL FAILED FOR MOD XXXXXXXX.**
Explanation: The program issued a BLDL for mod xxxxxxxx and it failed.
System Action: Program aborts.
User Response: Ensure the proper JOBLIB/STEPLIB is included.

Messages and Codes

- FSIVSHAR-134 CMCB ADDRESS NOT FOUND.**
Explanation: The program was looking for the CMCB control block and could not find it.
System Action: Program aborts.
User Response: Call CSI technical support.
- FSIVSHAR-135 ACB DID NOT OPEN.**
Explanation: The program checked the ACB after open, and it was not open.
System Action: Program aborts.
User Response: Check the IBM messages to see why the file did not open.
- FSIVSHAR-136 FSI-VSHARE I/O ROUTINE NOT LOADED.**
Explanation: The program got the address of the VSHARE VSAM I/O routine and it was zero.
System Action: Program aborts.
User Response: Call CSI technical support.
- FSIVSHAR-137 ERROR TYPE IN VSAM RECORD.**
Explanation: The program encountered an invalid record type in the control file.
System Action: Program aborts.
User Response: Call CSI technical support.
- FSIVSHAR-138 INVALID RECORD IN CONTROL FILE.**
Explanation: The program found an invalid record in the control file.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-139 SWA ERROR INCOUNTERED.**
Explanation: The program tried to access the system work area and encountered an error.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-140 VSAM ERROR.**
Explanation: The program tried to execute a VSAM function and failed.
System Action: Program aborts.
User Response: Refer to other related messages.
- FSIVSHAR-141 SEND/RECEIVE FAILED - RETRY BEING ATTEMPTED.**
Explanation: The program issued a send or receive and failed.
System Action: Program retries.
User Response: None.
- FSIVSHAR-142 SEND/RECEIVE FAILED.**
Explanation: The program issued a send or receive and failed.
System Action: Program aborts.
User Response: Contact CSI technical support.

Messages and Codes

- FSIVSHAR-143 EXLST ERROR - LIST ADDR XXXXXXXX.**
Explanation: VSAM encountered an error with the EXLST at address xxxxxxxx.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-144 REQUEST ERROR - REQUEST XXXXXXXX.**
Explanation: An internal invalid request was issued at address xxxxxxxx.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-145 KEY LENGTH ZERO.**
Explanation: A key length of zero was found for the file.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-146 SUBROUTINE ERROR (SUBROUTINE).**
Explanation: An error was encountered in the subroutine.
System Action: Program aborts.
User Response: Contact CSI technical support.
- FSIVSHAR-147 RACF ERROR RETURN - XXXXXXXX REASON - XXXXXXXX.**
Explanation: The security check returned a return and reason code.
System Action: Program aborts.
User Response: Make sure access is permitted to the data set.
- FSIVSHAR-148 RECEIVE ERROR RETURNED.**
Explanation: The communication program got an error during receive.
System Action: Program aborts.
User Response: Refer to other related messages.
- FSIVSHAR-149 INVALID REQUEST CODE IN USER CALL.**
Explanation: The user program made an invalid direct call to VSHARE.
System Action: The program aborts.
User Response: Correct the request and rerun.
- FSIVSHAR-150 FSIVSHARE NOT ACTIVE, DIRECT CALLS WILL BE IGNORED.**
Explanation: The user program made a direct call to VSHARE without it being active.
System Action: None.
User Response: Either activate VSHARE or ignore the message.
- FSIVSHAR-154 USER EXCLUDED DATASET**
Explanation: DD specified 99/333 for expiration date.
System Action: Dataset is excluded from VSHARE.
User Response: None.

Messages and Codes

FSIVSHAR-199 (Informational Messages).

Explanation: These are informational messages only.

System Action: None.

User Response: None.

Journal Messages

FSIVSHAR-200 **WARNING - UPDATE IN PROGRESS INDICATED AT TERMINATION BUT I/O TYPE WAS NOT DELETE, ADD, OR REWRITE.**

Explanation: The batch program indicated that an I/O was in progress at the time of termination. However, the I/O operation was not one that would have changed the user file.

System Action: No journal record will be created for the I/O in progress at termination. This is only a warning and termination processing will continue normally.

User Response: None. If the problem persists, contact CSI technical support.

FSIVSHAR-201 **RECOVERY PARAMETER VALUES FOR THIS STEP ARE: JOBNAME=xxxxxxx
JOBNUM=xxxx PGMSTEP=xxxxxxx PROCSTEP=xxxxxxx.**

Explanation: Some or all of the parameters listed could be used as input to the recovery job to back out changes up to the beginning of this step. Note that PGMSTEP is the name of the step containing the EXEC PGM= statement. PROCSTEP is the name of the step containing the EXEC PROC statement, if used.

System Action: Informational only.

User Response: None

FSIVSHAR-202 **JOURNAL MODULE CALLED WITH INVALID OPERATION CODE.**

Explanation: The journaling module was called by the batch program with an invalid operation code in its parameter list. The parameter list storage may have been corrupted.

System Action: Return code set and control returned to the batch program.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-203 **JOURNAL FILE OPEN REQUEST FAILED.**

Explanation: This module called the journal I/O module to open the journal file but received a nonzero return code. The job log may show further messages.

System Action: Set return code and return to caller.

User Response: Review the job log for related messages. Review the journal information online to ensure that journal file definitions are correct. Also, try browsing the journal file or displaying its attributes using some other means (for instance with TSO) to ensure that it was created properly. Review the system message log for the allocated dataset name. If the problem persists, contact CSI technical support.

FSIVSHAR-204 **JOURNAL FILE WRITE REQUEST FAILED.**

Explanation: This module called the journal I/O module to write to the journal file but received a nonzero return code. The job log may show further messages.

System Action: Set return code and return to caller.

User Response: See instructions under open request failure above. In addition, ensure that enough space was allocated for the file. If not, change the journal space allocation online. Note that if records have already been written to the journal file, it may be necessary to perform recovery before rerunning the application. Refer to the chapter "Recovery" on page 18 for more information. If the problem persists, contact CSI technical support.

FSIVSHAR-205 JOURNAL FILE CLOSE REQUEST FAILED.

Explanation: This module called the journal I/O module to close the sequential journal file but received a nonzero return code. The job log may show further messages.

System Action: Set return code and return to caller.

User Response: See instructions under open request failure above. If the problem persists, contact CSI technical support.

FSIVSHAR-206 REQUEST FOR SYSTEM TIME FAILED.

Explanation: This module attempted to get the current time and date from the operating system via the TIME macro. The request failed as indicated by a nonzero return code in register 15.

System Action: Abend with user 999.

User Response: Rerun application after performing recovery functions if necessary. If the problem persists, contact CSI technical support.

FSIVSHAR-207 DD=<ddname> NOT FOUND IN TIOT.

Explanation: While processing a journal request, this module attempted to look up the DDNAME in the Task I/O Table in order to locate the corresponding dataset name. However, the supplied DDNAME could not be located. The internal control block containing the DDNAME may have been corrupted.

System Action: Abend with user 999.

User Response: Rerun application after performing recovery functions if necessary. If the problem persists, contact CSI technical support.

FSIVSHAR-208 FOUND INVALID JFCB FOR TIOT DD=<ddname>.

Explanation: While processing a journal request, this module located the appropriate DDNAME in the Task I/O Table however the corresponding Job File Control Block contained an invalid control block identifier. Storage may have been corrupted.

System Action: Abend with user 999.

User Response: Rerun application after performing recovery functions if necessary. If the problem persists, contact CSI technical support.

FSIVSHAR-210 OPENING JOURNAL DATASET <dsname>.

Explanation: This message is issued during recovery processing by the journal I/O module just prior to opening the journal dataset.

System Action: Informational message only.

User Response: None.

FSIVSHAR-211 ALLOCATING <OLD/NEW> JOURNAL <dsname>.

Explanation: This message is issued by the journal I/O module just prior to dynamically allocating the journal dataset. 'OLD' indicates that the journal dataset is already cataloged and new records will be appended to the end of the dataset. 'NEW' indicates that the journal dataset was not cataloged and will be created in the current step.

System Action: Informational message only.

User Response: None.

Messages and Codes

FSIVSHAR-212 I/O MODULE CALLED WITH INVALID OPERATION CODE.

Explanation: The parameters received by this module contained an invalid operation code. Parameter list storage may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-213 JOURNAL FILE <OPEN/CLOSE> <FAILURE/WARNING> R15=nn ACBERROR=nn.

Explanation: During recovery processing, the journal I/O module opened the VSAM journal file for input and encountered a nonzero return code. The hex value of register 15 is shown as well as the value of the ERROR field from the ACB. Specific information about the cause of the warning or failure can be found in the OS/VS VSAM Programmer's Guide.

System Action: Failure will cause the task to abend with a user 999. Warnings will not interrupt processing.

User Response: Warnings may require no action. For instance, the dataset may have been implicitly verified on open. For failures, make sure that the VSAM journal dataset was defined properly when the QSAM journal file was copied to it. If the problem persists, contact CSI technical support.

FSIVSHAR-214 JOURNAL FILE <READ/POINT> FAILED - R15= nn FDBK= nn FTNCD= nn.

Explanation: The journal I/O module issued either a read or a point request for the VSAM recovery dataset and the request failed. The message includes hex values from the feedback field of the RPL used. Specific information about these fields can be found in the OS/VS VSAM Programmer's Guide.

System Action: Set return code and return to caller.

User Response: Check for related messages in the job log. Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. If the problem persists, contact CSI technical support.

FSIVSHAR-215 CATALOG ERROR TRYING TO LOCATE JOURNAL FILE - R15=nn.

Explanation: This module tried to locate the QSAM journal file dataset name in the catalog but the request failed. The return code in register 15 from the LOCATE macro was 'nn' (hex).

System Action: Set return code and return to caller.

User Response: Check to see if the QSAM journal file is defined and listed in the catalog. Refer to the chapter "Recovery" on page 18 for journal dataset naming. If the problem persists, contact CSI technical support.

FSIVSHAR-216 JOURNAL ALLOCATION ERROR - R15=nn INFO=nnnn REASON=nnnn.

Explanation: Dynamic allocation of the journal file failed. Register 15, information and reason codes were returned from the dynamic allocation function (in hex). Note that if the allocation module failed for a reason unrelated to the dynamic allocation call, the R15, INFO and REASON fields will be irrelevant.

System Action: Set return code and return to caller.

User Response: Check for messages from the allocation routine, CSIVSJ08. Review the journal information online to ensure that journal file definitions are correct. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-217 JOURNAL FILE (DD=JNLIN) NOT FOUND IN JCL.

Explanation: The recovery job was being run but no journal file (DD=JNLIN) was specified in the JCL.

System Action: Abend with user 999.

User Response: Ensure that the 'JNLIN' DDNAME is included in the recovery JCL for the journal dataset being recovered from. If the problem persists, contact CSI technical support.

FSIVSHAR-218 RDJFCB MACRO CALL FAILED FOR DD = <ddname>.

Explanation: This message occurs during recovery processing. The DDNAME entry (DD=JNLIN) was located in the TIOT but an error was encountered when accessing the corresponding Job File Control Block entry via the RDJFCB macro.

System Action: Abend with user 999.

User Response: Rerun recovery process. If the problem persists, contact CSI technical support.

FSIVSHAR-219 STORAGE MANAGER CALLED WITH INVALID OPERATION CODE.

Explanation: The parameters received by this module contained an invalid operation code. Parameter list storage may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-220 INVALID MIN STORAGE REQUESTED - LESS THAN OR EQUAL ZERO.

Explanation: The minimum storage request field of the parameter list was invalid. Parameter list storage may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-221 INVALID MAX STORAGE REQUESTED - LESS THAN OR EQUAL ZERO.

Explanation: The maximum storage request field of the parameter list was invalid. Parameter list storage may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-222 INVALID STORAGE REQUESTED - MINIMUM GREATER THAN MAXIMUM.

Explanation: The minimum storage request field parameter was greater than the maximum. This would result in an invalid GETMAIN request. Parameter list storage may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-223 GETMAIN FAILED - INSUFFICIENT VIRTUAL STORAGE.

Explanation: Not enough virtual storage was available for the storage manager to create its request buffer.

System Action: Set return code and return to caller.

User Response: Increase storage available to recovery job using the REGION parameter on either the JOB card or EXEC card.

Messages and Codes

FSIVSHAR-224 FREEMAIN FAILED - NOT ALL STORAGE RELEASED.

Explanation: This module received a request to release the request buffer, but the storage FREEMAIN failed.

System Action: Set return code and return to caller.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-225 ELEMENT SIZE EXCEEDS TOTAL BUFFER SIZE.

Explanation: The storage manager was asked to store an element that was larger than the total buffer acquired when the request queue was created. The buffer size field or the element size field may have been corrupted.

System Action: Set return code and return to caller.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-226 RECOVERY MANAGER HAS COMPLETED <normally/abnormally>.

Explanation: The recovery manager finished processing all records in the VSAM recovery dataset.

System Action: Return control to operating system.

User Response: Informational message only. If abnormally was indicated, review prior messages in the log for error conditions.

FSIVSHAR-227 RECOVERY MANAGER WAITING - REQUEST BUFFER IS FULL.

Explanation: The recovery manager tried to enqueue a request block but the storage manager indicated that the buffer was full.

System Action: When wait is finished, recovery manager tries again to enqueue the request block.

User Response: Informational message only.

FSIVSHAR-228 <attaching/detaching> I/O SUBTASK.

Explanation: The I/O subtask is about to be attached or detached by the recovery manager.

System Action: Informational only.

User Response: None.

FSIVSHAR-229 RID - DD=<ddname> DATE=<yyddd> TIME=<hh:mm:ss:ht> SEQ=<seq#>.

Explanation: This message is used to supplement other messages where it is useful to identify the journal file record being processed. The DDNAME identifies the file for which the update was recorded. The Julian date and time when the journal record was created as well an internal sequence number are included. The journal record layout included in the Recovery chapter shows the location of each of the fields.

System Action: Informational message only.

User Response: See message prior to this one in log for more information.

Messages and Codes

- FSIVSHAR-230 I/O SUBTASK ENDED <NORMALLY/ABNORMALLY> - RC = nnn.**
- Explanation:** The recovery manager has received notification that the I/O subtask has finished with the return code shown.
- System Action:** If the subtask finished normally, the recovery manager continues termination normally. If the subtask finished with a nonzero return code, the recovery manager issues a user abend.
- User Response:** If the subtask terminated abnormally, review prior messages in the log for further information. If sufficient information is not found, review the CICS job log for messages pertaining to the job, and also the job log for other messages, and act accordingly.
- FSIVSHAR-231 PROCESSING STOPPED ON FOLLOWING JOURNAL RECORD:**
- Explanation:** This message is used to supplement other messages where it is useful to identify the current journal file record being processed. It is followed by a message identifying the specific journal record.
- System Action:** Informational message only.
- User Response:** See message prior to this one in log for more information.
- FSIVSHAR-232 RECOVERY <STARTED/FINISHED> FOR PGMSTEP=xxxxxxx PROCSTEP=xxxxxxx
JOBNAME=xxxxxxx JOBNUM=xxxx.**
- Explanation:** Recovery has either started or finished for the jobstep identified in the message.
- System Action:** Continue normally.
- User Response:** Informational only.
- FSIVSHAR-233 <REWRITE/DELETE> RECORD JOURNALED DURING ABEND AND DOES NOT MATCH
BEFORE IMAGE - RECORD BYPASSED.**
- Explanation:** A rewrite or delete journal record was found but its key did not match that of the prior read update (before image) record found. This message is accompanied by additional messages identifying both the before image record and the rewrite or delete record.
- System Action:** This would normally cause the recovery process to abend but the rewrite or delete was journaled during an abend situation so the record pair is bypassed.
- User Response:** If more information is required about the records in question, find them in the journal file using the information from the messages.
- FSIVSHAR-234 CURRENTLY PROCESSING FOLLOWING JOURNAL RECORD.**
- Explanation:** This message is followed by another message identifying the current journal record being processed. It is issued in circumstances where the position in the journal file may be pertinent.
- System Action:** Continue normally.
- User Response:** Informational only. See accompanying messages.
- FSIVSHAR-235 RECOVERY STOPPING AT LAST USER SYNCPOINT.**
- Explanation:** Recovery has found a user SYNCPOINT record and is discontinuing its processing.
- System Action:** Processing stops.
- User Response:** None.

- FSIVSHAR-236 ROLLBACK RECORD FOUND< SKIPPING TO NEXT SYNCPOINT.**
Explanation: Recovery found a rollback record and will assume CICS has recovered back to the SYNCPOINT record.
System Action: Processing continues.
User Response: None.
- FSIVSHAR-237 I/O ERROR ENCOUNTERED OPENING JOURNAL FILE.**
Explanation: The journal I/O module was called to open the VSAM journal recovery file but the request failed.
System Action: Abend with user 999.
User Response: Look for further messages from the Journal I/O module. Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Look for further messages in the job log relating to the recovery dataset. If the problem persists, contact CSI technical support.
- FSIVSHAR-238 END OF FILE ON START OF JOURNAL FILE.**
Explanation: The Journal I/O module was called to position the VSAM journal recovery file on the last record and end of file was encountered.
System Action: Abend with user 999.
User Response: Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Also, make sure the QSAM journal file is not empty. If the problem persists, contact CSI technical support.
- FSIVSHAR-239 I/O ERROR STARTING JOURNAL FILE.**
Explanation: The Journal I/O module was called to position the VSAM journal recovery file on the last record and an unrecoverable I/O error was encountered.
System Action: Abend with user 999.
User Response: Look for further messages from the Journal I/O module. Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Look for further messages in the job log relating to the recovery dataset. If the problem persists, contact CSI technical support.
- FSIVSHAR-240 STORAGE MGR CREATE OPERATION FAILED.**
Explanation: The storage manager was called to create the request buffer but failed to complete the operation normally.
System Action: Abend with user 999.
User Response: Check for further messages from the storage manager. If the problem persists, contact CSI technical support.
- FSIVSHAR-241 I/O ERROR DURING JOURNAL FILE READ OPERATION.**
Explanation: The journal I/O module was called to read a record from the VSAM journal recovery file but the request failed. End of file was not reached.
System Action: Abend with user 999.
User Response: Look for further messages from the Journal I/O module. Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Look for further messages in the job log relating to the recovery dataset. If the problem persists, contact CSI technical support.

FSIVSHAR-242 TYPE FIELD ON JOURNAL RECORD IS UNKNOWN.

Explanation: A journal record was found with an unknown type field in the header section. The QSAM journal file or the VSAM journal recovery file may be corrupted.

System Action: Abend with user 999.

User Response: Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Rebuild the VSAM recovery dataset if necessary. If the problem persists, contact CSI technical support.

FSIVSHAR-243 BAD RETURN CODE FROM STIMER SERVICES (WAIT).

Explanation: This module requested timer services from the operating system but the request could not be completed normally.

System Action: Abend with user 999.

User Response: Check job log for other messages that may have been issued by the operating system. Rerun recovery process and if problem persists, contact technical support.

FSIVSHAR-244 STORAGE MGR ENQUEUE OPERATION FAILED.

Explanation: The storage manager failed trying to enqueue an I/O request block for the I/O subtask.

System Action: Abend with user 999.

User Response: Check for other messages from the storage manager. Rerun recovery process and if problem persists, contact technical support.

FSIVSHAR-245 STORAGE MGR DESTROY OPERATION FAILED.

Explanation: The recovery manager and I/O request manager had finished using the request storage queue but the storage manager failed to free the storage properly.

System Action: Return normally.

User Response: This message is informational only and will not impact the recovery operation. If message recurs, contact technical support.

FSIVSHAR-246 <REWRITE/DELETE> TYPE JOURNAL RECORD FOUND WITH NO MATCHING BEFORE IMAGE.

Explanation: Two journal records are expected for each rewrite or delete type I/O performed: a before image and an after image. In this case, only the after image was found. This message is accompanied by additional messages identifying both the before image record and the rewrite or delete record.

System Action: Abend with user 999.

User Response: Make sure that the VSAM recovery dataset was defined properly when the QSAM journal file was copied to it. Rebuild the VSAM recovery dataset if necessary. If the problem persists, find the rewrite or delete record in the journal file using the information from the messages and the journal record layout included in the Recovery chapter. This record can be deleted from the journal and the process can be rerun. The contents of the record should be saved for further investigation. If the problem persists, contact CSI technical support.

FSIVSHAR-247 ERROR ON CALL TO JOURNAL I/O MODULE.

Explanation: The Journal I/O module could not complete a request because of an unexpected error. Program storage may have been corrupted.

System Action: Abend with user 999.

User Response: Look for messages from the journal I/O module. If rerunning recovery produces the same message, contact technical support.

FSIVSHAR-248 ERROR ON CALL TO STORAGE MANAGER.

Explanation: The storage manager could not complete a request because of an unexpected error. Program storage may have been corrupted.

System Action: Abend with user 999.

User Response: Look for messages from the storage manager module. If rerunning recovery produces same message, contact technical support.

FSIVSHAR-249 ERROR ON CALL TO CHANGE LIST FORMATTER.

Explanation: The change list formatter module could not complete a request because of an unexpected error. Program storage may have been corrupted.

System Action: Abend with user 999.

User Response: Look for messages from the change list formatter module. If rerunning recovery produces same message, contact technical support.

FSIVSHAR-250 GETMAIN <nnnnnnnn> HEX BYTES FAILED FOR <area>.

Explanation: Not enough virtual storage was available for a GETMAIN request. The number of bytes (in hex) requested is shown as well as the data area for which the GETMAIN was being performed.

System Action: Abend with user 999.

User Response: Increase storage available to recovery job using the region parameter on either the jobcard or the step card. If the problem persists and number of bytes requested appears excessive, contact CSI technical support.

FSIVSHAR-251 CALL TO TABLE MANAGER FAILED ON JOURNAL RECORD.

Explanation: The table manager was called to locate or create an entry for the dataset specified in the current journal record but could not complete the request. A message identifying the journal record accompanies this one.

System Action: Abend with user 999.

User Response: Review the message log for further messages from the table manager and act accordingly. If the problem persists contact technical support.

FSIVSHAR-252 UNABLE TO PROCESS RECOVERY PARAMETERS.

Explanation: The parameter list processor was invoked from this program but did not complete normally.

System Action: Abend with user 999.

User Response: Review the message log for further messages from the parameter list processor. If the problem persists contact technical support.

FSIVSHAR-253 CHANGE LIST EXCEEDS ALLOCATED CHANGE LIST BUFFER.

Explanation: The internal buffer acquired to hold a change list element to be passed to the storage manager was not large enough to hold the current change list being built.

System Action: Abend with user 999.

User Response: Retry the recovery process. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-254 I/O SUBTASK WAITING FOR WORK.

Explanation: The recovery I/O subtask tried to dequeue a request block but the storage manager indicated that the buffer was empty.

System Action: When wait is finished, recovery I/O subtask tries again to dequeue a request block.

User Response: Informational message only.

FSIVSHAR-255 I/O SUBTASK HAS BEEN STARTED.

Explanation: The I/O subtask has been successfully attached to the main recovery manager task.

System Action: Continue normally.

User Response: Informational message only.

FSIVSHAR-256 RID - DD=<ddname> DATE=<yyddd> TIME=<hh:mm:ss:ht> SEQ=<seq#>.

Explanation: This message is used to supplement other messages where it is useful to identify the journal file record being processed. The DDNAME identifies the file for which the update was recorded. The Julian date and time when the journal record was created as well an internal sequence number are included. The journal record layout included in the Recovery chapter shows the location of each of the fields.

System Action: Informational message only.

User Response: See message prior to this one in log for more information.

FSIVSHAR-257 WARNING - FREEMAIN FAILED FOR READ BUFFER.

Explanation: While increasing the read buffer to accommodate a larger record length, the recovery manager tried to free the old buffer but the request failed.

System Action: Continue normally.

User Response: This is only a warning and will not impact processing unless subsequent storage requests fail due to lack of virtual storage. If the problem persists, contact CSI technical support.

FSIVSHAR-258 ERROR ON CALL TO STORAGE MANAGER.

Explanation: The storage manager could not complete a request because of an unexpected error. Program storage may have been corrupted.

System Action: Abend with user 999.

User Response: Look for messages from the storage manager module. If rerunning recovery produces same message, contact technical support.

FSIVSHAR-259 BAD RETURN CODE FROM STIMER SERVICES (WAIT).

Explanation: This module requested timer services from the operating system but the request could not be completed normally.

System Action: Abend with user 999.

User Response: Check job log for other messages that may have been issued by the operating system. Rerun recovery process and if problem persists, contact technical support.

Messages and Codes

FSIVSHAR-260 FOUND UNKNOWN REQUEST BLOCK TYPE.

Explanation: A request block was returned by the storage manager that had an unknown type field in the header section. Storage may have been corrupted.

System Action: Abend with user 999.

User Response: Retry the recovery process. If the problem persists, contact CSI technical support.

FSIVSHAR-261 ERROR ON CALL TO CHANGE LIST PROCESSOR.

Explanation: The change list processor module could not complete a request because of an unexpected error. Program storage may have been corrupted.

System Action: Abend with user 999.

User Response: Look for messages from the change list processor module. If rerunning recovery produces same message, contact technical support.

FSIVSHAR-262 WARNING, <operation> FAILED - R15=<nnn> FDBK=<nnn> FTNCD=<nnn>.

Explanation: The I/O subtask executed a VSAM I/O request that failed. The message identifies the operation being performed as well as the VSAM return code fields (decimal). Specific information about these fields can be found in the OS/VS VSAM Programmer's Guide. This message should be followed by another message identifying the journal record for which the I/O request was being performed.

System Action: This is only a warning and recovery is continued.

User Response: If most or all of the requests failed, check to ensure that the proper journal file is being used for the recovery process. Otherwise, check within context of application to see if online activity could account for the failures. If the problem persists, contact CSI technical support.

FSIVSHAR-263 GETMAIN <nnnnnnnn> HEX BYTES FAILED FOR <area>.

Explanation: Not enough virtual storage was available for a GETMAIN request. The number of bytes (in hex) requested is shown as well as the data area for which the GETMAIN was being performed.

System Action: Abend with user 999.

User Response: Increase storage available to recovery job using the region parameter on either the jobcard or the step card. If the problem persists and number of bytes requested appears excessive, contact CSI technical support.

FSIVSHAR-264 INVALID CHANGE LIST LENGTH: <nnnn> (HEX).

Explanation: The change list processor was passed a change list with an invalid length. The length is printed in hexadecimal.

System Action: Abend with user 999.

User Response: Possible storage corruption may have occurred. If the problem persists contact technical support.

FSIVSHAR-265 GETMAIN FOR DYNAMIC ALLOCATION PARM LIST FAILED.

Explanation: Not enough virtual storage was available when this module tried to acquire storage for building its dynamic allocation parameter list.

System Action: Return to caller with bad return code.

User Response: Increase storage available to recovery job using the REGION parameter on either the JOB card or EXEC card.

Messages and Codes

FSIVSHAR-266 FREEMAIN OF DYNALLOC PARM LIST AREA FAILED.

Explanation: This message is a warning only. The dynamic allocation completed normally but the storage for the parameter list was not freed.

System Action: Continue processing.

User Response: This message should not recur. If it persists, contact technical support.

FSIVSHAR-267 DYNAMIC FILE ALLOCATION FAILED.

Explanation: This module was called to dynamically allocate a file but the allocation failed.

System Action: Information regarding the failure is returned to the caller.

User Response: Check for other messages from the caller of this module. If the problem persists, contact CSI technical support.

FSIVSHAR-268 <OPENED/CLOSED> <ddname> NORMALLY.

Explanation: The VSAM dataset identified by 'DDNAME' was either opened or closed without errors.

System Action: Continue normally.

User Response: Informational only.

FSIVSHAR-269 GENCB FAILED FOR DD=<ddname>.

Explanation: A GENCB macro was executed to build an ACB for a file being recovered and the request failed.

System Action: Abend with user 999.

User Response: This condition is most likely to occur if there is not enough virtual space allocated to the recovery job to build all the necessary control blocks. This may occur if a large number of files are being recovered. Increase the space requested in the REGION parameter of the JOB card or the EXEC card and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-270 <OPEN/CLOSE> FAILED FOR <ddname> R15=<nnn> ACBERROR=<nnn>.

Explanation: An open or close operation failed for the dataset identified. Register 15 and the VSAM ACBERROR field values are included (in decimal).

System Action: For open requests, abend the process. For close requests, continue closing the rest of the files.

User Response: See the OS/VS VSAM Programmer's Guide for further information about the return codes shown. If the operation was a close, all I/O for recovery has already been performed and depending upon the severity of the close error, no action may be required. If the operation was an open, the file error should be resolved and the recovery process rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-271 SEGMENT TABLE FULL.

Explanation: The table used to record file information is full and processing cannot continue.

System Action: Abend with user 999.

User Response: Program storage may have been corrupted or an excessively large number of datasets are being recovered. Rerun recovery and if problem persists, contact technical support.

Messages and Codes

FSIVSHAR-272 GETMAIN <nnnnnnnn> HEX BYTES FAILED FOR <area>.

Explanation: Not enough virtual storage was available for a GETMAIN request. The number of bytes (in hex) requested is shown as well as the data area for which the GETMAIN was being performed.

System Action: Abend with user 999.

User Response: Increase storage available to recovery job using the REGION parameter on either the JOB card or EXEC card. If the problem persists and number of bytes requested appears excessive, contact CSI technical support.

FSIVSHAR-273 MISSING DSN=<dsname>.

Explanation: This module was called to open a dataset (dsname) for recovery but the dataset was not included in the recovery JCL.

System Action: Abend with user 999.

User Response: Ensure that the proper journal file and recovery JCL are being used. If so, add the necessary file to the recovery job. If the problem persists contact technical support.

FSIVSHAR-274 RDJFCB MACRO CALL FAILED FOR DD = <ddname>.

Explanation: The recovery manager was trying to locate the DDNAME for a VSAM dataset listed in the Journal file. The RDJFCB macro call failed for <DDNAME> in the recovery JCL.

System Action: Abend with user 999.

User Response: Rerun recovery process. If the problem persists, contact CSI technical support.

FSIVSHAR-275 PROCESSING PARAMETER CARDS FROM DD=<ddname>.

Explanation: This message is issued prior to processing the recovery parameter cards from the dataset associated with 'DDNAME'. This DDNAME is typically SYSIN.

System Action: Continue normally.

User Response: Informational message only.

FSIVSHAR-276 'RECOVERALL' OPTION SPECIFIED - ENTIRE JOURNAL WILL BE RECOVERED.

Explanation: The entire journal file will be processed during this recovery run.

System Action: Continue normally.

User Response: Informational message only.

FSIVSHAR-277 PARMS ACCEPTED - <JOBNAME=xxxxxxxxx JOBNUM=xxxx PGMSTEP=xxxxxxxxxPROCSTEP=xxxxxxxxx>.

Explanation: The parameter values listed in this message were included in the parameter cards supplied to the recovery job. These are used to set the stop conditions for the recovery process.

System Action: Informational only.

User Response: None.

FSIVSHAR-278 PROCESSING STOPPED ON CARD <nnn> (INCLUDING COMMENT CARDS).

Explanation: This message accompanies other messages when errors or inconsistencies are found with the input parameter cards for the recovery process. It identifies the parameter card being processed when the error was detected.

System Action: Informational only.

User Response: Review other messages in the recovery message log for more information on the cause of the error.

Messages and Codes

FSIVSHAR-279 SYNCPOINT OPTION SPECIFIED - RECOVER TO LAST.

Explanation: The parameter specified recovery back to last syncpoint.

System Action: Recovery to last syncpoint.

User Response: None.

FSIVSHAR-280 CALL TO PARM CARD READER FAILED.

Explanation: The parameter card reader module was called by this module to retrieve the next parameter card or to close the parameter dataset, and it did not complete normally.

System Action: Abend with user 999.

User Response: Check for further messages from the parameter card reader module. Review the JCL being used to run the recovery process and ensure that the SYSIN parameter dataset is supplied. If the problem persists, contact CSI technical support.

FSIVSHAR-281 INVALID PARAMETER CARD FOUND.

Explanation: The parameter card retrieved is invalid or unknown. A message identifying the card in error accompanies this one.

System Action: Abend with user 999.

User Response: Review the input parameter cards and correct any format or spelling errors.

FSIVSHAR-282 DUPLICATE '<keyword>' PARAMETER CARD FOUND.

Explanation: A keyword parameter card was used more than once in the card input dataset. A message identifying the duplicate card accompanies this one.

System Action: Abend with user 999.

User Response: Correct the input parameter cards and rerun.

FSIVSHAR-283 'RECOVERALL' OPTION NOT ALLOWED WHEN OTHER PARMS ARE USED.

Explanation: The RECOVERALL option card was found but other stop condition cards were found also.

System Action: Abend with user 999.

User Response: The RECOVERALL option cannot be used with other options. Correct the input cards and rerun.

FSIVSHAR-284 <keyword> PARAMETER IS TOO LONG.

Explanation: The value for the keyword parameter specified is longer than allowed. A message identifying the card in error accompanies this one.

System Action: Abend with user 999.

User Response: Correct the card in error and rerun.

FSIVSHAR-285 <keyword> PARAMETER VALUE IS INVALID.

Explanation: The value specified with the keyword parameter contains invalid characters. A message identifying the card in error accompanies this one.

System Action: Abend with user 999.

User Response: Correct the card in error and rerun.

Messages and Codes

FSIVSHAR-286 JOBNAME OR JOBNUM PARAMETER REQUIRED BUT NOT FOUND.

Explanation: Neither the JOBNAME nor the JOBNUM parameter card was found. At least one is required for specifying stop conditions.

System Action: Abend with user 999.

User Response: Add the appropriate card(s) to the input parameter dataset and rerun.

FSIVSHAR-287 ERROR <OPENING/CLOSING/READING> PARAMETER FILE DD=<ddname>.

Explanation: The parameter card input module failed on an open, close, or read of the input parameter dataset.

System Action: Abend with user 999.

User Response: Ensure that the parameter dataset is properly specified in the recovery JCL. Look for further messages in the job log.

CMS (Cross Memory Services) Messages

FSIVSHAR-301 CMS CONVERSATION ALLOCATION FAILED.

Explanation: The program tried to establish a CMS conversation and the allocation failed.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-302 CMS SEND/RECEIVE FAILED.

Explanation: CMS was trying to perform a send/receive function and failed.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-303 CMS CONVERSATION DEALLOCATION FAILED.

Explanation: The program tried to end a CMS conversation and the deallocation failed.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-304 ATTEMPTED SEND ON UNALLOCATED CONVERSATION.

Explanation: The program tried to issue a send when the conversation had not been allocated or had been deallocated.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-305 INVALID REQUEST CODE.

Explanation: A program issued a request code to CMS that was invalid.

System Action: Program abends.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-306 (CMS INFORMATION MESSAGES).

Explanation: The program issued informational messages about its processing.

System Action: None.

User Response: None.

FSIVSHAR-307 INVALID /MISSING PARM: function parm.

Explanation: The program encountered a function request that was invalid.

System Action: Program aborts.

User Response: Review and correct the function request and resubmit the job.

FSIVSHAR-308 GETMAIN XXXXXXXX BYTES FROM SUBPOOL XXX FAILED AT XXXXXXXX.

Explanation: The program issued a GETMAIN for storage from a specified subpool and failed at the program address.

System Action: Program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

Messages and Codes

- FSIVSHAR-309 INVALID LENGTH OF SVC MODULE XXXXXXXX YYYYYYYY.**
Explanation: The program encountered an error in the length of module xxxxxxxx, where yyyyyyyy is the length.
System Action: Program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-310 SVCUPDTE REQUEST FAILED, R15=XX, SVC INSTALL TERMINATED.**
Explanation: The program tried to issue the SVCUPDTE macro and failed with register 15 equal to xx.
System Action: Program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-311 INITIALIZATION FAILED FOR MODULE AT XXXXXXXX.**
Explanation: A module was loaded and tried to initialize and failed at location xxxxxxxx.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-312 LOAD OF SVC MODULE INTO ACQUIRED STORAGE FAILED.**
Explanation: The CMS SVC was being loaded into storage acquired by the program and failed.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-313 FAILURE EXECUTING CMS OPERATION XXXXX.**
Explanation: CMS was executing an operation xxxxx and failed.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-314 SST UTILITY, REQUEST CODE PARM IS INVALID.**
Explanation: SST utility of CMS was called with an invalid parm.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-315 GETMAIN FOR SSCVT ENTRY, SP 241 FAILED.**
Explanation: The program issued a GETMAIN for storage out of subpool 241 to build an SSCVT control block and the GETMAIN failed.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-316 SSCVT ADD FAILED, CONTENTION FOR TABLE, TRY AGAIN.**
Explanation: The program tried to add an SSCVT while another job was doing the same.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-317 NO JES COMMUNICATION TABLE FOUND, CVTJESCT = 0.**
Explanation: The program was looking for the CVTJESCT but the load address was 0.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.

Messages and Codes

- FSIVSHAR-318 NO ENT JES COMMUNICATION TABLE FOUND, JESSCT = 0.**
Explanation: The program was looking for the JESSCT and the load address was 0.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-319 SSCVT NAME: XXXXXXXX IS INVALID.**
Explanation: The program was passed an invalid name for the SSCVT.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-320 REQUIRED MESSAGE ROUTINE: XXXXXXXX NOT AVAILABLE.**
Explanation: The program tried to load message routine xxxxxxxx, and found it not available.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-321 SSCVT CHAIN INVALID, MULT. ENTRIES ARE MISSING SSCT ID.**
Explanation: The program found invalid SSCVT entries.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-322 NO OR INVALID PARMS SPECIFIED ON EXECUTE CARD.**
Explanation: The program could not recognize the parms on EXEC card.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-323 INVALID UTILITY REQUEST NUMBER.**
Explanation: The program requested a function of another program, but the function number was invalid.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-324 SVC NUMBER XXXXXXXXXXXXX.**
Explanation: The SVC numbered in the message had a problem also identified in the message.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-325 SYSTEM NAME XXXXXXXXXXXXX.**
Explanation: The system name in the message had a problem also identified in the message
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-326 TIME OPTION XXXXXXXXXXXXX.**
Explanation: The time option identified in the message had a problem also identified in the message.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.

Messages and Codes

- FSIVSHAR-327 ERROR EXECUTING MODULE: XXXXXXXX.**
Explanation: The program received a bad return code from module xxxxxxxx.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-328 XXXXXXXX OF SUBSYSTEM NAME TABLE ENT XXXXXXXX.**
Explanation: An addition or deletion of the subsystem name table entry has either completed or failed.
System Action: Dependent.
User Response: Dependent.
- FSIVSHAR-329 XXXX FAILED, OPERATION TERMINATED XXXXXXXX.**
Explanation: A system request failed, and the operation that terminated is described.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-330 ERROR ACCESSING FSI-VSHARE SYSTEM INFORMATION BLOCK.**
Explanation: The program tried to access a control block and failed.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-331 XXXX ALREADY INSTALLED.**
Explanation: The program tried to install the CMS SVC and it was already found.
System Action: The program ends.
User Response: None.
- FSIVSHAR-332 CONTENTION FOR FSI-VSHARE SYSTEM TABLE, OPERATION ABORTED.**
Explanation: The program tried to access the system table and it was being updated by another job.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-333 SYSTEM NOT INSTALLED YET, CANNOT EXECUTE REQUEST.**
Explanation: The program tried to make a request to CMS, but CMS was not found to be installed.
System Action: The program aborts.
User Response: Install CMS.
- FSIVSHAR-334 INSTALLATION FACILITY IN USE, TRY LATER.**
Explanation: The program tried to use the installation facility and found another user had it.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.
- FSIVSHAR-335 A TIMER INITIATED REQUEST FOR XXXX IS OUTSTANDING.**
Explanation: An enqueue request was made and one was found to be outstanding.
System Action: The program aborts.
User Response: Retry the job; if the error occurs again contact CSI technical support.

Messages and Codes

FSIVSHAR-336 INVALID REQUEST CODE IN SVC PARMLIST.

Explanation: The program made a request to the SVC module, and this module found an error in the request code.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-337 SVC PARMLIST IS INVALID.

Explanation: The program made a request to the SVC module, and this module found an error in the parmlist.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

FSIVSHAR-338 MODULE XXXXXXXX NOT FOUND.

Explanation: The program made a load request and could not find the module.

System Action: The program aborts.

User Response: Retry the job; if the error occurs again contact CSI technical support.

Presentation Services Messages

FSIVSHAR-401 FSM WAS PASSED AN INVALID FSM ID.

Explanation: The Finite State Machine module was invoked with an unknown Finite State Machine ID. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-402 FSM WAS PASSED AN INVALID INPUT STATE.

Explanation: The input state for the specified Finite State Machine was not within the range of possible values. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-403 FSM WAS PASSED AN INVALID OPERATION.

Explanation: The Finite State Machine module received a request to perform an undefined operation. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-404 FSM ENCOUNTERED AN EXECUTION TIME ERROR.

Explanation: The specified input to the Finite State Machine module was invalid given the current state of the specified Finite State Machine. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-405 INVALID CONTROL BLOCK NUMBER REQUESTED.

Explanation: The control block formatter module received a request for the address of an unknown control block. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-406 CONTROL FILE PROGRAM FAILED ON <oper>. FILE STATUS = <nn>.

Explanation: An I/O error occurred accessing the Session Information Control File. <oper> indicates the operation that failed (open, close, or read) and <nn> is the COBOL file status code. Refer to a COBOL programmer's reference guide for further status code information.

System Action: Session termination.

User Response: File status code should reveal the nature of the problem (that is, control record not found, wrong file definition, etc.).

FSIVSHAR-407 CONTROL FILE PROGRAM FAILED - RETURN CODE WAS NOT SET.

Explanation: The Control File Program was invoked by this module but did not set a return code. The return code field may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-408 CONTROL FILE PROGRAM FAILED - UNKNOWN RETURN CODE.

Explanation: The Control File Program was invoked by this module but finished with an unknown return code. Program code field may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-409 REQUEST TO EXECUTE UNKNOWN ROUTINE.

Explanation: The verb router was invoked from the Transaction Program but the parameter list contained an unknown routine number. Program storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-410 INVALID RETURN CODE FROM CONTROL BLOCK FORMATTER.

Explanation: The control block formatter program was invoked by this module but terminated abnormally.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-411 INVALID PARM - LU NAME.

Explanation: The remote application ID passed as a parameter to this module contained invalid characters. The session information record on the control file may be invalid or program storage may have been corrupted.

System Action: Session termination.

User Response: Check control file session information record to ensure validity of session information. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-412 LU_NAME NOT ON PARTNER LU LIST.

Explanation: The remote application ID passed as a parameter to this module was not found on the list of valid partners for this LU. The session information record on the control file may be bad or program storage may have been corrupted.

System Action: Session termination.

User Response: Check control file session information record to ensure validity of session information. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-413 INVALID PARM - MODE_NAME.

Explanation: The mode table entry name ID passed as a parameter to this module contained invalid characters. The session information record on the control file may be bad or program storage may have been corrupted.

System Action: Session termination.

User Response: Check control file session information record to ensure validity of session information. Refer to the Installation manual for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If problem still persists, contact your technical support group.

FSIVSHAR-414 MODE NAME NOT ON PARTNER LU MODELIST.

Explanation: The mode table entry name ID passed as a parameter to this module was not listed as a valid mode entry for the specified remote application ID. The session information record on the control file may be bad or program storage may have been corrupted.

System Action: Session termination.

User Response: Check control file session information Record to ensure validity of session information. Refer to the Installation manual for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If problem still persists, contact technical support.

FSIVSHAR-415 NO PARMLIST RETURNED FROM RM.

Explanation: An invalid parameter list address was returned from a call to the Resource Manager.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-416 WRONG REC TYPE RETURNED FROM RM.

Explanation: This module received an unknown or invalid record type from the Resource Manager. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-417 UNKNOWN RETCODE FROM RM.

Explanation: This module received an unknown/invalid return code from the Resource Manager. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-418 ABEND REQUESTED BY RM.

Explanation: The return code on a call to the Resource Manager indicates that an abend situation was encountered.

System Action: Session termination.

User Response: Look for further messages from the Resource Manager and take action accordingly. If none are found, storage may have been corrupted. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-419 INVALID SESSION ID SPECIFIED.

Explanation: The session ID parameter passed to this module contained illegal characters. The transaction program's storage area may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-420 TYPE PARAMETER CONTAINS INVALID VALUE.

Explanation: The parameter passed to this module specifying type of deactivation was unknown value. The transaction program's storage area may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-421 NO PARMLIST RETURNED FROM RM.

Explanation: An invalid parameter list address was returned from a call to the Resource Manager.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-422 WRONG REC TYPE RETURNED FROM RM.

Explanation: This module received an unknown or invalid record type from the Resource Manager. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-423 UNKNOWN RETCODE FROM RM.

Explanation: This module received an unknown/invalid return code from the Resource Manager. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-424 SESSION DEACT MAY NOT HAVE COMPLETED.

Explanation: The call to the Resource Manager to deactivate the session completed with a nonzero return code. This is only a warning since the session will terminate anyway when the transaction program finishes. Related messages may be generated by the remote LU.

System Action: Continue normally - warning only.

User Response: If the problem persists, contact CSI technical support.

FSIVSHAR-425 RECEIVED INVALID ADDRESS OF PARM LIST.

Explanation: The parameter list address passed to this program was invalid. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-426 OWN (LOCAL) IS UNSUPPORTED PARAMETER.

Explanation: The 'own' parameter specifies that the transaction program to be allocated resides in the local LU. This is an unsupported option but may have been erroneously specified due to storage corruption.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-427 LUNAME PARM CONTAINS INVALID CHARACTERS.

Explanation: The remote application ID passed as a parameter to this module contained invalid characters. Since this error was not encountered during session activation, program storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-428 MODE PARM CONTAINS INVALID CHARACTERS.

Explanation: The mode table entry name ID passed as a parameter to this module contained invalid characters. Since this error was not encountered during session activation, program storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-429 TPNAME PARM CONTAINS INVALID CHARACTERS.

Explanation: The remote transaction ID passed as a parameter to this module contained invalid characters. The session information record on the control file may be invalid or program storage may have been corrupted.

System Action: Session termination.

User Response: Check control file session information record to ensure validity of session information. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If problem still persists, contact technical support.

FSIVSHAR-430 MODE=SNASVCMG INVALID FOR THIS PROGRAM.

Explanation: The mode specified is only valid for LU service transaction programs.

System Action: Session termination.

User Response: Put valid mode entry name on control file session information record. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, contact technical support.

FSIVSHAR-431 UNSUPPORTED PARM - MAPPED CONVERSATION.

Explanation: This program received a request for a mapped conversation which is an unsupported option. Parameter list storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

- FSIVSHAR-432 RM RETURNED RECORD WITH INVALID SOURCE.**
- Explanation:** This module called the Resource Manager which returned a record with an invalid source field. Storage corruption may have occurred.
- System Action:** Session termination.
- User Response:** Check application program for possible storage violations. If the problem persists, contact CSI technical support.
- FSIVSHAR-433 NONZERO RETURN CODE FOUND IN RECORD FROM RM.**
- Explanation:** The Resource Manager encountered an abnormal condition while processing a request from this module.
- System Action:** Session termination.
- User Response:** Look for further messages from the Resource Manager and take action accordingly. If none are found, storage may have been corrupted. If the problem persists, contact CSI technical support.
- FSIVSHAR-434 INVALID RECORD ADDR RETURNED FROM RM.**
- Explanation:** An invalid parameter list address was returned from a call to the Resource Manager.
- System Action:** Session termination.
- User Response:** If the problem persists, contact CSI technical support.
- FSIVSHAR-435 LUNAME PARM NOT FOUND IN PCB.**
- Explanation:** The remote application ID passed as a parameter to this module was not listed as a valid partner LU in the Partner Control Block. The session information record on the control file may be invalid or program storage may have been corrupted.
- System Action:** Session termination.
- User Response:** Check control file session information record to ensure validity of session information. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If problem still persists, contact technical support.
- FSIVSHAR-436 MODE PARM NOT FOUND IN MCB.**
- Explanation:** The mode entry name passed as a parameter to this module was not listed in the Mode Control Block for this LU. The session information record on the control file may be invalid or program storage may have been corrupted.
- System Action:** Session termination.
- User Response:** Check control file session information record to ensure validity of session information. Refer to the Installation chapter for the format of this record. If record appears valid and error persists, check application program for possible storage violations. If problem still persists, contact technical support.
- FSIVSHAR-437 ALLOCATE FAILED - RCB NOT CREATED.**
- Explanation:** The Resource Manager returned a conversation failure record on an allocate request and the conversation ID was not found in the Resource Control Block. Therefore, no further error processing was performed for the conversation.
- System Action:** Session termination.
- User Response:** Look for further messages from the Resource Manager and take action accordingly. If none are found and the problem persists, contact technical support.

FSIVSHAR-438 ALLOCATE FAILED - PROTOCOL VIOLATION.

Explanation: The Resource Manager returned a conversation failure record on an allocate request and the reason code indicated a protocol violation was detected.

System Action: Session termination.

User Response: If the problem persists, contact technical support.

FSIVSHAR-439 ALLOCATE FAILED - SESSION OUTAGE.

Explanation: The Resource Manager returned a conversation failure record on an allocate request and the reason code indicated a session outage was detected.

System Action: Session termination.

User Response: If the problem persists, contact technical support.

FSIVSHAR-440 ALLOCATE FAILED RETURNED FROM RM.

Explanation: The Resource Manager set the return code on an allocate request to 'allocate failed'.

System Action: The allocate verb return code is set to 'allocate failed' and control is returned to the transaction program.

User Response: Check for further messages. If none are found and problem persists, contact technical support.

FSIVSHAR-441 SYNC LEVEL UNSUPPORTED ON ALLOCATE.

Explanation: The Resource Manager set the return code on an allocate request to 'synclevel unsupported'. The parameter list may have been corrupted.

System Action: The allocate verb return code is set to 'synclevel unsupported' and control is returned to the transaction program.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-442 UNKNOWN RETCODE FROM RM ON ALLOCATE.

Explanation: The return code set by the Resource Manager on an allocate request is unknown. Storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-443 FSM CONVERSATION ENCOUNTERED ABEND.

Explanation: The state of the conversation could not be changed to 'RCB allocated' from its current state. This may happen if the return code area or the Finite State Machine storage has been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-444 HS FAILED SENDING FMH-5 RECORD - NONZERO RETCODE FOUND.

Explanation: This module called Half Session to send an attach type Function Management Header (FMH-5) to the remote LU to initiate a remote TP. Half Session could not complete this request normally.

System Action: Session termination.

User Response: Check for further messages from Half Session. If none are found, check the application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-445 RETURN CONTROL PARM NOT WHEN_SESS_ALLOC (REQUIRED).

Explanation: The return control parameter of the allocate verb was not set to 'when session allocated' which is required by the Batch LU. The parameter list may have been corrupted.

System Action: Session termination.

User Response: Check the application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-446 FSM CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-447 INVALID RECORD ADDR RETURNED FROM HS.

Explanation: The address of the record expected from half session manager on a deallocation request was unusable. Possible storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-448 NONZERO RETCODE FROM DEALLOCATION_CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-449 INVALID PARM LIST ADDRESS FOUND.

Explanation: The transaction program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-450 RCBID PARM DOES NOT MATCH ALLOCATED RCB.

Explanation: The conversation identifier (RCBID) parameter from the transaction program did not match the RCBID of the currently allocated conversation. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-451 TYPE PARM NOT IN VALID RANGE.

Explanation: The deallocation type parameter passed by the transaction program was not within the range of possible values. Storage corruption may have occurred.

System Action: Session termination.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-452 RM RETURNED OTHER THAN RCB-DEALLOCATED REC.

Explanation: This module called the resource manager requesting deallocation of the conversation and did not receive the expected RCB-DEALLOCATED record type back from the resource manager. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-453 INVALID SOURCE FOUND ON RECORD FROM RM.

Explanation: The record returned from the resource manager on a deallocation request contained an unexpected or unknown value in the 'source' field. Storage corruption may have occurred.

System Action: Session termination

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-454 NONZERO RETURN CODE FROM CALL TO RM.

Explanation: The resource manager failed trying to deallocate the conversation. This was indicated by a nonzero return code set on the record received from the resource manager.

System Action: Session termination.

User Response: Check for further messages from the resource manager. If the problem persists, contact CSI technical support.

FSIVSHAR-455 INVALID REC ADDRESS RETURNED FROM RM.

Explanation: The address of the record expected from the resource manager on a deallocation request was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-456 UNSUPPORTED PARM: TYPE = CONFIRM.

Explanation: The transaction program requested deallocation of the conversation with confirm level synchronization. This type is unsupported. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-457 UNSUPPORTED PARM: TYPE = SYNCLEVEL.

Explanation: The transaction program requested deallocation of the conversation with 'synclevel' synchronization. This type is unsupported. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-458 UNSUPPORTED PARM: TYPE = ABEND SVC.

Explanation: The type of deallocation requested is invalid for the transaction program since it is only used by LU service transactions. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-459 UNSUPPORTED PARM: TYPE = ABEND TIMER.

Explanation: The type of deallocation requested is invalid for the transaction program since it is only used by LU service transactions. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-460 UNSUPPORTED PARM: TYPE = LOCAL.

Explanation: Local deallocation is unsupported. This type is only used when the transaction program wants to clean up the local LU after being informed that deallocation took place during a previously issued verb. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-461 UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.

Explanation: A state check was encountered while trying to change the state of the conversation to deallocated. In this case, the only expected action to take (as indicated by the output code) is to perform deallocation cleanup. This code was not received possibly due to corrupted storage.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-462 DEALLOCATION CLEANUP PERFORMED.

Explanation: Deallocation cleanup is performed when an error is encountered for which the conversation must be deallocated. Other messages indicating the nature of the problem should appear also.

System Action: Session termination.

User Response: Check for other messages to further identify problem. If the problem persists, contact CSI technical support.

FSIVSHAR-463 SEND DATA NOT ON LOGICAL RECORD BOUNDARY.

Explanation: The send length field indicated that some data remained to be sent when the deallocation request was made by the transaction program. Storage corruption may have occurred.

System Action: Deallocation cleanup will be performed and the session will be terminated.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-464 INVALID REC ADDRESS RETURNED FROM HS.

Explanation: The address of the record expected from half session manager on a deallocation request was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-465 NONZERO RETURN CODE FROM CALL TO HS.

Explanation: The half session manager failed sending a deallocate request to the remote LU. This was indicated by a nonzero return code set on the record received from the half session.

System Action: Session termination.

User Response: Check for further messages from the resource manager. If the problem persists, contact CSI technical support.

FSIVSHAR-466 FSM CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-467 FSM CONVERSATION ENCOUNTERED ABEND.

Explanation: The conversation Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-468 NONZERO RETCODE FROM DEALLOCATION CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage has possibly been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-469 INVALID PARM LIST ADDRESS FOUND.

Explanation: The transaction program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-470 RCBID PARM DOES NOT MATCH ALLOCATED RCB.

Explanation: The conversation identifier (RCBID) parameter from the transaction program did not match the RCBID of the currently allocated conversation. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-471 FSM CONVERSATION ENCOUNTERED ABEND.

Explanation: The conversation Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-472 UNKNOWN OUTPUT FROM FSM CONVERSATION.

Explanation: The output of the conversation Finite State Machine not one of the possible values (state check, normal or abend). The parameter list storage or Finite State Machine storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-473 FSM CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-474 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.

Explanation: The error/failure Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-475 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.

Explanation: The error/failure Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-476 LL FIELD OF SEND DATA IS INVALID.

Explanation: The two byte length field preceding the data to be sent can be set to up to 32765. This value includes the length field so it must be at least 2. Also, the first bit of the length is reserved so one of the following invalid values was encountered: x'0000', x'0001', x'8000' or x'8001'. The length bytes may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-477 HS FAILED SENDING TP DATA RECORD (NONZERO RETCODE).

Explanation: The half session manager failed during a request to send data to the remote LU. This was indicated by a nonzero return code set on the record received from the half session.

System Action: Session termination.

User Response: Check for further messages from the half session manager. If the problem persists, contact CSI technical support.

FSIVSHAR-478 UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.

Explanation: A state check was encountered while trying to change the state of the conversation to 'send'. In this case, the only expected action to take (as indicated by the output code) is to perform deallocation cleanup. This code was not received possibly due to corrupted storage.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-479 INVALID RECORD ADDR RETURNED FROM HS.

Explanation: The address of the record expected from half session manager was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-480 NONZERO RETCODE FROM DEALLOCATION CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage has possibly been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-481 STATE CHECK CHANGING FSM TO SEND STATE.

Explanation: A state check was encountered while trying to change the state of the conversation to 'send'. This message is only a warning since the program will attempt to process the output code from the state machine. Other messages may be generated.

System Action: Start error processing leading to eventual session termination.

User Response: Check for other related messages and act accordingly. If the problem persists, contact CSI technical support.

FSIVSHAR-482 FOUND NONZERO VTAM SENSE CODE IN LULUCB.

Explanation: A session problem was encountered prior to invocation of this module. The session was probably deactivated abnormally as a result.

System Action: Continue with error processing leading to eventual program termination.

User Response: Check for other related messages and act accordingly. If the problem persists, contact CSI technical support.

FSIVSHAR-483 INVALID PARM LIST ADDRESS FOUND.

Explanation: The transaction program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-484 RCBID PARM DOES NOT MATCH ALLOCATED RCB.

Explanation: The conversation identifier (RCBID) parameter from the transaction program did not match the RCBID of the currently allocated conversation. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-485 INVALID TYPE PARAMETER FOUND.

Explanation: The type parameter passed to this module was invalid (not FLUSH, CONFIRM, or SYNC_LEVEL).

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-486 TYPE = CONFIRM NOT SUPPORTED.

Explanation: The type parameter passed to the Prepare-to-Receive module was CONFIRM. This option is not supported.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-487 TYPE UNSUPPORTED (SYNCPT AND RCB.LEVEL NOT NONE).

Explanation: The type parameter passed to the Prepare-to-Receive module was SYNCPT. This option is only supported if the RCB SYNCLEVEL is none.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-488 DEALLOCATION CLEANUP INVOKED.

Explanation: Deallocation cleanup is performed when an error is encountered for which the conversation must be deallocated. Other messages indicating the nature of the problem should accompany this one.

System Action: Session termination.

User Response: Check for other messages to further identify the problem. If the problem persists, contact CSI technical support.

FSIVSHAR-489 UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.

Explanation: A state check was encountered while trying to change the state of the conversation. In this case, the only expected action to take (as indicated by the output code) is to perform deallocation cleanup. This code was not received possibly due to corrupted storage.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-490 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.

Explanation: The error/failure Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the finite state machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-491 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.

Explanation: The Finite State Machine processor encountered an execution time error changing the ERROR_OR_FAILURE FSM to the RESET state.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-492 FSM_CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the finite state machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-493 STATE CHECK CHECKING FSM WITH PREP_TO_RCV_FLUSH.

Explanation: The state of the conversation could not be changed to PREPARE_TO_RECEIVE_FLUSH so the corresponding output code is executed. The output code probably implies deallocation cleanup.

System Action: Continue processing output code from FSM.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-494 INVALID RECORD ADDR RETURNED FROM HS.

Explanation: The address of the record expected from half session manager was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-495 HS FAILED HANDLING PREP_TO_RCV_FLUSH RECORD.

Explanation: HS was called to handle a PREPARE_TO_RECEIVE_FLUSH request but did not complete the request normally.

System Action: Session termination.

User Response: Check for further messages from Half Session. If the problem persists, contact CSI technical support.

FSIVSHAR-496 FSM_CONVERSATION ENCOUNTERED ABEND.

Explanation: The conversation Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-497 NONZERO RETCODE FROM DEALLOCATION CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage has possibly been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-498 INVALID PARM LIST ADDRESS FOUND.

Explanation: The transaction program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-499 RCBID PARM DOES NOT MATCH ALLOCATED RCB.

Explanation: The conversation identifier (RCBID) parameter from the transaction program did not match the RCBID of the currently allocated conversation. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

Presentation Messages

FSIVSHAR-501 FSM CONVERSATION ENCOUNTERED ABEND.

Explanation: The conversation Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-502 UNKNOWN OUTPUT FROM FSM_CONVERSATION.

Explanation: The output of the conversation Finite State Machine not one of the possible values (state check, normal or abend). The parameter list storage or Finite State Machine storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-503 FSM_CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-504 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.

Explanation: The error/failure Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-505 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.

Explanation: The error/failure Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-506 NONZERO RETURN FROM PS_PROTOCOL_ERROR.

Explanation: The protocol error processing program was called to process a protocol error encountered during a receive operation, and did not complete normally. Program storage has possibly been corrupted.

System Action: Session termination.

User Response: Check for further messages relating to the problem. Also check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-507 NONZERO RETCODE FROM HS AFTER RECEIVE.

Explanation: The half session manager failed during a request to receive data from the remote LU. This was indicated by a nonzero return code set on the record received from the half session.

System Action: Session termination.

User Response: Check for further messages from the half session manager. If the problem persists, contact CSI technical support.

FSIVSHAR-508 NONZERO RETCODE FROM PERFORM_RECEIVE_PROC.

Explanation: The receive processing program was called to process a record received but it did not complete normally (R15 was nonzero). Further messages may have been generated.

System Action: Session termination.

User Response: Check for further messages. If the problem persists, contact CSI technical support.

FSIVSHAR-509 NONZERO RETCODE FROM DEALLOCATION_CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage has possibly been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-510 UNKNOWN OUTPUT CODE FROM FSM_CONVERSATION.

Explanation: A state check was encountered and the program was trying to process the resulting output code from the conversation Finite State Machine. However, the output code had an unknown value, possibly due to storage corruption.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-511 INVALID RECORD ADDR RETURNED FROM HS.

Explanation: The address of the record expected from half session manager was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-512 INVALID SOURCE FOUND ON RECORD FROM HS.

Explanation: The record returned from the half session manager on a receive request contained an unexpected or unknown value in the 'source' field. Storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-513 PROTOCOL ERROR AFTER RECEIVE BY HS.

Explanation: The protocol error processing program was invoked based on the data received from the half session. Further messages should be included.

System Action: Session termination.

User Response: Check for further messages to help identify problem. If the problem persists, contact CSI technical support.

FSIVSHAR-514 PROTOCOL ERROR - DIDNT GET EXPECTED FMH7.

Explanation: A receive error occurred for which this module expected an error type function management header record (FMH7). The data received was not an FMH7.

System Action: Session termination.

User Response: Check for further messages that may identify the nature of the error. Messages may indicate why the FMH7 was being sent. If the problem persists, contact CSI technical support.

FSIVSHAR-515 NONZERO RETCODE FROM PROCESS_FMH7_PROC.

Explanation: This module received an FMH7 from the remote LU and called the FMH7 processing module to handle it. The FMH7 processor did not complete normally and it may have generated further messages.

System Action: Return to batch program with appropriate return codes.

User Response: Check for further messages to identify the problem. If the error persists, contact technical support.

FSIVSHAR-516 INVALID PARM LIST ADDR FROM PROCESS_FMH7.

Explanation: An error function management header (FMH7) was received from the remote LU and was processed by the FMH7 processing module. The return parameter list address from the FMH7 processor was unusable.

System Action: Session termination.

User Response: Check for further messages that may identify the source of the problem. If the error persists, contact technical support.

FSIVSHAR-517 TWO REQ_2_SEND SIGNALS RECEIVED FROM HS.

Explanation: A request-to-send signal was received from the remote LU after which, this program expected to receive a data record. However, another request-to-send signal was received instead.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-518 UNEXPECTED CONFIRMED REC RECVD FROM HS.

Explanation: Data is not sent with confirmation requests so this module does not expect to receive confirm acknowledgments (confirmed records). However, one was received by this module. Record storage area may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-519 UNEXPECTED RTS RESPONSE RECVD FROM HS.

Explanation: Request to send was not issued so program did not expect to receive a response to one. Record data area may be corrupted or remote LU may be experiencing problems.

System Action: Session termination.

User Response: Check for messages that further identify the problem including the CICS log. Also, check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-520 UNKNOWN RECORD TYPE FROM RECVD FROM HS.

Explanation: A record was received from the half session but its type was unknown and therefore could not be processed. The record storage area may have been corrupted.

System Action: Session termination.

User Response: Check application program for storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-521 STATE CHECK CHANGING TO RCV_IMMED STATE.

Explanation: A state check was encountered while trying to change the state of the conversation to 'receive immediate'. This message is only a warning since the program will attempt to process the output code from the state machine. Other messages may be generated.

System Action: Start error processing leading to eventual session termination.

User Response: Check for other related messages and act accordingly. If the problem persists, contact CSI technical support.

FSIVSHAR-522 FOUND NONZERO VTAM SENSE CODE IN LULUCB.

Explanation: A session problem was encountered either prior to invocation of this module or during receive processing. The session was probably deactivated abnormally as a result.

System Action: Continue with error processing leading to eventual program termination.

User Response: Check for other related messages and act accordingly. If the problem persists, contact CSI technical support.

FSIVSHAR-523 INVALID PARM LIST ADDRESS FOUND.

Explanation: The batch program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-524 INVALID REC ADDRESS RETURNED FROM RM.

Explanation: The address of the record expected from the resource manager was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-525 NONZERO RETURN CODE FROM CALL TO RM.

Explanation: The resource manager failed processing an unbind-protocol-error record. This was indicated by a nonzero return code set on the record received from the resource manager.

System Action: Session termination.

User Response: Check for further messages from the resource manager. If the problem persists, contact CSI technical support.

- FSIVSHAR-527 INVALID SOURCE FOUND ON RECORD FROM RM.**
Explanation: The record returned from the resource manager on a receive request contained an unexpected or unknown value in the 'source' field. Storage corruption may have occurred.
System Action: Session termination.
User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.
- FSIVSHAR-528 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.**
Explanation: The error/failure Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.
System Action: Session termination.
User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.
- FSIVSHAR-529 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.**
Explanation: The error/failure Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.
System Action: Session termination.
User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.
- FSIVSHAR-530 PROTOCOL ERROR WITH SENSE = nnnnnnnn.**
Explanation: Presentation services detected a protocol error. This message should be accompanied by others further describing the error. The sense code values and their meanings can be found in the IBM publication 'SNA Format and Protocol Manual for LU Type 6.2.'
System Action: The resource manager is called to deactivate the session. Control is returned to the transaction program following session termination.
User Response: Check for further messages from presentation services. If the problem persists, contact CSI technical support.
- FSIVSHAR-531 INVALID PARM LIST ADDRESS FOUND.**
Explanation: The transaction program passed a parameter list address that was negative or zero.
System Action: Session termination.
User Response: Check for other related messages. If the problem persists, contact CSI technical support.
- FSIVSHAR-532 INVALID REC ADDRESS RETURNED FROM HS.**
Explanation: The address of the record expected from half session manager was unusable. Program storage corruption may have occurred.
System Action: Session termination.
User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-533 NONZERO RETURN CODE FROM CALL TO HS.

Explanation: The half session manager failed while trying to receive an error type function management header (FMH7). This was indicated by a nonzero return code set on the record received from half session.

System Action: Session termination.

User Response: Check for further messages relating to the problem. If the problem persists, contact CSI technical support.

FSIVSHAR-534 INVALID SOURCE FOUND ON RECORD FROM HS.

Explanation: When trying to receive an FMH7 record from half session, this module received a record with an unexpected or unknown value in the 'source' field. Storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-535 FSM_CONVERSATION LOGIC OR PARM ERROR.

Explanation: The conversation Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-536 FSM_CONVERSATION ENCOUNTERED AN ABEND.

Explanation: The error/failure Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-538 NONZERO RETCODE FROM PS_PROTOCOL_ERROR.

Explanation: The protocol error processing program was called to process a protocol error encountered in this module but did not complete normally. Program storage was possibly corrupted.

System Action: Session termination.

User Response: Check for further messages relating to the problem. Also check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-539 NONZERO RETCODE FROM DEALLOCATION CLEANUP.

Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

Messages and Codes

FSIVSHAR-540 UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.

Explanation: A state check was encountered and the program was trying to process the resulting output code from the conversation Finite State Machine. However, an the output code had an unknown value possibly due to storage corruption.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-541 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.

Explanation: FSM_OR_FAILURE LOGIC OR PARM ERROR.

System Action: Program aborts.

User Response: Look for any previous error messages and fix as required. If the problem persists call CSI technical support.

FSIVSHAR-542 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.

Explanation: FSM_OR_FAILURE ENCOUNTERED ABEND.

System Action: Program aborts.

User Response: Look for any previous error messages and fix as required. If the problem persists call CSI technical support.

FSIVSHAR-543 INVALID REC ADDRESS RETURNED FROM RM.

Explanation: The address of the record expected from the resource manager after requesting deallocation was unusable. Program storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-544 INVALID SOURCE FOUND ON RECORD FROM RM.

Explanation: When checking the record returned from the resource manager after a deallocation request, this module found an unexpected or unknown value in the 'source' field. Storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-545 NONZERO RETURN CODE FROM CALL TO RM.

Explanation: The resource manager failed while processing a deallocation request. This was indicated by a nonzero return code set on the record received from half session.

System Action: Session termination.

User Response: Check for further messages from the resource manager. If the problem persists, contact CSI technical support.

FSIVSHAR-546 RM RETURNED OTHER THAN RCB-DEALLOCATED REC.

Explanation: After requesting that the resource manager deallocate the conversation, this program expected a 'RCB-DEALLOCATED' type record to be returned from the resource manager. Another unexpected or unknown record type was returned. Storage corruption may have occurred.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-547 DEALLOCATION CLEANUP INVOKED.

Explanation: Deallocation cleanup is performed when an error is encountered for which the conversation must be deallocated. Other messages indicating the nature of the problem should appear also.

System Action: Session termination.

User Response: Check for other messages to further identify problem. If the problem persists, contact CSI technical support.

FSIVSHAR-548 INVALID ADDR OF HS RECORD PASSED IN REG01.

Explanation: The transaction program passed a parameter list address that was negative or zero.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-549 INVALID PARM LIST ADDRESS PASSED IN PARMBAR.

Explanation: This module expected to get the address of the receive verb parameter list in a register ('parmbar') but the address in that register was invalid.

System Action: Session termination.

User Response: Check for other related messages. If the problem persists, contact CSI technical support.

FSIVSHAR-550 NONZERO RETCODE FROM PS_PROTOCOL_ERROR.

Explanation: The protocol error processing program was called to process a protocol error encountered in this module but did not complete normally. Program storage may have been corrupted.

System Action: Session termination.

User Response: Check for further messages relating to the problem. Also check application program for possible storage violations. If the problem persists, contact technical support.

FSIVSHAR-551 RECEIVED PS_HEADER - UNSUPPORTED.

Explanation: The type field of the record received indicated that it was a PS_HEADER which is used for syncpoint services. These record types are unsupported and may indicate that the record area was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-552 PROTOCOL ERROR - INV LL IN RECEIVED DATA.

Explanation: The two byte length field preceding the received data can be set to up to 32765. This value includes the length field so it must be at least 2. Also, the first bit of the length is reserved so one of the following invalid values was encountered: x'0000', x'0001', x'8000' or x'8001'. The length bytes may have been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-553 NONZERO RETURN FROM PS_PROTOCOL_ERROR.

Explanation: The protocol error processing program was called to process a protocol error encountered in this module but did not complete normally. Program storage was possibly corrupted.

System Action: Session termination.

User Response: Check for further messages relating to the problem. Also check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-554 FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.

Explanation: The error/failure Finite State Machine failed to complete the requested operation either because the parameters passed were bad or the Finite State Machine data was corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-555 FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.

Explanation: The error/failure Finite State Machine flagged an abend situation based on the current state of the conversation and the state it was being changed to. Either some internal operations were performed out of sequence or program storage was been corrupted.

System Action: Session termination.

User Response: Check application program for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-556 NONZERO RETURN FROM PROCESS_FMH7_PROC.

Explanation: This module was passed an FMH7 and called the FMH7 processing module to handle it. The FMH7 processor did not complete normally and it may have generated further messages.

System Action: Session termination.

User Response: Check for further messages to identify the problem. If the error persists, contact technical support.

FSIVSHAR-557 BAD RETURN PARM ADDR FROM PROCESS_FMH7.

Explanation: An error type function management header (FMH7) was received from the remote LU and was processed by the FMH7 processing module. The return parameter list address from the FMH7 processor was unusable.

System Action: Session termination.

User Response: Check for further messages that may identify the source of the problem. If the error persists, contact technical support.

CICS Messages

FSIVSHAR-601 JOB <jobname> WAS CANCELLED BY OPERATOR <user-id> FROM TERMID <termid> <date/time>.

Explanation: The batch job was canceled from the online system via the active job display screen. The user-id is the user identifier field from CICS which is assigned at sign on time and termid is the terminal identifier from CICS.

System Action: Abend batch job.

User Response: Informational message only.

FSIVSHAR-602 ON-LINE END-OF-SESSION <normal/abnormal> <date/time>.

Explanation: The batch TP indicated that the session was to be ended, usually at termination of the batch job. <normal/abnormal> indicates that the batch job was ending normally or abending. Normal termination includes committing changes since the last syncpoint but abnormal termination does not.

System Action: Abend batch job.

User Response: Informational message only.

FSIVSHAR-603 <user/auto><syncpoint/rollback><succeeded/failed> IN THE ON-LINE PARTNER <date/time>.

Explanation: A syncpoint or rollback was attempted by the online task. It was initiated either automatically or by request from the application program. The request was either processed successfully or failed.

System Action: Continue unless failed.

User Response: Informational message only.

FSIVSHAR-604 ON-LINE FILE RESOURCES HELD LEVEL HAS REACHED <nnnnnn>.

Explanation: This is a warning indicating that possibly excessive file resources are being held by the task running in CICS. It can occur if many updates have been performed without execution of a syncpoint, either user initiated or automatic.

System Action: Continue normally.

User Response: This is a warning only. If online response is being degraded, review the syncpoint options for the job and adjust accordingly.

FSIVSHAR-605 ON-LINE FILE RESOURCES FREED BY <user/auto> <syncpoint/rollback>.

Explanation: Excessive file resources were being held within CICS. The bottleneck was relieved by a syncpoint or rollback initiated either automatically or from the application program.

System Action: Continue normally.

User Response: Informational only.

FSIVSHAR-606 SYNCPOINTS PERFORMED - USER: nnnn AUTO: nnnn ENQ FREED: nnnnn.

Explanation: At job completion, VSHARE displays the number of automatic syncpoints taken, the number of user syncpoints taken, and the number of syncpoints taken to free unused enqueue elements in CICS.

System Action: Informational only.

User Response: None.

Messages and Codes

- FSIVSHAR-607** **USER ROLLBACK FOLLOWED AUTO SYNCPOINT, ROLLBACK IGNORED.**
- Explanation:** The application program requested a user syncpoint immediately following a rollback. Since there are no changes to commit, the rollback is ignored.
- System Action:** Continue normally.
- User Response:** Informational only.
- FSIVSHAR-608** **CURRENT CPU/LPAR ID is <xxxxx>.**
- Explanation:** This program has determined the identity of the CPU that it is running on.
- System Action:** None.
- User Response:** Informational message only.
- FSIVSHAR-610** **FILE TYPE UNSUPPORTED, DDNAME= '<ddname>' FCT= '<fctname>', RETURN CODE = <xxxx>, ignored.**
- Explanation:** The file identified by batch DDNAME and CICS FCT entry name is a type which is unsupported at this time.
- System Action:** Abend batch job.
- User Response:** Check the file definition to ensure it is a supported type. If it is, a storage violation may have occurred. If the problem persists, contact CSI technical support.
- FSIVSHAR-615** **CMS INITIALIZATION HAS BEEN STARTED <date/time>.**
- Explanation:** The CICS CMS initialization program has executed the necessary request to start the cross memory service for VSHARE.
- System Action:** None.
- User Response:** Informational message only.
- FSIVSHAR-616** **CMS INITIALIZATION HAS BEEN COMPLETED <date/time>.**
- Explanation:** The CICS CMS initialization program has complete the necessary request to start the cross memory service for VSHARE.
- System Action:** None.
- User Response:** Informational message only.
- FSIVSHAR-617** **ABEND EXIT ENTERED WITH ERROR CODE '<abcode>'.**
- Explanation:** The CICS CMS initialization program has encountered an abnormal condition. The HANDLE ABEND exit has been entered. The "abcode" displayed is the CICS abend encountered.
- System Action:** Program terminated.
- User Response:** Determine from the abend code what caused the failure. Correct failure condition and re-execute the transaction.
- FSIVSHAR-618** **SVC NUMBER FOR CMS INITIALIZATION IS INVALID PROCESSING TERMINATED.**
- Explanation:** The CICS CMS initialization program has encountered an abnormal condition. The SVC number known to the CICS system is not defined in the operating system.
- System Action:** Program terminated.
- User Response:** Determine why the SVC numbers do not match. Correct the discrepancy and re-execute the transaction.

Messages and Codes

FSIVSHAR-619 CMS INITIALIZATION REQUEST REFUSED BY OS, REASON: SEE MESSAGE ABOVE. PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. The request for Cross Memory Services was refused by the operating system. The reason for the refusal is explained in the message displayed above this one in the SYSLOG.

System Action: Program terminates.

User Response: Determine from the above message what caused the refusal. Correct the discrepancy and re-execute the transaction.

FSIVSHAR-620 SYSTEM GETMAIN FOR '<function>' FAILED WITH RC='<xxxx>' PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. A request for CICS storage has failed with a return code of xxxx in the "function" routine.

System Action: Program terminates.

User Response: Determine why CICS refused the request by examination of the return code. Correct the problem and re-execute the transaction.

FSIVSHAR-621 LOCATED DCB: <dcbname>.

Explanation: The CICS CMS initialization program has located the Data Control Block for file "dcbname".

System Action: Processing continues.

User Response: Informational message only.

FSIVSHAR-622 WARNING -- UNSUCCESSFUL FREEMAIN FOR IRC.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. A request to free storage returned with an error condition.

System Action: Processing continues.

User Response: Informational message only. CICS storage may be reduced because of the failed FREEMAIN.

FSIVSHAR-623 <read><write> TO TEMPSTOR QUEUE <queue name> FAILED, PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. A read or write request to the identified temp storage queue has failed.

System Action: Process terminates.

User Response: Determine from associated CICS messages why the read may have failed. Correct the problem and re-execute the transaction.

FSIVSHAR-624 LOAD FAILED FOR MODULE <modname>.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. An operating system load has failed for module "modname".

System Action: Processing continues.

User Response: Determine from associated CICS messages why the load may have failed. The load failure will not terminate the process until after all required loads have been attempted.

Messages and Codes

FSIVSHAR-625 LOAD OF ALL REQUIRED MODULES FAILED. PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. The operating system loads for required cross memory modules have failed. Message FSIVSHAR-624 will have identified the modules not loaded.

System Action: Processing terminates.

User Response: Determine from associated CICS messages why the load may have failed. Correct the condition and re-execute the transaction.

FSIVSHAR-627 EXECUTION of GIVEN SVC FAILED, PROCESSING TERMINED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. The given SVC for Cross Memory Services has not executed properly or returned abnormal conditions upon execution.

System Action: Program terminates.

User Response: Determine from associated message why the SVC execution failed. Correct the problem and re-execute the transaction.

FSIVSHAR-628 CMS INITIALIZATION ALREADY <accomplished/failed>.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. Another attempt to initiate CMS services has been executed. Only one execution per CICS initialization is allowed.

System Action: Processing continues.

User Response: If CMS initialization was accomplished, continue with operations. If CMS initialization has failed previously, cycle CICS or manually clear the Temp Storage queues held.

FSIVSHAR-629 OPEN FOR DFHRPL FAILED WITH RC='<nnnn>' PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. During an open request for the DDNAME DFHRPL a failure was detected.

System Action: Program terminates. Required modules cannot be loaded.

User Response: Determine from associated messages why the open attempt failed. Correct the problem and re-execute the transaction.

FSIVSHAR-630 CLOSE FOR DFHRPL FAILED WITH RC='<nnnn>', PROCESSING TERMINATED.

Explanation: The CICS CMS initialization program has encountered an abnormal condition. During an close request for the DDNAME DFHRPL a failure was detected.

System Action: Program terminates.

User Response: Determine from associated messages why the open attempt failed. Correct the problem and recycle CICS as necessary.

FSIVSHAR-640 INITIAL GETMAIN FAILED FOR <nnnn> BYTES REQUESTED.

Explanation: The online transaction program executed a CICS GETMAIN for an internal data area.

System Action: Abend batch job.

User Response: If the number of hex bytes is not excessive, consult with the CICS system programmer about the availability of storage in the CICS region. If the problem persists and amount of storage requested appears excessive, contact technical support.

FSIVSHAR-641 UNKNOWN INTERNAL REQUEST CODE, <xxxxxx> WAS USED.

Explanation: An internal request code passed from the batch transaction program to the online transaction program is unknown.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations. If the problem persists, contact CSI technical support.

FSIVSHAR-642 REQUEST CODE <xxxxxx> IS INVALID AT START OF SESSION.

Explanation: The internal request code <nnnn> passed from the batch transaction program to the online transaction program is invalid at the start of a session.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-643 INVALID CHARACTERS RECEIVED IN DDNAME: <ddname>.

Explanation: The online program received a request from the batch program that could not be processed since the DDNAME used to identify the target dataset contained invalid characters.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-644 LINK FAILED. PROGRAM NAME '<pname>' IS INVALID.

Explanation: An attempt was made to link to the indicated program but the link failed.

System Action: Abend batch job.

User Response: Check the CICS definitions for the program to see that it is properly installed in the CICS region being accessed.

FSIVSHAR-645 ACKNOWLEDGEMENT MISSING FOR "READY-TO-SYNC" REQUEST.

Explanation: The online program indicated that it was ready to issue a syncpoint but the batch program did not confirm that it was also ready.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-646 MAX ON-LINE RESOURCES HELD LEVEL HAS BEEN REACHED. TASKID '<tasknr>' ABENDING.

Explanation: The batch CICS transaction noticed that VSHARE is holding an abnormally high level of record locks.

System Action: Informational only.

User Response: If this was not expected, consider releasing resource through syncpointing. Refer to the chapter "Recovery" on page 18 for details. This is only informational for a warning mechanism. Too many locks take up CICS storage.

FSIVSHAR-647 **END-SESSION REQUESTED MORE THAN ONCE. TRANSACTION <trandid> ABENDING.**

Explanation: The online program received two requests to end the session from the batch program.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-648 **CMS NOT AVAILABLE IN TARGET CICS REGION.**

Explanation: During step initialization, the batch job was defined to use Cross Memory Services but after connecting to the CICS region, it was found that CMS was not installed in the CICS region.

System Action: Abend the job.

User Response: Initialize CMS in the CICS region and rerun the job. Refer to the section "Install the Cross Memory Services Interface" on page 105.

FSIVSHAR-650 **CMS RECEIVE EXIT TAKEN ON XXXX ABEND.**

Explanation: The CICS cross memory service manager receive exit was called after the abend XXXX was issued.

System Action: Display the message and perform error processing.

User Response: Determine the reason for the abend and perform your recovery procedure before rerunning the job.

FSIVSHAR-651 **<trandid> <termid> <taskid> <connecting/disconnecting> JOB: <jobname> <jobnum> <step> <procstep> <errors??>.**

Explanation: This is the message in the CICS log explaining the starting or completion of a VSHARE batch job. The CICS transaction ID used, the terminal ID, the time, the job name, the JES job number, and the step name are listed in the message.

System Action: Informational only.

User Response: None. This is an audit trail to determine batch activity during the day or job start/stop times.

FSIVSHAR-652 **CICS REGION IS RUNNING ON CPU/LPAR ID(<CPUID>) USING LU62 CONNECTIVITY.**

Explanation: The batch job was defined to use CMS but VSHARE determined that the batch job and the CICS region were running on a different CPU or LPAR. The CICS CPU ID is displayed. Because of this, VSHARE cannot use CMS; therefore, it automatically switches to LU 6.2.

System Action: Switch to LU 6.2 mode.

User Response: None unless the job is supposed to run on the same CPU.

FSIVSHAR-653 **CICS FUNCTION (<function ident>) FAILED WITH "<msg>" ERROR.**

Explanation: A CICS online function as identified has failed with return code translated to "msg".

System Action: Processing continues if function was not critical to the intended operation. Error recovery will be attempted. If failure is critical, processing will terminate with other messages.

User Response: Informational message. Monitor the CICS log and SYSLOG for subsequent messages affecting the continuity of processing.

Messages and Codes

- FSIVSHAR-654** **JOB HISTORY RECORDING TO CONTROL FILE "<file>" FOR JOB <jobname>, JOBNUM J<jobnum>, FAILED.**
- Explanation:** An attempt to write job history to the control file has failed for given job name and number.
- System Action:** Processing continues.
- User Response:** Informational message. Monitor the CICS log and SYSLOG for subsequent messages affecting the continuity of processing.
- FSIVSHAR-655** **GDS SEND TO TRANSACTION PARTNER FAILED, RETURN CODES GIVEN WERE: '<nnnnn>'.**
- Explanation:** The LU62 send request from CICS to the batch application failed with return codes of "nnnn".
- System Action:** Program continues. If during CMS initialization an attempt will be made to switch to CMS services. If not then the transaction will wait upon a receive from the batch application.
- User Response:** Informational message only. This may be a warning of abnormal system conditions that will cause a subsequent failure and termination of the connection.
- FSIVSHAR-661** **APPC FREE SESSION FAILED WITH RC='<nnnnn>'.**
- Explanation:** The CICS online connection received a failing return code from an LU62 request to free the APPC session.
- System Action:** Processing terminates with an abend to force batch application down.
- User Response:** Determine from related messages why the request for session termination failed. Correct the problem and restart the application.
- FSIVSHAR-662** **INITIAL CAPTURE OF INITIATING PARTNER INFO FAILED RETURN CODE WAS: '<nnnnn>'.**
- Explanation:** The CICS online connection received a failing return code from a request for CICS system information on the LU62 session.
- System Action:** Processing terminated in the absence of required information.
- User Response:** Determine from related messages why the request for CICS system information failed. Correct the problem and restart the application.
- FSIVSHAR-666** **ERROR OCCURRED DURING STORAGE MANAGEMENT, RETURN CODE WAS <nnnn>.**
- Explanation:** The CICS online connection received a failing return code from a request for CICS storage.
- System Action:** Processing terminates with sufficient signals to abend the batch application.
- User Response:** Determine from related messages why the request for CICS storage failed. Correct the problem and restart the application.
- FSIVSHAR-667** **ROLLEDBACK CONDITION TRIPPED ON CURRENT SYNCPOINT. PROCESSING TERMINATED.**
- Explanation:** The CICS online connection received a failing return code from a syncpoint request that indicated a rollback condition existed.
- System Action:** Processing terminates with a 0C1 abend.
- User Response:** Determine from related messages what other conditions may have caused the failure. Correct the problem and restart the application.

FSIVSHAR-671 FILE ACCESS FOR DDNAME '<ddname>' FCTENTRY '<fctname>', RETURNED ERROR CONDITION '<nnnn>'.

Explanation: An online I/O module encountered a CICS non recoverable error condition. The error condition will typically coincide with one of the CICS handle condition names. The file for which the I/O was being performed is identified by batch DDNAME and CICS FCT entry name.

System Action: Abend batch job.

User Response: Look up the error condition in the CICS Programmer's Reference and act accordingly. Consult with the CICS system programmer for system related errors (e.g., ISC errors, SYSID errors, etc.).

FSIVSHAR-672 KEY CHANGED ON DUPKEY REPOSITION REQUEST DD=<ddname> FCT=<fctname>.

Explanation: The online KSDS I/O module was sequentially processing duplicate keys on an alternate index. The batch TP requested repositioning amongst the duplicate keys but the key of reference was changed from the prior request. The file for which the I/O was being performed is identified by batch DDNAME and CICS FCT entry name.

System Action: Abend batch job.

User Response: Storage may have been corrupted either on the batch or the online side. Check application for possible storage violations and rerun. If the problem persists, contact CSI technical support.

FSIVSHAR-673 (<function ident>) FAILED ON CONN(<connection>), SESS(<session id>) WITH A "<msg>" ERROR.

Explanation: A CICS communications function, as identified, failed on the identified connection and session. If this is an ALLOCATE, SESS will be blank as no sessions are yet established.

System Action: Processing continues. Error returns will be given to requesting functions signifying lack of communications capability.

User Response: Informational message. Monitor the CICS log and SYSLOG for subsequent messages affecting the continuity of processing. If this is a ALLOCATE that fails with a "SYSIDERR", the most probable cause is in the remote region. The remote region's connection is or has been out of service. The remote regions connection must be re-installed to place it back into service and re-establish its VTAM identity.

FSIVSHAR-674 CONNECTION PROTOCOL DEFINED AS LU61 BUT ACCESS METHOD NOT IRC, VTAM OR XM.

Explanation: A CICS communications function has detected an invalid access method for given connection protocol.

System Action: Processing continues. Error returns will be given to requesting functions signifying lack of communications capability.

User Response: Informational message. Monitor the CICS log and SYSLOG for subsequent messages affecting the continuity of processing.

FSIVSHAR-675 CONNECTION (<id>) IS NOT IN SERVICE, NO SESSIONS CAN BE ALLOCATED.

Explanation: A attempt to establish a session on the given connection has failed because the connection is out of service.

System Action: Processing continues. Error returns will be given to requesting functions signifying lack of communications capability.

User Response: Informational message. Determine why the connection is out of service, correct the situation and rerun the job.

Messages and Codes

FSIVSHAR-676 (<function ident>) FAILED ON CONN (<connection>), SESS (<session id>) WITH A "<msg>" ERROR.

Explanation: A CICS communications function, as identified, failed on the identified connection and session. If this is an ALLOCATE, SESS will be blank as no sessions are yet established.

System Action: Processing continues. Error returns will be given to requesting functions signifying lack of communications capability.

User Response: Informational message. Monitor the CICS log and SYSLOG for subsequent messages affecting the continuity of processing. If this is a ALLOCATE that fails with a "SYSIDERR", the most probable cause is in the remote region. The remote region's connection must be re-installed to place it back into service and re-establish its VTAM identity.

FSIVSHAR-693 VTAM-LU62 ERROR FOUND (CONV RESOURCE FAILED)

Explanation: VSHARE issued a CICS GDS RECEIVE, ISSUE ERROR, or ISSUE ABEND command, but CICS detected an error in the received data stream or session. VSHARE reports the error that CICS returned.

System Action: The batch job ABENDS.

User Response: As part of the problem determination of the underlying issue, locate the return code value (CDBERRCD) in the IBM block DFHCDBLK.

Half Session Messages

FSIVSHAR-701 ILLEGAL RECORD TYPE WAS PASSED TO HS.

Explanation: The Half Session manager was passed a bad record type.

System Action: The Half Session manager returns to the calling program passing a bad return code.

User Response: Possible storage corruption by the application program. Check for potential problems with the application. If the problem persists, contact your technical support group.

FSIVSHAR-702 ADDRESS IN RCB DOES NOT POINT TO A HS TO RM RECORD.

Explanation: The Half Session Manager did not find the address of a HS to RM Record in the RCB.

System Action: The Half Session Manager returns to the calling program with a bad return code.

User Response: Possible storage corruption by the application program. Check for potential problems with the application. If problems persist, contact your technical support group.

FSIVSHAR-703 A RM TO HS RECORD WAS RECEIVED BUT THIS TYPE IS NOT SUPPORTED.

Explanation: The Half Session Manager received a Resource Manager Record but currently this is not supported by the Batch Logical Unit.

System Action: The Half Session Manager returns to the calling program with a bad return code.

User Response: Contact your technical support group.

FSIVSHAR-704 ILLEGAL RECORD CODE WAS PASSED TO HS.

Explanation: The Half Session Manager received a record with a erroneous major code.

System Action: The Half Session Manager returns to the calling program with a bad return code.

User Response: Possible storage corruption by the application program. Check for potential problems with the application. If this problem persists, contact your technical support group.

FSIVSHAR-705 ILLEGAL RECORD CODE WAS PASSED TO HS.

Explanation: While trying to process a record from Presentation Services, an erroneous minor code was found in the record.

System Action: The Half Session Manager returns a bad return code to Presentation Services.

User Response: Possible storage corruption by the application program. Check for potential problems with the application. If the problem persists, contact your technical support group.

FSIVSHAR-706 STRAY SIGNAL RECEIVED. IGNORED.

Explanation: The Half Session Manager received a stray signal from the other Logical Unit.

System Action: The Half Session Manager returns with a bad return code.

User Response: Try operation again. If message persists, contact your technical support group.

FSIVSHAR-707 THE LENGTH OF THE LARGEST BUFFER IS ERRONEOUS. IT IS EQUAL TO ZERO.

Explanation: The Half Session Manager found the length of the largest buffer to be zero.

System Action: The Half Session Manager returns with a bad return code.

User Response: Contact your technical support group.

Messages and Codes

- FSIVSHAR-708 THE LENGTH OF THE LARGEST BUFFER IS ERRONEOUS. IT IS A NEGATIVE.**
Explanation: The Half Session Manager found the length of the largest buffer to be a negative number.
System Action: The Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-709 BAD SEND DATA RECORD TYPE WAS PASSED TO HALF SESSION.**
Explanation: The Half Session Manager was processing a send data record and found the type to be unknown.
System Action: The Half Session Manager returns with a bad return code.
User Response: Possible storage corruption by the application program. Check for potential application problem. If the message persists, contact your technical support group.
- FSIVSHAR-710 CALL FSM_CHAIN_RCV_FMP19 CAUSED A STATE CHECK ERROR.**
Explanation: The Half Session Manager was trying to change the state of the receive Finite State Machine and received a state error.
System Action: Half Session manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-711 THE FSM #8 IS BAD. TRIED A VALID CHANGE, BUT RECEIVED ERROR.**
Explanation: The Half Session Manager was trying to change the state of the Chain Receive Finite State Machine and received a state check.
System Action: Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-712 TRIED TO CHANGE FSM #9 WITH BC BUT RECEIVED PROTOCOL ERROR.**
Explanation: The Half Session Manager was trying to change the state of the Chain Send Finite State Machine and received a state check.
System Action: Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-713 TRIED TO CHANGE FSM #9 WITH EC BUT RECEIVED PROTOCOL ERROR.**
Explanation: The Half Session Manager was trying to change the state of the Chain Send Finite State Machine with an End Chain input and received a state check.
System Action: Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-714 TRIED TO CHANGE FSM #8 WITH A RESPONSE BUT RECEIVED FSM ERROR.**
Explanation: The Half Session Manager was trying to change the state of the Change Receive Finite State Machine with a response input and received a bad return code.
System Action: Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.

Messages and Codes

FSIVSHAR-716 SEND FAILED. RC=XX FDBK2=XX IN SENSE=XXYYYY OUT SENSE=XXYYYY.

Explanation: While issuing a VTAM send, a bad response was received from VTAM. The return code, feedback code, and sense data are displayed in the message.

System Action: Half Session Manager returns with a bad return code.

User Response: Determine the reason of the bad send by looking up the return codes in the VTAM Messages and Codes chapter. If problem is not understood, contact your technical support group.

FSIVSHAR-717 ERROR WITH ACB DURING A VTAM RECEIVE. ACB MAY NOT BE OPENED.

Explanation: After a VTAM receive was issued, the error indicated a problem with the VTAM ACB.

System Action: Half Session Manager returns with a bad return code.

User Response: Contact your technical support group.

FSIVSHAR-718 NON-ZERO SENSE CODE WAS RECEIVED. SENSE = XXXXXXXX.

Explanation: After the receive, the Half Session Manager noticed either a bad sense code received from the other Logical Unit or the Half Session Manager found erroneous information and issued the bad sense code.

System Action: Half Session Manager returns with a bad return code.

User Response: Contact your technical support group.

FSIVSHAR-719 RECEIVE FAILED RC=XX FDBK2=XX IN SENSE=XXYYYY OUT SENSE=XXYYYY.

Explanation: The Half Session Manager received an error from VTAM after issuing a receive.

System Action: Half Session Manager returns with a bad return code and returns the sense information.

User Response: Determine the meaning of the message by referencing the VTAM Messages and Codes chapter. If problems are encountered deciphering the message, contact your technical support group.

The most common occurrences for this message and their causes are as follows:

RECEIVE FAILED RC=04 FDBK2=04 IN SENSE=460000

- 1) VSHARE background transaction, TCCB, has security defined. Refer to the section "PCT Definitions" on page 102 for an explanation on this transaction and security.
- 2) VSHARE background transaction, TCCB, is disabled.
- 3) VSHARE background transaction, TCCB, is not defined.
- 4) VSHARE program CSIVSC00 is not defined.
- 5) VSHARE program CSIVSC00 is disabled.
- 6) VSHARE CICS load library is not defined to the RPL.

FSIVSHAR-720 STRAY RESPONSE WAS RECEIVED. IGNORED.

Explanation: The Half Session received a stray response from the other logical unit.

System Action: Stray was ignored and another receive is issued.

User Response: None.

Messages and Codes

- FSIVSHAR-721 RESPONSE PROTOCOL ERROR DETECTED. SESSION FREED.**
Explanation: The Half Session Manager was trying to update the finite machine and received a protocol error.
System Action: The Half Session Manager returns with a bad return code.
User Response: Call your technical support group.
- FSIVSHAR-722 FUNCTION MANAGEMENT HEADER #X WAS SENT TO THE LU. NOT SUPPORTED.**
Explanation: The Half Session received a function management header from CICS that is not supported by the Batch LU.
System Action: The Half Session Manager returns with a bad return code.
User Response: This should not occur. Contact your technical support group.
- FSIVSHAR-723 TRIED TO DETERMINE TYPE OF RECORD RECEIVED BUT COULD NOT. FATAL.**
Explanation: The Half Session Manager received a record from CICS that it could not identify.
System Action: The Half Session Manager returns a bad return code.
User Response: This should not occur. Contact your technical support group.
- FSIVSHAR-724 STATE ERROR WHILE CHANGING FSM_CHAIN_RCV WITH BEGIN_CHAIN.**
Explanation: The Half Session Manager was trying to change the receive chain Finite State Machine with a Begin Chain input and receive a state error.
System Action: The Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.
- FSIVSHAR-725 STATE ERROR WHILE CHANGING FSM_CHAIN_RCV WITH END_CHAIN.**
Explanation: The Half Session Manager received a state error while trying to change the receive chain finite state machine with an End Chain input.
System Action: The Half Session Manager returns with a bad return code.
User Response: Contact your technical support group.

Resource Manager Messages

FSIVSHAR-800 ILLEGAL RECORD TYPE WAS PASSED TO RM.

Explanation: The Resource Manager received an unidentifiable record type.

System Action: The Resource Manager returns with a bad return code.

User Response: Possible storage corruption. Check the application for a possible error. If the problem persists, call the technical support.

FSIVSHAR-801 ILLEGAL RECORD CODE WAS PASSED TO RM.

Explanation: While interrogating a passed record, the Resource Manager could not identify the record code.

System Action: The Resource Manager returns with a bad return code.

User Response: Possible storage corruption. If the problem persists, call the technical support.

FSIVSHAR-802 HS_IDS DO NOT MATCH. SCB = XXXXXXXX RM RECORD=XXXXXXXX.

Explanation: The Resource Manager received a request from the Half Session Manager, but the Session Control Block ID is not known.

System Action: The Resource Manager returns with a bad return code.

User Response: Possible storage corruption. If the problem persists and a storage corruption has not occurred, call the technical support.

FSIVSHAR-803 ILLEGAL RECORD CODE WAS PASSED TO RM.

Explanation: The Resource Manager received an unknown record type from Presentation Services.

System Action: The Resource Manager returns a bad return code to the Presentation Manager.

User Response: Possible storage corruption. If the problem persists and a storage corruption has not occurred, call the technical support.

FSIVSHAR-804 DEALLOCATE FAILURE. BAD RCBID PASSED.

Explanation: While processing a request to deallocate a session, the Resource Manager received an ID for a unknown Resource Control Block.

System Action: The Resource Manager returns with a bad return code.

User Response: Possible storage corruption. If the problem persists and a storage corruption has not occurred, call the technical support.

FSIVSHAR-805 RCB IS ALREADY ALLOCATED.

Explanation: While processing an allocate request, the Resource Manager noticed that the Resource Control was already in use.

System Action: The Resource Manager Returns with a bad return code.

User Response: This should not occur. Contact the technical support.

FSIVSHAR-806 SYNC LEVEL PASSED IN ALLOCATE RCB IS NOT SUPPORTED.

Explanation: The Resource Manager received a request to allocate a session with an unsupported sync level. The Batch LU does not support any sync point processing.

System Action: The Resource Manager returns with a bad return code.

User Response: This should not occur. Contact the technical support.

Messages and Codes

- FSIVSHAR-807 RCB_STATUS STATE ERROR WHILE CHANGING THE STATE WITH INPUT ALLOCATE_RCB.**
Explanation: While trying to change the state of the Finite State Machine #4 with ALLOCATE_RCB as input, the Resource Manager received a state error.
System Action: The Resource Manager stops processing the Allocation and returns a bad return code.
User Response: This should not occur. Contact the technical support.
- FSIVSHAR-808 SCB_STATUS STATE ERROR WHILE CHANGING THE STATE WITH INPUT GET_SESSION.**
Explanation: While trying to change the state of the Finite State Machine #6 with GET_SESSION, the Resource Manager received a state error.
System Action: The Resource Manager stops processing the Allocation and returns a bad return code.
User Response: This should not occur. Contact the technical support.
- FSIVSHAR-809 DEACTIVATE SESSION ID PASSED IS NOT THE SAME AS THE ID IN THE SCB.**
Explanation: While trying to Deactivate a session for the Presentation Service Manager, the Resource Manager noticed that the session ID passed did not match the ID in the Session Control Block.
System Action: The Resource Manager stops the deactivation process and returns with a bad return code.
User Response: Possible storage corruption. If this problem persists and a storage corruption did not occur, contact the technical support.
- FSIVSHAR-810 ERROR WITH RECORD PASSED FROM LNS. EXPECTED 1,2 BUT RECEIVED X,X.**
Explanation: After receiving control from the LU Network Service Manager, the Resource Manager received a record type denoted by X,X; but was expecting a record type of 1,2.
System Action: The Resource Manager stops processing the request and returns with a bad return code.
User Response: This should not occur. Contact the technical support.
- FSIVSHAR-811 SESSION ID PASSED FROM LNS DURING DEACTIVATION DOES NOT EQUAL SCB ID.**
Explanation: While processing a deactivation request from the LU Network Service Manger, the Resource Manager noticed that the Session Control Block ID passed was not known.
System Action: The Resource Manager stops the deactivation process and returns with a bad return code.
User Response: This should not occur. Contact the technical support.
- FSIVSHAR-812 RCB_IDS DO NOT MATCH IN RCB AND SCB. RCB=XXXXXXXX SCB=XXXXXXXX.**
Explanation: While processing a deactivate request, there was a mismatch between the Resource Control Block ID located in the RCB and the SCB.
System Action: The Resource Manager stops the deactivation process and returns with a bad return code.
User Response: This should not occur. Contact the technical support.

Messages and Codes

FSIVSHAR-813 TRYING TO DEACTIVATE A SESSION THAT IS NOT IN_USE.

Explanation: While processing a deactivate request, the Resource Manager could not find the session in the table.

System Action: The Resource Manager returns with a bad return code.

User Response: This should not occur. Contact the technical support.

LU 6.2 Network Services Messages

FSIVSHAR-900 UNKNOWN MAJOR RECORD TYPE WAS PASSED.

Explanation: The LU Network Manager received an unknown record type.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible storage corruption by the application. Check for possible application error. If the problem persists, contact CSI technical support.

FSIVSHAR-902 UNKNOWN MINOR RECORD TYPE WAS PASSED.

Explanation: The LU Network Manager received a record with an unknown minor code.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible storage corruption by the application. Check for possible application error. If the problem persists, contact CSI technical support.

FSIVSHAR-903 THE DEACTIVATE CORRELATOR PASSED DOES NOT EQUAL VALUE IN LULUCB.

Explanation: The LU Network Manager received an identification of a session to be deactivated that is not equal to the value known by the Manager.

System Action: The LU Network Manager returns with a bad return code.

User Response: This should not occur. Contact the technical support.

FSIVSHAR-904 CAN NOT DEACTIVATE BECAUSE FSM_STATUS DOES NOT INDICATE RIGHT STATE.

Explanation: The LU Network Manager received a deactivate request but the state of the session was not correct.

System Action: The LU Network Manager returns with a bad return code.

User Response: Contact technical support.

FSIVSHAR-905 MODE NAME PASSED IS UNKNOWN. PASSED NAME=XXXXXXXX, MODE=XXXXXXXX.

Explanation: During session activation, the LU Network Manager received a Mode Name that was unknown to the one defined to the Batch LU.

System Action: The LU Network Manager returns with a bad return code.

User Response: Make sure that the Mode Name contained in the control file is identical to the one in the mode entry table. If the problem persists, contact CSI technical support.

FSIVSHAR-906 PARTNER LU NAME PASSED IS UNKNOWN. PASSED=XXXXXXXX, LU NAME=XXXXXXXX.

Explanation: During a session activation, the LU Network Manager received a name of a CICS region not known to the Batch LU.

System Action: The LU Network Manager returns with a bad return code.

User Response: Make sure that the APPLID of the CICS region is correctly located in the control file. If the problem persists, contact the technical support.

FSIVSHAR-907 THE VTAM ACB IS ALREADY OPEN.

Explanation: When attempting to open the VTAM ACB, the LU Network Manager found that the ACB was already opened.

System Action: The LU Network Manager returns with a bad return code.

User Response: Contact technical support.

FSIVSHAR-908 ERROR CREATING THE VTAM EXIT LIST. ERROR = XXXX.

Explanation: While trying to create the VTAM Exit List, the LU Network Manager received an error code. XXXX is the return code from the GENCB Macro.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible region size problem. Increase region size and try the operation again. If the problem persists, contact CSI technical support.

FSIVSHAR-909 ERROR CREATING THE VTAM ACB. ERROR = XXXX.

Explanation: While trying to create the VTAM ACB, the LU Network Manager received an error code. XXXX is the return code from the GENCB Macro.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible region size problem. Increase the region size and try the operation again. If the problem persists, call the technical support.

FSIVSHAR-910 ERROR CREATING THE VTAM NIB. ERROR = XXXX.

Explanation: While trying to create the VTAM NIB, the LU Network Manager received an error code. XXXX is the return code from the GENCB Macro.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible region size problem. Increase the region size and try the operation again. If the problem persists, call the technical support.

FSIVSHAR-911 ERROR CREATING THE VTAM RPL. ERROR = XXXX.

Explanation: While trying to create the VTAM RPL, the LU Network Manager received an error code. XXXX is the return code from the GENCB Macro.

System Action: The LU Network Manager returns with a bad return code.

User Response: Possible region size problem. Increase the region size and try the operation again. If the problem persists, call the technical support.

FSIVSHAR-912 THE OPENING OF THE VTAM ACB FAILED. REASON = XXYY.

Explanation: While opening the ACB, the LU Network Manager received the reason code XXYY.

System Action: The LU Network Manager returns with a bad return code.

User Response: Look up the reason code in the VTAM Programmers Guide and determine the reason. If the problem persists, call the technical support.

FSIVSHAR-913 VTAM FAILURE. RC=XX FDBK2=XX IN SENSE=XXXXYYYY OUT SENSE=XXXXYYYY.

Explanation: While attempting to connect to the CICS region, the LU Network Manager received a bad return code. The return code, feedback code, and the sense code information are displayed in the message.

System Action: The LU Network Manager returns with a bad return code.

User Response: Look up the codes displayed in the VTAM Messages and Codes chapter and determine the cause of the error. If the problem persists, call the technical support.

The most common occurrences for this message and their causes are as follows:

RC=10 FDBK2=01 IN SENSE=08570003

- 1) The CICS APPLID is inactive.
- 2) The CICS ACB is closed in CICS.
- 3) The CICS region is down.

Messages and Codes

RC=10 FDBK2=01 IN SENSE=087D0001

- 1) The CICS APPLID is not defined to VTAM.
- 2) The CICS APPLID defined to a job in the maintenance screen is incorrect.

RC=10 FDBK2=01 IN SENSE=08063426

- 1) The mode table containing the mode entry is not available to CICS and the batch APPLID.
- 2) The mode entry defined to the session is not the same as the mode name defined to the CICS region in VSHARE control file.
- 3) ACF2 external security is turned on for the sessions and connections.
- 4) The batch APPLIDs do not have related session and connection definitions in CICS.
- 5) The connections in CICS were defined as being parallel. They should be single sessions.

RC=14 FDBK2=55 IN SENSE=00000000

- 1) The batch APPLID was defined to not have the authority to acquire a session. Specify ACQ in definition.
- 2) The batch APPLIDs are not defined to RACF. The batch APPLIDs should be defined as users.

RC=10 FDBK2=01 IN SENSE=10033418

- 1) ISC has not been installed in the CICS region.

RC=14 FDBK2=53 IN SENSE=00000000

- 1) The definition for the CICS region in VSHARE System Maintenance Screen was defined with the ACBNAME of the CICS APPLID definition not the label NAME. Refer to the figure explaining the interrelationship between all of the definitions.

FSIVSHAR-914 THE CICS REGION XXXXXXXX WAS NOT AVAILABLE DURING ACTIVATION.

Explanation: While making the connection to the CICS region called XXXXXXXX, the LU Network Manager received a sense code indicating that the region was not available.

System Action: The LU Network Manager returns with a bad return code.

User Response: Make sure that the CICS region is available and that the CICS ACB is open.

FSIVSHAR-915 THE APPLID XXXXXXXX IS ALREADY IN USE.

Explanation: While trying to open the batch APPLID, the LU Network Manager received an error message stating that the APPLID is already in use by another program.

System Action: The LU Network Manager returns a bad return code and stops the activation process.

User Response: Another batch program is using the APPLID. VSHARE allows for multiple APPLID definitions. Consult the Installation chapter for details.

Messages and Codes

FSIVSHAR-916 THE APPLID XXXXXXXX IS NOT DEFINED TO VTAM. CHECK THE DEFINITION.

Explanation: The LU Network Manager was given an APPLID to open that is not defined to VTAM.

System Action: The LU Network Manager returns a bad return code and stops the activation process.

User Response: Refer to the section “Environment Definitions” on page 106 for details on defining the APPLIDS.

FSIVSHAR-917 VTAM FAILURE. RC=XX FDBK2=XX IN SENSE=XXXXYYYY OUT SENSE=XXXXYYYY.

Explanation: While sending a BIS to the CICS region, the LU Network Manager received a bad return code from VTAM. The error information is located in the message.

System Action: The LU Network Manager returns with a bad return code.

User Response: Look up the message in the VTAM Messages and Codes chapter and determine the cause of the problem. If the problem continues, contact the technical support.

FSIVSHAR-918 THE CICS REGION DID NOT SEND A BIS.

Explanation: While waiting for a BIS to be sent from the CICS region during session deactivation, the LU Network Manager received another command.

System Action: The LU Network Manager returns with a bad return code.

User Response: This should not occur. Contact the technical support.

FSIVSHAR-919 TPEND EXIT TAKEN, VTAM HALT MAY HAVE BEEN ISSUED.

Explanation: The VTAM exit routing was entered via an error.

System Action: See other messages.

User Response: Contact Technical Support.

FSIVSHAR-920 NSEXIT ROUTINE WAS ENTERED.

Explanation: A VTAM network services request was received by the LU Network Service Manager. This is usually entered during clean up processing after a session termination.

System Action: The sense information is updated and indicators are set to inform the Half Session Manager that the session has been terminated.

User Response: Other messages should accompany this message. Determine the cause of the session outage either through other network messages or messages written in CICS System Log.

FSIVSHAR-921 LOSTERM EXIT WAS EXECUTED WITH A REASONCODE = XX.

Explanation: The LOSTERM exit was entered. This occurs when the session has been lost for various reasons. The reason code is displayed in the message.

System Action: The sense information is updated and indicators are set to inform the Half Session Manager that the session has been terminated.

User Response: Determine the reason for the lost session by looking up the reason code in the VTAM Programmers Guide. Other messages that may help will be located in the system log or the CICS system log. If the problem continues, contact technical support.

FSIVSHAR-922 TPEND EXIT TAKEN.

Explanation: The TPEND exit was triggered as the result of an operator either issuing a VTAM HALT or VARY NET,INACT command.

System Action: The session is terminated abnormally and the system abends with a user 900.

User Response: Rerun the operation and determine why the session was ended by the operator.

Messages and Codes

FSIVSHAR-9998 WARNING - PRODUCT AUTHORIZATION EXPIRES IN 30 DAYS OR LESS

This message is displayed if the copy of VSHARE you are using is about to expire. Contact your CSI International sales representative for a new Product Key.

FSIVSHAR-9999 PRODUCT AUTHORIZATION HAS EXPIRED

VSHARE is protected with a Product Key. This message is displayed if that Product Key has expired. Contact your CSI International sales representative for a new Product Key.

Batch Abend Codes

U1000

Explanation: This abend code is issued every time an error is detected by VSHARE in module TCCBT001.

System Action: The job abends with a code of U1000.

User Response: Check the message which accompanies the U1000 abend and take the action suggested for the error concerned.

U2000

Explanation: This abend code is issued every time an error is detected after session activation.

System Action: The job abends with a code of U2000.

User Response: Check the message which accompanies the U2000 abend and take the action suggested for the error concerned.

U3000

Explanation: This abend code occurs during syncpoint/rollback interface calls or during CMS processing.

System Action: The job abends with a code of U3000.

User Response: Check the message which accompanies the U3000 abend and take the action suggested for the error concerned.

U0999

Explanation: This abend code is issued during journal recovery processing.

System Action: The job abends with a code of U0999.

User Response: Check the message which accompanies the U0999 abend and take the action suggested for the error concerned.

CICS Abend Codes

TC01

Explanation: An error was encountered by the batch TCC transaction after a receive was issued.

System Action: Abend task with abend TC01.

User Response: Refer to the VSHARE batch log or the CICS log to determine why the VTAM or CMS receive failed.

Module: CSIVSCOn

TC02

Explanation: An error was encountered by the batch TCC transaction after a send was issued.

System Action: Abend task with abend TC02.

User Response: Refer to the VSHARE batch log or the CICS log to determine why the VTAM or CMS send failed.

Module: CSIVSCOn

TC03

Explanation: A “signal” command was received by the VSHARE online transaction. This is not part of the TCC protocol.

System Action: Abend task with abend TC03.

User Response: This is an internal error that should not have occurred. Contact CSI technical support.

Module: CSIVSCOn

TC04

Explanation: The online TCC transaction received two consecutive receives. This is not part of the TCC communications protocol.

System Action: Abend task with abend TC04.

User Response: This is an internal error that should not have occurred. Contact CSI technical support.

Module: CSIVSCOn

TC05

Explanation: The TCC online transaction received a free request before an end session was sent by the batch job.

System Action: Abend task with abend TC05.

User Response: Refer to the batch log to determine if the batch job abended. If the problem persists, contact CSI technical support.

Module: CSIVSCOn

TC06

Explanation: During CMS processing, the online TCC transaction receive an error during a receive.

System Action: Abend task with abend TC06.

User Response: Refer to the VSHARE batch log or the CICS log to determine why the CMS receive failed. If the problem cannot be isolated, contact technical support.

Module: CSIVSCOn

Messages and Codes

TC07

Explanation: The TCC online transaction received a request to use LU 6.2 when the LU 6.2 session was already freed.

System Action: Abend task with abend TC07.

User Response: Refer to the VSHARE batch log or the CICS log to determine if the CMS interface was activated correctly. If the problem cannot be isolated, contact technical support.

Module: CSIVSCOn

TC08

Explanation: During CMS processing, the TCC control areas were corrupted.

System Action: Abend task with abend TC08.

User Response: A storage violation has probably occurred in the CICS region. Determine the cause for the violation. If a storage violation did not occur, contact CSI technical support.

Module: CSIVSCOn

TC09

Explanation: An error was encountered by the online TCC transaction after a CMS resume.

System Action: Abend task with abend TC09.

User Response: Refer to the VSHARE batch log or the CICS log to determine why the VTAM or CMS resume failed.

Module: CSIVSCOn

TC0A

Explanation: After trapping an AICA in the TCC online transaction, the CMS interface determined that the batch job was no longer running.

System Action: Abend task with abend TC0A.

User Response: Determine the cause for the batch job terminating. Possible user abend occurred. If the messages point to VSHARE having problems, contact CSI technical support.

Module: CSIVSCOn

TC10

Explanation: An error was encountered by the online TCC table manager during open validation. A DSIDERR was encountered opening the dataset. Storage must have been detected.

System Action: Abend task with abend TC10.

User Response: Check for possible storage violations in CICS. If nothing can be found, contact technical support.

Module: CSIVSCOn

TC61

Explanation: While trying to find a remotely defined dataset, a request was issued before anything was sent by the TCC online transaction.

System Action: Abend task with abend TC61.

User Response: The TCC online transaction abended or there was an error in the ISC or MRO link used for remote file access. Refer to the CICS log to determine which occurred, and react based on those error messages.

Module: CSIVSCOn

TC62

Explanation: The TCC remote file locator received two consecutive receives from the TCC online transaction. This is not part of the communications protocol.

System Action: Abend task with abend TC62.

User Response: Internal error; contact CSI technical support.

Module: CSIVSCOn