



B I MOYLE ASSOCIATES, INC.

5788 LINCOLN DRIVE • MINNEAPOLIS, MN 55438 • 612-933-2885

BIM-VSHARE/VSE

"The CICS Connection"

VSAM FILE SHARING PRODUCT

INSTALLATION AND OPERATIONS GUIDE

Release 2.5

(COPYRIGHT © 1998, B I MOYLE ASSOCIATES, INC.)

BIM-VSHARE is a proprietary product of B I Moyle Associates, Inc. It cannot be reproduced, changed, copied, or stored in any form (including, but not limited to, copies on magnetic media) without the express prior written permission of B I Moyle Associates, Inc.

This documentation applies to
Release 2.5 of the program product
BIM-VSHARE/VSE.

Original Printing 12/01/98
Last Revised..... 09/21/99

TABLE OF CONTENTS

GETTING STARTED	1
Current Environment	3
The Problem	4
Past Solutions.....	4
How does a DBMS Solve the Problem?	5
BIM-VSHARE	5
Features	6
Ease of Use	7
About This Manual	7
New For This Release	8
PROGRAMMERS GUIDE	9
Overview.....	11
BIM-VSHARE Architecture	11
A Sample BIM-VSHARE Execution	12
Steps to Define a Job to BIM-VSHARE	13
Cross Memory Services Versus LU 6.2.....	14
Not All the Files Have to Be Defined to CICS	14
Accessing Files on a Different CPU	15
VSAM File Security	15
Running a Program in a Shared Environment	15
Recovery.....	16
Performance.....	16
Problem Determination	16
Assembler Support	17
Non-Unique Keys With Alternate Index Files	17
Conclusion.....	18
RECOVERY	19
Overview.....	21
Recovery In a Shared Environment	21
Is Delete, Define, Repro Practical For Recovery?	21
Who Else Can Update My Record?	21
How Long Should I Lock a Record?	21
What Happens If My Program Abends?	22
Locking and Releasing Records Within CICS.....	22
Multiple Programs Updating the Same Record.....	22
BIM-VSHARE 3 Tiers of Recovery	24
Tier 1: No Syncpointing	25
What is Tier 1?	25
What Type of Job Should Use Tier 1?	25
How is Tier 1 Implemented?.....	25
How Are the Files Recovered After an Abend?	26
Tier 2: Implied Syncpointing	27
What is Tier 2?	27
What Type of Job Should Use Tier 2?	27
How is Tier 2 Implemented?.....	27
What is Journaling?	27
Why Use Journaling?	28
What is Backward Recovery?	28
Journaling and Recovery with Tier 2.....	28

TABLE OF CONTENTS

Journaling Example with Tier 2.....	28
The Steps in the Tier 2 Recovery Process.....	29
Define the Job to Utilize Journaling.....	30
Define the Syncpoint Level.....	30
Create a BIM-VSHARE Recovery Procedure.....	31
Creating BIM-VSHARE Recovery Parameters.....	31
What is Journalled and What is not Journalled?.....	32
Recovery Example.....	32
Tier 3: Explicit Syncpointing.....	33
What is Tier 3?.....	33
What Type of Job Should Use Tier 3?.....	33
How is Tier 3 Implemented?.....	33
Determine If BIM-VSHARE Journaling is Required.....	33
Modify The Program to Issue the Syncpoints or Rollbacks.....	34
Batch-Only Mode.....	34
Requesting a Syncpoint or Rollback.....	34
Sample Syncpoint Request.....	34
Issuing Syncpoints and Rollbacks Through A Batch Exit Program.....	36
Syncpoint and Rollback Journal Records.....	38
Create an Application Recovery Scheme.....	39
Tier 3 Recovery Scenario.....	39
Recovery and ESDS Datasets.....	40
ON-LINE CONTROL.....	41
Overview.....	43
BIM-VSHARE Main Menu.....	44
Dynamic Monitoring of Batch Activity.....	45
Active Job Details.....	47
Obtaining Dataset Statistics on Active Jobs.....	48
Defining What Jobs Can Access the Files.....	50
Displaying a List of Jobs Already Defined.....	54
Reviewing History of Past Job Executions.....	55
Detailed History Information.....	57
Detailed Dataset History Information.....	59
Defining CICS Partitions to BIM-VSHARE.....	61
Listing Defined CICS Partitions.....	65
VSAM PERFORMANCE.....	67
Overview.....	69
Concepts.....	69
Storage Mediums.....	70
Synchronous Versus Asynchronous Access.....	71
Basic I/O Concepts.....	72
The Mechanics of a VSAM KSDS I/O.....	73
Local Shared Versus Non Shared Resources.....	74
How to Improve VSAM Performance.....	74
Avoid Control Interval and Control Area Splits.....	75
Reorganize Datasets With Multiple Extents.....	75
Spread Heavily Accessed Datasets.....	76
Use Index and Data Buffering.....	76
Index and Data Control Interval Sizes.....	76
The One Cylinder VSAM Dataset.....	77
Analyze CICS End of Day Statistics.....	77

TABLE OF CONTENTS

Use VSAM Local Shared Resources	78
Beware of Datasets Using Share Option 3 or 4.....	79
Reduce Program I/O	79
Place Critical Files on Faster Storage Mediums.....	80
INSTALLATION	81
Overview for VSE Installation	83
CICS and VTAM Levels	83
Installation for VSE	83
Router Modules	83
Batch Modules	83
CICS Modules	84
Sample JCL and Source Members.....	84
Installing BIM-VSHARE Tape	84
Define BIM-VSHARE Control Dataset.....	84
Definition of Control Dataset.....	85
Update the BIMVSBDS Parameter Member	85
ISC Installation.....	86
Library Containing CICS Phases	86
Environment Definitions.....	86
VTAM Definitions	90
CICS Definitions.....	91
TCT Definitions	91
PCT Definitions	92
PPT Definitions	93
FCT Definition	93
Define the Eligible CICS Partitions and Jobs.....	94
BIM-VSHARE Activation and Deactivation.....	95
File Security.....	97
MESSAGES AND CODES	99
BIMVSBAI Messages	101
On-Line Maintenance Messages.....	101
Batch Messages	106
Journal Messages.....	111
Presentation Services Messages	121
Presentation Messages	133
CICS Messages	140
Half Session Messages	147
Resource Manager Messages.....	150
LU 6.2 Network Services Messages.....	152
BIMVSR-nnn (Routing Facility)	156
CICS Abend Codes	157

TABLE OF CONTENTS

GETTING STARTED

Current Environment

VSAM has been chosen by many companies to be the data access method for their computer applications. The ease of use, ease of understanding, and its efficiency makes VSAM appealing over a more complicated database management system. Unfortunately, VSAM places constraints on the systems using it. The major constraint in VSAM involves limitations on both on-line and batch programs updating the same VSAM file simultaneously with full data integrity. VSE/VSAM provides Share-Option(4), but it effectively eliminates all buffering and other performance characteristics of VSAM, even when only one partition is accessing a SHR(4) file. With the current work environment, this constraint is unacceptable. BIM-VSHARE lifts the constraint by providing VSAM applications simultaneous access to the SAME files with full data integrity.

The single update constraint and VSAMs lack of functionality raises concerns over how VSAM systems fit into the long range data processing strategy. How can these critical VSAM systems fit into the SAA platform or how can these systems be utilized in a distributed environment? BIM-VSHARE provides a solution to these issues by extending VSAM's functionality. It not only allows simultaneous access but it also breaks down the physical CPU barrier by allowing access to any VSAM file within an SNA network.

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.

- MIS Departments have invested millions of dollars and many thousands of hours in VSAM CICS applications.
- Computer applications are being required to provide longer on-line availability because of the current dynamic business environment. Many corporate goals include 24 hour on-line availability.
- Longer on-line availability leads to higher volumes to be processed in a shrinking batch time frame.
- Applications are so constricted in batch cycle processing that any type of error during a cycle means lost on-line availability to the end users resulting 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

VSE/VSAM only permits two partitions to open a file for update simultaneously if the file is specified with SHR(4), but the performance implications of that are so severe that it is rarely justified. Usually files that must be updated by both on-line (CICS) and batch are given update access by CICS only, except for specific periods when they are closed to CICS so batch jobs can update them, then they are again given to CICS and denied update access by batch jobs. This is the classic VSAM conflict of On-line vs. Batch.

VSAM's single update limitation has caused many limitations in your VSAM systems. The on-line and batch conflict has caused many problems in the past.

- No 24 Hour On-Line Availability
- Long Batch Cycles Delaying the Next On-Line Day
- No Batch Processing During the On-Line Day
- Expensive Rewrites to Use a DBMS
- Read Integrity Problems for Batch Reports
- Double the DASD Costs to Support Shadow Databases
- Poor On-Line Response from Background CICS Transactions
- No Access to Remote Files

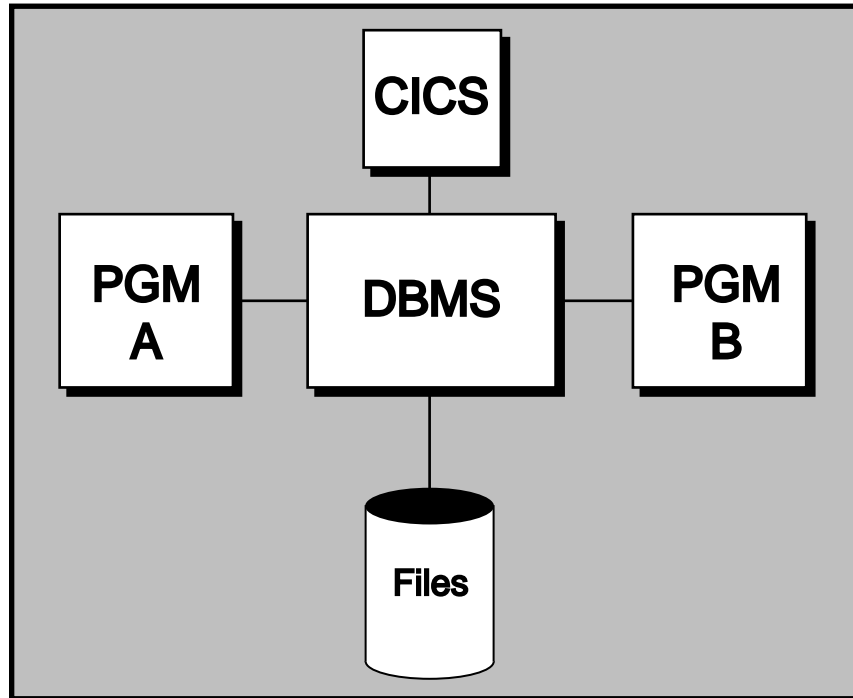
Past Solutions

In the past there have been few options to handle the pressure of longer on-line or batch availability. The solutions were either too time consuming, too expensive, or too awkward to maintain. This single update constraint has forced companies to design VSAM systems based on technical constraints rather than designing systems based on business needs.

- Re-develop the application with a database management system. This solution is timely and expensive.
- Develop CICS transactions to perform batch functions. This results in poor response time for the on-line 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 on-line files for short periods of time. This causes on-line outages and is not really a long term solution.
- Home grown software to access files remotely.

How does a DBMS Solve the Problem?

BIM-VSHARE solves the single update constraint by asking a simple question. How does a database management system 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 database management system runs continuously in its own address space receiving I/O requests from batch jobs or on-line systems. The DBMS in turn performs all the I/O on behalf of these programs. The Database Manager in essence becomes the traffic cop for database access. It manages requests efficiently with full data integrity.



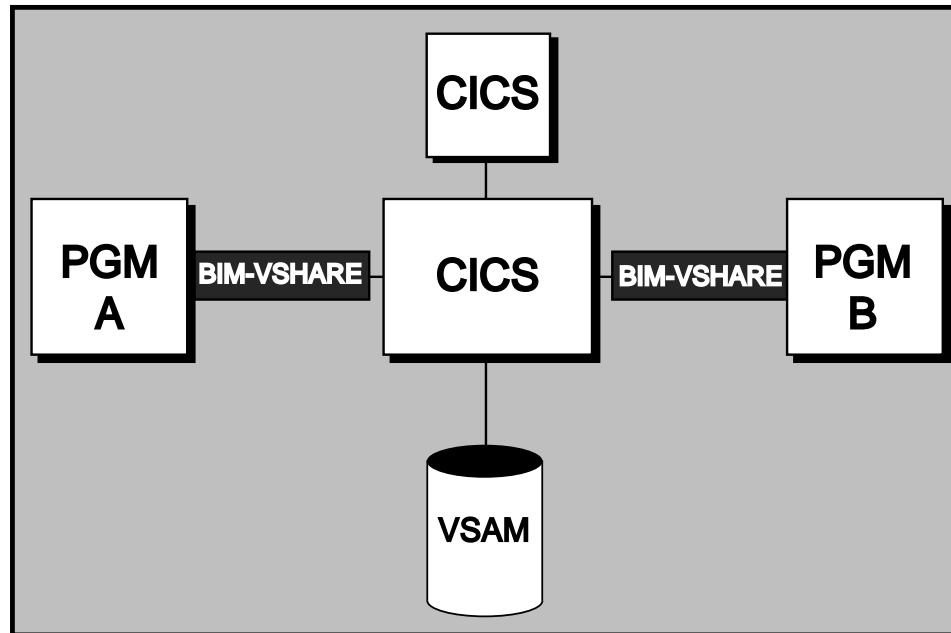
BIM-VSHARE

BIM-VSHARE was designed 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 on-line programs. It centralizes all on-line I/O guaranteeing full data integrity. BIM-VSHARE capitalized on CICS's multiprogramming environment giving the same file sharing capability to batch programs. It provides a transparent vehicle that intercepts all VSAM requests, communicates the request to BIM-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 on-line programs can update the same set of VSAM files. CICS therefore becomes the traffic cop for both on-line and batch programs.

BIM-VSHARE provides two methods of communication between the batch programs and CICS. The first is LU 6.2 which allows remote file access. If the batch job and CICS are running within the same CPU, BIM-VSHARE utilizes its Cross Memory Service interface (XPCC) in order to maximize performance.

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

CICS has been the VSAM traffic cop for on-line programs for decades. BIM-VSHARE provides a vehicle giving batch programs the same advantages of using CICS as the VSAM traffic cop.



Features

BIM-VSHARE was designed to be 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

- On-line 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.
- No new interface to learn.

3) Total Recovery

- BIM-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 on-line programs are lost.

4) Total Control

- You determine which jobs can utilize BIM-VSHARE.
- BIM-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 partition at one time for better control.
- Dynamic control is also provided allowing you the ability 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 the batch job.
- History of past executions of a job can be viewed on-line through BIM-VSHARE on-line system.

Ease of Use

BIM-VSHARE was designed to provide all the functionality required to execute a VSAM system in a multi-programming environment while being easy to install and easy to use.

- Total transparency.
- No source code modifications.
- No recompiling of current code.
- No need for any system software modifications.
- No proprietary address space to support.
- All programming languages are supported.
- No need to learn a new interface. The application programmer codes with the familiar I/O verbs.
- Easy installation, activation, and deactivation.

BIM-VSHARE is a solution not a tool.

About This Manual

BIM-VSHARE provides a complete solution to the VSAM single update constraint. BIM-VSHARE creates a shared VSAM environment by allowing multiple batch and on-line programs to update the SAME VSAM files simultaneously with full data integrity. This manual covers all the information required to install and use BIM-VSHARE. It also addresses issues that arise from executing in a shared environment. Issues like new recovery requirements, the importance of VSAM file tuning, and the other issues with executing a program in a shared environment.

New For This Release

The following changes have been made to this release of BIM-VSHARE:

- The Job Maintenance and System Maintenance Screens of the Online Maintenance facility have been redesigned to more logically group the information being displayed.
- Selection of batch Applids is now controlled with an option and selection mask on the System Maintenance Screen to assist in filtering the available applids to only those applicable to BIM-VSHARE.
- A ROLLBACK control option has been added to more closely manage actions during batch job abort processing. The ROLLBACK option may be controlled at the System or Job level.

PROGRAMMERS GUIDE

Overview

BIM-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 BIM-VSHARE. BIM-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 BIM-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. This includes recovery considerations, resource sharing, performance tuning, and problem determination.

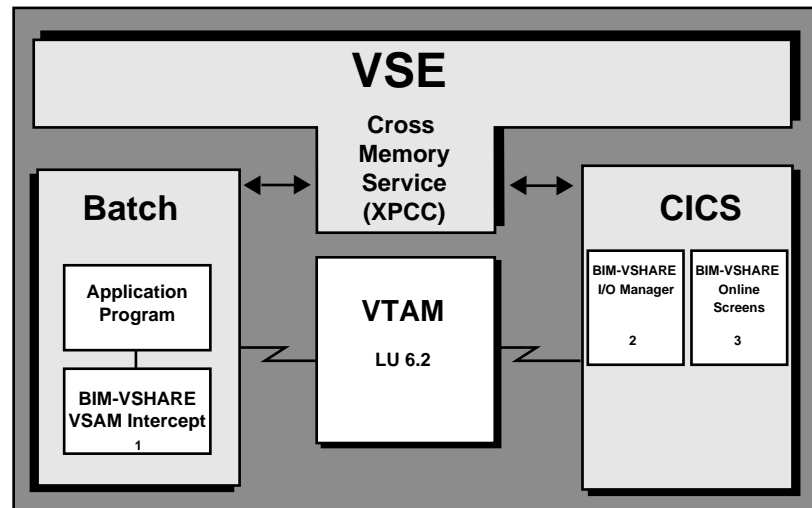
BIM-VSHARE Architecture

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

BIM-VSHARE is broken down into three basic components:

- 1) The first component executes in the batch program's partition. It intercepts the I/O requests and ships them to the second component of BIM-VSHARE residing in CICS.
- 2) The second component of BIM-VSHARE resides in the CICS partition 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 (XPCC). VTAM may be used when the batch job and the CICS partition reside either on the same CPU or on different CPUs. Cross Memory Services provides a performance boost but the batch job and the CICS partition must be on the same CPU.
- 3) The third component of BIM-VSHARE also resides in the CICS partition. It is comprised of a series of on-line screen programs providing the following functions:
 - Define CICS partitions to utilize BIM-VSHARE.
 - Define jobs eligible to utilize BIM-VSHARE.
 - Monitor active jobs currently utilizing BIM-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 partition in the event of heavy on-line activity.

- Review prior executions of batch jobs.



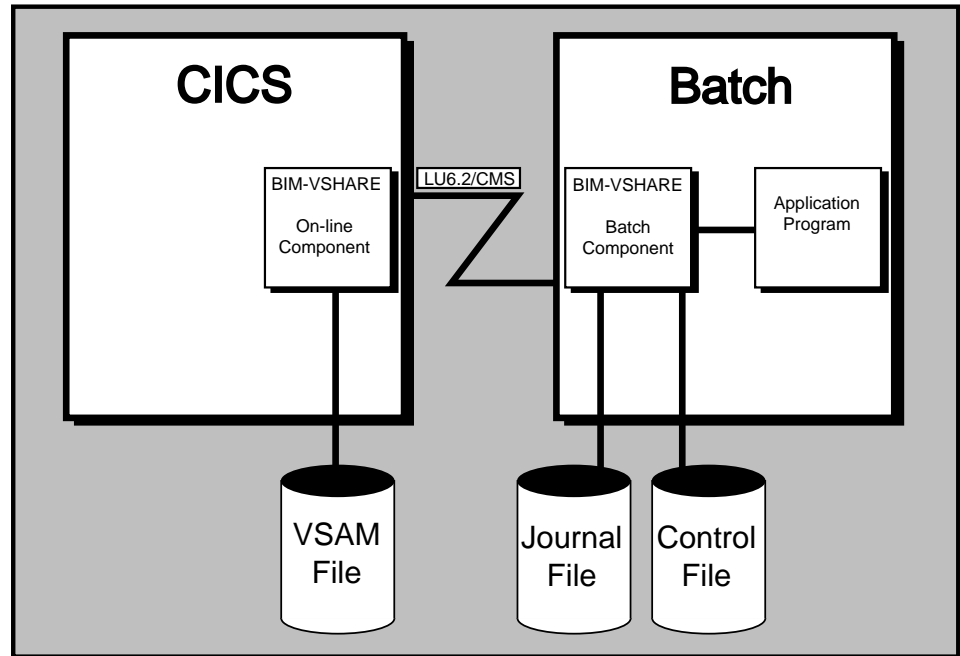
Refer to the chapter “ON-LINE CONTROL” on page 41 for details pertaining to the on-line functions.

A Sample BIM-VSHARE Execution

Following is an example of a program execution using BIM-VSHARE. It describes the steps taken and it gives a better understanding of how BIM-VSHARE executes.

- The program begins execution.
- The program opens a file.
- During the first open of a file, BIM-VSHARE gets control and reads the BIM-VSHARE control file for a job record.
- After the job record is read, BIM-VSHARE retrieves the APPLID of the CICS partition owning the files. This CICS APPLID is located on the job record.
- BIM-VSHARE reads the system record for the CICS partition and retrieves the information necessary to communicate to CICS.
- BIM-VSHARE communicates to the CICS partition and starts the BIM-VSHARE on-line component. If the job is defined to use Cross Memory Services, VTAM is not used for the rest of the job's execution.
- BIM-VSHARE sends the dataset name to the on-line component to determine if the file is allocated to CICS.
- The on-line 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 BIM-VSHARE and shipped to the CICS partition. The CICS component of BIM-VSHARE performs the I/O on behalf of the batch program and returns the records and/or status information.
- When the batch program finishes, BIM-VSHARE stops the CICS component and terminates the communication to the CICS partition.



Steps to Define a Job to BIM-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 BIM-VSHARE without impacting other jobs. Because BIM-VSHARE needs no system software modifications, it is easily deactivated from an application.

Following are the steps needed to define a job to BIM-VSHARE.

- Logon to the CICS partition executing BIM-VSHARE and execute the BIM-VSHARE on-line system (CICS transaction 'TCC'). Select the Job Maintenance option from the main menu. From the Job Maintenance screen a job or set of jobs is defined. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on job definition. Note that 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 utilize BIM-VSHARE, only the name PROD is defined.
- Concatenate the BIM-VSHARE batch load library at the beginning of the PHASE LIBDEF.
- Submit the batch job.

Notice the ease of implementation. Deactivation is accomplished by removing BIM-VSHARE batch load library from the LIBDEF concatenation.

Cross Memory Services Versus LU 6.2

BIM-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 in a different CPU than the CICS partition. 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 utilize PR/SM, the batch job can be executing on one image while the CICS partition is running on another image.

The second communications vehicle that BIM-VSHARE can use is Cross Memory Services (XPCC). Cross Memory Services is a faster communications vehicle than LU 6.2 but there is one limitation. The batch job and the CICS partition has to be executing in the same physical VSE image.

You specify which communications vehicle to use in the job maintenance screen. If the job could possibly be submitted on either the same CPU or a different one than the CICS partition, define the job to use Cross Memory Services. If BIM-VSHARE determines that the batch job and the CICS partition are on different CPUs or PR/SM images, it will automatically utilize LU 6.2.

Not All the Files Have to Be Defined to CICS

During the opening of a file, BIM-VSHARE determines if the file is eligible to be intercepted. It compares the cluster name contained in the DLBL card with the CICS VSAM files allocated. If a cluster is located in the CICS partition 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. BIM-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.

BIM-VSHARE does not require the FCT CICS name to match the filename in the batch program. BIM-VSHARE inquires against the CICS FCT on full cluster name.

Also, a file with an expiration date of 93/333 on its batch job VSE DLBL statement will be ignored by BIM-VSHARE. This provides a method for temporarily excluding a file which is normally processed by BIM-VSHARE. It should be used only if there is certainty that the file is not being updated by CICS during the excluded batch run.

Accessing Files on a Different CPU

Accessing a remote file is easy. First, the remote CICS partition owning the remote file needs to be defined to the batch owning system. This is accomplished through the on-line system maintenance screen. Refer to the chapter “ON-LINE CONTROL” on page 41 for details on how to define a CICS partition to BIM-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 BIM-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 anything with file security. BIM-VSHARE fits into your current security scheme. Before every file is intercepted, the batch component of BIM-VSHARE interfaces with your security system. If the security system determines that the userid associated with the batch job does not have authority to update or read the file, BIM-VSHARE abends the batch job. By reading the BIM-VSHARE log, the dataset causing the problem can be isolated. For more information, refer to the section “File Security” on page 97.

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 with file sharing in mind. Since BIM-VSHARE provides a level of file sharing identical to a database management system, a programmer has to be aware of a new set of issues arising from accessing files in a shared environment.

- 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, what functions can execute simultaneously and what functions can not execute simultaneously?
- Will the records I lock with a read for update impact performance of other programs? How long should a record be locked?

These issues are covered in greater detail in the chapter “RECOVERY” starting on page 19.

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. BIM-VSHARE gives many recovery alternatives to recover from an abend in a shared VSAM environment. The Recovery chapter discusses all the recovery options in great detail.

Performance

BIM-VSHARE was designed with performance in mind. An objective of BIM-VSHARE is to give the programming staff total control over performance. First, if the on-line system needs performance priority, the BIM-VSHARE on-line transaction can be given a lower priority within CICS. In this way, the on-line 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 partition'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 BIM-VSHARE on-line transaction utilizes 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 on-line system will be at a minimum. Fifth, when a batch job is not accessing files owned by the CICS partition, BIM-VSHARE has no impact to your CICS partition. When BIM-VSHARE is not executing you can expect the same on-line throughput achieved today.

Now that CICS programs and batch programs access the same files simultaneously, the need to tune a VSAM file under CICS becomes important. Refer to the chapter "VSAM PERFORMANCE" on page 67 for hints on how to improve VSAM performance within your CICS partition and within your application system. Now that CICS is the database manager of VSAM, 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 partition is tuned for VSAM the faster BIM-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 partition? These questions are important and easy to answer. BIM-VSHARE is designed to provide much information about the Batch/CICS connection. In the event of an abend, perform 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. BIM-VSHARE writes messages to this log when jobs start and complete as well as other messages dealing with Cross Memory Services.
- If BIM-VSHARE forces an abend, one or more messages will be generated explaining why. All of the messages are documented in the Messages and Codes chapter of this manual. 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 if the problem was in CICS or the batch partition. Also look up the sense codes displayed in the VTAM Messages and Codes chapter for more information. It may be necessary to consult a systems programmer since many of the VTAM message explanations are cryptic.
- BIM-VSHARE was designed with 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 on-line screens to aid BIM technical support to solve any problem.

Assembler Support

Since Assembler allows direct access to certain VSAM control areas, some Assembler capabilities are not supported by BIM-VSHARE because the VSAM control areas reside in the CICS rather than batch address space. Following is a list of Assembler functions not supported.

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

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 returned indicating 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.

Conclusion

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

BIM-VSHARE also provides the system programmers with a desirable and familiar architecture. BIM-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 BIM-VSHARE. BIM-VSHARE offers a tiered solution to recovery providing three possible recovery alternatives. The one chosen 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-Repro 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 also programmers 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 Repro 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? BIM-VSHARE provides the capabilities of recovering a file in a shared VSAM environment. In order to explain recovery with BIM-VSHARE, some 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.

The unlocking of the records is accomplished 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.

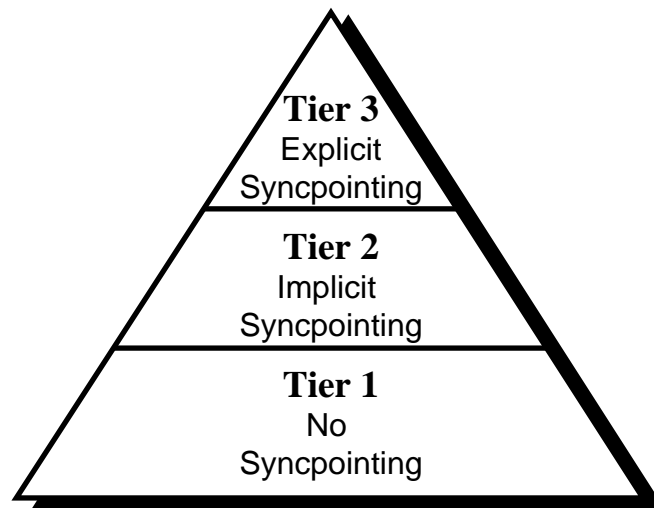
If the first program does not issue a syncpoint until it completes execution, there would be no integrity exposure because all of the program's work is completed before the updates are committed. If the program abends during execution, CICS would back out the changes through DTB with 100% confidence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

This seems the ideal solution. Why worry about recovery? The side effect of not releasing records for a long period of time can have an impact on performance and CICS DSA. Other programs are forced to wait for locked records until they are released, and CICS will consume DSA storage for each record being held. 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, and CICS may run short on DSA storage. For these reasons, 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?

BIM-VSHARE 3 Tiers of Recovery

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. BIM-VSHARE provides three alternatives to recovery. They give you the flexibility to choose the one right for your situation. The method chosen depends on how long the program runs, how many records are updated, and who is impacted by locked records. The first tier provides the ability to lock all the records for the entire job while the second two 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.

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 step completes normally, there is no danger of any other program changing the records updated. 100% data integrity is provided.

What Type of Job Should Use Tier 1?

Any job can theoretically use Tier 1 but there is a practical limit to what 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 step runs for 1 hour updating many records in a critical file, other programs accessing any of the records will wait for up to one hour. For a production on-line 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 (see below, refer to the chapter "ON-LINE CONTROL" on page 41 for details), the field called Sync Level is set to zero. BIM-VSHARE does not issue any syncpoints until the job completes normally. Other than this, there is nothing else to do.

```

+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC15   J O B   M A I N T E N A N C E   TIME: 12:10:08 |
+-----+
| FUNCTION: ( A = ADD, C = CHANGE, D = DELETE ) |
|
| JOB NAME : |
|-----|
| JOB INFORMATION | HISTORY INFORMATION | PROCESSING INFORMATION |
| CICS REGION: BIMCICST | RECORD?: Y | ABEND IF NO CICS?: |
| TRANID: VSCB | GENERATIONS: 002 | DO INPUT FILES?: |
| SYNC LEVEL: 000 |-----| ABEND IF DISABLED?: |
| REQS/MINUTE: 00000 | JOURNAL INFORMATION | TURN TRACE ON?: |
| ACCESS TYPE: LU62 | JOURNAL UPDATES?: N | FORCE ROLLBACK?: N |
|-----|
+-----+

```

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 BIM-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. What this means is that BIM-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. In order 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 BIM-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. BIM-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. Since BIM-VSHARE issues syncpoints periodically, there is no danger of locking out programs for a long period of time. Since BIM-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 BIM-VSHARE recovery backs out the bytes on the record that change, other fields updated on the record will not be 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 BIM-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

BIM-VSHARE contains its own journaling system for Tier 2 recovery. BIM-VSHARE records all the file changes via BIM-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 after images saved, the specific bytes changed are identifiable.

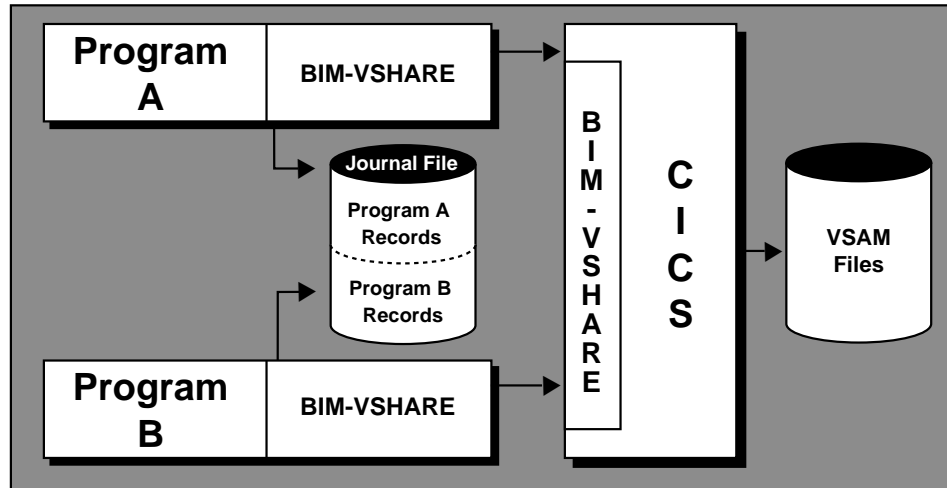
BIM-VSHARE also contains its own recovery system. Added records are deleted and deleted records are added. Updates are recovered at the field level instead of at the record level. The BIM-VSHARE recovery system compares the before and after image from the journal file and restores only those bytes changed.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

Journaling Example with Tier 2

The following example shows the steps taken during the journaling process:

- A job stream starts to execute containing application programs A and B.
- Program A starts to run thus activating the BIM-VSHARE interface.
- BIM-VSHARE activates journal processing. Since program A is the first program in the job stream, the DLBL for the journal dataset (BIMVSHR) should have a DISP=(NEW,KEEP) operand to create a new journal.
- As program A makes I/O requests, BIM-VSHARE calls upon the journaling system to record every add, change or delete. The journaling system writes journal records for each of these requests.
- Program A finishes processing and BIM-VSHARE terminates journaling activity and the journal file is closed.
- Program B starts to process thus activating BIM-VSHARE interface.

- Once again, journal processing is activated by BIM-VSHARE. Since this is the second step, the DLBL for the journal dataset (BIMVSHR) should have a DISP=(OLD,KEEP) operand to add these journal records to the existing journal.
- 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.



The Steps in the Tier 2 Recovery Process

The following example shows the steps taken during 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 BIM-VSHARE recovery job stream is started using the journal dataset.
- The recovery system processes the journal file backwards utilizing BIM-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.

Note that it would also be possible to back out only Program B's changes, and then restart the job at Program B beginning. This is discussed later in this chapter.

Define the Job to Utilize Journaling

Journaling of changes is optional with BIM-VSHARE. The setting of this option is at the job or job group level. Place a Y next to the Journal Updates question on the Job Maintenance screen program.. Refer to the chapter “ON-LINE CONTROL” on page 41 for details on how to define a job to BIM-VSHARE.

BIM-VSHARE journals all changes to a disk journal using a DLBL name of “BIMVSHR”. The journal can be defined as SAM or VSAM-Managed-SAM. The following DLBL is an example of a VSAM-Managed-SAM definition:

```
// DLBL BIMVSHR, 'your.journal.name', 99/365, VSAM,      X
DISP=(NEW,KEEP), RECORDS=(1000,1000),                X
RECSIZE=9000, CAT=IJSYSCT
```

Define the Syncpoint Level

In order for BIM-VSHARE to know when to issue an implied syncpoint, a syncpoint level needs to be defined in the job definition screen. The Sync Level number informs BIM-VSHARE how many locks are permitted before a syncpoint is taken.

```

=====
APPLID:  BIMCICST  BIM-VSHARE - THE CICS CONNECTION      DATE:  11/24/1998
PROGID:  BIMVSC15      J O B   M A I N T E N A N C E    TIME:  12:10:08
=====
FUNCTION:  ( A = ADD, C = CHANGE, D = DELETE )

JOB NAME:

JOB INFORMATION          HISTORY INFORMATION          PROCESSING INFORMATION
CICS REGION: BIMCICST   RECORD?: Y
TRANID:  VSCB          GENERATIONS: 002
SYNC LEVEL: 005
REQS/MINUTE: 00000     JOURNAL INFORMATION
ACCESS TYPE: LU62      JOURNAL UPDATES?: Y
ABEND IF NO CICS?:
DO INPUT FILES?:
ABEND IF DISABLED?:
TURN TRACE ON?:
FORCE ROLLBACK?: N
=====

```

Note that BIM-VSHARE can only issue a syncpoint when there are no read for updates outstanding. If it did issue a syncpoint when a read for update was outstanding, the record read for update would be released and not 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 it is not updated until the end of the job, BIM-VSHARE cannot issue a syncpoint. If messages in the BIM-VSHARE log inform you that extensive locks are being held and you defined the job with implicit syncpointing, this situation may be the cause for the locks being held after the syncpoint threshold has been met.

Create a BIM-VSHARE Recovery Procedure

In the installation library there is sample JCL for recovery called BIMVSBRC.A. During the installation process this JCL is customized; therefore, contact the person responsible for installing BIM-VSHARE for the name of the JCL member or procedure created for recovery. If no customizing was performed, copy the member BIMVSBRC.A from the BIM-VSHARE installation library. The recovery procedure contains three steps.

- Step 1: The first step deletes and re-defines an ESDS VSAM recovery dataset.
- Step 2: The second step loads the sequential journal created during the batch job into the VSAM dataset created in step 1.
- Step 3: The third step executes the BIM-VSHARE recovery program to back out the changes based on the recovery commands.

The recovery program accepts parameters that control the recovery process. The next section explains the format and meaning of these recovery parameters.

Creating BIM-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 BIM-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 BIM-VSHARE, depending upon the application's requirements.

The control cards are 80 characters each and are read by the recovery program. Each control card can contain one recovery parameter. The format of these parameters are as follows:

JOBNAME=XXXXXXXX

This is the VSE job name from the job card. It is useful if journal files are combined as jobs are run.

PARTID=XX

This is the VSE partition id that the job was executed in. It is only required if a journal contains records for multiple job executions and the same job is run in multiple partitions.

PGMSTEP=XXXXXXXX

This is the "step name" assigned by BIM-VSHARE for each step executed in a job. Message BIMVSHAR-201 is generated for each step of a job that has recovery active. This message contains the step name assigned to each step.

PGMNAME=XXXXXXXX

This is the name of the program that was executed for a specific step.

RECOVERALL

This parameter causes the recovery program to process the entire journal file. Other parameter cards will not be accepted with this one since the combination would be inconsistent.

*** Comment**

If the first character of a control card is an asterisk, the line is treated as comments.

Whenever a job step is executed using BIM-VSHARE, a message log is included in the SYSLST 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?

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

Recovery Example

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 the third step. After some analysis it is determined to restart the job at step 3. In order to restart the job at this step, the following steps are taken:

- Add the VSAM datasets updated in step 3 to the recovery JCL.
- Add the following control cards to the recovery JCL:

```
// EXEC PGM=BIMVSJ04
*
* Back out changes through step 3 of ADDCUST
*
JOBNAME=ADDCUST
PGMNAME=PROG3
/*
```

- Run the recovery job.
- Back up the ADDCUST sequential journal if needed.
- Delete the sequential journal.
- Restart ADDCUST at step 3.

Tier 3: Explicit Syncpointing

What is Tier 3?

Tier 3 recovery provides the capability for a program to directly call BIM-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. In order 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 allowing a program to call BIM-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 has to 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 in order to provide full data integrity.

How is Tier 3 Implemented?

The first requirement is to make sure that the Sync Level field in the Job Definitions screen is **set to zero** or else implied syncpointing is invoked. This is not desirable since the program would lose control over syncpoint activity.

Determine If BIM-VSHARE Journaling is Required

If there is no way to determine where the last good syncpoint was taken during recovery, turn BIM-VSHARE journaling on. BIM-VSHARE journals any syncpoint that is taken. The application Tier 3 recovery procedure can read the journal backwards in order 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, BIM-VSHARE journaling is not required and it should be turned off for Tier 3 jobs.

Modify The Program to Issue the Syncpoints or Rollbacks

BIM-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 wish to define their own logical units of work. Requesting either a syncpoint or a rollback via the interface causes the BIM-VSHARE transaction associated with the batch program to execute the respective request under CICS. The request will be executed regardless if file locks are held or not. 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 responsibility of the programmer to insure 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 (i.e., without the BIM-VSHARE interface). In this case, the syncpoint or rollback requests have no effect on program execution.

Requesting a Syncpoint or Rollback

To request a syncpoint or rollback, the program sets up a request parameter list and then calls the BIM-VSHARE application program interface module, BIMVSBAL. On return from BIMVSBAL, the application program should then interrogate the return code area to insure 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 BIMVSBAL.C and BIMVSBAL2.C in the installation library.

Sample Syncpoint Request

```
WORKING-STORAGE SECTION.  
  
(A)      COPY BIMVSBAL.  
         COPY BIMVSBAL2.  
  
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.
```

- A. Copy parameter list and parameter list values into program working storage. See below.
- B. Set request type to syncpoint.

- C. Call the program BIMVSBAI, passing the parameter list.
- D. Display any messages returned.
- E. Check to see that the request was completed successfully. For critical errors, BIM-VSHARE interface will abend the process immediately without returning control to the application program.

```
*****
*   BIM-VSHARE - THE CICS CONNECTION                               *
*                                                                 *
*   BIMVSBA1 - APPLICATION INTERFACE COBOL PARAMETER LIST *
*****
01  CICS-CONN-PARM-LIST.
    05  CCON-VERSION                PIC X(04)  VALUE 'R240'.
    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 BIMVSBA1
```

```
*****
*   BIM-VSHARE - THE CICS CONNECTION                               *
*   BIMVSBA2 - 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 BIMVSBA2
```

Issuing Syncpoints and Rollbacks Through A Batch Exit Program

In the event that the batch program's source code can not be modified to directly issue syncpoints and rollbacks, BIM-VSHARE can be notified to issue a syncpoint or rollback by a user written batch exit program. After each VSAM I/O, BIM-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 BIM-VSHARE, BIM-VSHARE interrogates the parameter list and issues a syncpoint or rollback depending on the request setting.

BIMVSBE2 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-SYNCPOINT 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-PART      PIC X(02).
05 FILLER             PIC X(03).
05 CCON-DATASET-NAME PIC X(44).
```

The copybook BIMVSBE2 provides the layout for the BIM-VSHARE batch exit parameter list. This copybook is located in the installation library. It can be used to copy into your application programs. The Assembler version is called BIMVSBE1.

The user exit can be written in COBOL or Assembler. An example of one 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 DLBL name being processed, the step name, the cluster name, and job number. The exit program can then interrogate the information and determine if 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. BIM-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 note that the exit program is dynamically loaded at execution time by BIM-VSHARE; therefore, each application can have their own exit routine by placing their program named BIMVSBEX at the beginning of the concatenation of load libraries. If an exit program does not exist, BIM-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 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 BIMVSBAI, BIM-VSHARE will be invoked at the beginning resulting in a recursive call.

Example of a Batch Exit Program BIMVSBEX

```

*****
*           L I N K A G E   S E C T I O N           *
*****
LINKAGE SECTION.
  COPY BIMVSB2.
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.

```

Syncpoint and Rollback Journal Records

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

```

*****
*
*   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
JCRLDSN  DS   CL49        RESERVED
          DS   CL44        BASE CLUSTER NAME
          DS   CL45        RESERVED
*
*   CONTROL INFORMATION
*
JCRLCNTL EQU  *           START OF CONTROL INFO
*...STOP KEY INFO
JCRLPART DS   CL2         JOB PARTITION
          DS   CL3         RESERVED
JCRLJOB  DS   CL8         JOBNAME
JCRLPGMN DS   CL8         EXEC PGM= NAME
JCRLPGMS DS   CL8         EXEC PGM= STEPNAME (GENERATED)
          DS   CL3         RESERVED
*...RECORD IDENTIFICATION
JCRLDDN  DS   CL8         FILE DDNAME
JCRLDATE DS   PL4         00YYDDDF (JULIAN UNSIGNED)
JCRLTIME DS   CL4         HHMMSS (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 a way 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. BIM-VSHARE's journal can be utilized 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

Following is what occurs 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 a data exception.
- BIM-VSHARE detects the abend and issues a Rollback in CICS. CICS in turn backs out the 5 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 committed to the files. BIM-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 it 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. In Tier 1, CICS DTB does not back out records added to an ESDS dataset. In Tier 2, the BIM-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.

ON-LINE CONTROL

Overview

BIM-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 on-line programs provided with BIM-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 BIM-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.

BIM-VSHARE on-line programs provide four basic functions:

- 1) Monitor active jobs utilizing BIM-VSHARE.
- 2) Define jobs that can utilize BIM-VSHARE.
- 3) Browse statistics on previously executed jobs.
- 4) Define all the CICS partitions running BIM-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 BIM-VSHARE on-line system, sign on to the CICS partition where BIM-VSHARE is executing and enter the transaction TCC. The first screen displayed is BIM-VSHARE Main Menu.

All BIM-VSHARE screens have the same format. The heading of each screen contains the CICS applid, the BIM-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 information messages are displayed. For an explanation of the valid attention keys, always read the message line for details.

BIM-VSHARE Main Menu

```
-----+-----
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC12      M A I N   M E N U               TIME: 12:31:41 |
|-----+-----|
|                                     |
|          1      ACTIVE JOB DISPLAY  |
|          2      JOB MAINTENANCE    |
|          3      JOB HISTORY        |
|          4      SYSTEM MAINTENANCE |
|                                     |
|          ENTER SELECTION =====>|
|-----+-----
```

The BIM-VSHARE on-line screen program is executed by entering the transaction id TCC. The main menu allows the selection of the four basic functions:

1. ACTIVE JOB DISPLAY

- a) View statistics of currently executing jobs.
- b) Cancel a job utilizing BIM-VSHARE.
- c) Temporarily suspend a job.
- d) Restart a job previously suspended.
- e) Change the number of I/Os being sent to CICS by a batch job.

2. JOB MAINTENANCE

- a) Define a new job eligible to utilize BIM-VSHARE and it's processing options.
- b) Change processing options of a job previously defined to use BIM-VSHARE.
- c) Delete a job previously defined to use BIM-VSHARE.

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 partition running BIM-VSHARE.
- b) Change default processing options of a CICS partition previously defined to use BIM-VSHARE.
- c) Remove a partition from using BIM-VSHARE.

Dynamic Monitoring of Batch Activity

```

+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC13   A C T I V E   J O B S               TIME: 12:32:49 |
+-----+
|  CMD  JOB NAME  STEP NAME  REQS/  REGION  START  VSAM  ACCESS |
|      |            |          | MINUTE |        | TIME  | REQUESTS | STATUS | TYPE |
+-----+
|  -   | PRODJOB1  | STEP5122 | 00000 | PRODCICS | 08:51:22 | 00057080 | ACTIVE | LU62 |
|  -   | PRODJOB4  | STEP5510 | 04001 | PRODCICS | 08:55:10 | 00002301 | ACTIVE | CMS  |
|  -   | USER001C | STEP0112 | 01101 | PRODCICS | 09:01:12 | 00030704 | ACTIVE | CMS  |
|  S   | MASTERUP  | STEP0156 | 00000 | PRODCICS | 09:01:56 | 00021000 | ACTIVE | LU62 |
|  -   | TESTJOB1  | STEP1121 | 00000 | PRODCICS | 09:11:21 | 00002306 | ACTIVE | CMS  |
|  -   | PRODFIX2  | STEP1534 | 00000 | PRODCICS | 09:15:34 | 00001106 | ACTIVE | LU62 |
|  -   | CC045RMA  | STEP2429 | 00000 | PRODCICS | 09:24:29 | 00001002 | ACTIVE | CMS  |
|  -   | SYS1221A  | STEP3217 | 00000 | PRODCICS | 09:32:17 | 00000454 | ACTIVE | CMS  |
|  -   | ITIDKVMA  | STEP4034 | 00010 | PRODCICS | 09:40:34 | 00000059 | ACTIVE | CMS  |
+-----+
|  CMD: S=SEL X=STOP/STRT R=REQ C=CANCEL   PF7=BACK   PF9=HOME |
+-----+

```

The active job summary screen displays all the currently running BIM-VSHARE jobs attached to the CICS in which this transaction is run. You will not see BIM-VSHARE jobs that are attached to other CICS systems.

This screen allows certain processing options to be changed dynamically and it allows the monitoring of the VSAM shared environment. Each detail line on the screen displays information about each job. If more detailed information on a job is desired, enter an S in the command field next to the job and press Enter. This displays the Active Job Detail Screen.

From this screen a job can be stopped, restarted, canceled, or modified with a new I/O rate. These operations can be 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 described below.

CMD: 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, by placing an X in all the CMD fields all the jobs displayed on the screen are stopped.

- S Displays a new screen called the detail job screen. The detail job screen displays more detailed information about the job step chosen.
- X Sends a request to BIM-VSHARE to stop or restart a job based on its current status.
- R Allow the operator to change the batch I/O request rate. By entering an R and pressing Enter, the request rate field becomes unprotected allowing the value to be changed.
- C Send a request to BIM-VSHARE to cancel a batch job.

JOB NAME: The name of the job as it appears on the VSE JOB card.

STEP NAME: This value is generated by BIM-VSHARE. It allows recovery to a specific job step. Refer to message BIMVSHAR-201.

<u>REQS/MIN:</u>	The maximum number of I/O requests sent per minute to the CICS partition. After R is placed in the command line and Enter is pressed, the field is unprotected. Enter any number representing the new rate. If zero is entered, the batch job sends requests as quickly as possible.
<u>REGION:</u>	The APPLID of the CICS partition executing BIM-VSHARE. Note that this is not necessarily the same as the CICS job name.
<u>START TIME:</u>	The start time of the step being displayed.
<u>VSAM REQUESTS:</u>	The total number of VSAM requests processed in the job step.
<u>STATUS:</u>	The status of the job executing. This message changes as commands are entered from this screen. Possible values follow.
ACTIVE	The batch job is active with no pending commands.
STOPPED	The batch job has been suspended. The job keeps this status until another command is entered.
START PEND	The starting of the job is pending.
C PENDING	A cancel was entered but the batch job has not received the request.
UPDATE REQ	This message is displayed after an R is placed in the command line. At this point the request rate field is unprotected.
REQ UPDATE	The new request rate has been entered but the batch job has not processed the new rate.
<u>ACCESS TYPE:</u>	The type of communications used to communicate between batch job and the CICS partition. CMS is the quickest communications type but it can only be used when both the batch job and the CICS partition are executing on the same CPU. LU62 - LU 6.2 is being utilized CMS - Cross Memory Services is being utilized

Attention Keys

<u>ENTER:</u>	Process the entered command(s) or refresh the screen with up to date information.
<u>CLEAR:</u>	Return the operator to the main menu.
<u>PF7:</u>	Scroll backward through the active job list.
<u>PF8:</u>	Scroll forward through the active job list.
<u>PF9:</u>	Display the beginning of the active job list.

Active Job Details

```

+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC14   A C T I V E   J O B S               TIME: 05:49:01 |
+-----+
|
| JOB NAME: MASTERUP           STEP NAME: STEP1121           PART ID: BG
|                               CICS NAME: PRODCICS           START TIME: 09:11:21
| USER SYNCPOINTS: 0002000   AUTO SYNCPOINTS: 000000
| DATASET LIST
|
| _ CUSTOMER   _ SHARE           S MASTER
|
+-----+

```

Based on the selection made on the active job 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 giving dataset specific information. The information displayed is explained below.

JOB NAME: The job name as it appears in the VSE JOB card of the batch JCL.

STEP NAME: Generated by BIM-VSHARE.

CICS NAME: The CICS APPLID of the partition where BIM-VSHARE is executing.

PART ID: VSE partition id of the active job.

OF DELETES: The number of records deleted in the current 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

The dataset statistics screen displays information about the dataset chosen from the detailed active job screen. Along with the number of VSAM operations performed on the dataset, dataset characteristics are also displayed. Note that if this or any other active job screen is entered and the job step displayed 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 57 or the section “Defining What Jobs Can Access the Files” on page 50 for details. The information display is defined below.

DDNAME: The DLBL/DDNAME of the dataset as it appears in the batch JCL.

FCTNAME: The CICS FCT name of the dataset. Notice that the DDNAME and the FCT name do not have to match. BIM-VSHARE matches the two by 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, BIM-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.
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.

If any of the above letters are not displayed, the operation is not allowed against the dataset.

LSR POOL: The number of the LSR pool where the dataset is defined. Refer to the chapter “VSAM PERFORMANCE” on page 67 for details on LSR in CICS. If NSR is utilized, 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; therefore, if a dataset is defined to have 5 strings then 5 different operations are allowed on the file at one point in time.

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.

```
+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC18  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 )
|
+-----+
|
+-----+
```

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 job screen. If more than one DDNAME was selected on the previous detail active job screen, display the chosen DDNAME's statistics.

Defining What Jobs Can Access the Files

```

+=====+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC15   J O B   M A I N T E N A N C E       TIME: 12:46:47 |
+=====+
| FUNCTION: ( A = ADD, C = CHANGE, D = DELETE )          |
|                                                       |
| JOB NAME:                                             |
|-----|-----|-----|
| JOB INFORMATION | HISTORY INFORMATION | PROCESSING INFORMATION |
| CICS REGION: BIMCICST | RECORD?: Y | ABEND IF NO CICS?: |
| TRANID: VSCB | GENERATIONS: 002 | DO INPUT FILES?: |
| SYNC LEVEL: 000 |-----| ABEND IF DISABLED?: |
| REQS/MINUTE: 00000 | JOURNAL INFORMATION | TURN TRACE ON?: |
| ACCESS TYPE: LU62 | JOURNAL UPDATES?: N | FORCE ROLLBACK?: N |
|-----|-----|-----|
|                                                       |
+=====+

```

The Job Maintenance program (option 2 on main menu) allows the definition of the jobs or job groups that can access the CICS system that owns the shared VSAM files. This screen allows the control over what jobs can access the VSAM files. This definition process allows the tailoring of job processing options like the type of recovery used or the communications type to use.

Each field in the Job Information and Journal Information headings have CICS partition 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.

Note that when the default values are changed on the system definition screen, all jobs defined to that CICS partition obtain the changed default value. In this way all of the jobs do not have to be changed when a global change is desired.

If the job name is forgotten, enter a command along with the starting character of the job name. BIM-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 defined below.

FUNCTION: Determine which function to perform. The possible values are:
 A - Add a new job to BIM-VSHARE.
 C - Change the values of a previously defined job.
 D - Delete a job from BIM-VSHARE control file.

JOB NAME: The partial or full name of the VSE job(s) eligible to utilize BIM-VSHARE. Note that the VSE job name defined to BIM-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 utilize BIM-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.

JOB INFORMATION

CICS REGION: The CICS APPLID where BIM-VSHARE is defined and where all the files utilized by the job are defined.

TRANID: This field informs BIM-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 partition to have their own BIM-VSHARE transaction IDs for charge back purposes. If each application has its own tranid, CICS charge back for batch I/Os can be charged using current on-line charge back systems. Make sure that each tranid executes BIM-VSHARE program BIMVSC00. Refer to the section "PCT Definitions" on page 92 for details.

SYNC LEVEL: This field controls the implicit (automatic) syncpoint feature. It is the number of CICS file enqueues allowed before BIM-VSHARE attempts a syncpoint. Note that this is only a threshold level. Before BIM-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 the implicit syncpoint feature is used by the application program with Tier 3 recovery or if no syncpointing is used with Tier 1 recovery.

REQS/MINUTE: The maximum number of I/O requests that the batch job can send to the CICS partition per minute. If 0 is entered, the batch job sends the requests as quickly as possible. Note that BIM-VSHARE gives the CICS programs priority. Therefore, if a CICS program needs to be serviced, CICS services the on-line program before the batch job. For this reason, this field needs a non-zero value only when the CICS partition is very active.

ACCESS TYPE: This is the communications type used to send the VSAM requests between the batch job and BIM-VSHARE transaction. The possible values can be LU62 for LU 6.2 communications and CMS for Cross Memory Service usage. Note that CMS can only be used when the batch job and the CICS partition are executing under the same VSE system. LU 6.2 can be used either when the batch job and CICS are on the same VSE system or on different VSE systems. If CMS is chosen and BIM-VSHARE determines that the batch job and the CICS partition are on different VSE systems, BIM-VSHARE automatically

utilizes LU 6.2. CMS is the faster access type; therefore, if a batch job is executing in the same CPU as the CICS partition, choose CMS.

HISTORY INFORMATION

RECORD?: The possible values are Y or N. This informs BIM-VSHARE to keep history information about the execution of the job.

GENERATIONS: The number of generations of history desired. Note that there is one history record in BIM-VSHARE Control File per step. Therefore, if 5 generations are entered and there are 6 steps in the job, only 5 records will appear in the control file. The first step will not be there.

JOURNAL INFORMATION

JOURNAL UPDATES?: A Y informs BIM-VSHARE to journal all changes made during the jobs execution. By specifying Y, all changes can be backed out in the event of an abend. An N informs BIM-VSHARE not to journal file changes. Refer to the chapter "RECOVERY" on page 19 for more details on BIM-VSHARE journaling.

PROCESSING INFORMATION

ABEND IF NO CICS?: If a Y is entered and the job is executed while CICS is not running, BIM-VSHARE will abend the job. If this option is defined with a N and CICS is not running, BIM-VSHARE passes control directly to VSAM.

ABEND IF DISABLED?: If a Y is entered, BIM-VSHARE abends the batch job if any files opened by the batch job are disabled in the CICS partition. If an N is entered, BIM-VSHARE will not abend the job. Instead the job will allocate the file directly.

DO INPUT FILES?: If a Y is entered, BIM-VSHARE reads all files open for input through the CICS partition. If N is entered, the batch job reads the VSAM file directly.

TURN TRACE ON?: This should always be set to a N unless BIM Technical Support requests a trace for problem resolution. This option informs BIM-VSHARE in the batch partition to trace certain events during a program's execution.

FORCE ROLLBACK?: If Y is entered, a ROLLBACK will be forced if the batch job abends during processing.

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: BIMCICST  BIM-VSHARE - THE CICS CONNECTION  DATE: 11/24/1998 |
| PROGID: BIMVSC11  J O B  M A I N T E N A N C E  TIME: 12:55:12 |
+-----+
|
| JOB NAME: A
|
| CMD          JOB NAME          DATE          USER ID
|
|  -           BIMJOB0          05/08/1998
|  -           BIMJOB1          05/08/1998
|  -           BIMJOBG          05/08/1998
|  S           BIMJOBH          05/08/1998
|  -           BIMSJJT2         02/16/1998
|  -           BIMSJJT         08/11/1998
|  -           BIMVSBJX         10/06/1998
|  -           BIMVSMXT         08/14/1998
|  -           BIMVSR1          07/07/1995
|  -           BIMVSR3          07/31/1995
|
+-----+
| PLACE S FOR SELECTION AND PRESS ENTER.  PF7=BACK  PF8=FORWARD  PF9=HOME |
+-----+

```

If an incorrect job name is entered in the detail job summary screen, BIM-VSHARE displays this screen with a list of jobs starting with the characters of the job name entered. This list can be scrolled to find the desired job. By placing a 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 jobs defined.

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

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 detail 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 detail job maintenance screen.

PF7: Scroll backward in the list.

PF8: Scroll forward in the list.

PF9: Displays the list starting with the name entered in the JOB NAME field.

Reviewing History of Past Job Executions

```

+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC16   J O B   H I S T O R Y   S U M M A R Y   TIME: 12:56:48 |
+-----+
|
| JOB NAME: A
|
|  CMD      JOB NAME      STEP NAME      PART ID      DATE      TIME
|  ---      ---          ---          ---          ---      ---
|  -        BIMVSMXT    STEP2449      F5           10/22/1996 15:24:51
|  -        BIMVSMXT    STEP4918      F5           10/24/1996 13:49:20
|  -        BIMVSR1     STEP1320      F5           02/13/1998 13:13:21
|  -        BIMVSR1     STEP1949      F6           07/29/1998 11:19:53
|  -        BIMVSR3     STEP3946      F5           07/31/1995 11:39:47
|  -        BIMVSR3     STEP0228      F5           07/31/1995 12:02:29
|  -        MEAVSHR     STEP4310      F7           10/06/1998 12:43:12
|  -        MEAVSHR     STEP4154      F7           10/06/1998 14:41:56
|  -        MEAVSHR     STEP3409      F7           10/07/1998 12:34:12
|  -        MEAVSHR     STEP1943      F7           10/14/1998 13:19:47
|
+-----+
| S + ENTER = DETAIL  S + PF5 = DELETE.    PF7=BACK  PF8=FORWARD  PF9=HOME |
+-----+
    
```

If a job is defined to save history, the job history summary screen is used to browse all the generations saved by BIM-VSHARE. Obtaining the desired history information can be accomplished by choosing option 3 on BIM-VSHARE main menu. The 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. In order 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 What Jobs Can Access the Files” on page 50 for details on how to save history information. The information displayed on the history summary screen is described below.

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 selections are displayed.

JOB NAME: The name of the job as it appeared on the VSE JOB card.

STEP NAME: Generated by BIM-VSHARE.

PART ID: VSE partition id.

DATE: The date the job was executed.

TIME: The start time of the job.

Attention Keys

ENTER: Refresh the screen, display a new list when the JOB NAME field is changed.

<u>CLEAR:</u>	Return to the main menu.
<u>PF7:</u>	Scroll backward through the list.
<u>PF8:</u>	Scroll forward through the list.
<u>PF9:</u>	Display the list starting with the job contained in the JOB NAME field.

Detailed History Information

```

+-----+
| APPLID: BIMCICST  BIM-VSHARE - THE CICS CONNECTION  DATE: 11/24/1998 |
| PROGID: BIMVSC17  J O B  H I S T O R Y  D E T A I L  TIME: 12:57:45 |
+-----+
| JOB NAME:  MEAVSHR          PART ID:  F7          DATE: 10/14/1998 |
| STEP NAME:  STEP1943          TIME: 13:19:47 |
|
| JOURNAL INFORMATION:  UPDATES JOURNALED?  N |
|
|
|
| SYSTEM INFORMATION:  CICS REGION: BIMCICST          REQ/MIN: 00000 |
|                   FILES OPENED: 001          ACCESS TYPE: CMS |
|                   USER SYNCPOINTS: 0000000    AUTO SYNCPOINTS: 0000001 |
|                   TOTAL VSAM REQUESTS: 00000001 |
|
| DDNAMES:  _  MOSSFIL |
+-----+
| SELECT A DDNAME AND PRESS ENTER.          PF7=BACK  PF8=FORWARD  PF9=HOME |
+-----+

```

Based on the selection made in the history summary screen, this screen displays more details including a list of DDNAMEs opened during execution. By selecting one of the DDNAMEs, a detail screen is displayed giving dataset specific information. The information display on this screen is described below.

- JOB NAME: The name of the job as it appeared on the VSE JOB card.
- PART ID: VSE partition id.
- DATE: The date of the job execution.
- STEP NAME: Generated by BIM-VSHARE.
- TIME: The start time of the step.
- JOURNAL UPDATES?: 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. Refer to the section "Defining What Jobs Can Access the Files" on page 50 and the section "Defining CICS Partitions to BIM-VSHARE" on page 61 for details.
- CICS REGION: The VTAM APPLID of the CICS where BIM-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 allowed to be processed by BIM-VSHARE per minute.
- FILES OPENED: The number of files opened in the step. Note that this is the number of files under the control of BIM-VSHARE. Batch only files are not listed.

ACCESS TYPE: The type of communications utilized during this job's execution. The values can be LU62 when LU 6.2 was used or CMS when the Cross Memory Services interface was used.

USER SYNCPOINTS: The number of syncpoints issued by the user program by calling BIM-VSHARE directly for that purpose. A non-zero value indicates Tier 3 recovery is being used.

AUTO SYNCPOINTS: The number of syncpoints issued by BIM-VSHARE automatically on behalf of the user's program. A value of 1 indicates Tier 1 recovery is being used. A value greater than 1 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 BIM-VSHARE intercepted and issued on behalf of the batch program.

DDNAMES: A list of DDNAMES opened during the step execution and under control of BIM-VSHARE. If more detailed information on a dataset is desired, place an S next to the DDNAME and a detailed DDNAME screen is displayed.

Attention Keys

ENTER: Process the selected DDNAMEs or refresh the screen.

CLEAR: Return to the summary history screen or display the next step selected on the previous summary history 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: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC18  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. Along with the number of VSAM operations performed on the dataset, dataset characteristics are also displayed.

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.
 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.

If any of the above letters are not displayed, the operation is not allowed against the dataset.

LSR POOL: The number of the LSR pool where the dataset was defined. Refer to the chapter "VSAM PERFORMANCE" on page 67 for details on LSR in CICS. If NSR is utilized, the number is zero.

STRINGS: The number of strings defined to this dataset. A string allows one I/O operation at a given time; therefore, if a dataset is defined to have 5 strings then 5 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.

<u># OF ADDS:</u>	The number of adds performed against the dataset in the step.
<u># OF UPDATES:</u>	The number of updates performed against the dataset in the step.
<u># OF DELETES:</u>	The number of records deleted in the step.
<u># OF STARTS:</u>	The number of starts requested by the step.
<u># OF ENDS:</u>	The number of end requests processed.
Attention Keys	
<u>CLEAR:</u>	Return to the detail job history screen. If more than one DDNAME was selected on the previous detail history screen, display the next dataset statistics screen.

Defining CICS Partitions to BIM-VSHARE

```

+=====+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC19  S Y S T E M   M A I N T E N A N C E   TIME: 13:00:01 |
+=====+
| 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: N       |
+-----+-----+
|
|
+-----+-----+
| BATCH APPLIDS:   BIMAPPC1   BIMAPPC2   BIMAPPC3   WBYBTCH1   BIMAPPL |
|                  TR20A      LI2C       LI5C      |
|                  P390CICS   P390CICT   BIMCICSP   P390APP1   P390APP2 |
+-----+-----+
| ENTER=EDIT   CLEAR=EXIT   PF3=PROCESS   PF7=BACK   PF8=FORWARD   PF9=HOME |
+=====+

```

The System Maintenance screen defines what CICS partitions can use BIM-VSHARE. When BIM-VSHARE finds a job definition record in the control file, it retrieves the CICS partition 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 BIM-VSHARE to communicate to the CICS partition. Along with this control information, all of the default journaling and history fields are defined for the CICS partition. These defaults are used during job definition when no override values are entered.

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

FUNCTION: This determines which function to execute. The possible values are:

- A Add a new system definition to BIM-VSHARE.
- C Change the values of a previously defined system.
- D Delete a system definition from BIM-VSHARE file.

CICS APPLID: The full CICS APPLID eligible to utilize BIM-VSHARE.

SYSTEM INFORMATION

<u>TRANID:</u>	The name of the transaction defined to execute BIM-VSHARE program BIMVSC00. Refer to the section “CICS Definitions” on page 91 for details on defining this transaction. Each application in a CICS partition can have its own BIM-VSHARE tranid by defining multiple transactions that execute BIMVSC00 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.
<u>MODE NAME:</u>	The mode entry name as defined in CICS and VTAM. Refer to the section “Environment Definitions” on page 86 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 BIM-VSHARE. SMSNA100 is a definition in the default mode entry table.
<u>ROLLBACK:</u>	If Y is entered, a ROLLBACK will be forced if batch jobs abend during processing..
<u>APPLID SEL:</u>	If Y is entered in this field, and an APPLID selection mask is entered in the SELECT MSK: field, then only those APPLIDS matching the mask will be displayed in the BATCH APPLIDS fields.
<u>ACCESS TYPE:</u>	The system default communications vehicle. This field informs BIM-VSHARE how to communicate between the batch job and the CICS partition. The possible values are LU62 or CMS (Cross Memory Services). Note that even if CMS is defined, LU 6.2 is used to start BIM-VSHARE transaction; therefore, the batch APPLIDs are still required if CMS is solely utilized. CMS can only be used when the batch job and the CICS partition are executing in the same VSE system. LU62 should be used if the batch job and the CICS partition are on different CPUs. CMS provides a faster communications access method between the batch job and CICS.
<u>SYNC LEVEL:</u>	This field controls the implicit (automatic) syncpoint feature. It is the number of CICS file enqueues allowed before BIM-VSHARE attempts a syncpoint. Note that this is only a threshold level. Before BIM-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) are used.

REQ/MIN:

The maximum number of requests that the batch job can send to the CICS partition per minute. If 0 is entered, the batch job sends the requests as quickly as possible. Note that BIM-VSHARE is developed to give the CICS programs priority. Therefore, if a CICS program needs to be serviced, CICS services the on-line program before the batch job. For this reason, this field needs a non-zero value only when the CICS partition 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

RECORD HISTORY:

The possible values are Y or N. This informs BIM-VSHARE to keep history information about the execution of the job.

GENERATIONS:

Enter the default number of generations of history desired. Note that there is one history record added to the BIM-VSHARE Control File per step execution. Therefore, if 5 generations are entered and there are 6 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. This informs BIM-VSHARE to journal all changes made during the jobs execution. By specifying Y, all changes can be backed out in the event of a program abend. Refer to the chapter "RECOVERY" on page 19 for more details on BIM-VSHARE journaling.

BATCH APPLIDS:

BATCH APPLIDS:

The VTAM APPLIDS defined during installation. These are the APPLIDS that BIM-VSHARE opens in order to communicate to the CICS partition through a LU 6.2 link. Refer to the section "Environment Definitions" on page 86 for details on the batch APPLID definitions. If more than a page of APPLIDS are desired, the list can be scrolled forward and backward. BIM-VSHARE allows a maximum of 90 APPLIDS allowed per CICS system. Note that batch APPLIDS are required even if CMS is chosen as the access type. LU 6.2 is always used to start BIM-VSHARE transaction in CICS. It is not until a connection is made to CICS through LU 6.2 that the CMS access is utilized.

Attention KeysENTER:

Perform edits on the fields entered.

<u>CLEAR:</u>	Return to the main menu.
<u>PF3:</u>	Process the desired function.
<u>PF7:</u>	Scroll backward through the APPLID list.
<u>PF8:</u>	Scroll forward through the APPLID list.
<u>PF9:</u>	Display the beginning of the APPLID list.

Listing Defined CICS Partitions

```

+-----+
| APPLID: BIMCICST   BIM-VSHARE - THE CICS CONNECTION   DATE: 11/24/1998 |
| PROGID: BIMVSC11  S Y S T E M   M A I N T E N A N C E   TIME: 13:08:16 |
+-----+
|
|  APPLID: A
|
|  CMD          CICS APPLID          DATE          USER ID
|  --          -
|  --          BIMCICST             03/31/1998
|  --          BIMDOIT             02/03/1998
|  --          BIMDOIT8            03/31/1998
|  --          P390CIC             02/16/1998
|  --          VSE1CICS            08/11/1998
|  --          VSE3CICS            08/11/1998
|
+-----+
| PLACE S FOR SELECTION AND PRESS ENTER.   PF7=BACK   PF9=HOME |
+-----+

```

If an incorrect system name is entered in the detail system maintenance screen, BIM-VSHARE displays this screen with a list of CICS partitions defined starting with the characters of the name entered. The list can be scrolled to find the appropriate CICS name or the system search name can be changed in order to re-position the list. By placing an S next to a system name, the detail system screen is displayed with the chosen definition. The system maintenance screen can also be used to define what CICS partitions are defined to a particular BIM-VSHARE control file. By choosing the Change function and placing a letter in the CICS APPLID field, a list of all the systems defined will be displayed on this summary screen.

APPLID: The initial name entered is displayed. The name can be changed to re-position the list.

CMD: By placing a S next to a CICS name, the detail definition for the system is displayed.

Attention Keys

ENTER: Process the newly entered 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

Utilization of BIM-VSHARE focuses importance on VSAM I/O performance because more programs can access the same file simultaneously. This section addresses VSAM performance from both a application and system point of view so that an installation can tune their VSAM databases as efficiently as possible. This section initially discusses basic I/O and VSAM concepts for those unfamiliar with them. These basic concepts allow the programmer to fully understand the impact of the tuning recommendations. Following the concepts, a list of tuning recommendations are discussed.

Note that the majority of the tuning options are focused on KSDS files since it is the most commonly utilized VSAM file structure.

Concepts

In most cases the number of I/O operations is the single most important item in program performance. The major percentage of program elapsed time is due to I/O wait time. The elimination or reduction of this time through I/O tuning or the removal of 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 mid-range 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 for maximizing the CPU resource.

Before performance options are discussed, some basic I/O concepts are covered so that those unfamiliar with these 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.

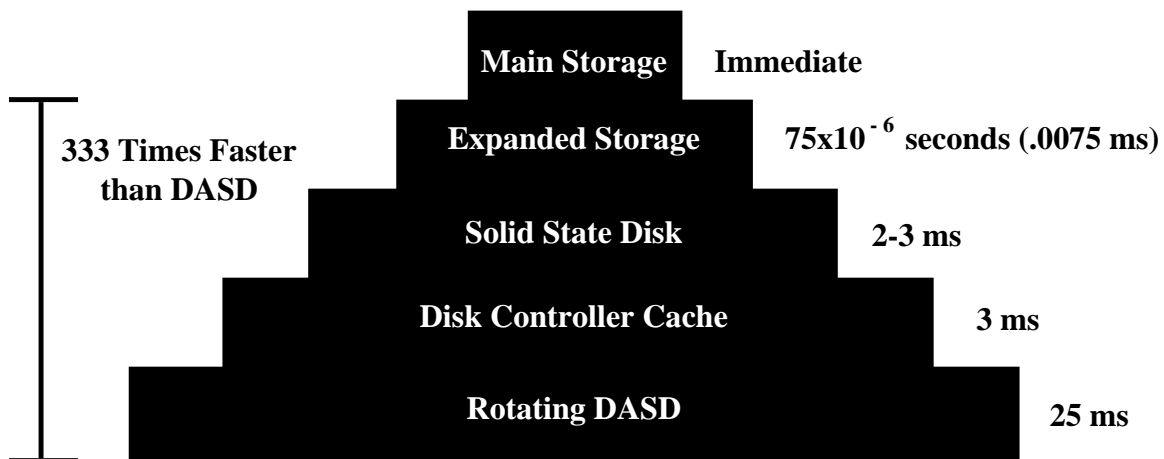
- 1) Storage Mediums
- 2) Concept of Asynchronous Versus Synchronous I/O
- 3) The Mechanics of a Physical I/O
- 4) The Mechanics of a VSAM KSDS I/O Request
- 5) Non Shared Resources Versus Local Shared Resources

Storage Mediums

Until recently data could be located either in main storage or on an external disk. Data can 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. Because of this, the SSD delivers a constant low response time for the reading and writing of 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:** The reading and writing of data from rotating disks is the slowest operation since it relies on mechanical movement.

The data retrieval times are dramatically different depending on the storage medium utilized.



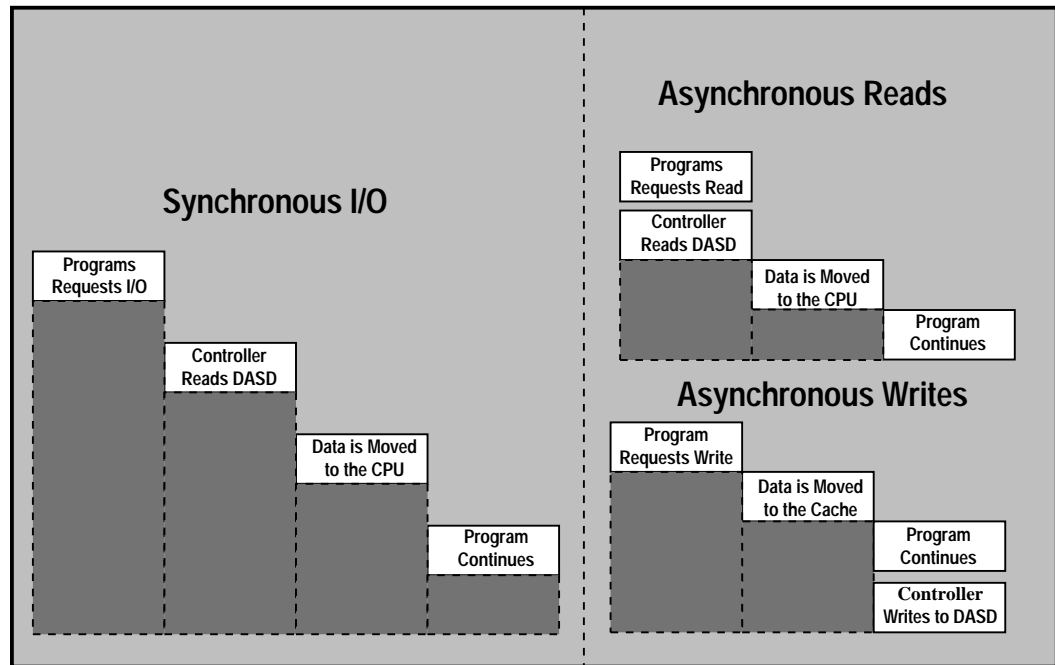
Synchronous Versus Asynchronous Access

Retrieval of data from the storage mediums can be performed synchronously or asynchronously. Following is a description of 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 main storage buffer to the user program resulting in considerable time savings.

Writing of data to disk has always been a synchronous process because of the importance of data integrity. This changed with the advent of the IBM 3990 disk controller. In the past, disk controllers did not have battery backup; therefore, the controller had to wait for the completion of the physical I/O before informing the CPU of the I/O completion. This was important to insure data integrity. The new 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 disk I/O. As the program continues, the controller completes the physical I/O.



Basic I/O Concepts

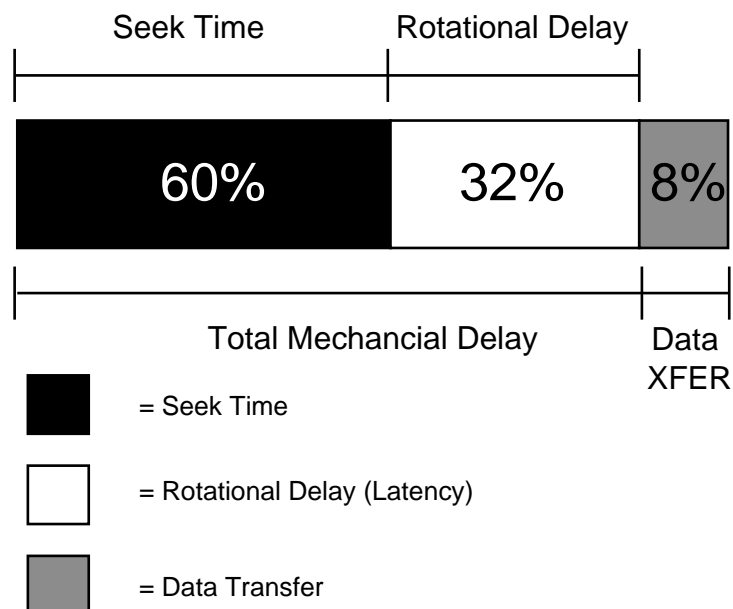
A normal disk I/O can be divided into 3 operations:

1. The first operation the disk performs is the movement of the disk head over the surface of the disk in order to place the head over the appropriate cylinder where the data resides. The average time to complete this seek operation is approximately 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 for any reason 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 for an additional 16 ms rotation of the disk 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 data transfer rate (currently 3 to 4.5 MB per second).

Data transfer represents 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.



The Mechanics of a VSAM KSDS I/O

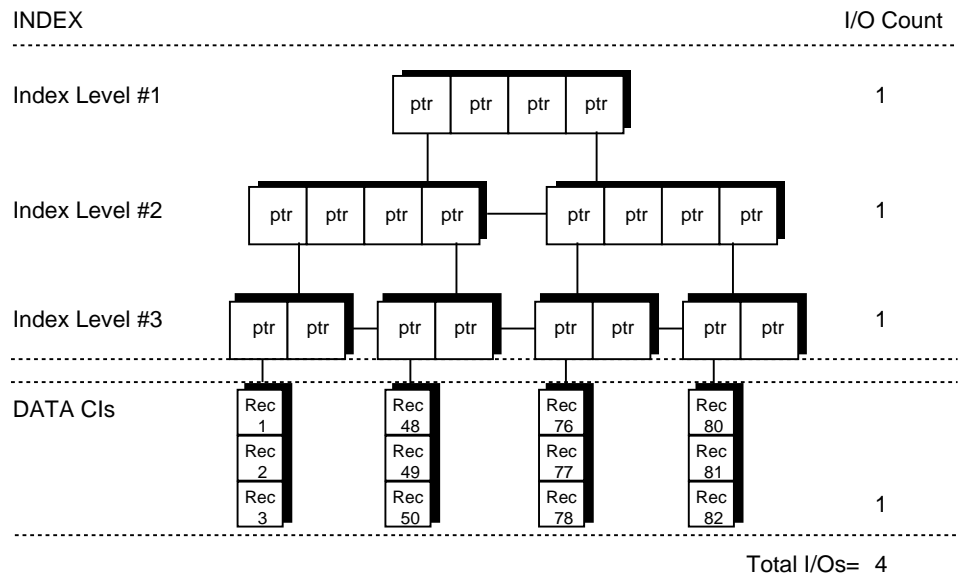
It is important to understand the steps taken by VSAM when a program requests an I/O. The first point to understand is the difference between a logical I/O and a physical I/O. When a program issues a read, this is referred to as a logical I/O. A physical I/O is a request given by the operating system to a controller to retrieve data from DASD. Since VSAM accesses keyed data via an index, there is a need to perform several physical I/Os in order to obtain the appropriate record. The figure below shows an example of a VSAM KSDS file with 3 index levels. To satisfy one logical I/O, VSAM may have to perform 4 physical I/Os to retrieve the requested record. Therefore, the program 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.

In most cases a VSAM index consists of 2 to 3 levels. Assuming 3 levels, 75% of the physical I/O related to a logical I/O is for the index and only 25% for the data. This assumes that no levels of the index are buffered in storage. The key to excellent I/O performance is the reduction of the physical I/Os through efficient buffering.

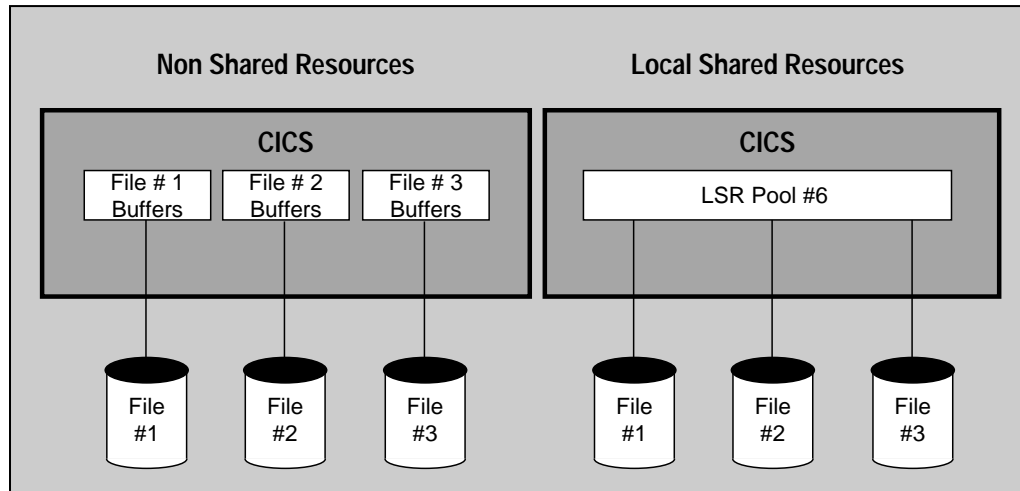
Also note that during sequential or skip sequential processing, the index sequence set is used to locate the next logical control interval. Therefore, browsing requires the reading of the index as well.

Here is an example of a VSAM KSDS file with an index level of 3. To perform 1 logical I/O, VSAM may have to perform 4 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 NSR or Non Shared Resources. 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 the buffers. The second method is called LSR or Local Shared Resources. This method creates a series of buffer pools which can be shared by multiple VSAM files. The figure below shows the two methods as they relate to CICS. More details about each technique are included later in this section.



How to Improve VSAM Performance

With the basic concepts understood, the remainder of this section explains guidelines often ignored but which, if applied correctly, can improve VSAM dataset performance.

- 1) Avoid Control Interval and Control Area Splits
- 2) Reorganize Datasets that are in Multiple Extents
- 3) Spread Heavily Accessed Datasets Over Several Devices
- 4) Utilize Index and Data Buffering
- 5) Index and Data Control Interval Sizes
- 6) The One Cylinder VSAM Dataset
- 7) Analyze CICS End of Day Statistics
- 8) Use VSAM Local Shared Resources
- 9) Use CICS VSAM Subtasking
- 10) Beware of Datasets Using Share Options 3 or 4
- 11) Reduce Program I/O
- 12) Place Critical Files on Faster Storage Mediums

Avoid Control Interval and Control Area Splits

Control area and interval splits are easily detectable and can result in poor performance. The number of CI and CA splits are located under the title SPLITS-CI and SPLITS-CA of an IDCAMS LISTCAT. 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. Note that splits cause large amounts of data to be moved around on DASD and can cause considerable delays.

Possible remedies for high split activity are allocation of sufficient freespace and regular reorganization of the dataset.

```

CLUSTER-INTERV. CTL. CLUSTER
ATTRIBUTES
  KEYLEN-----24          AVGLRECL-----80          BUFSPACE-----0
  RKP-----0            MAXLRECL-----80          EXCPEXIT-----0
  SHROPTNS(2,3)  SPEED    UNIQUE           NOERASE          INDEXED          UNORDERED
  NOREUSE        NONSPANNED
STATISTICS
  REC-TOTAL-----934523    SPLITS-CI-----1365    EXCPS-----204
  REC-DELETED-----0      SPLITS-CA-----457    EXTENTS-----12
  REC-INSERTED-----69350  FREESPACE-%CI-----2    SYSTEM-TIMES-----0
  REC-UPDATED-----398    FREESPACE-%CA-----10    X'A1450
  REC-RETRUEVED----2946099  FREESPC-BYTES-----569344  ALLOCATION-----0

```

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 limited by keeping datasets in as few extents as possible.

When multiple extents are identified by analyzing the IDCAMS LISTCAT report, 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.

```

-----0          EXCPS-----204          INDEX:
-----0          EXTENTS-----12         LEVELS-----3
-----0          SYSTEM-TIMESTAMP:        ENTRIES/SECT-----11
-----0          X'A1450108B5EDCEC0'     SEQ-SET-RBA-----2048
--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 utilized 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. Seldom are the heads 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 time in excess of this should be investigated. These times are obtained from RMF or similar products. Consistent extended seek times produce a critical performance bottleneck. When extended seek times occurs 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 utilizing the NSR buffering technique. BUFNI should be at least equal to the number of levels in the index of the dataset + 1. Never enter a number less, because performance suffers.

The use of buffering should be moderated because over-buffering causes paging to increase thus diminishing the effect of the buffering. 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 partition in a wait, affecting all programs, while a dataset I/O affects only the program requesting the I/O.

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 utilized 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 allow VSAM to choose the index CISIZE. VSAM generally determines the best size based on the data CISIZE, key length, and record length.

The One Cylinder VSAM Dataset

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. Note that 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 value 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% more I/O than necessary is being performed. To determine the number of levels contained in a VSAM dataset, refer to the IDCAMS LISTCAT.

```

-----0      EXCPS-----204      INDEX:
-----0      EXTENTS-----12      LEVELS-----3
-----0      SYSTEM-TIMESTAMP:      ENTRIES/SECT-----11
-----0      X'A1450108B5EDCEC0'    SEQ-SET-RBA-----2048
--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 provide a good source for detection of 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.

***** 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
BATCNIL	852	2	0	0	2	0	854
ITIVSCIL	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 if the control interval is already in storage. If it is available CICS does not issue the read. This buffer lookaside capability reduces I/O 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 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 elapsed processing time of a program is in the reduction of I/O. VSAM LSR can help achieve this.

The use of LSR must be analyzed because all files are not good candidates for the utilization of LSR. 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 utilizing NSR. This is because the browsing causes other dataset control intervals to be flushed from the buffers and upsets the LSR lookaside capability.

The following is a suggested technique for organizing and tuning the LSR pool. Assume that for every VSAM request there are typically 3 physical I/O's 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% of the I/O).

By ensuring that there is sufficient buffer space to hold the index in storage and ensuring 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. However, this is not the case and another solution must be examined.

First, create several LSR pools. Each pool is used by datasets with a particular index CISIZE. Second, define the datasets with different index and data CISIZES. Third, 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.

When specifying the number of 512, 1024, 2048 and 4096 buffers make sure they are a multiple of the VSE page size. If this is not specified, the remaining space on the page is wasted.

To monitor the results of the LSR tuning refer to the CICS shutdown statistics. It supplies the buffer lookaside hit rates achieved. A rate of 90% plus for index and 80% plus for data should be the target. Note that this is only a general rule and may vary greatly depending on data access patterns and frequency.

```

LSR POOL CREATED: 20.16.09

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 CONF. STRINGS  2

```

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

Beware of Datasets Using Share Option 3 or 4

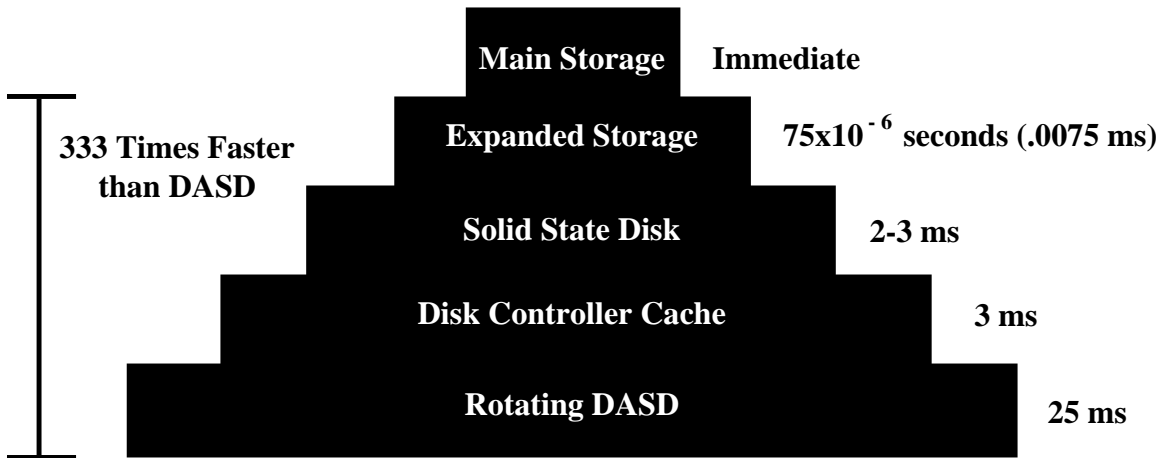
VSAM datasets defined using Share Option 3 or 4 should be avoided. Apart from the integrity exposures that Share Option 3 imposes, it also results in a degradation of performance within CICS. Also, the use of Share Option 4 forces a buffer refresh for every read request. This increases the physical I/O rate and negates the benefit of buffer lookaside. In general the utilization of these Share Options causes performance degradation in all circumstances.

Reduce Program I/O

Up to this point all discussion has centered around performance improvement through the tuning of the environment by buffer utilization, dataset placement, etc. When all of these options are exhausted, the only remaining way to reduce I/O is within the application. Analysis of the application to remove unnecessary I/O is required. By reducing application I/O requests there are 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. The effort needed to analyze the application and remove the I/O is time consuming and expensive. It should be performed only after the environment tuning is exhausted and further savings are desired.

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 traditional rotating DASD. By utilizing these devices, performance can be greatly improved.



INSTALLATION

Overview for VSE Installation

This section describes the installation process for BIM-VSHARE/VSE. The process can be broken down into a few basic steps:

- Insure that the VTAM and CICS levels are appropriate.
- Review module types and load the contents to DASD.
- Create the BIM-VSHARE control dataset.
- If not already implemented, define the CICS partition to utilize ISC.
- Add a LIBDEF and DLBL to CICS JCL to point to BIM-VSHARE library.
- Create the VTAM and CICS definitions for BIM-VSHARE.
- Define eligible CICS partitions and batch jobs to BIM-VSHARE.
- Review the BIM-VSHARE Activation and Deactivation section.
- Review security needs, and modify the BIMVSBSC module.

CICS and VTAM Levels

BIM-VSHARE executes under CICS/VSE 2.2 or greater. Earlier releases may work, but testing has not been performed on them.

BIM-VSHARE can run on VTAM level 3.2 or greater.

Installation for VSE

BIM-VSHARE/VSE is shipped on an unlabeled tape or cartridge containing the following modules.

Router Modules

These modules are prefixed BIMVSR. They are used by the Router facility to intercept VSAM open and close functions. The Router also gets control at end of job.

Batch Modules

These modules are prefixed BIMVSB, BIMVSJ and BIMXMC. They are needed by the batch application enabling BIM-VSHARE to intercept the I/O requests.

CICS Modules

These modules are prefixed BIMVSC, BIMVSM and BIMXMC. They are needed by CICS to process the I/O requests from the batch application, and also to perform the on-line processing of control information for BIM-VSHARE.

Sample JCL and Source Members

These modules are prefixed BIMVSB and have member types 'A' and 'C'. These members are sample JCL needed to install and start BIM-VSHARE. They also include examples of definitions required for the installation, and examples of source code for several BIM-VSHARE application interfaces.

Installing BIM-VSHARE Tape

BIM-VSHARE/VSE is distributed on a LIBR BACKUP tape or cartridge. It consists on one sublibrary, named BIMLIB.BIMVSHAR. Use a job similar to the following one to restore the sublibrary:

```
* $$ JOB JNM=INSTALL
// JOB      INSTALL
// ASSGN   SYS006, CUU                INSTALL TAPE
// DLBL    USRLIB1  etc.
// EXTENT   etc.
// EXEC    PGM=LIBR, SIZE=256K
//         RESTORE SUBLIB=BIMLIB.BIMVSHAR:USRLIB1.BIMVSHAR TAPE=SYS006
/*****/
/* Optional step to move members to a permanent */
/* sublibrary and delete sublibrary created above. */
/*****/
//         CONNECT SUBLIB=USRLIB1.BIMVSHAR:USRLIB.PROD
//         MOVE *.* LIST=YES REPLACE=YES
//         DELETE SUBLIB=USRLIB1.BIMVSHAR
/*****/
/*
/&
* $$ EOJ
```

Define BIM-VSHARE Control Dataset

The control dataset contains all the system, job and history information. The number of these control datasets depends on the user. There can be one per CICS partition, or one for all CICS partitions. If one dataset is used for many partitions, all the CICS partitions utilizing the control dataset must have access to the dataset through MRO or ISC. The installation library contains a member called BIMVSB1.A that defines this VSAM KSDS dataset. Change the dataset names to conform to installation standards.

Definition of Control Dataset

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

Update the BIMVSBDS Parameter Member

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

The module BIMVSBDS is loaded by BIM-VSHARE at initialization time to determine which control dataset to open. If multiple control files are used, a BIMVSBDS phase needs to be assembled, and link edited into different libraries. You will also need a separate activation job for each control dataset.

If changes are made to a BIMVSBDS module, the job used to activate BIM-VSHARE will need to be run again with the PARM=STRT changed to PARM=REFR. This will refresh the table for that control dataset.

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

```
BIMVSBDS  CSECT
          DC    CL44 'BIMVSHAR.TCCVCTL'
          END
```

ISC Installation

BIM-VSHARE utilizes LU 6.2 to communicate with the CICS partition. The CICS partition must be defined to utilize 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.

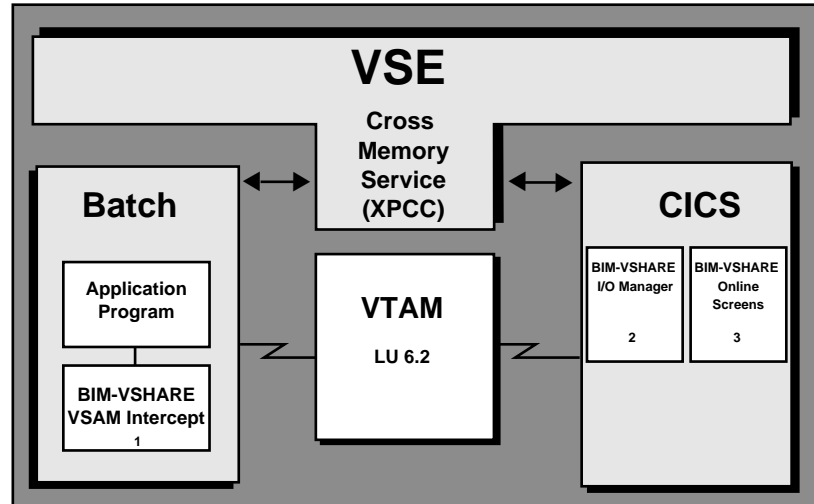
Library Containing CICS Phases

The BIM-VSHARE sub library must be accessible to any CICS partition using BIM-VSHARE. This means a DLBL in the CICS JCL or in Standard labels, and a LIBDEF (PHASE, SEARCH=) in the CICS JCL or permanently set for the partition.

Environment Definitions

The figure below shows the general design of BIM-VSHARE. BIM-VSHARE is located in both the CICS partition owning the VSAM files and the batch partition. The CICS and batch programs can communicate the VSAM request through LU 6.2 or Cross Memory Services (XPCC). LU 6.2 allows the batch partition and the CICS partition to be running on different CPUs, while Cross Memory Services provides for faster data transfer when the batch partition and the CICS partition reside on the same CPU. BIM-VSHARE always uses LU 6.2 to start its conversation between the batch partition and CICS. After the communications link has been established between the batch partition and CICS, the Cross Memory Services interface may be utilized, if it has been requested, and both parties are on the same CPU. With this in mind, the following definitions are needed:

- 1) VTAM:
Define an APPLID for the batch programs. This APPLID is opened by BIM-VSHARE batch programs in order to establish a LU 6.2 session with the CICS partition. The number of APPLIDs determine the number of concurrent jobs that can execute.
- 2) CICS:
Define CICS entries for BIM-VSHARE's transactions, programs, and the LU 6.2 sessions.
- 3) BIM-VSHARE on-line updates:
Define the CICS partition and default processing options to BIM-VSHARE through the BIM-VSHARE on-line panels. This will be discussed in greater detail later in this chapter as well as in the On-Line Control chapter.



The definitions needed to allow BIM-VSHARE to communicate between the batch partition and CICS are interrelated. If certain parameters are not properly linked together, the connection between the two partitions will be rejected by CICS or VTAM. Before describing each definition in detail, the following figures show the relations 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 Sessions and Connections are Used in CICS

BIM-VSHARE System Maintenance Screen:

```

=====
| APPLID: UAPPLID  BIM-VSHARE - THE CICS CONNECTION  DATE: 04/21/1998 |
| PROGID: BIMVSC19  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: [PRODCICS] |
=====
|          SYSTEM INFORMATION          |          HISTORY INFORMATION          |
| TRANID: TCCB   ACCESS TYPE:  CMS    | RECORD HISTORY:  Y                   |
| MODE NAME: LU62MODE  SYNC LEVEL:  005 | GENERATIONS:    003                  |
| ROLLBACK: N      REQ/MIN: 00000     | JOURNAL INFORMATION                  |
| APPLID SEL: N     SELECT MSK: ***** | JOURNAL UPDATES:  N                   |
=====
|          BATCH APPLIDS:  [BATCH01] |
=====
| ENTER=EDIT  CLEAR=EXIT  PF3=PROCESS  PF7=BACK  PF8=FORWARD  PF9=HOME |
=====
    
```

CICS APPLID Definition:

```

[PRODCICS]  APPL AUTH=(ACQ,PASS,VPACE),
            VPACING=20,EAS=20,
            ACBNAME=PRODCICS,
            PARSESS=YES,
            SONSCIP=YES
    
```

Batch APPLID Definition:

```

[BATCH01]  APPL AUTH=(ACQ,VPACE),
            VPACING=20,
            EAS=1,PARSESS=NO,
            SONSCIP=YES
    
```

Session Definition:

```

OBJECT CHARACTERISTICS
CEDA View
Sessions      : [BATCH01]
Group        : TCC
SESSION IDENTIFIERS
Connection    : [BAT1]
SESSName     :
NETNameq     :
MODEname     : SMSNA100
SESSION PROPERTIES
Protocol     : Appc
Maximum      : 00001,00001
RECEIVEPfx  :
RECEIVECount : No
SENDPfx     :
SENDCount   : No
SENDSize    : 30720
RECEIVESize : 30720
OPERATOR DEFAULTS
OPERId      :
OPERPriority : 000
OPERSrl    : 0
OPERSecurity : 1
USERId     :
SESSION USAGES
Transaction :
SESSPriority : 000
OPERATIONAL PROPERTIES
Autoconnect : No
INservice   :
Buildchain  : Yes
USERarealen : 000
IOarealen  : 00000 , 00000
RELreq     : No
Discreq    : No
NEPclass   : 000
RECOVERY
RECOVOption : Sysdefault
RECOVNotify : None
    
```

Connection Definition:

```

OBJECT CHARACTERISTICS
CEDA View
Connection   : [BAT1]
Group       : TCC
CONNECTION IDENTIFIERS
Netname     : [BATCH01]
INDsys     :
CONNECTION PROPERTIES
Accessmethod : Vtam
Protocol     : Appc
Singlesess  : Yes
Datastream  : User
Recordformat : U
OPERATIONAL PROPERTIES
Autoconnect : No
INService   : Yes
SECURITY
Securityname :
Attachsec   : Local
Bindpassword :
    
```

Definitions Required When Terminal Definitions are Used in CICS

BIM-VSHARE System Maintenance Screen:

```

=====
| APPLID: UAPPLID   BIM-VSHARE - THE CICS CONNECTION   DATE: 04/21/1998 |
| PROGID: BIMVSC19 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: PRODCICS |
|-----|
| SYSTEM INFORMATION | HISTORY INFORMATION |
| TRANDID: TCCB   ACCESS TYPE: CMS | RECORD HISTORY: Y |
| MODE NAME: LU62MODE   SYNC LEVEL: 005 | GENERATIONS: 003 |
| ROLLBACK: N   REQ/MIN: 00000 | JOURNAL INFORMATION |
| APPLID SEL: N   SELECT MSK: ***** | JOURNAL UPDATES: N |
|-----|
| BATCH APPLIDS: BATCH01 |
|-----|
| ENTER=EDIT   CLEAR=EXIT   PF3=PROCESS   PF7=BACK   PF8=FORWARD   PF9=HOME |
=====
    
```

CICS APPLID Definition:

```

PRODCICS  APPL AUTH=(ACQ,VPASS,VPACE),
          VPACING=20,EAS=20,
          ACBNAME=PRODCICS,
          PARSESS=YES,
          SONSCIP=YES,
          MODETAB=MODETAB1
    
```

Batch APPLID Definition:

```

BATCH01  APPL AUTH=(ACQ,VPASS,VPACE),
          VPACING=20,
          EAS=1,PARSESS=NO,
          SONSCIP=YES,
          MODETAB=MODETAB1,
          DLOGMODE=LU62MODE
    
```

Mode Entry Definition When SMSNA100 is Not Used:

```

MODETAB1  MODETAB
          .
          .
          LU62MODE LOGMODE=LU62MODE
          MODEEND
    
```

CICS Typeterm Definition:

```

OBJECT CHARACTERISTICS
CEDA View
TypeTerm : DFHLU62T
Group : BIM
RESOURCE TYPE
Device : APPC
TERmmodel :
SESSiontype :
LDclst :
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
APLKybd : No No|Yes
.
Backtrans : No No|Yes
CGcsqid : 00000,00000 0-65535
SESSION PROPERTIES
AScii : No No|7|8
SENdsiz : 30720 0-30720
RECEivesiz : 30720 0-30720
BRacket : Yes Yes|No
LOGMode :
DIAGNOSTIC DISPLAY
ERRLastline : No No|Yes
ERRIntensify : No No|Yes
ERRColor : NO Blue|Red|Pink|Green|
Turquoise|Yellow|NEutral
ERRHilght : No No|Blink|Reverse|Underline
OPERATIONAL PROPERTIES
AUTOConnect : No No|Yes|All
ATI : Yes No|Yes
TTI : Yes No|Yes
CReatesess : No No|Yes
RELreq : No No|Yes
DIScreq : Yes No|Yes
Nepclass : 000 0-255
Signoff : Yes Yes|No|Logoff
MESSAGE RECEIVING PROPERTIES
Routedmsg : None All|None|Specific
LOGOnmsg : No No|Yes
APPLICATION FEATURES
BUildchain : Yes No|Yes
USerarealen : 000 0-255
LOsarealen : 30720,30720 0-32767
UCxran : No No|Yes
RECOVERY
RECOVOption : Sysdefault Sysdefault|Clearconv|
Releaseses|Uncondrel|None
RECOVNotify : None None|Message|Transaction
    
```

CICS Terminal Definition:

```

OBJECT CHARACTERISTICS
CEDA View
TERminal : LU6T
Group : BIM
AUTINSTModel : No No|Yes|Only
AUTINSTName :
TERMINAL IDENTIFIERS
TypeTerm : DFHLU62T
Netname : BATCH01
Console : No No|0-99
REMOTESystem :
REMOTENAME :
Modename : LU62MODE
ASSOCIATED PRINTERS
PRINTER :
PRINTERCopy : No No|Yes
ALTPRINTER :
ALTPRINTCopy : No No|Yes
PIPELINE PROPERTIES
Pool :
Tasklimit : No No|1-32767
OPERATOR DEFAULTS
OPERID :
OPERPriority : 000 0-255
OPERRsl : 0 0-24..
OPERSecurity : 1 1-64..
Userid :
TERMINAL USAGES
TRANSACTION :
TERmpriority : 000 0-255
Inservice : Yes Yes|No
PRINTER DATA
SESSION SECURITY
Securityname :
ATTachsec : Local Local|Identify|Verify
Bindpassword : PASSWORD NOT SPECIFIED
TERMINAL USAGES
TRANSACTION :
TERmpriority : 000 0-255
Inservice : Yes Yes|No
PRINTER DATA
SESSION SECURITY
Securityname :
ATTachsec : Local Local|Identify|Verify
Bindpassword : PASSWORD NOT SPECIFIED
    
```

VTAM Definitions

There are two definitions required for BIM-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 BIM-VSHARE is as follows:

```
name          APPL          ACBNAME=applid1,
                                AUTH=(ACQ,VPACE),
                                EAS=1,
                                MODETAB=tablename,
                                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 partition. You can also share the same batch APPLIDs between CICS partitions, 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.

CICS Definitions

TCT Definitions

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

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

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

Remember that 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 DEFINE      SESSIONS(session name)
                  GROUP(BIMVSHAR)
                  CONNECTION(connection name)
                  MODENAME(modeent)
                  PROTOCOL(APPC)
                  MAXIMUM(1,1)
                  RECEIVECOUNT(NO)
                  SENDCOUNT(NO)
                  SENDSIZE(30720)
                  RECEIVESIZE(30720)
```

Remember that there is one session definition per batch APPLID.

PCT Definitions

The definition for all BIM-VSHARE transactions are listed below.

CEDA DEFINE	TRANSACTION(TCC) GROUP(BIMVSHAR) PROGRAM(BIMVSC10)
CEDA	DEFINE TRANSACTION(TCCB) GROUP(BIMVSHAR) PROGRAM(BIMVSC00)
CEDA	DEFINE TRANSACTION(TCCR) GROUP(BIMVSHAR) PROGRAM(BIMVSC06)

The first PCT definition, TCC, is for the BIM-VSHARE on-line system.

The second definition, TCCB, is for the LU 6.2 program that communicates to the batch partition. This transaction may be given any name desired. Any number of these batch tranids may be defined. By having each application within CICS partition 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. BIM-VSHARE provides dataset security by providing the user the security exit module, BIMVSBSC. See discussion on the security exit for more details.

By defining the batch tranid TCCB with a priority level of 0, CICS services the on-line programs before the batch programs, and the impact to the on-line 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 partitions owning remote files. For example, a batch job is connected to CICSPRD1 and it is accessing a VSAM file owned by a CICS partition called CICSPRD2. The transaction TCCR must be defined to CICSPRD2 so that BIM-VSHARE can obtain information about the file. The remote file can be defined to utilize either ISC or MRO.

PPT Definitions

BIM-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 partition that is running BIM-VSHARE, the program BIMVSC06 needs to be defined in all partitions owning remotely defined VSAM files that will be accessed using BIM-VSHARE.

CEDA	DEFINE	PROGRAM(name1)	CEDA	DEFINE	PROGRAM(name2)
		GROUP (BIMVSHAR)			GROUP (BIMVSHAR)
		LANGUAGE (ASSEMBLER)			LANGUAGE (COBOL)
		RELOAD (NO)			RELOAD (NO)
		RESIDENT (NO)			RESIDENT (NO)
		STATUS (ENABLE)			STATUS (ENABLE)
		Assembler			BMS Maps
		BIMVSC00		COBOL	BIMVSM1
		BIMVSC01		BIMVSC10	BIMVSM2
		BIMVSC02		BIMVSC11	BIMVSM3
		BIMVSC03		BIMVSC12	BIMVSM4
		BIMVSC04		BIMVSC13	BIMVSM5
		BIMVSC05		BIMVSC14	BIMVSM6
		BIMVSC06		BIMVSC15	BIMVSM7
		BIMVSCDM		BIMVSC16	BIMVSM8
		BIMVSCM0 (letter 'O')		BIMVSC17	BIMVSM9
				BIMVSC18	
				BIMVSC19	

FCT Definition

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

BIMVSFL	DFHFCT	TYPE=DATASET,	*
		DATASET=BIMVSFL,	*
		ACCMETH=VSAM,	*
SERVREQ=	(READ,ADD,BROWSE,UPDATE,DELETE),	*
		LSRPOOL=1,	*
		RECFORM=(VARIABLE,BLOCKED),	*
		STRNO=3,	*
		FILSTAT=(ENABLE,CLOSED)	

Define the Eligible CICS Partitions and Jobs

The next step is to define all the CICS partitions and job names eligible to execute BIM-VSHARE. This is accomplished through the BIM-VSHARE on-line system. Refer to the chapter “ON-LINE 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 partition. Remember to define the CICS APPLID, not the CICS job name.

```

+-----+
| APPLID: BIMCICST  BIM-VSHARE - THE CICS CONNECTION  DATE: 11/24/1998 |
| PROGID: BIMVSC19  S Y S T E M   M A I N T E N A N C E  TIME: 13:12:12 |
+-----+
| FUNCTION:      C ( A = ADD, C = CHANGE, D = DELETE ) |
| CICS APPLID:  PRODCICS |
+-----+
|          SYSTEM INFORMATION          |          HISTORY INFORMATION          |
| TRANID: TCCB  ACCESS TYPE:          | RECORD HISTORY:          |
| MODE NAME: SMSNA100 SYNC LEVEL: 000 | GENERATIONS: 000 |
| ROLLBACK: N    REQ/MIN: 00000       | JOURNAL INFORMATION     |
| APPLID SEL: N  SELECT MSK: ***** | JOURNAL UPDATES: N     |
+-----+
|          BATCH APPLIDS:  BATCH01          |
+-----+
| ENTER=EDIT  CLEAR=EXIT  PF3=PROCESS  PF7=BACK  PF8=FORWARD  PF9=HOME |
+-----+

```

The Job Maintenance screen is used to define the eligible batch job names. (VSE job name, not the POWER job name.) Note that not all of the individual job names need defining. For example, if all job names starting with BIM are eligible to use BIM-VSHARE, only define a job name in the Job Maintenance screen with the name BIM. BIM-VSHARE then intercepts any job name starting with BIM.

```

+-----+
| APPLID: BIMCICST  BIM-VSHARE - THE CICS CONNECTION  DATE: 11/24/1998 |
| PROGID: BIMVSC15  J O B   M A I N T E N A N C E  TIME: 14:23:53 |
+-----+
| FUNCTION: C ( A = ADD, C = CHANGE, D = DELETE ) |
+-----+
| JOB NAME: BIMJOBH |
+-----+
|          JOB INFORMATION          |          HISTORY INFORMATION          |          PROCESSING INFORMATION          |
| CICS REGION: BIMCICST | RECORD?: Y | ABEND IF NO CICS?: N |
| TRANID: VSCB          | GENERATIONS: 002 | DO INPUT FILES?: N |
| SYNC LEVEL: 010       |          | ABEND IF DISABLED?: N |
| REQS/MINUTE: 00000   | JOURNAL INFORMATION | TURN TRACE ON?: N |
| ACCESS TYPE: LU62    | JOURNAL UPDATES?: N | FORCE ROLLBACK?: N |
+-----+
|          BIMVSC15-008 PRESS PF3 TO PROCESS THE MODIFICATION.          |
+-----+

```

BIM-VSHARE Activation and Deactivation

In the installation library there is a member named BIMVSBJ3.A. Executing this JCL will cause the phases listed in the JCL to be loaded into the SVA:

```
// JOB BIMVSBJ3    LOAD SVA PHASES FOR BIM-VSHARE
/* THIS JOB MUST EXECUTE IN PARTITION BG
// OPTION LOG
// LIBDEF PHASE,SEARCH=USRLIB1.BIMVSHAR
SET SDL
BIMVSRQP,SVA      (BIM VSAM Router)
BIMVSRCL,SVA      (BIM VSAM Router)
BIMVSRPO,SVA      (BIM VSAM Router)
BIMVSRPC,SVA      (BIM VSAM Router)
BIMVSRTB,SVA      (BIM VSAM Router)
BIMVSRCM,SVA      (BIM VSAM Router)
BIMVSRJE,SVA      (BIM VSAM Router)
BIMVSR0E,SVA      (BIM VSAM Router)      ('0' IS A ZERO)
BIMVSBTB,SVA
BIMVSB01,SVA
BIMVSB02,SVA
BIMVSB0J,SVA
/*
/ &
```

The SVA modules with the prefix BIMVSR, are common to several BIM products, and may already be loaded into your SVA. They belong to the BIM VSAM Router. Care should be taken not to reload these SVA phases while the BIM VSAM Router is active for another product.

These phases must reside in the SVA. It is recommended after testing has been completed and BIM-VSHARE is moved into production, that this member be incorporated into the ASI PROC.

In addition to the SVA modules, the following modules must be in a VSE library that is accessible to all partitions via a permanent LIBDEF:

```
BIMVSRSC.PHASE
BIMVSRSO.PHASE
BIMXMC00.PHASE
```

This can be accomplished by copying these modules to a VSE library already in a permanent LIBDEF, or by adding the BIM-VSHARE sublibrary to your permanent LIBDEF.

In the installation library there are also members BIMVSBJ4.A, BIMVSBJ5.A, and BIMVSBJ6.A.

BIMVSBJ4.A

This job will activate BIM-VSHARE. You will need have a separate step for each Control Dataset created, with the DLBL BIMVSFL pointing to the Control Dataset:

```
// JOB BIMVSBJ4   START BIM-VSHARE
// DLBL BIMVSFL, 'BIMVSHAR.TCCVCTL', , VSAM
// LIBDEF PHASE,SEARCH=USRLIB1.BIMVSHAR
// EXEC BIMVSBCM,PARM='STRT'
/*
/ &
```

If you make changes to a Control Dataset after BIM-VSHARE has been started, you will need to run this job again with PARM='REFR' to update the BIM-VSHARE incore table.

BIMVSBJ5.A

This job will deactivate BIM-VSHARE:

```
// JOB BIMVSBJ5   STOP BIM-VSHARE
// DLBL BIMVSFL, 'BIMVSHAR.TCCVCTL', , VSAM
// LIBDEF PHASE,SEARCH=USRLIB1.BIMVSHAR
// EXEC BIMVSBCM,PARM='STOP'
/*
/ &
```

BIMVSBJ6.A

This job will reset all SDL pointers that were modified to bring the system back to its original state:

```
// JOB BIMVSBJ6   RESET SDL/SVA
// LIBDEF PHASE,SEARCH=USRLIB1.BIMVSHAR
// EXEC BIMVSRM,PARM='SHUT11'
/*
/ &
```

Job Activation and Deactivation

To activate BIM-VSHARE for a particular job, a LIBDEF must be included in the JCL, pointing to the library containing the BIM-VSHARE phases.

Deactivation may be accomplished in two ways:

- 1) The first way is the removal of the LIBDEF from the JCL.
- 2) The second way is by running the job BIMVSBJ5.A, which will simply stop BIM-VSHARE from being invoked. It is not recommended that the BIMVSBJ6.A job be run. This resets SDL pointers and may stop other vendor products from running if they were started after BIM-VSHARE.

File Deactivation

BIM-VSHARE processing may also be excluded for specific files within a job by coding an expiration date of 93/333 on their VSE DLBL statements. If it is necessary to reactivate BIM-VSHARE processing for such files in subsequent steps of that job, then a new DLBL statement will need to be included for that step (without the 93/333 date). Per VSE JCL rules, that will mean that other DLBL or TLBL statements in the prior step's JCL that are also needed in current or subsequent steps will have to be included again also.

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, BIM-VSHARE will call module BIMVSBSC. This module will be passed the dataset name, the type of access, and the volume the dataset resides on. If BIMVSBSC sets a return code greater than 4, a message will be displayed, and the batch job will be canceled.

There is an example of this source (BIMVSBSC.A) in the install library that may be modified to adhere to your security needs. When modified, the link edited phase must be placed in the BIM-VSHARE execution library.

MESSAGES AND CODES

BIMVSBAl Messages

BIMVSBAl01 **BIM-VSHARE NOT ACTIVE, DIRECT CALLS WILL BE IGNORED.**
Explanation: During the batch run, an explicit syncpoint or rollback was requested but BIM-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.

On-Line Maintenance Messages

BIMVSC1n-001 **COMMAREA LENGTH IS INVALID.**
Explanation: BIM-VSHARE on-line programs received a COMMAREA length error.
System Action: Processing stops.
User Response: This is an internal BIM-VSHARE message. Contact BIM support.

BIMVSC1n-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 "ON-LINE CONTROL" on page 41 for details.

BIMVSC1n-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 "ON-LINE CONTROL" on page 41 for details on the field.

BIMVSC1n-004 **INVALID PF KEY. MAKE SELECTION AND PRESS ENTER.**
Explanation: A PF key was entered that is not utilized by the on-line program.
System Action: Re-display the screen with the message.
User Response: Enter proper PF key. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on proper attention keys.

BIMVSC1n-005 **PROGRAM <progid> WAS NOT FOUND. CONTACT SYSTEMS PROGRAMMER.**
Explanation: BIM-VSHARE program PROGID was not available when another BIM-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 BIM-VSHARE library may not be available to CICS.

BIMVSC1n-006 **MAP FAILURE IN PROGRAM <progid>.**
Explanation: BIM-VSHARE on-line program PROGID received a map failure response from CICS.
System Action: Processing stops.
User Response: Contact CICS systems programmer. The CICS definition for the map may not be completed or BIM-VSHARE library may not be available to CICS.

BIMVSC1n-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 "ON-LINE CONTROL" on page 41 for details.

- BIMVSC1n-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 "ON-LINE CONTROL" on page 41 for details.
- BIMVSC1n-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 "ON-LINE CONTROL" on page 41 for details.
- BIMVSC1n-010** **ENTER REQUIRED FIELD.**
Explanation: A required field was not entered during a system or job definition.
System Action: Informational only. Definition will not be processed until the value for the highlighted field is entered.
User Response: Enter the required field. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on the field.
- BIMVSC1n-011** **FIELD MUST BE NUMERIC.**
Explanation: The field highlighted must contain a numeric value.
System Action: Informational only. Definition will not be processed until the value is numeric.
User Response: Enter a numeric value. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on the field in question.
- BIMVSC1n-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. Definition will not be processed until the value is correct.
User Response: Enter a Y or N. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on the field in question.
- BIMVSC1n-013** **THIS FUNCTION KEY IS NOT ACTIVE.**
Explanation: A PF key was entered that is not utilized by the on-line program.
System Action: Re-display the screen with the message.
User Response: Enter proper PF key. Refer to the chapter "ON-LINE CONTROL" on page 41 for details on proper attention keys.
- BIMVSC1n-014** **THE CONTROL DATASET <fctname> IS NOT DEFINED TO CICS.**
Explanation: A BIM-VSHARE on-line program received a DSIDERR from CICS while accessing BIM-VSHARE control dataset.
System Action: Processing stops.
User Response: Contact CICS systems programmer. The CICS definition for the dataset may not be completed.
- BIMVSC1n-015** **THE CONTROL DATASET <fctname> IS DISABLED.**
Explanation: A BIM-VSHARE on-line program received a DISABLED response from CICS while accessing BIM-VSHARE control dataset.
System Action: Processing stops.
User Response: Determine why the dataset is disabled and enable the BIM-VSHARE control dataset.
- BIMVSC1n-016** **I/O ERROR WAS DETECTED WHILE READING CONTROL FILE.**
Explanation: An I/O error was detected by BIM-VSHARE on-line screens while accessing the control dataset.
System Action: Processing stops.
User Response: Determine the cause of the control dataset I/O error.
- BIMVSC1n-017** **NOT AUTHORIZED TO ACCESS THE CONTROL DATASET.**
Explanation: The user executing BIM-VSHARE on-line 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.

BIMVSC1n-018	<p>THE CONTROL DATASET <fctname> IS NOT OPENED.</p> <p>Explanation: BIM-VSHARE on-line programs received a not open response from CICS when accessing the control dataset.</p> <p>System Action: Processing stops.</p> <p>User Response: Determine the reason why the control dataset is closed and open it.</p>
BIMVSC1n-019	<p>THE <JOB> <SYSTEM> <HISTORY> RECORD FOR THIS <JOB> <SYSTEM> <HISTORY> WAS NOT FOUND.</p> <p>Explanation: If defining a job, a CICS partition name was entered that was not previously defined. If reviewing history, a job name was entered that had no corresponding history.</p> <p>System Action: Screen is re-displayed and the program waits for new data.</p> <p>User Response: When defining a job, make sure the CICS name was entered properly or that the partition was previously defined. If reviewing history, there is nothing required.</p>
BIMVSC1n-020	<p>THIS <JOB> <SYSTEM> IS ALREADY DEFINED IN THE CONTROL FILE.</p> <p>Explanation: An attempt was made to add a job or CICS partition previously defined to BIM-VSHARE.</p> <p>System Action: Re-display the screen with the message.</p> <p>User Response: Make sure the job or CICS partition name is correct or look at the current definition for the job/CICS partition.</p>
BIMVSC1n-021	<p>A VSAM LENGERR HAS BEEN DETECTED.</p> <p>Explanation: A BIM-VSHARE program received a LENGERR from CICS while accessing the control dataset.</p> <p>System Action: Processing stops.</p> <p>User Response: This is an internal error. Contact BIM support.</p>
BIMVSC1n-022	<p>THE CONTROL FILE HAS RUN OUT OF SPACE.</p> <p>Explanation: A BIM-VSHARE program received an out of space condition from CICS when writing to the control file.</p> <p>System Action: Processing stops.</p> <p>User Response: The control file needs to be defined larger.</p>
BIMVSC1n-023	<p>THE <JOB> <CICS PARTITION> WAS SUCCESSFULLY ADDED TO THE CONTROL FILE.</p> <p>Explanation: The job or CICS partition entered was successfully added to BIM-VSHARE.</p> <p>System Action: Informational only.</p> <p>User Response: None</p>
BIMVSC1n-024	<p>THE <JOB> <CICS PARTITION> DATA WAS SUCCESSFULLY MODIFIED.</p> <p>Explanation: The job or CICS partition being modified was successfully processed.</p> <p>System Action: Informational only.</p> <p>User Response: None</p>
BIMVSC1n-025	<p>THE <JOB> <CICS PARTITION> WAS SUCCESSFULLY DELETED.</p> <p>Explanation: While performing system maintenance, the job or partition definition selected was successfully deleted.</p> <p>System Action: Informational only.</p> <p>User Response: None.</p>
BIMVSC1n-026	<p>THE CICS PARTITION NAME ENTERED IS NOT DEFINED.</p> <p>Explanation: The partition name entered while defining a job was not previously defined to BIM-VSHARE.</p> <p>System Action: Informational only. The job will not be added until a valid partition is entered.</p> <p>User Response: Make sure the name is spelled correctly or add the system to BIM-VSHARE through the System Maintenance Screen.</p>
BIMVSC1n-027	<p>THE BLOCK SIZE IS INCORRECT BASED ON THE LRECL.</p> <p>Explanation: The block size chosen for the journal file is not correct.</p> <p>System Action: Informational only. The new definition of the journal file will not be updated without a valid block size.</p> <p>User Response: Enter a value that is a multiple of the record size entered plus 4 bytes.</p>

BIMVSC1n-029	<p>INVALID ACCESS TYPE. ENTER CMS OR LU62 ONLY.</p> <p>Explanation: The access type entered is not valid.</p> <p>System Action: Informational only. A valid type must be entered before the definition is applied.</p> <p>User Response: Enter LU62 or CMS. Refer to the chapter “ON-LINE CONTROL” on page 41 for details on these types.</p>
BIMVSC1n-031	<p>I/O ERROR WAS DETECTED WHILE READING TEMP STORAGE.</p> <p>Explanation: BIM-VSHARE received an I/O error response from CICS after accessing temporary storage.</p> <p>System Action: Processing stops.</p> <p>User Response: Contact system programmers. There is something wrong with the CICS temporary storage.</p>
BIMVSC1n-032	<p>LENGTH ERROR WAS DETECTED WHILE READING TEMP STORAGE.</p> <p>Explanation: BIM-VSHARE received a length error response from CICS after accessing temporary storage.</p> <p>System Action: Processing stops.</p> <p>User Response: This is an internal BIM-VSHARE error. Contact BIM support.</p>
BIMVSC1n-033	<p>QID ERROR WAS DETECTED WHILE READING TEMP STORAGE.</p> <p>Explanation: BIM-VSHARE received a QID error from CICS after accessing temporary storage.</p> <p>System Action: Processing stops.</p> <p>User Response: This is an internal BIM-VSHARE error. Contact BIM support.</p>
BIMVSC1n-034	<p>NOT AUTHORIZED TO ACCESS TEMPORARY STORAGE.</p> <p>Explanation: The person executing BIM-VSHARE on-line programs does not have proper authorization to read temporary storage.</p> <p>System Action: Processing stops.</p> <p>User Response: Contact the CICS systems programmer to find out why this is the case.</p>
BIMVSC1n-035	<p>THE REQUEST TO STOP THIS JOB HAS BEEN ISSUED.</p> <p>Explanation: A request was given in the active job screen to stop a job.</p> <p>System Action: Informational only.</p> <p>User Response: None. Refer to the chapter “ON-LINE CONTROL” on page 41 for details.</p>
BIMVSC1n-036	<p>THE REQUEST TO CANCEL THIS JOB HAS BEEN ISSUED.</p> <p>Explanation: A request was given in the active job screen to cancel a job.</p> <p>System Action: Informational only.</p> <p>User Response: None. Refer to the chapter “ON-LINE CONTROL” on page 41 for details.</p>
BIMVSC1n-037	<p>THE REQUEST TO RESTART THIS JOB HAS BEEN ISSUED.</p> <p>Explanation: A request was given in the active job screen to restart a previously stopped job.</p> <p>System Action: Informational only.</p> <p>User Response: None. Refer to the chapter “ON-LINE CONTROL” on page 41 for details.</p>
BIMVSC1n-038	<p>THIS JOB IS NO LONGER ACTIVE.</p> <p>Explanation: A job was being analyzed in BIM-VSHARE active job process and the job completed.</p> <p>System Action: The information about the job is no longer available. A screen with no information is displayed.</p> <p>User Response: None. Refer to the chapter “ON-LINE CONTROL” on page 41 for details.</p>
BIMVSC1n-039	<p>THE REQUEST RATE HAS BEEN SUCCESSFULLY CHANGED.</p> <p>Explanation: A request to change the request rate for a job was processed successfully.</p> <p>System Action: Information only.</p> <p>User Response: None. Refer to the chapter “ON-LINE CONTROL” on page 41 for details.</p>
BIMVSC1n-040	<p>THE REQUEST RATE MUST BE NUMERIC.</p> <p>Explanation: A non-numeric request rate was entered.</p> <p>System Action: Highlight the field and re-display the screen.</p> <p>User Response: Enter a numeric value.</p>

- BIMVSC1n-042** NOT AUTHORIZED TO INQUIRE ON FILE CONTROL TABLE.
Explanation: While trying to display a dataset statistics screen, BIM-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.
- BIMVSC1n-043** THIS FCT NAME IS NOT IN THE CICS TABLES.
Explanation: While trying to display a dataset statistics screen in history, BIM-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 BIM-VSHARE history record.
- BIMVSC1n-044** AN INVALID REQUEST RETURNED AFTER A TS READ.
Explanation: BIM-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 problem persists, contact BIM support.
- BIMVSC1n-045** BIM-VSHARE HAS COMPLETED.
Explanation: BIM-VSHARE on-line system completed successfully.
System Action: This is an informational message. No error occurred.
User Response: None.

Batch Messages

BIMVSHAR-100	<p>NOT ENOUGH PCBS DEFINED. Explanation: A module requested a PCB and found they were all used. System Action: Program aborts. User Response: Contact BIM support.</p>
BIMVSHAR-101	<p>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.</p>
BIMVSHAR-103	<p>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 error occurs again call BIM support with dumps.</p>
BIMVSHAR-104	<p>ERROR INITIALIZING JOURNALING. Explanation: Journal initialization encountered a problem. System Action: Dependent on problem. User Response: Refer to other related messages.</p>
BIMVSHAR-106	<p>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.</p>
BIMVSHAR-107	<p>GETMAIN ERROR. Explanation: A program issued a storage GETMAIN/GETVIS and VSE returned an error. System Action: Program aborts. User Response: Refer to the IBM message relating to this. If none, contact BIM support.</p>
BIMVSHAR-108	<p>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.</p>
BIMVSHAR-109	<p>FREEMAIN FAILED. Explanation: A program issued a storage FREEMAIN/FREVIS and VSE returned an error. System Action: Dependent. User Response: If program aborts, call BIM support. If program ends normal, ignore.</p>
BIMVSHAR-110	<p>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.</p>
BIMVSHAR-111	<p>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.</p>
BIMVSHAR-115	<p>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 error occurs again call BIM support.</p>

BIMVSHAR-116	<p>INTERNAL ERROR: (DESCRIPTION). Explanation: An internal error occurred. System Action: Program aborts. User Response: Contact BIM support.</p>
BIMVSHAR-117	<p>UPDATE HAS/HAS NOT OCCURED - RECOVERY REQUIRED/NOT REQUIRED ON ABEND. Explanation: Describes weather or not an update has occurred, and as a result whether recovery will be required if an abend occurs. System Action: None. User Response: None.</p>
BIMVSHAR-118	<p>MODULE BIMVSBSG FOUND CMCB ADDRESS 0. Explanation: The storage module was found but the CMCB address was 0. System Action: None. User Response: None.</p>
BIMVSHAR-119	<p>ENQ FAILED RETURN=XXXXXXXX. Explanation: An enqueue function failed. System Action: Program aborts. User Response: Contact BIM support.</p>
BIMVSHAR-120	<p>BIMVSBER FORCING ABEND. Explanation: The error handling module forced an abend. System Action: Program aborts. User Response: Refer to other related messages.</p>
BIMVSHAR-121	<p>LOAD ERROR IN MODULE BIMVBLD FOR MOD XXXXXXXX. Explanation: BIMVBLD program encountered an error loading module xxxxxxxx. System Action: Program aborts. User Response: Refer to other related messages.</p>
BIMVSHAR-122	<p>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 BIM support.</p>
BIMVSHAR-123	<p>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.</p>
BIMVSHAR-124	<p>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.</p>
BIMVSHAR-125	<p>NO JOB/CICS RECORD FOUND IN BIMVSBTB. 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.</p>
BIMVSHAR-126	<p>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 BIM support.</p>
BIMVSHAR-127	<p>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.</p>

BIMVSHAR-128	<p>NOT ABLE TO ACTIV/ALLOC/SEND BATCH APPL. Explanation: The program encountered an error with the batch APPLID. System Action: None. User Response: None.</p>
BIMVSHAR-129	<p>ERROR FOR MOD XXXXXXXXX RTC WAS XXXXXXXXX RSN XXXXXXXXX. Explanation: A module encountered an error, and the return and reason code's were returned. System Action: Dependent. User Response: Refer to other related messages.</p>
BIMVSHAR-130	<p>DSN INPUT ONLY. Explanation: The open was intercepted, and the data set was open for input only. System Action: None. User Response: None.</p>
BIMVSHAR-131	<p>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.</p>
BIMVSHAR-132	<p>XXXXXXXX NOT INTERCEPTED. Explanation: xxxxxxxx DDNAME was not intercepted. System Action: None. User Response: None.</p>
BIMVSHAR-134	<p>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 BIM support.</p>
BIMVSHAR-135	<p>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.</p>
BIMVSHAR-136	<p>BIM-VSHARE I/O ROUTINE NOT LOADED. Explanation: The program got the address of the BIM-VSHARE VSAM I/O routine and it was zero. System Action: Program aborts. User Response: Call BIM support.</p>
BIMVSHAR-137	<p>ERROR TYPE IN VSAM RECORD. Explanation: The program encountered an invalid record type in the control file. System Action: Program aborts. User Response: Call BIM support.</p>
BIMVSHAR-138	<p>INVALID RECORD IN CONTROL FILE. Explanation: The program found an invalid record in the control file. System Action: Program aborts. User Response: Contact BIM support.</p>
BIMVSHAR-140	<p>VSAM ERROR. Explanation: The program tried to execute a VSAM function and failed. System Action: Program aborts. User Response: Refer to other related messages.</p>
BIMVSHAR-141	<p>SEND/RECEIVE FAILED - RETRY BEING ATTEMPTED. Explanation: The program issued a send or receive and failed. System Action: Program retries. User Response: None.</p>

BIMVSHAR-142	SEND/RECEIVE FAILED. Explanation: The program issued a send or receive and failed. System Action: Program aborts. User Response: Contact BIM support.
BIMVSHAR-143	EXLST ERROR - LIST ADDR XXXXXXXX. Explanation: VSAM encountered an error with the EXLST at address xxxxxxxx. System Action: Program aborts. User Response: Contact BIM support.
BIMVSHAR-144	REQUEST ERROR - REQUEST XXXXXXXX. Explanation: An internal invalid request was issued at address xxxxxxxx. System Action: Program aborts. User Response: Contact BIM support.
BIMVSHAR-145	KEY LENGTH ZERO. Explanation: A key length of zero was found for the file. System Action: Program aborts. User Response: Contact BIM support.
BIMVSHAR-146	SUBROUTINE ERROR (SUBROUTINE). Explanation: An error was encountered in the subroutine. System Action: Program aborts. User Response: Contact BIM support.
BIMVSHAR-148	RECEIVE ERROR RETURNED. Explanation: The communication program got an error during receive. System Action: Program aborts. User Response: Refer to other related messages.
BIMVSHAR-149	INVALID REQUEST CODE IN USER CALL. Explanation: The user program made an invalid direct call to BIM-VSHARE. System Action: The program aborts. User Response: Correct the request and rerun.
BIMVSHAR-150	BIMVSHARE NOT ACTIVE, DIRECT CALLS WILL BE IGNORED. Explanation: The user program made a direct call to BIM-VSHARE without it being active. System Action: None. User Response: Either activate BIM-VSHARE or ignore the message.
BIMVSHAR-151	TABLE xxx NOT FOUND Explanation: The table defined in BIMVSBDS was not loaded. System Action: Job Aborts. User Response: Make sure the table specified in BIMVSBDS is loaded. Refer to the section "Update the BIMVSBDS Parameter Member" on page 85 for more information.
BIMVSHAR-152	LABEL MACRO ERROR R15=xxxxxxx PROCESSING ABENDED Explanation: The VSE LABEL macro returned an error. System Action: Job Aborts. User Response: Contact BIM support.
BIMVSHAR-153	ERROR RETURNED FROM BIMXMCMD xxxxxxxx (xm) Explanation: The cross memory services module returned an error. System Action: Job Aborts. User Response: Retry. If problem persists, contact BIM support.
BIMVSHAR-154	USER EXCLUDED DATASET Explanation: DLBL specified 99/333 for expiration date. System Action: Dataset is excluded from BIM-VSHARE. User Response: None.

BIMVSHAR-199

(Informational Messages).

Explanation: These are informational messages only.

System Action: None.

User Response: None.

Journal Messages

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-201** **RECOVERY PARAMETER VALUES FOR THIS STEP ARE: JOBNAME=xxxxxxx PARTID=xx PGMSTEP=xxxxxxx PGMNAME=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.
System Action: Informational only.
User Response: None
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 on-line to insure 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 insure that it was created properly. Review the system message log for the allocated dataset name. If problem persists, contact technical support.
- BIMVSHAR-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, insure that enough space was allocated for the file. If not, change the journal space allocation on-line. 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 19 for more information. If problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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: Program aborts.
User Response: Rerun application after performing recovery functions if necessary. If problem persists, contact technical support.

- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 PARTITION parameter on either the JOB card or EXEC card.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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 problem persists, contact BIM technical support.
- BIMVSHAR-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.

BIMVSHAR-227	<p>RECOVERY MANAGER WAITING - REQUEST BUFFER IS FULL.</p> <p>Explanation: The recovery manager tried to enqueue a request block but the storage manager indicated that the buffer was full.</p> <p>System Action: When wait is finished, recovery manager tries again to enqueue the request block.</p> <p>User Response: Informational message only.</p>
BIMVSHAR-228	<p><attaching/detaching> I/O SUBTASK.</p> <p>Explanation: The I/O subtask is about to be attached or detached by the recovery manager.</p> <p>System Action: Informational only.</p> <p>User Response: None.</p>
BIMVSHAR-229	<p>RID - DD=<ddname> DATE=<yyddd> TIME=<hh:mm:ss:ht> SEQ=<seq#>.</p> <p>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.</p> <p>System Action: Informational message only.</p> <p>User Response: See message prior to this one in log for more information.</p>
BIMVSHAR-230	<p>I/O SUBTASK ENDED <NORMALLY/ABNORMALLY> - RC = nnn.</p> <p>Explanation: The recovery manager has received notification that the I/O subtask has finished with the return code shown.</p> <p>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 an abend.</p> <p>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.</p>
BIMVSHAR-231	<p>PROCESSING STOPPED ON FOLLOWING JOURNAL RECORD:</p> <p>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.</p> <p>System Action: Informational message only.</p> <p>User Response: See message prior to this one in log for more information.</p>
BIMVSHAR-232	<p>RECOVERY <STARTED/FINISHED> FOR PGMSTEP=xxxxxxx PGMNAME=xxxxxxx JOBNAME=xxxxxxx PARTID=xx.</p> <p>Explanation: Recovery has either started or finished for the jobstep identified in the message.</p> <p>System Action: Continue normally.</p> <p>User Response: Informational only.</p>
BIMVSHAR-233	<p><REWRITE/DELETE> RECORD JOURNALED DURING ABEND AND DOES NOT MATCH BEFORE IMAGE - RECORD BYPASSED.</p> <p>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.</p> <p>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.</p> <p>User Response: If more information is required about the records in question, find them in the journal file using the information from the messages.</p>
BIMVSHAR-234	<p>CURRENTLY PROCESSING FOLLOWING JOURNAL RECORD.</p> <p>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.</p> <p>System Action: Continue normally.</p> <p>User Response: Informational only. See accompanying messages.</p>
BIMVSHAR-235	<p>RECOVERY STOPPING AT LAST USER SYNCPOINT.</p> <p>Explanation: Recovery has found a user SYNCPOINT record and is discontinuing its processing.</p> <p>System Action: Processing stops.</p> <p>User Response: None.</p>

- BIMVSHAR-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.
- BIMVSHAR-237** **RBA/RRN ADDS BYPASSED:**
Explanation: RBA/RRN records were bypassed.
System Action: Processing continues.
User Response: None.
- BIMVSHAR-238** **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: Program aborts.
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 problem persists, contact technical support.
- BIMVSHAR-239** **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: Program aborts.
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 problem persists, contact technical support.
- BIMVSHAR-240** **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: Program aborts.
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 problem persists, contact technical support.
- BIMVSHAR-241** **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: Program aborts.
User Response: Check for further messages from the storage manager. If problem persists, contact technical support.
- BIMVSHAR-242** **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: Program aborts.
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 problem persists, contact technical support.
- BIMVSHAR-243** **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: Program aborts.
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 problem persists, contact technical support.

BIMVSHAR-244	<p>BAD RETURN CODE FROM STIMER SERVICES (WAIT).</p> <p>Explanation: This module requested timer services from the operating system but the request could not be completed normally.</p> <p>System Action: Program aborts.</p> <p>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.</p>
BIMVSHAR-245	<p>STORAGE MGR ENQUEUE OPERATION FAILED.</p> <p>Explanation: The storage manager failed trying to enqueue an I/O request block for the I/O subtask.</p> <p>System Action: Program aborts.</p> <p>User Response: Check for other messages from the storage manager. Rerun recovery process and if problem persists, contact technical support.</p>
BIMVSHAR-246	<p>STORAGE MGR DESTROY OPERATION FAILED.</p> <p>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.</p> <p>System Action: Return normally.</p> <p>User Response: This message is informational only and will not impact the recovery operation. If message recurs, contact technical support.</p>
BIMVSHAR-247	<p><REWRITE/DELETE> TYPE JOURNAL RECORD FOUND WITH NO MATCHING BEFORE IMAGE.</p> <p>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.</p> <p>System Action: Program aborts.</p> <p>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 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 problem persists, contact technical support.</p>
BIMVSHAR-248	<p>ERROR ON CALL TO JOURNAL I/O MODULE.</p> <p>Explanation: The Journal I/O module could not complete a request because of an unexpected error. Program storage may have been corrupted.</p> <p>System Action: Program aborts.</p> <p>User Response: Look for messages from the journal I/O module. If rerunning recovery produces the same message, contact technical support.</p>
BIMVSHAR-249	<p>ERROR ON CALL TO STORAGE MANAGER.</p> <p>Explanation: The storage manager could not complete a request because of an unexpected error. Program storage may have been corrupted.</p> <p>System Action: Program aborts.</p> <p>User Response: Look for messages from the storage manager module. If rerunning recovery produces same message, contact technical support.</p>
BIMVSHAR-250	<p>ERROR ON CALL TO CHANGE LIST FORMATTER.</p> <p>Explanation: The change list formatter module could not complete a request because of an unexpected error. Program storage may have been corrupted.</p> <p>System Action: Program aborts.</p> <p>User Response: Look for messages from the change list formatter module. If rerunning recovery produces same message, contact technical support.</p>
BIMVSHAR-251	<p>GETMAIN <nnnnnnnn> HEX BYTES FAILED FOR <area>.</p> <p>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.</p> <p>System Action: Program aborts.</p> <p>User Response: Increase storage available to recovery job using the partition parameter on either the jobcard or the step card. If problem persists and number of bytes requested appears excessive, contact BIM technical support.</p>

- BIMVSHAR-252** 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: Program aborts.
User Response: Review the message log for further messages from the table manager and act accordingly. If problem persists contact technical support.
- BIMVSHAR-253** UNABLE TO PROCESS RECOVERY PARAMETERS.
Explanation: The parameter list processor was invoked from this program but did not complete normally.
System Action: Program aborts.
User Response: Review the message log for further messages from the parameter list processor. If problem persists contact technical support.
- BIMVSHAR-254** 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: Program aborts.
User Response: Retry the recovery process. If problem persists, contact technical support.
- BIMVSHAR-255** 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.
- BIMVSHAR-256** 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.
- BIMVSHAR-257** 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 as 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.
- BIMVSHAR-258** 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 problem persists, contact technical support.
- BIMVSHAR-259** 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: Program aborts.
User Response: Look for messages from the storage manager module. If rerunning recovery produces same message, contact technical support.
- BIMVSHAR-260** 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: Program aborts.
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.

- BIMVSHAR-261** 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: Program aborts.
User Response: Retry the recovery process. If problem persists, contact technical support.
- BIMVSHAR-262** 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: Program aborts.
User Response: Look for messages from the change list processor module. If rerunning recovery produces same message, contact technical support.
- BIMVSHAR-263** 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 insure that the proper journal file is being used for the recovery process. Otherwise, check within context of application to see if on-line activity could account for the failures. If problem persists, contact technical support.
- BIMVSHAR-264** 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: Program aborts.
User Response: Increase storage available to recovery job using the partition parameter on either the jobcard or the step card. If problem persists and number of bytes requested appears excessive, contact BIM technical support.
- BIMVSHAR-265** 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: Program aborts.
User Response: Possible storage corruption may have occurred. If problem persists contact technical support.
- BIMVSHAR-269** <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.
- BIMVSHAR-270** 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: Program aborts.
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 PARTITION parameter of the JOB card or the EXEC card and rerun. If problem persists, contact technical support.

- BIMVSHAR-271 <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 problem persists, contact technical support.
- BIMVSHAR-272 SEGMENT TABLE FULL.
Explanation: The table used to record file information is full and processing cannot continue.
System Action: Program aborts.
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.
- BIMVSHAR-273 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: Program aborts.
User Response: Increase storage available to recovery job using the PARTITION parameter on either the JOB card or EXEC card. If problem persists and number of bytes requested appears excessive, contact BIM technical support.
- BIMVSHAR-274 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: Program aborts.
User Response: Insure that the proper journal file and recovery JCL are being used. If so, add the necessary file to the recovery job. If problem persists contact technical support.
- BIMVSHAR-275 LABEL MACRO ERROR R15=xxxxxxx yyyyyyy
Explanation: Empty label area, Dataset not found. "xxxxxxx" is the return code, "yyyyyyy" is the location.
System Action: Program aborts.
User Response: Insure DLBLs are supplied for all files to be recovered.
- BIMVSHAR-276 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.
- BIMVSHAR-277 '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.
- BIMVSHAR-278 PARMS ACCEPTED - <JOBNAME=xxxxxxx PARTID=xx PGMSTEP=xxxxxxx PGMNAME=xxxxxxx>.
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.
- BIMVSHAR-279 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.

BIMVSHAR-280	<p>SYNCPOINT OPTION SPECIFIED - RECOVER TO LAST. Explanation: The parameter specified recovery back to last syncpoint. System Action: Recovery to last syncpoint. User Response: None.</p>
BIMVSHAR-281	<p>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: Program aborts. User Response: Check for further messages from the parameter card reader module. Review the JCL being used to run the recovery process and insure that the SYSIN parameter dataset is supplied. If problem persists, contact technical support.</p>
BIMVSHAR-282	<p>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: Program aborts. User Response: Review the input parameter cards and correct any format or spelling errors.</p>
BIMVSHAR-283	<p>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: Program aborts. User Response: Correct the input parameter cards and rerun.</p>
BIMVSHAR-284	<p>'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: Program aborts. User Response: The RECOVERALL option cannot be used with other options. Correct the input cards and rerun.</p>
BIMVSHAR-285	<p><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: Program aborts. User Response: Correct the card in error and rerun.</p>
BIMVSHAR-286	<p><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: Program aborts. User Response: Correct the card in error and rerun.</p>
BIMVSHAR-287	<p>JOBNAME OR PARTID PARAMETER REQUIRED BUT NOT FOUND. Explanation: Neither the JOBNAME nor the PARTID parameter card was found. At least one is required for specifying stop conditions. System Action: Program aborts. User Response: Add the appropriate card(s) to the input parameter dataset and rerun.</p>
BIMVSHAR-288	<p>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: Program aborts. User Response: Insure that the parameter dataset is properly specified in the recovery JCL. Look for further messages in the job log.</p>
BIMVSHAR-289	<p>ERROR <OPEN/CLOSE/WRITE>FAILED FOR JOURNAL DATASET ERROR RETURNED=xx Explanation: The operation specified for the journal dataset failed with 'xx' being the reason. System Action: Program aborts. User Response: Insure the journal dataset was defined properly and a DLBL for BIMVSHR is present in the jobs JCL.</p>

The Return code can be one of the following:

- 01 GETMAIN/GETVIS failed
- 02 FREEMAIN/FREEVIS failed
- 03 FREEMAIN/FREEVIS wrong area
- 04 Bad block length field
- 05 Block too large
- 07 VARBLK record too large
- 10 End of file
- 11 Misc write error
- 12 OPEN/CLOSE error
- 20 File already closed

Presentation Services Messages

BIMVSHAR-401	<p>FSM WAS PASSED AN INVALID FSM ID.</p> <p>Explanation: The Finite State Machine module was invoked with an unknown Finite State Machine ID. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-402	<p>FSM WAS PASSED AN INVALID INPUT STATE.</p> <p>Explanation: The input state for the specified Finite State Machine was not within the range of possible values. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-403	<p>FSM WAS PASSED AN INVALID OPERATION.</p> <p>Explanation: The Finite State Machine module received a request to perform an undefined operation. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-404	<p>FSM ENCOUNTERED AN EXECUTION TIME ERROR.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-405	<p>INVALID CONTROL BLOCK NUMBER REQUESTED.</p> <p>Explanation: The control block formatter module received a request for the address of an unknown control block. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-406	<p>CONTROL FILE PROGRAM FAILED ON <oper>. FILE STATUS = <nn>.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: File status code should reveal the nature of the problem (i.e., control record not found, wrong file definition, etc.).</p>
BIMVSHAR-407	<p>CONTROL FILE PROGRAM FAILED - RETURN CODE WAS NOT SET.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-408	<p>CONTROL FILE PROGRAM FAILED - UNKNOWN RETURN CODE.</p> <p>Explanation: The Control File Program was invoked by this module but finished with an unknown return code. Program code field may have been corrupted.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 insure 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 persists, contact technical support.
- BIMVSHAR-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 insure 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.
- BIMVSHAR-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 insure 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.
- BIMVSHAR-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 insure 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.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

BIMVSHAR-417	<p>UNKNOWN RETCODE FROM RM.</p> <p>Explanation: This module received an unknown/invalid return code from the Resource Manager. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-418	<p>ABEND REQUESTED BY RM.</p> <p>Explanation: The return code on a call to the Resource Manager indicates that an abend situation was encountered.</p> <p>System Action: Session termination.</p> <p>User Response: Look for further messages from the Resource Manager and take action accordingly. If none are found, storage may have been corrupted. If problem persists, contact technical support.</p>
BIMVSHAR-419	<p>INVALID SESSION ID SPECIFIED.</p> <p>Explanation: The session ID parameter passed to this module contained illegal characters. The transaction program's storage area may have been corrupted.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-420	<p>TYPE PARAMETER CONTAINS INVALID VALUE.</p> <p>Explanation: The parameter passed to this module specifying type of deactivation was unknown value. The transaction program's storage area may have been corrupted.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-421	<p>NO PARMLIST RETURNED FROM RM.</p> <p>Explanation: An invalid parameter list address was returned from a call to the Resource Manager.</p> <p>System Action: Session termination.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-422	<p>WRONG REC TYPE RETURNED FROM RM.</p> <p>Explanation: This module received an unknown or invalid record type from the Resource Manager. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-423	<p>UNKNOWN RETCODE FROM RM.</p> <p>Explanation: This module received an unknown/invalid return code from the Resource Manager. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-424	<p>SESSION DEACT MAY NOT HAVE COMPLETED.</p> <p>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.</p> <p>System Action: Continue normally - warning only.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-425	<p>RECEIVED INVALID ADDRESS OF PARM LIST.</p> <p>Explanation: The parameter list address passed to this program was invalid. Possible storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 insure 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.
- BIMVSHAR-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.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

BIMVSHAR-434	<p>INVALID RECORD ADDR RETURNED FROM RM.</p> <p>Explanation: An invalid parameter list address was returned from a call to the Resource Manager.</p> <p>System Action: Session termination.</p> <p>User Response: If problem persists, contact technical support.</p>
BIMVSHAR-435	<p>LUNAME PARM NOT FOUND IN PCB.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check control file session information record to insure 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.</p>
BIMVSHAR-436	<p>MODE PARM NOT FOUND IN MCB.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check control file session information record to insure 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.</p>
BIMVSHAR-437	<p>ALLOCATE FAILED - RCB NOT CREATED.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>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.</p>
BIMVSHAR-438	<p>ALLOCATE FAILED - PROTOCOL VIOLATION.</p> <p>Explanation: The Resource Manager returned a conversation failure record on an allocate request and the reason code indicated a protocol violation was detected.</p> <p>System Action: Session termination.</p> <p>User Response: If the problem persists, contact technical support.</p>
BIMVSHAR-439	<p>ALLOCATE FAILED - SESSION OUTAGE.</p> <p>Explanation: The Resource Manager returned a conversation failure record on an allocate request and the reason code indicated a session outage was detected.</p> <p>System Action: Session termination.</p> <p>User Response: If the problem persists, contact technical support.</p>
BIMVSHAR-440	<p>ALLOCATE FAILED RETURNED FROM RM.</p> <p>Explanation: The Resource Manager set the return code on an allocate request to 'allocate failed'.</p> <p>System Action: The allocate verb return code is set to 'allocate failed' and control is returned to the transaction program.</p> <p>User Response: Check for further messages. If none are found and problem persists, contact technical support.</p>
BIMVSHAR-441	<p>SYNC LEVEL UNSUPPORTED ON ALLOCATE.</p> <p>Explanation: The Resource Manager set the return code on an allocate request to 'synclevel unsupported'. The parameter list may have been corrupted.</p> <p>System Action: The allocate verb return code is set to 'synclevel unsupported' and control is returned to the transaction program.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

BIMVSHAR-451	<p>TYPE PARM NOT IN VALID RANGE.</p> <p>Explanation: The deallocation type parameter passed by the transaction program was not within the range of possible values. Storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-452	<p>RM RETURNED OTHER THAN RCB-DEALLOCATED REC.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-453	<p>INVALID SOURCE FOUND ON RECORD FROM RM.</p> <p>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.</p> <p>System Action: Session termination</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-454	<p>NONZERO RETURN CODE FROM CALL TO RM.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for further messages from the resource manager. If problem persists, contact technical support.</p>
BIMVSHAR-455	<p>INVALID REC ADDRESS RETURNED FROM RM.</p> <p>Explanation: The address of the record expected from the resource manager on a deallocation request was unusable. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-456	<p>UNSUPPORTED PARM: TYPE = CONFIRM.</p> <p>Explanation: The transaction program requested deallocation of the conversation with confirm level synchronization. This type is unsupported. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-457	<p>UNSUPPORTED PARM: TYPE = SYNCLEVEL.</p> <p>Explanation: The transaction program requested deallocation of the conversation with 'synclevel' synchronization. This type is unsupported. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-458	<p>UNSUPPORTED PARM: TYPE = ABEND SVC.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-459	<p>UNSUPPORTED PARM: TYPE = ABEND TIMER.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

BIMVSHAR-460	<p>UNSUPPORTED PARM: TYPE = LOCAL.</p> <p>Explanation: Local deallocation is unsupported. This type is only used when the transaction program wants to cleanup the local LU after being informed that deallocation took place during a previously issued verb. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-461	<p>UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-462	<p>DEALLOCATION CLEANUP PERFORMED.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for other messages to further identify problem. If problem persists, contact technical support.</p>
BIMVSHAR-463	<p>SEND DATA NOT ON LOGICAL RECORD BOUNDARY.</p> <p>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.</p> <p>System Action: Deallocation cleanup will be performed and the session will be terminated.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-464	<p>INVALID REC ADDRESS RETURNED FROM HS.</p> <p>Explanation: The address of the record expected from half session manager on a deallocation request was unusable. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-465	<p>NONZERO RETURN CODE FROM CALL TO HS.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for further messages from the resource manager. If problem persists, contact technical support.</p>
BIMVSHAR-466	<p>FSM CONVERSATION LOGIC OR PARM ERROR.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-467	<p>FSM CONVERSATION ENCOUNTERED ABEND.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

BIMVSHAR-468	<p>NONZERO RETCODE FROM DEALLOCATION CLEANUP.</p> <p>Explanation: The deallocation cleanup program was called to deallocate the conversation abnormally but could not complete normally either. Program storage has possibly been corrupted.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-469	<p>INVALID PARM LIST ADDRESS FOUND.</p> <p>Explanation: The transaction program passed a parameter list address that was negative or zero.</p> <p>System Action: Session termination.</p> <p>User Response: Check for other related messages. If problem persists, contact technical support.</p>
BIMVSHAR-470	<p>RCBID PARM DOES NOT MATCH ALLOCATED RCB.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-471	<p>FSM CONVERSATION ENCOUNTERED ABEND.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-472	<p>UNKNOWN OUTPUT FROM FSM CONVERSATION.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-473	<p>FSM CONVERSATION LOGIC OR PARM ERROR.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-474	<p>FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-475	<p>FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

BIMVSHAR-484	<p>RCBID PARM DOES NOT MATCH ALLOCATED RCB.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-485	<p>INVALID TYPE PARAMETER FOUND.</p> <p>Explanation: The type parameter passed to this module was invalid (not FLUSH, CONFIRM, or SYNC_LEVEL).</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-486	<p>TYPE = CONFIRM NOT SUPPORTED.</p> <p>Explanation: The type parameter passed to the Prepare-to-Receive module was CONFIRM. This option is not supported.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-487	<p>TYPE UNSUPPORTED (SYNCPT AND RCB.LEVEL NOT NONE).</p> <p>Explanation: The type parameter passed to the Prepare-to-Receive module was SYNCPT. This option is only supported if the RCB SYNCLEVEL is none.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-488	<p>DEALLOCATION CLEANUP INVOKED.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for other messages to further identify the problem. If problem persists, contact technical support.</p>
BIMVSHAR-489	<p>UNKNOWN OUTPUT CODE FROM FSM CONVERSATION.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-490	<p>FSM_ERROR_OR_FAILURE LOGIC OR PARM ERROR.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-491	<p>FSM_ERROR_OR_FAILURE ENCOUNTERED ABEND.</p> <p>Explanation: The Finite State Machine processor encountered an execution time error changing the ERROR_OR_FAILURE FSM to the RESET state.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

Presentation Messages

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 error persists, contact technical support.
- BIMVSHAR-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 error persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

BIMVSHAR-518	<p>UNEXPECTED CONFIRMED REC RECVD FROM HS.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-519	<p>UNEXPECTED RTS RESPONSE RECVD FROM HS.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for messages that further identify the problem including the CICS log. Also, check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-520	<p>UNKNOWN RECORD TYPE FROM RECVD FROM HS.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-521	<p>STATE CHECK CHANGING TO RCV_IMMED STATE.</p> <p>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.</p> <p>System Action: Start error processing leading to eventual session termination.</p> <p>User Response: Check for other related messages and act accordingly. If problem persists, contact technical support.</p>
BIMVSHAR-522	<p>FOUND NONZERO VTAM SENSE CODE IN LULUCB.</p> <p>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.</p> <p>System Action: Continue with error processing leading to eventual program termination.</p> <p>User Response: Check for other related messages and act accordingly. If problem persists, contact technical support.</p>
BIMVSHAR-523	<p>INVALID PARM LIST ADDRESS FOUND.</p> <p>Explanation: The batch program passed a parameter list address that was negative or zero.</p> <p>System Action: Session termination.</p> <p>User Response: Check for other related messages. If problem persists, contact technical support.</p>
BIMVSHAR-524	<p>INVALID REC ADDRESS RETURNED FROM RM.</p> <p>Explanation: The address of the record expected from the resource manager was unusable. Program storage corruption may have occurred.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-525	<p>NONZERO RETURN CODE FROM CALL TO RM.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check for further messages from the resource manager. If problem persists, contact technical support.</p>
BIMVSHAR-527	<p>INVALID SOURCE FOUND ON RECORD FROM RM.</p> <p>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.</p> <p>System Action: Session termination.</p> <p>User Response: Check application program for possible storage violations. If problem persists, contact technical support.</p>

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

- BIMVSHAR-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 being corrupted.
System Action: Session termination.
User Response: Check application program for possible storage violations. If problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists call BIM technical support.
- BIMVSHAR-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 problem persists call BIM technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 error persists, contact technical support.
- BIMVSHAR-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 error persists, contact technical support.

CICS Messages

- BIMVSHAR-601** JOB <jobname> WAS CANCELLED BY OPERATOR <user-id> FROM TERMID <termid> <date/time>.
Explanation: The batch job was canceled from the on-line 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.
- BIMVSHAR-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.
- BIMVSHAR-603** <user/auto><syncpoint/rollback><succeeded/failed> IN THE ON-LINE PARTNER <date/time>.
Explanation: A syncpoint or rollback was attempted by the on-line 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.
- BIMVSHAR-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 on-line response is being degraded, review the syncpoint options for the job and adjust accordingly.
- BIMVSHAR-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.
- BIMVSHAR-606** SYNCPOINTS PERFORMED - USER: nnnn AUTO: nnnn ENQ FREED: nnnn.
Explanation: At job completion, BIM-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.
- BIMVSHAR-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.
- BIMVSHAR-608** CURRENT CPU/LPAR ID is <xxxxx>.
Explanation: This program has determined the identity of the CPU that is it running on.
System Action: None.
User Response: Informational message only.

BIMVSHAR-610	<p>FILE TYPE UNSUPPORTED, DDNAME= '<ddname>' FCT= '<fctname>', RETURN CODE = <xxxx>, ignored.</p> <p>Explanation: The file identified by batch DDNAME and CICS FCT entry name is a type which is unsupported at this time.</p> <p>System Action: Abend batch job.</p> <p>User Response: Check the file definition to insure it is a supported type. If it is, a storage violation may have occurred. If problem persists, contact technical support.</p>
BIMVSHAR-615	<p>CMS INITIALIZATION HAS BEEN STARTED <date/time>.</p> <p>Explanation: The CICS CMS initialization program has executed the necessary request to start the cross memory service for BIM-VSHARE.</p> <p>System Action: None.</p> <p>User Response: Informational message only.</p>
BIMVSHAR-616	<p>CMS INITIALIZATION HAS BEEN COMPLETED <date/time>.</p> <p>Explanation: The CICS CMS initialization program has complete the necessary request to start the cross memory service for BIM-VSHARE.</p> <p>System Action: None.</p> <p>User Response: Informational message only.</p>
BIMVSHAR-617	<p>ABEND EXIT ENTERED WITH ERROR CODE '<abcode>'.</p> <p>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.</p> <p>System Action: Program terminated.</p> <p>User Response: Determine from the abend code what caused the failure. Correct failure condition and re-execute the transaction.</p>
BIMVSHAR-618	<p>SVC NUMBER FOR CMS INITIALIZATION IS INVALID PROCESSING TERMINATED.</p> <p>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.</p> <p>System Action: Program terminated.</p> <p>User Response: Determine why the SVC numbers do not match. Correct the discrepancy and re-execute the transaction.</p>
BIMVSHAR-619	<p>CMS INITIALIZATION REQUEST REFUSED BY OS, REASON: SEE MESSAGE ABOVE. PROCESSING TERMINATED.</p> <p>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.</p> <p>System Action: Program terminates.</p> <p>User Response: Determine from the above message what caused the refusal. Correct the discrepancy and re-execute the transaction.</p>
BIMVSHAR-620	<p>SYSTEM GETMAIN FOR '<function>' FAILED WITH RC='<xxxx>' PROCESSING TERMINATED.</p> <p>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.</p> <p>System Action: Program terminates.</p> <p>User Response: Determine why CICS refused the request by examination of the return code. Correct the problem and re-execute the transaction.</p>
BIMVSHAR-621	<p>LOCATED DCB: <dcbname>.</p> <p>Explanation: The CICS CMS initialization program has located the Data Control Block for file "dcbname".</p> <p>System Action: Processing continues.</p> <p>User Response: Informational message only.</p>
BIMVSHAR-622	<p>WARNING -- UNSUCCESSFUL FREEMAIN FOR IRC.</p> <p>Explanation: The CICS CMS initialization program has encountered an abnormal condition. A request to free storage returned with an error condition.</p> <p>System Action: Processing continues.</p> <p>User Response: Informational message only. CICS storage may be reduced because of the failed FREEMAIN.</p>

- BIMVSHAR-623 <read><write> TO TEMPSTOR QUEUE <queuname> 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.
- BIMVSHAR-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.
- BIMVSHAR-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 BIMVSHAR-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.
- BIMVSHAR-627 EXECUTION of GIVEN SVC FAILED, PROCESSING TERMINATED.
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.
- BIMVSHAR-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.
- BIMVSHAR-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 can not be loaded.
User Response: Determine from associated messages why the open attempt failed. Correct the problem and re-execute the transaction.
- BIMVSHAR-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.
- BIMVSHAR-640 INITIAL GETMAIN FAILED FOR <nnnn> BYTES REQUESTED.
Explanation: The on-line 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 partition. If problem persists and amount of storage requested appears excessive, contact technical support.

BIMVSHAR-641	<p>UNKNOWN INTERNAL REQUEST CODE, <xxxxxx> WAS USED.</p> <p>Explanation: An internal request code passed from the batch transaction program to the on-line transaction program is unknown.</p> <p>System Action: Abend batch job.</p> <p>User Response: Storage may have been corrupted either on the batch or the on-line side. Check application for possible storage violations. If problem persists, contact technical support.</p>
BIMVSHAR-642	<p>REQUEST CODE <xxxxx> IS INVALID AT START OF SESSION.</p> <p>Explanation: The internal request code <nnnn> passed from the batch transaction program to the on-line transaction program is invalid at the start of a session.</p> <p>System Action: Abend batch job.</p> <p>User Response: Storage may have been corrupted either on the batch or the on-line side. Check application for possible storage violations and rerun. If problem persists, contact technical support.</p>
BIMVSHAR-643	<p>INVALID CHARACTERS RECEIVED IN DDNAME: <ddname>.</p> <p>Explanation: The on-line 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.</p> <p>System Action: Abend batch job.</p> <p>User Response: Storage may have been corrupted either on the batch or the on-line side. Check application for possible storage violations and rerun. If problem persists, contact technical support.</p>
BIMVSHAR-644	<p>LINK FAILED. PROGRAM NAME '<pgname>' IS INVALID.</p> <p>Explanation: An attempt was made to link to the indicated program but the link failed.</p> <p>System Action: Abend batch job.</p> <p>User Response: Check the CICS definitions for the program to see that it is properly installed in the CICS partition being accessed.</p>
BIMVSHAR-645	<p>ACKNOWLEDGEMENT MISSING FOR "READY-TO-SYNC" REQUEST.</p> <p>Explanation: The on-line program indicated that it was ready to issue a syncpoint but the batch program did not confirm that it was also ready.</p> <p>System Action: Abend batch job.</p> <p>User Response: Storage may have been corrupted either on the batch or the on-line side. Check application for possible storage violations and rerun. If problem persists, contact technical support.</p>
BIMVSHAR-646	<p>MAX ON-LINE RESOURCES HELD LEVEL HAS BEEN REACHED. TASKID '<tasknr>' ABENDING.</p> <p>Explanation: The batch CICS transaction noticed that BIM-VSHARE is holding an abnormally high level of record locks.</p> <p>System Action: Informational only.</p> <p>User Response: If this was not expected, consider releasing resource through syncpointing. Refer to the chapter "RECOVERY" on page 19 for details. This is only informational for a warning mechanism. Too many locks take up CICS storage.</p>
BIMVSHAR-647	<p>END-SESSION REQUESTED MORE THAN ONCE. TRANSACTION <tranid> ABENDING.</p> <p>Explanation: The on-line program received two requests to end the session from the batch program.</p> <p>System Action: Abend batch job.</p> <p>User Response: Storage may have been corrupted either on the batch or the on-line side. Check application for possible storage violations and rerun. If problem persists, contact technical support.</p>
BIMVSHAR-648	<p>CMS NOT AVAILABLE IN TARGET CICS PARTITION.</p> <p>Explanation: During step initialization, the batch job was defined to use Cross Memory Services but after connecting to the CICS partition, it was found that CMS was not installed in the CICS partition.</p> <p>System Action: Abend the job.</p> <p>User Response: Initialize CMS in the CICS partition and rerun the job. Refer to the section "Error! Reference source not found." on page Error! Bookmark not defined.</p>

- BIMVSHAR-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.
- BIMVSHAR-651 <trandid> <termid> <taskid> <connecting/disconnecting> JOB: <jobname> <errors??>.
Explanation: This is the message in the CICS log explaining the starting or completion of a BIM-VSHARE batch job. The CICS transaction ID used, the terminal ID, the time, the job name..
System Action: Informational only.
User Response: None. This is an audit trail to determine batch activity during the day or job start/stop times.
- BIMVSHAR-652 CICS PARTITION IS RUNNING ON CPU/LPAR ID(<CPUID>) USING LU62 CONNECTIVITY.
Explanation: The batch job was defined to use CMS but BIM-VSHARE determined that the batch job and the CICS partition were running on a different CPU or LPAR. The CICS CPU ID is displayed. Because of this, BIM-VSHARE can not 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.
- BIMVSHAR-653 CICS FUNCTION (<function ident>) FAILED WITH "<msg>" ERROR.
Explanation: A CICS on-line 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.
- BIMVSHAR-654 JOB HISTORY RECORDING TO CONTROL FILE "<file>" FOR JOB <jobname> 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.
- BIMVSHAR-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.
- BIMVSHAR-661 APPC FREE SESSION FAILED WITH RC='<nnnnn>'.
Explanation: The CICS on-line 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.
- BIMVSHAR-662 INITIAL CAPTURE OF INITIATING PARTNER INFO FAILED RETURN CODE WAS: '<nnnnn>'.
Explanation: The CICS on-line 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.

- BIMVSHAR-666** ERROR OCCURRED DURING STORAGE MANAGEMENT, RETURN CODE WAS <nnnn>.
Explanation: The CICS on-line 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.
- BIMVSHAR-667** ROLLEDBACK CONDITION TRIPPED ON CURRENT SYNCPOINT. PROCESSING TERMINATED.
Explanation: The CICS on-line 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.
- BIMVSHAR-671** FILE ACCESS FOR DDNAME '<ddname>' FCTENTRY '<fctname>', RETURNED ERROR CONDITION '<nnnn>'.
Explanation: An on-line 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.).
- BIMVSHAR-672** KEY CHANGED ON DUPKEY REPOSITION REQUEST DD=<ddname> FCT=<fctname>.
Explanation: The on-line 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 on-line side. Check application for possible storage violations and rerun. If problem persists, contact technical support.
- BIMVSHAR-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 partition. The remote partition's connection is or has been out of service. The remote partition's connection must be re-installed to place it back into service and re-establish its VTAM identity.
- BIMVSHAR-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.
- BIMVSHAR-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.

BIMVSHAR-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 partition. The remote partition's connection must be re-installed to place it back into service and re-establish its VTAM identity.

Half Session Messages

BIMVSHAR-701	<p>ILLEGAL RECORD TYPE WAS PASSED TO HS.</p> <p>Explanation: The Half Session manager was passed a bad record type.</p> <p>System Action: The Half Session manager returns to the calling program passing a bad return code.</p> <p>User Response: Possible storage corruption by the application program. Check for potential problems with the application. If problem persists, contact your technical support group.</p>
BIMVSHAR-702	<p>ADDRESS IN RCB DOES NOT POINT TO A HS TO RM RECORD.</p> <p>Explanation: The Half Session Manager did not find the address of a HS to RM Record in the RCB.</p> <p>System Action: The Half Session Manager returns to the calling program with a bad return code.</p> <p>User Response: Possible storage corruption by the application program. Check for potential problems with the application. If problems persist, contact your technical support group.</p>
BIMVSHAR-703	<p>A RM TO HS RECORD WAS RECEIVED BUT THIS TYPE IS NOT SUPPORTED.</p> <p>Explanation: The Half Session Manager received a Resource Manager Record but currently this is not supported by the Batch Logical Unit.</p> <p>System Action: The Half Session Manager returns to the calling program with a bad return code.</p> <p>User Response: Contact your technical support group.</p>
BIMVSHAR-704	<p>ILLEGAL RECORD CODE WAS PASSED TO HS.</p> <p>Explanation: The Half Session Manager received a record with a erroneous major code.</p> <p>System Action: The Half Session Manager returns to the calling program with a bad return code.</p> <p>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.</p>
BIMVSHAR-705	<p>ILLEGAL RECORD CODE WAS PASSED TO HS.</p> <p>Explanation: While trying to process a record from Presentation Services, an erroneous minor code was found in the record.</p> <p>System Action: The Half Session Manager returns a bad return code to Presentation Services.</p> <p>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.</p>
BIMVSHAR-706	<p>STRAY SIGNAL RECEIVED. IGNORED.</p> <p>Explanation: The Half Session Manager received a stray signal from the other Logical Unit.</p> <p>System Action: The Half Session Manager returns with a bad return code.</p> <p>User Response: Try operation again. If message persists, contact your technical support group.</p>
BIMVSHAR-707	<p>THE LENGTH OF THE LARGEST BUFFER IS ERRONEOUS. IT IS EQUAL TO ZERO.</p> <p>Explanation: The Half Session Manager found the length of the largest buffer to be zero.</p> <p>System Action: The Half Session Manager returns with a bad return code.</p> <p>User Response: Contact your technical support group.</p>
BIMVSHAR-708	<p>THE LENGTH OF THE LARGEST BUFFER IS ERRONEOUS. IT IS A NEGATIVE.</p> <p>Explanation: The Half Session Manager found the length of the largest buffer to be a negative number.</p> <p>System Action: The Half Session Manager returns with a bad return code.</p> <p>User Response: Contact your technical support group.</p>
BIMVSHAR-709	<p>BAD SEND DATA RECORD TYPE WAS PASSED TO HALF SESSION.</p> <p>Explanation: The Half Session Manager was processing a send data record and found the type to be unknown.</p> <p>System Action: The Half Session Manager returns with a bad return code.</p> <p>User Response: Possible storage corruption by the application program. Check for potential application problem. If the message persists, contact your technical support group.</p>

- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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 enciphering 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) BIM-VSHARE background transaction, TCCB, has security defined. Refer to the section "PCT Definitions" on page 92 for an explanation on this transaction and security.
- 2) BIM-VSHARE background transaction, TCCB, is disabled.
- 3) BIM-VSHARE background transaction, TCCB, is not defined.
- 4) BIM-VSHARE program BIMVSC00 is not defined.
- 5) BIM-VSHARE program BIMVSC00 is disabled.
- 6) BIM-VSHARE CICS load library is not defined to the RPL.

BIMVSHAR-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.
BIMVSHAR-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.
BIMVSHAR-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.
BIMVSHAR-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.
BIMVSHAR-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.
BIMVSHAR-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

- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.

- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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

- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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 problem persists, contact technical support.
- BIMVSHAR-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 partition 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 partition is correctly located in the control file. If the problem persists, contact the technical support.
- BIMVSHAR-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.
- BIMVSHAR-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 partition size problem. Increase partition size and try the operation again. If problem persists, contact technical support.
- BIMVSHAR-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 partition size problem. Increase the partition size and try the operation again. If the problem persists, call the technical support.

- BIMVSHAR-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 partition size problem. Increase the partition size and try the operation again. If the problem persists, call the technical support.
- BIMVSHAR-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 partition size problem. Increase the partition size and try the operation again. If the problem persists, call the technical support.
- BIMVSHAR-912** **THE OPENING OF THE VTAM ACB FAILED. REASON = XYYY.**
Explanation: While opening the ACB, the LU Network Manager received the reason code XYYY.
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.
- BIMVSHAR-913** **VTAM FAILURE. RC=XX FDBK2=XX IN SENSE=XXXXXXXXX OUT SENSE=XXXXXXXXX.**
Explanation: While attempting to connect to the CICS partition, 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 partition is down.

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 partition in BIM-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 partition.

RC=14 FDBK2=53 IN SENSE=00000000

- 1) The definition for the CICS partition in BIM-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.

- BIMVSHAR-914** THE CICS PARTITION XXXXXXXXX WAS NOT AVAILABLE DURING ACTIVATION.
Explanation: While making the connection to the CICS partition called XXXXXXXXX, the LU Network Manager received a sense code indicating that the partition was not available.
System Action: The LU Network Manager returns with a bad return code.
User Response: Make sure that the CICS partition is available and that the CICS ACB is open.
- BIMVSHAR-915** THE APPLID XXXXXXXXX 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. BIM-VSHARE allows for multiple APPLID definitions. Consult the Installation chapter for details.
- BIMVSHAR-916** THE APPLID XXXXXXXXX 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 86 for details on defining the APPLIDS.
- BIMVSHAR-917** VTAM FAILURE. RC=XX FDBK2=XX IN SENSE=XXXXYYYY OUT SENSE=XXXXYYYY.
Explanation: While sending a BIS to the CICS partition, 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.
- BIMVSHAR-918** THE CICS PARTITION DID NOT SEND A BIS.
Explanation: While waiting for a BIS to be sent from the CICS partition 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.
- BIMVSHAR-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.
- BIMVSHAR-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.
- BIMVSHAR-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 problem continues, contact technical support.
- BIMVSHAR-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.
User Response: Rerun the operation and determine why the session was ended by the operator.

BIMVSHAR-9998 **WARNING - PRODUCT AUTHORIZATION EXPIRES IN 30 DAYS OR LESS**
This message is displayed if the copy of BIM-VSHARE you are using is about to expire. Contact your BIM product sales representative for a new Expiration Password.

BIMVSHAR-9999 **PRODUCT AUTHORIZATION HAS EXPIRED**
BIM-VSHARE is protected with an Expiration Password. This message is displayed if that Password has expired. Contact your BIM product sales representative for a new Expiration Password.

BIMVSR-nnn (Routing Facility)

The following messages are generated by the VSAM Routing Facility used by several BIM software products.

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

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 BIM-VSHARE batch log or the CICS log to determine why the VTAM or CMS receive failed.

Module: BIMVSCOn

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 BIM-VSHARE batch log or the CICS log to determine why the VTAM or CMS send failed.

Module: BIMVSCOn

TC03

Explanation: A "signal" command was received by BIM-VSHARE on-line 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 BIM support.

Module: BIMVSCOn

TC04

Explanation: The on-line 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 BIM support.

Module: BIMVSCOn

TC05

Explanation: The TCC on-line 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 BIM support.

Module: BIMVSCOn

TC06

Explanation: During CMS processing, the on-line TCC transaction receive an error during a receive.

System Action: Abend task with abend TC06.

User Response: Refer to the BIM-VSHARE batch log or the CICS log to determine why the CMS receive failed. If the problem can not be isolated, contact technical support.

Module: BIMVSCOn

TC07

Explanation: The TCC on-line 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 BIM-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: BIMVSCOn

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 partition. Determine the cause for the violation. If a storage violation did not occur, contact BIM support.

Module: BIMVSCOn

TC09

Explanation: An error was encountered by the on-line TCC transaction after a CMS resume.
System Action: Abend task with abend TC09.
User Response: Refer to BIM-VSHARE batch log or the CICS log to determine why the VTAM or CMS resume failed.
Module: BIMVSCOn

TC0A

Explanation: After trapping an AICA in the TCC on-line 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 BIM-VSHARE having problems, contact BIM support.
Module: BIMVSCOn

TC10

Explanation: An error was encountered by the on-line 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: BIMVSCOn

TC61

Explanation: While trying to find a remotely defined dataset, a request was issued before anything was sent by the TCC on-line transaction.
System Action: Abend task with abend TC61.
User Response: The TCC on-line transaction abended or there was an error in the ISC or MRO link used remote file access. Refer to the CICS log to determine which occurred and react based on those error messages.
Module: BIMVSCOn

TC62

Explanation: The TCC remote file locator received two consecutive receives from the TCC on-line transaction. This is not part of the communications protocol.
System Action: Abend task with abend TC62.
User Response: Internal error, contact BIM support.
Module: BIMVSCOn