CICS VSAM Recovery

Implementation Guide

Version 3 Release 1
First edition (December 2001)

This edition applies to Version 3 Release 1 of CICS VSAM Recovery, Program Number 5655-H91, and to all subsequent releases and modifications until otherwise indicated in new editions. The changes for this edition are summarized under "Summary of changes" on page xxiii.

This edition replaces SH19-6971-01.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
RCF Processing, Department M86/050
5600 Cottle Road
San Jose, CA 95193-0001
United States of America

IBMLINK from US: STAR PUBS at SJEV M5
IBMLINK from Canada: STAR PUBS at TORIBM
IBM Mail Exchange: USIB3VVD at IBMMail
Internet: starpubs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
• Title and order number of this book
• Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1991, 2001. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Figures .......................... xi
Tables ........................... xiii

About this book ........................ xv
How you use this book ................. xv
Notes on terminology .................... xvi
Operating environment ................... xvi
  Software requirements for CICS VR ....... xvi
  Optional software ................... xvii
  Hardware requirements ............... xviii
  Optional hardware requirements ....... xviii
The CICSVR library ..................... xviii
Referenced publications ................. xix
Accessing messages using LookAt ....... xx
How to send your comments ............. xx

Summary of changes .................... xxiii
Summary of Changes for SH26-4127-00 CICSVR Version 3 Release 1
  Implementation Guide ............... xxiii
  New information ................... xxiii
Summary of changes for SH19-6971-01 CICSVR Version 2 Release 3
  Implementation Guide ............... xxiv
  New Information ................... xxiv

Part 1. Getting Started .................... 1

Chapter 1. Introducing CICSVR .......... 3
  How CICSVR can help you ............. 3
  Understanding CICSVR functions ..... 3
    General CICSVR functions ............ 3
    CICSVR functions for CICS TS ....... 4
    CICSVR functions for CICS V4 ....... 5
  CICSVR capabilities and limitations .. 6
    Logging ........................ 7
    Alternate indexes (AIX) ............. 7
    Backout ......................... 7
    VSAM data set attributes .......... 7
    DFSMSShsm ........................ 7
    DFSMSdss ........................ 7
  CICS transaction backout concepts ..... 8
    When online backout fails in CICS Version 4 .... 9
    Preventing logical inconsistencies in CICS V4 .... 11

Chapter 2. Migration considerations .... 13
  Reviewing CICS V4 to CICS TS migration considerations . 13
  Reviewing CICSVR V2R3 to CICSVR V3R1 migration considerations . 13
    Migrating your RCDS from CICSVR V2R3 to CICSVR V3R1 . 14

Chapter 3. Activating the CICSVR server address space ... 17
  Understanding the CICSVR server address space requirements . 17
  Setting up PARMLIB for the CICSVR server address space .... 18
  Initializing the CICSVR server address space .... 18

© Copyright IBM Corp. 1991, 2001
Defining a CICSVR prefix ........................................ 18
Preallocating the CICSVR message data sets and the RCDS ......... 19
Preallocating the DWWMSG data sets ................................ 19
Preallocating the RCDSs ............................................. 20
Identifying preallocation errors ..................................... 22
Preallocating other CICSVR data sets ................................ 22
Understanding the CICSVR dynamically created data sets .......... 23
Using the DISPLAY command to display the CICSVR server status .. 23
Displaying the CICSVR server status ................................ 23
Displaying the log stream status ..................................... 24
Using the VARY command to activate or terminate the CICSVR server 24
Activate the CICSVR server .......................................... 25
Terminate the CICSVR server ......................................... 25
Authorizing userids to issue DFSMShsm commands .................. 25

Chapter 4. Setting up CICSVR VSAM batch logging .................. 27
Using CICSVR VSAM batch logging .................................. 27
Enabling CICSVR VSAM batch logging ................................ 28
Using ISMF data class define panel ................................. 28
Using NaviQuest batch job to define a data class .................. 29
Using IDCAMS DEFINE CLUSTER ................................... 29
Using IDCAMS ALTER .................................................. 30
Using the AMP JCL DD parameter to enable VSAM Batch Logging 31
Determining if VSAM batch logging is enabled ....................... 31
Setting up the logging environment for CICSVR VSAM batch logging 32
Understanding CICSVR and VSAM batch logging .................... 32
Planning log streams for forward recovery logs ..................... 32
Authorizing access to MVS log streams ................................ 33
Setting up the logging environment .................................. 33
Detecting batch logging error conditions ............................. 34

Part 2. CICSVR and CICS TS .......................................... 35

Chapter 5. Setting up CICSVR for CICS TS ......................... 37
Setting up your CICSVR ISPF environment .......................... 37
Allocating data sets to your TSO session ............................ 37
Running the CICSVR ISPF dialog interface ........................ 38
Editing the CICSVR JCL skeleton ..................................... 40
Understanding the recovery control data set ......................... 41
Creating and maintaining the RCDS .................................. 42
Deciding which VSAM spheres to protect ............................ 46
Registering or deregistering your log of logs data sets ............... 47
Registering your log of logs data sets ................................ 47
Deregistering your log of logs data sets ................................ 47
Setting up your retention periods for your log stream copies and log stream blocks ......................... 48
Retention periods for log stream copies ................................ 48
Retention periods for log stream blocks ................................ 48
Setting up the log of logs scan utility (scan) ......................... 49
Improving forward recovery performance using the scan utility .... 49
Setting up the scan utility to run as a batch job ..................... 50
Understanding the log of logs scan utility report ..................... 50
Setting up CICSVR change accumulation ............................ 52
Step 1: Set up change accumulation batch jobs ..................... 52
Step 2: Use the change accumulation data sets in a forward recovery 59
Deciding if you want to use shadow copies ........................... 60
Preparing for shadow forward recovery ........................................ 61
Replacing the original sphere with a shadow copy ...................... 62

Chapter 6. Setting up your CICS TS environment ....................... 65
Understanding logging with CICS TS for z/OS and OS/390 ............ 65
CICS logging and journaling .................................................... 65
Understanding MVS log streams ................................................. 66
Defining system log streams .................................................... 66
Defining forward recovery log streams ..................................... 66
Defining the log of logs ......................................................... 67
Defining recovery attributes by CICS resource definition ............ 67
Establishing procedures to backup and recover your VSAM spheres .. 69
  Deciding how often to make backups .................................. 69
  Deciding when to make additional backups ............................ 69
  Deciding how to group your backups .................................. 69
  Scheduling backups to minimize your CICS TS down time .......... 70
Determining your backup and restore products ......................... 70
  Using DFSMSHsm as your backup utility ............................... 70
  Using DFSMSdss as your backup utility ............................... 71
Understanding nonrecoverable versus recoverable resources ......... 79
  Understanding the backup-while-open (BWO) facility .............. 80
Understanding VSAM RLS ..................................................... 85
Making the CICS system definition (CSD) data set recoverable .... 89
Deciding on your naming conventions ..................................... 89

Chapter 7. Using CICSVR with CICS TS at your disaster recovery site 91
Preparing your primary site ................................................... 91
  Making backups of your VSAM spheres ................................ 91
  Running your change accumulation batch jobs to update the RCDS 91
  Running scan to update the RCDS ..................................... 92
  Using the RCDS utility to export your RCDS to the remote site 92
  Sending your shadow copies to the remote site ..................... 93
  Using the log stream copy utility to copy your log streams ...... 93
Preparing your remote site ................................................... 98
  Installing CICSVR at your remote disaster recovery site ....... 98
  Saving the backups of your VSAM spheres ......................... 98
  Saving the log stream copies ....................................... 99
  Saving the SAM RCDS ................................................. 99
  Saving the shadow copies ......................................... 100

Part 3. CICSVR and CICS V4 ................................................. 101

Chapter 8. Setting up CICSVR for CICS V4 ............................... 103
Setting up your CICSVR ISPF environment ............................. 103
  Allocating data sets to your TSO session ............................ 103
  Running the CICSVR ISPF dialog interface ......................... 104
  Editing the CICSVR JCL skeleton .................................... 106
Understanding the recovery control data set (RCDS) .................. 107
  Creating and maintaining your RCDS ................................ 108
  Deciding which VSAM spheres to protect ............................ 112
Setting up your retention periods for your logs ....................... 113
  Retention periods for archived logs ................................ 113
Setting up automatic archiving with CICS V4 .......................... 114
  Setting up the journal archive job skeleton ....................... 114
  Setting up the journal archive job skeleton with shadow forward recovery 119
Setting up CICSVR change accumulation ............................... 121
Chapter 13. CICSVR commands used with CICS V4

Chapter 14. Using the CICSVR archive exits
Chapter 15. Implementing exits for forward recovery and backout . . . . 261
   What you can use the exits for ............................................. 261
      User processing after an I/O error .................................. 261
      Recovering an ESDS ...................................................... 261
      Logical recovery ........................................................ 262
      Changing the CICSVR return code ................................... 262
   How the exits work ......................................................... 263
      Calling the exits ......................................................... 263
      Exit programs ............................................................ 263
      Register conventions .................................................... 263
      Parameter lists .......................................................... 264
      Parameters ............................................................... 264
      Sample assemble and link-edit JCL .................................. 264
   Error exit—pass I/O error information .................................. 265
      When the exit is called ................................................ 265
      Parameter list ............................................................ 266
      Parameters passed to the exit program .............................. 267
      Parameters returned from the exit program ....................... 269
   ESDS delete exit—pass ESDS deletion records ....................... 269
      When the exit is called ................................................ 269
      Parameter list ............................................................ 269
      Parameters Passed to the Exit Program ............................ 270
      Parameters Returned from the Exit Program ...................... 270
   Preapply Exit—Pass Log Records and VSAM Records ............... 271
      When the Exit Is Called ............................................... 271
      Parameter List .......................................................... 271
      Parameters passed to the exit program ............................ 272
      Parameters returned from the exit program ....................... 273
   Termination exit—pass CICSVR return code ......................... 273
      When the exit is called ................................................ 273
      Parameter list ............................................................ 273
      Parameters passed to the exit program ............................ 274
      Parameters returned from the exit program ....................... 274

Appendix A. CICSVR V3R1 ddnames ................................. 277

Appendix B. Diagnosing change accumulation and DFSORT problems 279
   How to get DFSORT messages .......................................... 279
   How to eliminate other common errors ............................... 279
   How to get a DFSORT dump ............................................ 280
   Common system abends related to storage .......................... 280

Appendix C. Diagnosing logging problems .......................... 281
   Logging problem categories ............................................ 281
   MVS system logger availability ....................................... 282
   Diagnosis of MVS System Logger Problems ......................... 282
   Collecting diagnosis information .................................... 285
   Restarting the MVS system logger address space .................. 285

Appendix D. Sample CLIST (DWWCLIST) ............................ 287

Notices ............................................................................. 293
Programming Interface Information .................................. 294
Trademarks ...................................................................... 294

Glossary .......................................................................... 295
Figures

1. Transaction backout concepts ........................................... 9
2. Online backout failure in CICS Version 4 .................................. 10
3. CICS V4 CEMT INQUIRE DSNAME Screen ..................................... 11
4. Sample JCL to run the CICSVR V3R1 migration utility ......................... 14
5. Preallocate the DWWMSG data sets for the CICSVR server address space .......... 19
6. AMS Commands to Define the CICSVR RCDS .................................. 21
7. DWW180I message due to preallocation error or invalid prefix ................... 22
8. Preallocate the DWWDUMP and DWWDMSG data sets for the CICSVR server address space 23
9. DISPLAY SMS,CICSVR results when the CICSVR server is not active ........... 24
10. DISPLAY SMS,CICSVR,ALL results when the CICSVR server is active ........... 24
11. DISPLAY SMS,CICSVR,LOGSTREAMS(ALL) results when the CICSVR server is active .... 24
12. VARY SMS,CICSVR,ACTIVE results ........................................ 25
13. VARY SMS,CICSVR,TERMINATESERVER results ................................ 25
14. Data Class Define Panel ................................................... 28
15. NaviQuest sample batch job ................................................. 29
16. Enabling CICSVR VSAM batch logging using IDCAMS ALTER and FRLOG ........... 30
17. Enabling CICSVR VSAM batch logging using AMP JCL DD ...................... 31
18. Disabling CICSVR VSAM batch logging using AMP JCL DD ...................... 31
19. Data Class List Panel ....................................................... 31
20. Modifying the ISPF selection menu for CICSVR .................................. 39
21. Main Menu Panel .......................................................... 40
22. JCL skeleton secondary window .............................................. 41
23. AMS Commands to Define the CICSVR RCDS ................................ 43
24. Allocating the RCDS for the CICSVR Run .................................... 44
25. Allocating the RCDS to the CICSVR ISPF Dialog Interface ......................... 44
26. Increasing the size of the RCDS (steps 1 and 2) ................................ 45
27. Increasing the size of the RCDS (steps 4 and 5) ................................ 46
28. AMS command to delete and define DWWCON3 .................................. 46
29. Log of logs register secondary window ....................................... 47
30. Log of logs deregister secondary window ..................................... 48
31. CICSVR automatic log stream deregister secondary window ..................... 49
32. Sample scan job you can run at regularly scheduled times ....................... 50
33. Log of logs scan report .................................................... 51
34. Change accumulation batch job using DFSMShsm or DFSMSdss backup ........... 53
35. Change accumulation batch job with no DFSMShsm or DFSMSdss backup ........ 54
36. Change accumulation output, page 1 of 2 .................................... 56
37. Change accumulation output, page 2 of 2 .................................... 57
38. Sample ISPF dialog generated forward recovery job with APPLYCA ............. 60
39. Sample scan and shadow forward recovery job .................................. 62
40. DFSMSdss data set copy for use with CICSVR .................................. 71
41. Sample JCL to put date and time in the JCL and execute DFSMSdss DUMP .......... 72
42. Sample JCL to start REXX procedure, JCL1 (input/output files in the proc) ........ 73
43. Sample JCL to start REXX procedure, JCL2 (input/output files in the JCL) ........ 73
44. Sample REXX procedure JCL1 for dynamic allocation ............................. 74
45. Sample REXX procedure JCL2 for dynamic allocation ............................. 76
46. Job submitted by REXX procedure JCL1 and JCL2 ................................ 78
47. Sample JCL to start a task .................................................. 78
48. Sample JCL to define a GDG ................................................ 78
49. Sample JCL to use the GDG ................................................ 79
50. Sample RCDS EXPORT job to copy the RCDS for the remote recovery site ......... 92
51. Sample JCL to run the CICSVR log stream copy utility .......................... 94
52. Log stream copy report ..................................................... 95
53. Sample LOGSTREAMCOPY command with STARTTIME and STOPTIME ......... 96
### Tables

1. Summary of CICSVR functions, by CICS level .............................................. 13
2. ISPF dialog data sets ....................................................................................... 37
3. Layout of the first log copy record ............................................................... 96
4. Layout of last record ....................................................................................... 97
5. Layout of all other records ............................................................................ 97
6. Spanned format for records using a SAM file .............................................. 98
7. ISPF dialog data sets ....................................................................................... 103
8. Command Synonyms ...................................................................................... 163
9. Keyword and Keyword Value Synonyms ....................................................... 164
10. CICSVR program names ................................................................................ 165
11. CICSVR commands used with CICS TS ...................................................... 167
12. ALLOCATE synonyms ................................................................................... 169
13. BLDVRP synonyms ....................................................................................... 172
14. DEFEXIT Synonyms ...................................................................................... 176
15. DEFEXIT synonyms ...................................................................................... 178
16. LOGOFLOGS Synonyms ............................................................................... 180
17. LOGSTREAMCOPY Synonyms ...................................................................... 185
18. MVSLOG synonyms ....................................................................................... 187
19. RCDS Synonyms ........................................................................................... 188
20. RECOVER synonyms ..................................................................................... 198
21. CICSVR commands used with CICS V4 ...................................................... 205
22. ALLOCATE synonyms ................................................................................... 208
23. ARCHIVE Synonyms .................................................................................... 212
24. BACKOUT synonyms .................................................................................... 217
25. BLDVRP synonyms ....................................................................................... 219
26. DEFEXIT Synonyms ...................................................................................... 223
27. DEFEXIT synonyms ...................................................................................... 226
28. FCTCOMP synonyms ................................................................................... 229
29. RCDS Synonyms ........................................................................................... 231
30. RECOVER synonyms ..................................................................................... 242
31. RELATE synonyms ....................................................................................... 244
32. SEQCHKL synonyms .................................................................................... 247
33. SEQCHKR synonyms .................................................................................... 250
34. CICSVR V3R1 ddnames ............................................................................... 277
35. Getting a dump ............................................................................................. 280
36. Common system abends related to storage ................................................ 280
About this book

This book describes the tasks you need to perform to set up CICS VSAM Recovery Version 3 Release 1 (CICSVR), Program Number 5655-H91. This publication is for people who are responsible for the recovery of lost or damaged VSAM data.

To implement CICSVR, you should know how to:
- Change CICS resource definitions
- Use job control language (JCL)
- Create VSAM data sets
- Use an Interactive System Productivity Facility (ISPF) dialog interface
- Write assembler programs for CICSVR exits (optional)

How you use this book

Use this book for guidance on how to set up CICSVR when you use CICS TS or CICS V4. This book contains the following information:

**Part 1. Getting Started** on page 1:
- "Chapter 1. Introducing CICSVR" on page 3 introduces CICSVR and explains the CICSVR functions and CICS transaction backout concepts.
- "Chapter 2. Migration considerations" on page 13 describes the CICSVR related functions if you are migrating from CICS V4 to CICS TS or from CICSVR V2R3 to CICSVR V3R1.
- "Chapter 3. Activating the CICSVR server address space" on page 17 describes the tasks you need to do to set up the CICSVR server address space.
- "Chapter 4. Setting up CICSVR VSAM batch logging" on page 27 describes the tasks you need to do to set up CICSVR VSAM batch logging.

**Part 2. CICSVR and CICS TS** on page 35:
- "Chapter 5. Setting up CICSVR for CICS TS" on page 37 describes tasks you need to do to set up CICSVR in your CICS TS environment.
- "Chapter 6. Setting up your CICS TS environment" on page 63 describes the tasks you need to do to set up CICS TS for CICSVR recovery.
- "Chapter 7. Using CICSVR with CICS TS at your disaster recovery site" on page 91 describes the tasks you need to do to prepare your primary site and your remote site for CICSVR recovery.

**Part 3. CICSVR and CICS V4** on page 101:
- "Chapter 8. Setting up CICSVR for CICS V4" on page 103 describes tasks you need to do to set up CICSVR in your CICS V4 environment.
- "Chapter 9. Setting up your CICS V4 environment" on page 133 describes the tasks you need to do to set up CICS V4 for CICSVR recovery.
- "Chapter 10. Using CICSVR with CICS V4 at your disaster recovery site" on page 151 describes the tasks you need to do to prepare your primary site and your remote site for CICSVR recovery.

**Part 4. CICSVR Command Reference** on page 159:
- "Chapter 11. Command reference" on page 161 describes CICSVR command rules and how to read the syntax diagrams.
- "Chapter 12. CICSVR commands used with CICS TS" on page 163 describes all the CICSVR commands you can use with CICS TS.
• "Chapter 13. CICSVR commands used with CICS V4" on page 205 describes all the CICSVR commands you can use with CICS V4.

• "Part 5. CICSVR Exits" on page 255:
  • "Chapter 14. Using the CICSVR archive exits" on page 257 describes the CICSVR exit programs that you can use with the archive utility and the log stream copy utility.
  • "Chapter 15. Implementing exits for forward recovery and backout" on page 261 describes the CICSVR exit programs that you can use with forward recovery and backout.

• "Appendix A. CICSVR V3R1 ddnames" on page 277 shows the ddnames for CICSVR V3R1.

• "Appendix B. Diagnosing change accumulation and DFSORT problems" on page 279 describes how to get DFSORT™ messages and dumps as well as how to identify and eliminate other common errors.

• "Appendix C, Diagnosing logging problems" on page 281 describes what to do if you encounter problems with MVS™ system logger.

• "Appendix D, Sample CLIST (DWWCLIST)" on page 287 shows a sample command list that you can use to start the CICSVR ISPF dialog interface.

Notes on terminology

CICS Transaction Server is used when referring to the CICS element of CICS Transaction Server for OS/390® or CICS Transaction Server for z/OS™.

CICS V4 is used for Customer Information Control System/Enterprise System Architecture Version 4.

CICS is used when referring to all versions (CICS V4 and CICS Transaction Server).

The term log is used to describe any of these:
• MVS log streams
• CICSVR SAM copies of MVS log streams
• CICS V4 forward-recovery logs
• CICS system log

Operating environment

CICSVR uses logs to recover your VSAM data. CICSVR supports:
• MVS log streams
• CICSVR SAM copies of MVS log streams
• CICS V4 forward-recovery logs
• CICS system log

Software requirements for CICSVR

OS/390 V2R6 or above is required for installation. However, if OS/390 V2R6 to OS/390 V2R9 is installed the following features are not available:
• CICSVR VSAM batch logging
• Complete data set forward recovery automation (using the CICSVR ISPF dialog interface) using DFSMSdss™ copies and dumps.

z/OS Version 1 or higher or OS/390 V2R10, with the following APARs, is required to take advantage of the new CICSVR functions being introduced with this release:

• OW44714
• OW45559
• OW45560
• OW46161
• OW47025
• OW47574
• OW47781
• OW48258
• OW48701
• OW49369
• OW49478
• OW49529
• OW49530
• OW49899
• OW50340
• OW50341
• OW50433
• OW50538
• OW50683
• OW51100
• OW51135
• OW51544
• OW51555
• OW52194
• PQ42845

The following CICS TS and CICS V4 APARs are required to take advantage of the new CICSVR VSAM batch logging function:

• PQ50896
• PQ50900
• PQ50901

Optional software

This software or subsequent levels provides optional capabilities:
• For running backout, you need:
  – CICS V4 Version 4 Release 1 (5655-018)
• For automating backout and restore, you need:
  – Data Facility Hierarchical Storage Manager (DFHSM) Version 2 Release 6 (5665-329) or DFSMS/MVS® Version 1.4 (5695-DF1) and higher
• For VSAM RLS support, all of these MVS environments:
  – CICS Transaction Server for OS/390 Release 1 (5655-147) and higher
  – DFSMS/MVS Version 1 Release 4 (5695-DF1)
For more information about VSAM RLS, see "Understanding VSAM RLS" on page 85.

- For the backup-while-open facility (BWO) and concurrent copy support, you need:
  - CICS V4 Version 4 Release 1 (5655-018)
  - DFSMS/MVS Version 1 Release 6 (5695-DF1) or higher

For more information about the backup-while-open facility, see "Understanding the backup-while-open (BWO) facility" on page 80 or "Understanding the backup-while-open (BWO) facility" on page 145. For more information about concurrent copy, see "Using the concurrent copy function with CICSVR" on page 85 or "BWO and concurrent copy" on page 148.

## Hardware requirements

You need this hardware to run CICSVR:

- A processor supporting the software mentioned above
- A tape unit or cartridge unit, disk drive space. Refer to the CICSVR Version 3 Release 1 Program Directory for required DASD space needed to install CICSVR

## Optional hardware requirements

This hardware provides optional capabilities:

- For concurrent copy:
  - An IBM® 3990 Model 3 with the Extended Platform and Licensed Internal Code or an IBM 3990 Model 6
- For VSAM RLS support:
  - A coupling facility device or OS/390 image running coupling facility support is required
- CICSVR requires a minimum private area of 4 MB below the 16 MB line to execute.
- For CICSVR VSAM batch logging and the user of DFSMSdss backups via the dialog interface:
  - A coupling facility device, z/OS or OS/390 image running coupling facility support is required.

## The CICSVR library

IBM provides access to unlicensed CICSVR softcopy books on the Internet. To find CICSVR books on the Internet, if you are using z/OS, first go to the z/OS home page at:

http://www.ibm.com/servers/eserver/zseries/zos

If you are using OS/390, go to the OS/390 home page at:

http://www.ibm.com/servers/s390/os390/

From either of these Web sites, you can link directly to the CICSVR softcopy books by selecting the Library icon.

<table>
<thead>
<tr>
<th>Publication Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSVR V3R1 Implementation Guide</td>
<td>SH26-4126</td>
</tr>
<tr>
<td>CICSVR V3R1 User's Guide and Reference</td>
<td>SH26-4127</td>
</tr>
<tr>
<td>CICSVR V3R1 Messages and Problem Determination</td>
<td>SH26-4128</td>
</tr>
<tr>
<td>CICSVR V3R1 Program Directory</td>
<td>GI11-1232</td>
</tr>
</tbody>
</table>
You can also link to IBM Direct to order additional copies of the *CICSVR V3R1 Implementation Guide*.

These online CICSVR books are distributed on CD-ROM:

<table>
<thead>
<tr>
<th>Publication Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Software Products Collection,</td>
<td>SK3T-4270</td>
</tr>
<tr>
<td>OS/390 Collection,</td>
<td>SK2T-6700</td>
</tr>
<tr>
<td>OS/390 PDF Library Collection,</td>
<td>SK2T-6718</td>
</tr>
<tr>
<td>IBM Transaction Processing and Data Collection,</td>
<td>SK2T-0730</td>
</tr>
</tbody>
</table>

### Referenced publications

The following publications are referenced in this book:

<table>
<thead>
<tr>
<th>Publication Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS MVS Setting Up a Sysplex</td>
<td>SA22-7626</td>
</tr>
<tr>
<td>OS/390 MVS Setting Up a Sysplex</td>
<td>GC28-1779</td>
</tr>
<tr>
<td>z/OS DFSMSdss Storage Administration Guide</td>
<td>SC35-0421</td>
</tr>
<tr>
<td>OS/390 DFSMSdss Storage Administration Guide</td>
<td>SC35-0388</td>
</tr>
<tr>
<td>z/OS DFSMSdss Storage Administration Reference</td>
<td>SC35-0422</td>
</tr>
<tr>
<td>OS/390 DFSMSdss Storage Administration Reference</td>
<td>SC35-0389</td>
</tr>
<tr>
<td>z/OS DFSMSdss User Commands Reference Summary</td>
<td>SX35-5063</td>
</tr>
<tr>
<td>OS/390 DFSMSdss User Commands Reference Summary</td>
<td>SX35-5053</td>
</tr>
<tr>
<td>z/OS DFSMSdss Storage Administration Guide</td>
<td>SC35-0423</td>
</tr>
<tr>
<td>OS/390 DFSMSdss Storage Administration Guide</td>
<td>SC35-0393</td>
</tr>
<tr>
<td>z/OS DFSMSdss Storage Administration Reference</td>
<td>SC35-0424</td>
</tr>
<tr>
<td>OS/390 DFSMSdss Storage Administration Reference</td>
<td>SC35-0394</td>
</tr>
<tr>
<td>z/OS DFSMSshm Managing Your Own Data</td>
<td>SC35-0420</td>
</tr>
<tr>
<td>OS/390 DFSMSshm Managing Your Own Data</td>
<td>SC35-0387</td>
</tr>
<tr>
<td>z/OS MVS Programming: Assembler Services Guide</td>
<td>SA22-7606</td>
</tr>
<tr>
<td>OS/390 MVS Programming: Assembler Services Guide</td>
<td>GC28-1762</td>
</tr>
<tr>
<td>z/OS MVS Programming: Assembler Services Reference ABE-HSP</td>
<td>SA22-7606</td>
</tr>
<tr>
<td>OS/390 MVS Programming: Authorized Assembler Services Reference ALE-DYN</td>
<td>GC28-1764</td>
</tr>
<tr>
<td>CICS Recovery and Restart Guide</td>
<td>SC33-1698</td>
</tr>
</tbody>
</table>
Accessing messages using LookAt

LookAt is an online facility that allows you to look up explanations for z/OS messages and system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

You can use LookAt on the Internet at:


To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on News and Help or from the z/OS Collection, SK3T-4269.

To find a message explanation from a TSO command line, simply enter: lookat message-id as in the following example:

lookat iec192i

This results in direct access to the message explanation for message IEC192I.

To find a message explanation from the LookAt Web site, simply enter the message ID. You can select the release if needed.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in z/OS MVS Routing and Descriptor Codes. For such messages, LookAt prompts you to choose which book to open.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other CICSVR documentation:

- Send your comments by e-mail to:
– IBMLink™ from US: starpubs@us.ibm.com
– IBMLink from Canada: STARPUBS at TORIBM
– IBM Mail Exchange: USIB3VVD at IBMMAIL
– Internet: starpubs@us.ibm.com

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

• Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, RCF Processing Department M86/050, 5600 Cottle Road, San Jose, California 95193-0001, U.S.A.
Summary of changes

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You may notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only or procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

Summary of Changes for SH26-4127-00 CICSVR Version 3 Release 1 Implementation Guide

The following sections summarize the changes to that information.

New information

This book has been completely restructured and is now divided into five major parts as described in the How you use this book on page xv. The contents of some chapters have been reorganized with new sections that describes specific tasks you will need to run CICSVR. The following are new chapters:

- “Chapter 3. Activating the CICSVR server address space” on page 17
- “Chapter 4. Setting up CICSVR VSAM batch logging” on page 27
- “Chapter 5. Setting up CICSVR for CICS TS” on page 37
- “Chapter 6. Setting up your CICS TS environment” on page 63
- “Chapter 7. Using CICSVR with CICS TS at your disaster recovery site” on page 91
- “Chapter 8. Setting up CICSVR for CICS V4” on page 103
- “Chapter 9. Setting up your CICS V4 environment” on page 133
- “Chapter 10. Using CICSVR with CICS V4 at your disaster recovery site” on page 151
- “Chapter 11. Command reference” on page 161, “Chapter 12. CICSVR commands used with CICS TS” on page 167, and “Chapter 13. CICSVR commands used with CICS V4” on page 205. Many of the commands described in these chapters were previously presented in IBM CICS VSAM Recovery MVS/ESA Version 2 Release 3 User’s Guide and Reference
- “Appendix A. CICSVR V3R1 ddnames” on page 277
- “Appendix B. Diagnosing change accumulation and DFSORT problems” on page 279
- “Appendix C. Diagnosing logging problems” on page 281
- “Appendix D. Sample CLIST (DWWCLIST)” on page 287

The following programming support is now provided in CICSVR Version 3 Release 1:

CICSVR server address space

The CICSVR server address space provides services that support change accumulation and CICSVR VSAM batch logging.
CICSVR VSAM batch logging
CICSVR VSAM batch logging provides forward recovery capability for VSAM data sets that are updated by batch applications.

CICSVR change accumulation utility
Change accumulation consolidates forward recovery log records into a change accumulation data set. CICSVR uses the change accumulation data set in conjunction with the forward recovery log to speed up forward recovery processing.

DFSMShsm backup change notification
This support notifies CICSVR when a new data set backup has been created or an existing data set backup has been deleted by DFSMShsm.

DFSMSdss backup change notification
This support notifies CICSVR when a new data set backup has been created by DFSMSdss.

CICSVR log stream copy utility
The log stream copy utility makes a sequential access method (SAM) copy of an MVS log stream.

CICSVR RCDS export/import utility
The RCDS export/import utility creates an RCDS copy that can be sent (exported) to a remote site and used to update (import) the RCDS at the remote site. This helps you to keep your remote site in synchronization with your local site.

Summary of changes for SH19-6971-01 CICSVR Version 2 Release 3 Implementation Guide

The following are changes to this publication. Except for editorial changes, updates to this edition are marked with a vertical bar (|) to the left of the change.

New Information
Here are the major changes:

General
The content of some chapters has been rearranged. These are now referred to as 'logs':
• MVS log streams
• CICSVR QSAM copies of MVS log streams
• CICS V4 forward-recovery logs
• CICS/MVS journal-format logs
• CICS system logs

Where sample CICS commands are shown, the format of the command for CICS V4 Version 4 Release 1 is used. For the equivalent command format in CICS Transaction Server, CICS V4 Version 3, or for CICS/MVS, refer to CICS—Supplied Transactions in the appropriate CICS library. The CICSVR ISPF dialog interface now supports VSAM record level sharing (RLS). CICSVR can now recover variable relative-record data sets (VRRDS).

Preface
The software and hardware requirements for CICSVR have been updated.

Chapter 1
New terminology has been added. A description of backout failure in CICS Transaction Server has been added. The descriptions of
the CICSvr Complete recovery and Forward recovery only functions have been updated. The section describing CICSvr capabilities and limitations has been updated.

Chapter 2
The CICSvr ISPF dialog interface has been updated. Information about concurrent copy has been added. A description about VSAM record level sharing (RLS) has been included. The CICSvr ISPF dialog interface has been updated.

Chapter 3
A section about logging with CICS Transaction Server has been added. A section about defining files in CICS Transaction Server has been added. Information about the CICSvr migration utility has been added.

Chapter 4
A new chapter about CICSvr migration has been created.

Chapter 5
The description of the CICSvr archive utility has been updated. A new keyword has been added to the ARCHIVE command. The LOGOFLLOGS command has been added. The LOGSTREAMCOPY function has been added. The sample archive jobs have been updated. A sample log stream copy utility job has been included. New reports have been added.

Chapter 6
The old-style preapply exit has been removed. The new preapply exit has been added. For details about the old-style preapply exit, refer to an earlier edition of this book.

Bibliography
Book references have been updated and new books have been added.

Glossary
Entries have been added and updated.
# Part 1. Getting Started

## Chapter 1. Introducing CICSVR

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How CICSVR can help you</td>
<td>3</td>
</tr>
<tr>
<td>Understanding CICSVR functions</td>
<td>3</td>
</tr>
<tr>
<td>General CICSVR functions</td>
<td>3</td>
</tr>
<tr>
<td>What is the CICSVR server address space?</td>
<td>3</td>
</tr>
<tr>
<td>What is CICSVR VSAM batch logging?</td>
<td>4</td>
</tr>
<tr>
<td>CICSVR functions for CICS TS</td>
<td>4</td>
</tr>
<tr>
<td>What is CICSVR forward recovery for CICS TS?</td>
<td>4</td>
</tr>
<tr>
<td>What is the CICSVR log of logs scan utility for CICS TS?</td>
<td>4</td>
</tr>
<tr>
<td>What is the CICSVR log stream copy utility for CICS TS?</td>
<td>4</td>
</tr>
<tr>
<td>What is the CICSVR RCDS export/import utility for CICS TS?</td>
<td>5</td>
</tr>
<tr>
<td>What is the CICSVR change accumulation utility for CICS TS?</td>
<td>5</td>
</tr>
<tr>
<td>CICSVR functions for CICS V4</td>
<td>5</td>
</tr>
<tr>
<td>What is the CICSVR archive utility for CICS V4?</td>
<td>5</td>
</tr>
<tr>
<td>What is CICSVR forward recovery for CICS V4?</td>
<td>6</td>
</tr>
<tr>
<td>What is CICSVR backout for CICS V4?</td>
<td>6</td>
</tr>
<tr>
<td>What is the CICSVR RCDS export/import utility for CICS V4?</td>
<td>6</td>
</tr>
<tr>
<td>What is the CICSVR change accumulation utility for CICS V4?</td>
<td>6</td>
</tr>
<tr>
<td>CICSVR capabilities and limitations</td>
<td>6</td>
</tr>
<tr>
<td>Logging</td>
<td>7</td>
</tr>
<tr>
<td>Alternate indexes (AIX)</td>
<td>7</td>
</tr>
<tr>
<td>Backout</td>
<td>7</td>
</tr>
<tr>
<td>VSAM data set attributes</td>
<td>7</td>
</tr>
<tr>
<td>DFSMShsm</td>
<td>7</td>
</tr>
<tr>
<td>DFSMSdss</td>
<td>7</td>
</tr>
<tr>
<td>CICS transaction backout concepts</td>
<td>8</td>
</tr>
<tr>
<td>When online backout fails in CICS Version 4</td>
<td>9</td>
</tr>
<tr>
<td>DTB failure in CICS Version 4</td>
<td>10</td>
</tr>
<tr>
<td>Preventing logical inconsistencies in CICS V4</td>
<td>11</td>
</tr>
</tbody>
</table>

## Chapter 2. Migration considerations

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reviewing CICS V4 to CICS TS migration considerations</td>
<td>13</td>
</tr>
<tr>
<td>Reviewing CICSVR V2R3 to CICSVR V3R1 migration considerations</td>
<td>13</td>
</tr>
<tr>
<td>Migrating your RCDS from CICSVR V2R3 to CICSVR V3R1</td>
<td>14</td>
</tr>
</tbody>
</table>

## Chapter 3. Activating the CICSVR server address space

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the CICSVR server address space requirements</td>
<td>17</td>
</tr>
<tr>
<td>Setting up PARMLIB for the CICSVR server address space</td>
<td>18</td>
</tr>
<tr>
<td>Initializing the CICSVR server address space</td>
<td>18</td>
</tr>
<tr>
<td>Defining a CICSVR prefix</td>
<td>18</td>
</tr>
<tr>
<td>Preallocating the CICSVR message data sets and the RCDS</td>
<td>19</td>
</tr>
<tr>
<td>Preallocating the DWWMSG data sets</td>
<td>19</td>
</tr>
<tr>
<td>Preallocating the RCDSs</td>
<td>20</td>
</tr>
<tr>
<td>Creating and maintaining the RCDSs</td>
<td>20</td>
</tr>
<tr>
<td>Identifying preallocation errors</td>
<td>22</td>
</tr>
<tr>
<td>Preallocating other CICSVR data sets</td>
<td>22</td>
</tr>
<tr>
<td>Understanding the CICSVR dynamically created data sets</td>
<td>23</td>
</tr>
<tr>
<td>Using the DISPLAY command to display the CICSVR server status</td>
<td>23</td>
</tr>
<tr>
<td>Displaying the CICSVR server status</td>
<td>23</td>
</tr>
<tr>
<td>Displaying the log stream status</td>
<td>24</td>
</tr>
<tr>
<td>Using the VARY command to activate or terminate the CICSVR server</td>
<td>24</td>
</tr>
<tr>
<td>Activate the CICSVR server</td>
<td>25</td>
</tr>
<tr>
<td>Terminate the CICSVR server</td>
<td>25</td>
</tr>
</tbody>
</table>
Chapter 4. Setting up CICSVR VSAM batch logging

Using CICSVR VSAM batch logging

Enabling CICSVR VSAM batch logging
   Using ISMF data class define panel
   Using NaviQuest batch job to define a data class
   Using IDCAMS DEFINE CLUSTER
   Using IDCAMS ALTER
   Using the AMP JCL DD parameter to enable VSAM Batch Logging
   Determining if VSAM batch logging is enabled

Setting up the logging environment for CICSVR VSAM batch logging
   Understanding CICSVR and VSAM batch logging
   Planning log streams for forward recovery logs
   Authorizing access to MVS log streams
   Setting up the logging environment
      DASD-only logging
      Coupling facility logging
      Combination of coupling facility logging and DASD-only logging
   Detecting batch logging error conditions
Chapter 1. Introducing CICSVR

CICS VSAM Recovery Version 3 Release 1 (CICSVR) recovers your lost or damaged VSAM data. CICSVR is for organizations where the availability and integrity of VSAM data is vital.

This chapter introduces you to the following information:
- How CICSVR can help you
- Understanding CICSVR functions
  - General CICSVR functions
  - CICSVR functions for CICS TS
  - CICSVR functions for CICS V4
- Understanding CICSVR ddnames
- CICSVR capabilities and limitations
- CICS transaction backout concepts

How CICSVR can help you

CICSVR helps you recover your VSAM data sets in the following environments:
- CICSVR VSAM batch logging (when the VSAM data sets are not accessed in record level sharing–mode).
- CICS TS V1 or V2
- CICS V4

Your VSAM data sets can be corrupted in the following situations:
- Physical loss or damage of VSAM data could occur because:
  - A batch application alters or deletes the data incorrectly.
  - A disaster occurs and the data at the entire site is destroyed.
  - A device failure makes part of a disk inaccessible or causes damage to certain files.
- Loss of logical data integrity because a failure occurred during CICS transaction backout or during an emergency restart backout with CICS V4.

Understanding CICSVR functions

CICSVR offers many different functions. Some functions are specific to a particular environment such as: VSAM batch logging, CICS TS, or CICS V4. Other functions work in all three environments. This section describes which functions are available in each of the different environments.

General CICSVR functions

This section describes the functions that can be used with CICS TS, CICS V4, or without CICS:
- CICSVR server address space
- VSAM batch logging

What is the CICSVR server address space?
The CICSVR server address space is an essential part of CICSVR V3R1. It is started when MVS is IPLed. The CICSVR server address space must be active if you want to use any of the following CICSVR V3R1 functions:
• CICSVR VSAM batch logging
• Selection of DFSMSdss backups using the CICSVR dialog.
• Automated change accumulation processing with DFSMSdss logical dumps and logical copies.

The DISPLAY and VARY commands are available to display information about the CICSVR server address space and to alter the status or the SYS1.PARMLIB parameters of the CICSVR server address space.

What is CICSVR VSAM batch logging?
CICSVR VSAM batch logging provides logging for your batch changes to VSAM data sets that are not accessed in RLS-mode. This means that your CICS transactions and your batch jobs that use CICSVR VSAM batch logging cannot update the same VSAM data set at the same time. Once you have disabled the file from CICS (have CICS deallocate it), you can use CICSVR VSAM batch logging to log your application’s batch updates. If a problem occurs with your VSAM file, you can then use CICSVR to forward recover the updates recorded on the log by CICSVR VSAM batch logging.

CICSVR functions for CICS TS
This section describes the CICSVR functions that are available in the CICS TS environment:
• Forward recovery
• Log of logs scan utility
• Log stream copy utility
• Recovery control data set (RCDS) export/import utility
• Change accumulation (CA) utility

What is CICSVR forward recovery for CICS TS?
The CICSVR forward recovery function helps you recreate a VSAM sphere from a backup copy of the sphere. Using the panel interface, you can restore a DFSMSHsm or DFSMSdss backup, reapply all changes made to the VSAM sphere since the last backup, and return the VSAM sphere back to the exact state before the data was lost. The sphere can be an entry-sequenced data set (ESDS), a key-sequenced data set (KSDS), a fixed-length relative record data set (RRDS), or a variable RRDS (VRRDS). CICSVR obtains the information that it needs to construct the recovery job from the CICSVR RCDS.

What is the CICSVR log of logs scan utility for CICS TS?
You can use the log of logs scan utility (also referred to as scan) in your CICS TS environment. The utility scans all the log of logs that are registered in the RCDS, then updates and stores the details needed for recovery in the RCDS. Scan is run automatically when a forward recovery job is constructed using the ISPF panel interface. The log of logs scan utility should be set up as a batch job which is regularly submitted with a production planning system such as OPC/ESA.

What is the CICSVR log stream copy utility for CICS TS?
You can use the log stream copy utility to copy an MVS log stream to up to nine sequential access method (SAM) data sets. You cannot use IDCAMS or CICSVR archive to copy an MVS log stream; you must use the CICSVR log stream copy utility to copy a log stream.

You can perform a CICSVR forward recovery using a log stream copy. When the log stream copy is run, information about the copy (or copies) is stored in the RCDS.
What is the CICSVR RCDS export/import utility for CICS TS?
You can use the CICSVR RCDS export/import utility to extract the information in your RCDS so that you can send it to your remote recovery site and incorporate the information into its RCDS. This utility helps you to keep your remote site’s RCDS in synchronization with your local site’s RCDS.

What is the CICSVR change accumulation utility for CICS TS?
CICSVR change accumulation (CA) is the process of consolidating forward recovery log records into a change accumulation data set. CICSVR uses the change accumulation data set in conjunction with the forward recovery log to speed up forward recovery processing.

Change accumulation is a two step process:
1. You must first create CA batch jobs that define CA groups and identify the spheres that are in each group. When you run one of these CA batch jobs, CA reads the forward recovery log for a certain log range, selects the log records for the spheres in that CA group that are important for forward recovery, and calls DFSORT to sort these log records. CA saves the last update for each record using the sorted log records. Then CA stores the consolidated records in the CA data set. Every time you run the CA batch job, the change accumulation data set is updated with the information from the next log range. You must perform the following:
   a. Run the CA batch job after a backup is taken for any of the spheres in the CA group.
   b. Set up a CA batch job so that it is regularly submitted with a production planning system, such as OPC/ESA.

   CA runs in parallel with CICS production runs or VSAM batch logging; so keeping the CA data set as current as possible reduces the amount of log data that must be read and applied to forward recover a VSAM sphere.
2. The second step occurs when you want to forward recover a VSAM sphere. Use the ISPF panels to restore the backed up version of the sphere and generate the necessary JCL using information in the CICSVR RCDS. The APPLYCA keyword is generated on the RECOVER command. APPLYCA tells CICSVR to apply the records in the CA data set to the restored VSAM sphere, and then to apply the remaining log range from the forward recovery log to pick up the most recent changes.

CICSVR functions for CICS V4
This section describes the following CICSVR functions that are available in the CICS V4 environment:
- Archive utility
- Forward recovery
- Backout
- RCDS export/import utility
- CA utility

What is the CICSVR archive utility for CICS V4?
The archive utility is used when the CICS Journal Control Table (JCT) is set up to journal to two disk journals (JTYPE=DISK2). When a disk journal becomes full, CICS V4 will switch disks and will start journaling to the alternate disk. CICS V4 will automatically submit a CICSVR archive job that makes a copy of the full journal and stores information that is needed for recovery in the recovery control data set.
**What is CICSVR forward recovery for CICS V4?**
The CICSVR forward recovery function helps you recreate a VSAM sphere from a backup copy of the sphere. Using the panel interface, you can restore a DFSMShsm or DFSMsdss backup, reapply all changes made to the VSAM sphere since the last backup, and return the VSAM sphere back to the exact state before the data was lost. The sphere can be an entry-sequenced data set (ESDS), a key-sequenced data set (KSDS), a fixed-length relative record data set (RRDS), or a variable-length RRDS. CICSVR obtains the information it needs to construct the recovery job from the CICSVR RCDS.

**What is CICSVR backout for CICS V4?**
The CICSVR backout function removes partially completed transactions from one or more VSAM files when CICS V4 online backout has failed. It is needed if a failure occurs during CICS dynamic transaction backout (DTB) or during CICS transaction backout at emergency restart.

**What is the CICSVR RCDS export/import utility for CICS V4?**
You can use the CICSVR RCDS export/import utility to extract the information in your RCDS so that you can send it to your remote recovery site and incorporate the information into its RCDS. This utility helps you to keep your remote site in synchronization with your local site. If your RCDS is lost at your primary site, the reverse process can be done to restore the RCDS at your local site to the same level as the recovery site.

**What is the CICSVR change accumulation utility for CICS V4?**
CICSVR accumulation utility (CA) is the process of consolidating forward recovery log records into a CA data set. CICSVR uses the CA data set in conjunction with the forward recovery log to speed up forward recovery processing.

CA is a two step process:
1. You must first create CA batch jobs that define CA groups and identify the spheres that are in each group. When you run one of these CA batch jobs, CA reads the forward recovery log for a certain log range, selects the log records for the spheres in that CA group that are important for forward recovery, and then calls DFSORT to sort these log records. CA saves the last update for each record using the sorted log records. Then CA stores the consolidated records in the CA data set. Every time you run the CA batch job, the change accumulation data set is updated with the information from the next log range. You must set up the batch CA job so that it is regularly submitted with a production planning system such as OPC/ESA.

   You must set up the CA job as a batch job that is regularly submitted with a production planning system such as OPC/ESA. CA runs in parallel with CICS production runs or VSAM batch logging; so keeping the CA data set as current as possible reduces the amount of log data that must be read and applied to forward recover a VSAM sphere.

2. The second step occurs when you want to forward recover a VSAM sphere. You must use the ISPF panels to restore the backed up version of the sphere and generate the necessary JCL using information in the CICSVR RCDS. The APPLYCA keyword is generated on the RECOVER command. APPLYCA tells CICSVR to apply the records in the CA data set to the restored VSAM sphere, and then to apply the remaining log range from the forward recovery log to pick up the most recent changes.

---

**CICSVR capabilities and limitations**

This section describes the capabilities and limitations of CICSVR.
Logging

CICSVR:
- Uses the after-images on the logs to forward recover your VSAM spheres; it cannot forward recover a VSAM sphere that does not have an associated log containing after-images.
- Accepts logs on disk, tape, or cartridge; the CICSVR archive utility and the ISPF dialog interface cannot process uncataloged logs on tape or cartridge. If you run CICSVR manually, without using the ISPF dialog interface, forward recovery and backout can process uncataloged logs.
- Can recover base clusters with path updates; it cannot recover a base cluster with path updates if the base and path updates are recorded on separate logs. Record base and path updates on the same log.

Alternate indexes (AIX)

CICSVR:
- Removes reusable alternate indexes (AIX) from the upgrade set prior to recovery and rebuilds the reusable AIXs after recovery; it cannot remove and rebuild non-reusable AIXs. Define the AIXs in your upgrade sets as reusable.
- Can forward recover a VSAM sphere with nonunique AIXs.

Backout

CICSVR:
- Removes uncommitted changes on a KSDS, ESDS, or RRDS, as indicated on the log with CICS V4 logs; it cannot back out a VSAM sphere using logs from a CICS TS system.
- Can run backout using logs on disk, cataloged on tape, or cataloged on cartridge; it cannot process disk logs that reside on multiple volumes. Do not define your disk logs across multiple volumes.

VSAM data set attributes

CICSVR:
- Can detect whether the VSAM attributes of the target VSAM sphere in a recovery run differ from the attributes of the sphere as recorded on the log.

DFSMShsm

CICSVR:
- Can list all DFSMShsm backups for a VSAM sphere; it cannot list DFSMShsm dump information. For more information about DFSMShsm and CICSVR, see "Using DFSMShsm as your backup utility" on page 70 if you are using CICS TS or "Using DFSMShsm as your backup utility" on page 136 if you are using CICS V4.

DFSMSoS

CICSVR:
- Can list all DFSMSdss copies or dumps for a VSAM sphere. For more information about DFSMSdss and CICSVR, see "Using DFSMSdss as your backup utility" on page 74 if you are using CICS TS or "Using DFSMSdss as your backup utility" on page 137 if you are using CICS V4.
CICS transaction backout concepts

When a failure occurs while a transaction is active, the data that is associated with the transaction might be left in an inconsistent state. The purpose of CICS transaction backout is to back out updates made by transactions that did not finish successfully. It is needed, for example, when a task or CICS abends. Online CICS runs two transaction backout programs when required:

- **Dynamic transaction backout (DTB)**, used when a task abends
  If the resources that are affected by the abending task are recoverable, CICS performs DTB. DTB backs out the effects of a transaction that terminated abnormally. DTB restores the recoverable resources to the state they were at the beginning of the interrupted logical unit of work (LUW), that is, at the most recent sync point or start of task.
  DTB is similar in effect to the backout of in-flight tasks during an emergency restart following a CICS system failure. The most important differences are that DTB operates on a single abnormally terminating transaction, and that DTB performs the backout online (that is, while the rest of the CICS system continues to run normally). DTB thus provides immediate recovery of data integrity following a transaction failure.

- **Transaction backout during an emergency restart of CICS**, used if the CICS system has abended.
  Following an abnormal shutdown of CICS, an emergency restart returns recoverable resources to their committed states. CICS emergency restart:
    - Repositions the latest log
    - Reads the log backward
    - Copies log records for those LUWs that were processing when the abnormal termination of CICS occurred, to the restart data set

CICS backout processing uses the information on the restart data set to remove the effects of data set modifications made by in-flight transactions.

During a transaction backout, CICS performs:
- Initialization
- Recovery of resource definitions
- Backout processing

**Attention**

Do not change recoverable resources between the abnormal shutdown and emergency restart. To do so endangers successful processing of emergency restart backout.

CICSVR provides the backout function so that, in CICS Version 4, transaction backout can also occur in batch mode if either of the online backout programs have failed to back out some VSAM records. Figure 1 on page 9 illustrates transaction backout.
Every CICS task that affects a recoverable resource consists of one or more LUWs. The LUWs each start and end with a sync point. Transaction backout is needed when a task cannot complete but has made uncommitted updates (U1 and U2 in Figure 1). The transaction backout program backs out incomplete LUWs, including all updates, to the last sync point.

When online backout fails in CICS Version 4

Typically CICS invokes online backout when a transaction encounters an I/O error on a VSAM sphere and then abends. CICS will invoke DTB or emergency restart backout, depending on the situation. If the online backout program also encounters an I/O error, the program will fail. This results in possible uncommitted updates on the VSAM sphere.

Once online backout has failed to back out a sphere, CICS can no longer back it out. At this point, if you have CICS V4, you can use the CICSVR backout function. Figure 2 on page 10 shows the sequence of events when online backout fails in CICS V4.
DTB failure in CICS Version 4

In Figure 2, a transaction encounters an I/O error on the PAYROLL.BASE VSAM data set and abends. CICS V4 invokes DTB, but DTB fails, because it also encounters an I/O error. CICS backout-failure control, which is part of CICS V4, now takes over. It ensures that an application does not use the data set until the CICS console operator indicates that the problem is fixed. CICS V4 closes all files that point to PAYROLL.BASE and issues messages to the console informing the operator that DTB has failed. These messages give the name of the failed data set and the names of all files for that data set. CICS V4 sets the data set status to FAILED, indicating that backout has failed. CICS V4 will maintain the FAILED status until the CICS console operator issues this command (see Figure 3 on page 11).
Fig. 3. CICS V4 CEMT INQUIRE DSNAME Screen

Attempts to open the files for this data set will fail until the operator issues this command. During the time the data set is in FAILED status, responsible personnel must correct the I/O problem and run the CICSVR backout function.

Preventing logical inconsistencies in CICS V4

Often in a backout failure, the failing LUW updates more than one VSAM sphere. CICS V4 will change the data set status to FAILED for all data sets that fail backout. These data sets cannot be used until their status is changed from FAILED to NORMAL. This prevents CICS applications from accessing data sets that are logically inconsistent.

For example, assume that the data sets EXPENSES and PAYROLL were updated by a transaction that failed. If DTB succeeds for EXPENSES but fails for PAYROLL, CICS V4 will set PAYROLL status to FAILED, but the status of EXPENSES will remain NORMAL. Applications are prevented from accessing PAYROLL until the data set status is set to NORMAL.
Chapter 2. Migration considerations

This chapter describes how CICSVR functions relate to both CICS V4 and CICS TS to help you migrate from CICS V4 to CICS TS. It also explains how to migrate from CICSVR V2R3 to CICSVR V3R1.

Reviewing CICS V4 to CICS TS migration considerations

Table 1 describes all of the CICSVR V3R1 functions and identifies the CICSVR commands, dialog selections, or invoking programs that are used to carry out functions for both CICS V4 and CICS TS.

Table 1. Summary of CICSVR functions, by CICS level

<table>
<thead>
<tr>
<th>Function</th>
<th>CICS V4</th>
<th>CICS TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Recovery</td>
<td>Forward recovery only</td>
<td>Forward recovery only or complete recovery ¹</td>
</tr>
<tr>
<td>Backout</td>
<td>Backout only</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Forward Recovery and Backout</td>
<td>Complete recovery</td>
<td>Complete recovery ²</td>
</tr>
<tr>
<td>Copy a Forward Recovery log</td>
<td>ARCHIVE</td>
<td>LOGSTREAMCOPY</td>
</tr>
<tr>
<td>Update your RCDS</td>
<td>ARCHIVE</td>
<td>SCAN</td>
</tr>
<tr>
<td></td>
<td>CICSVR server address space</td>
<td>LOGSTREAMCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CICSVR server address space</td>
</tr>
<tr>
<td>Migrate your RCDS</td>
<td>MIGRATION utility ²</td>
<td>MIGRATION utility ²</td>
</tr>
<tr>
<td>Change accumulation</td>
<td>CA command and APPLYCA keyword</td>
<td>CA command and APPLYCA keyword</td>
</tr>
<tr>
<td>CICSVR VSAM batch logging</td>
<td>CICSVR VSAM batch logging</td>
<td>CICSVR VSAM batch logging</td>
</tr>
<tr>
<td>Send your RCDS to a remote site</td>
<td>RCDS export/import utility</td>
<td>RCDS export/import utility</td>
</tr>
</tbody>
</table>

Note:

¹ With CICS TS, backout is not supported so a complete recovery results in a forward recovery and no backout.

² You do not need to migrate your RCDS when migrating from CICS V4 to CICS TS.

Reviewing CICSVR V2R3 to CICSVR V3R1 migration considerations

When you’re ready to migrate from CICSVR V2R3 to CICSVR V3R1, plan your migration carefully. Ensure that you maintain your existing ability to forward recover your VSAM spheres during the migration process by performing the following tasks:

- Reduce the number of changes that occur simultaneously.
- Test all proposed changes before you put them into production.
- Be prepared to back out the changes if it fails.
When you migrate from CICSVR V2R3 to CICSVR V3R1, the following will run on
CICSVR V3R1 as they did for CICSVR V2R3:

- The batch log of logs scan jobs that update your RCDS and are set up to run at
  regularly scheduled intervals
- VSAM data sets
- DFSMShsm™ backups
- CICS TS log of logs or log streams
- CICSVR user exits.

In most cases, only migrate the RCDS from the CICSVR V2R3 level to the CICSVR
V3R1 level. If you created log stream copies using CICSVR V2R3 and you used
MOD to append the next copy onto an existing log stream copy, convert this copy
from the CICSVR V2R3 format (RECFM=U with BLKSIZE=6000) to the CICSVR
V3R1 format (RECFM=VB with BLKSIZE=6144). This conversion must be done
before you can run the CICSVR V3R1 log stream copy and MOD onto the copy.

**Migrating your RCDS from CICSVR V2R3 to CICSVR V3R1**

You can migrate your CICSVR V2R3 RCDS to CICSVR V3R1 by running the
CICSVR V3R1 migration utility. Migrating your RCDS converts your CICSVR V2R3
RCDS to the CICSVR V3R1 format. If you do not migrate your existing RCDS,
CICSVR cannot use the information in the CICSVR V2R3 RCDS.

As a precaution, save a copy of your CICSVR V2R3 RCDS in case you need to
return to it due to some kind of failure.

Modify the sample JCL in **Figure 4** and submit it to run the CICSVR V3R1 migration
utility.

```assembly
//DWWMIGA JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=OM
//COMMAND EXEC PGM=DWWMIW 1
//STEP1 J1 DD DSN=DWW.SDWWLOAD,DISP=SHR 2
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR 3
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR
//*
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR 4
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR
//*
//DWWMSG DD SYSOUT=* 5
```

**Figure 4. Sample JCL to run the CICSVR V3R1 migration utility**

A description of each of the numbered statements is:

1. The migration program is DWWMIW. The prefix DWW always refers to a
   component of CICSVR V3R1.
2. This is the name of the CICSVR load library.
3. DWWCON1–DWWCON3 defines the three new (CICSVR V3R1) recovery
   control data sets. If you plan to activate the CICSVR server address space
   you must use the **hlq.s1q** that was defined in the IGDSMSxx PARMLIB
   member for these data sets.
4 DWWOCON1–DWWOCON3 defines the three old (CICSVR 2.3) recovery control data sets.

5 DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a sysout data set. The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.
Chapter 3. Activating the CICSVR server address space

The CICSVR server address space provides a communication vehicle for CICSVR and other applications.

It must be active if you want to use any of the following CICSVR functions:
- CICSVR VSAM batch logging.
- Selection of DFSMSdss backups via the CICSVR dialog.
- Automated change accumulation processing with DFSMSdss logical dumps and logical copies.

**Note:** The CICSVR server address space is not required to use DFSMShsm; however, if the CICSVR server address space is enabled, all userids running CICSVR must be authorized to issue DFSMShsm-authorized commands. See [Authorizing userids to issue DFSMShsm commands](#) on page 26.

The following topics are described in this section:
- Understanding the CICSVR server address space requirements.
- Setting up PARMLIB for the CICSVR server address space.
- Preallocating the CICSVR message data sets and RCDSs.
- Preallocating other CICSVR data sets.
- Understanding the CICSVR dynamically created data sets.
- Using the DISPLAY command to display the CICSVR server status.
- Using the VARY command to activate or terminate the CICSVR server.
- Authorizing userids to issue DFSMShsm commands.

### Understanding the CICSVR server address space requirements

To enable the CICSVR server address space, the following requirements must be met:
- All systems must be running as a sysplex; local mode is not allowed.
- CICSVR RCDSs must be defined (one must be defined, but three are recommended). The RCDSs must be part of the sysplex.
- CICSVR message data sets must be defined. One DWWMSG data set must be defined on each system in the sysplex.
- The SMS address space must be defined with a minimum of a null configuration.
- If you want to use CICSVR VSAM batch logging:
  - The CICSVR server address space must be enabled. See [Chapter 4. Setting up CICSVR VSAM batch logging](#) on page 27 for information on how to enable CICSVR VSAM batch logging.
  - The VSAM sphere must be an SMS-managed data set.
  - The LOGR Couple Data Set (LOGR policy) must be defined. The LOGR policy includes the following:
    - Log stream definitions
    - Coupling facility list structure definitions, if applicable.
    - Data containing the current state of a log stream (for example, whether a log stream is currently connected to the coupling facility structure).
For more information on the formatting a LOGR couple data set and defining the LOGR policy, refer to z/OS MVS Setting Up a Sysplex section “Preparing to Use System Logger Applications”.

Setting up PARMLIB for the CICSVR server address space

To enable the CICSVR server address space you must update the SYS1.PARMLIB member, IGDSMSxx. This member must be updated on each system in the sysplex in which the CICSVR server address space is active.

The next two sections describe the IGDSMSxx updates.

Initializing the CICSVR server address space

Specify CICSVR_INIT(YES | NO) in the SYS1.PARMLIB IGDSMSxx member to initialize the CICSVR server address space. YES indicates that the CICSVR server address space be started during system initialization. NO indicates that the CICSVR server address space not be started during system initialization. The default value is NO. An IGDSMSxx member must be created for each system in the SMS complex, so you can initialize CICSVR on some systems and not others.

If an error occurs with the CICSVR server address space during system initialization (for example, cannot allocate the RCDSs), the CICSVR server will terminate and will not automatically restart. You must first correct the problem and then use the VARY SMS,CICSVR,ACTIVE command to start the CICSVR server. See “Activate the CICSVR server address space” on page 22 for more information on the CICSVR automatic restart mechanism.

Note: You can also issue the SETSMS CICSVR_INIT(YES | NO) operator command to enable or disable the initialization of the CICSVR server address space. This is not a multi-system command so you must issue ROUTE *ALL,SETSMS CICSVR_INIT(YES | NO) to route the SETSMS command to all the systems in the sysplex. This command will take effect immediately. Please note the CICSVR_INIT parameter only applies to the initialization of the CICSVR address space (during the IPL or by the VARY SMS,CICSVR,ACTIVE command). Therefore, if you issue the SETSMS CICSVR_INIT(NO) command while the CICSVR server is active, it will remain active. Once the CICSVR server is terminated, you will not be able to initialize the CICSVR server again until the CICSVR_INIT parameter has been set to YES.

Defining a CICSVR prefix

You must decide if you want to define a prefix for all the CICSVR-related data sets or use the default value. You can either define a prefix by specifying CICSVR_DSNAME_PREFIX(qlq.sq) in the SYS1.PARMLIB IGDSMSxx member or use the default value of DWW. qlq is the high level qualifier for the data set name and sq is the second level qualifier for the data set name.

Once you have decided on your own prefix or the default prefix of DWW, you must specify this prefix on the DWWMSG data set and the DWWCONn RCDS data sets. Preallocating the CICSVR message data sets and the RCDS data sets on page 19 and Understanding the CICSVR dynamically created data sets on page 23 discuss the data sets that use this prefix.

Note: You can also issue the SETSMS CICSVR_DSNAME_PREFIX(qlq.sq) operator command to change the prefix for any CICSVR dynamically created...
data sets. This is not a multi-system command so you must issue the ROUTE *ALL,SETSMS CICSVR_DSNAME_PREFIX(hlq.slq) to route the SETSMS command to all the systems in the sysplex. If you issue this SETSMS command, CICSVR will immediately use the new prefix value for any new dynamically created data sets; CICSVR will continue to use data sets that were already allocated using the old prefix until the CICSVR server address space is terminated. To avoid confusion, IBM recommends that if you change the prefix, you also terminate and reactivate the CICSVR server address space.

### Preallocating the CICSVR message data sets and the RCDS

You must preallocate the CICSVR message data sets (DWWMSG) and the RCDSs (DWWCON1, DWWCON2, and DWWCON3) prior to bringing up the CICSVR server address space. The following topics are discussed:

- Preallocating the DWWMSG data sets
- Preallocating the RCDSs
- Identifying preallocation errors

#### Preallocating the DWWMSG data sets

The CICSVR message data sets (DWWMSG) must be preallocated before you activate the CICSVR server address space. One DWWMSG data set must be defined on each system in the sysplex for use by the CICSVR server address space. The CICSVR server address space will write CICSVR messages to this data set.

You must specify the same hlq.slq as you did in CICSVR_DSNAME_PREFIX(hlq.slq) or use the default of DWW if you did not specify CICSVR_DSNAME_PREFIX(hlq.slq). You must also use the system name as the lowest level qualifier.

Figure 5 shows an example of how to preallocate a DWWMSG data set on a system for use by the CICSVR server address space.

```plaintext
//*******************************************************//
//* PREALLOCATE A DWWMSG DATA SET */
//*******************************************************//
EXEC PGM=IEFBR14
//DWWMSG DD DSN=hlq.slq.DWWMSG.systemname,DISP=(NEW,CATLG),
// SPACE=(CYL,(1,1)),
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=27930)
```

Figure 5. Preallocate the DWWMSG data sets for the CICSVR server address space

.hlq.slq is the high level qualifier and second level qualifier of the DWWMSG data set and systemname is the name of the system the DWWMSG data set is located. Replace the hlq.slq with the same values you defined in CICSVR_DSNAME_PREFIX(hlq.slq) in the SYS1.PARMLIB IGDSMSxx member or use the default of DWW if you did not define a prefix. Replace the systemname with the name of the system. The systemname is specified in the IEASYSxx PARMLIB member. Each system must have its own CICSVR message data set.
Attention: You also need to define a DWWMSG message data set in your batch jobs and when you allocate the data sets for the CICSVR dialog. IBM recommends:

- You use SYSOUT=* for your batch jobs
- You use a data set name that is different than the one you used for the CICSVR server address space when you allocate the data sets for the CICSVR dialog.

If you inadvertently use the same data set name for more than one DWWMSG data set, the information in the data set can be overwritten unexpectedly.

Preallocating the RCDSs

The RCDSs (DWWCON1, DWWCON2, DWWCON3) must be preallocated before you activate the CICSVR server address space. You must specify the same hlq.s1q as you did in CICSVR_DSNNAME_PREFIX(hlq.s1q) or use the default of DWW if you did not specify CICSVR_DSNNAME_PREFIX(hlq.s1q).

Creating and maintaining the RCDSs

The three RCDSs are linear VSAM data sets. They contain identical copies of essential information CICSVR needs to recover your VSAM spheres. To reduce the possibility of losing all three copies of the RCDSs, IBM recommends that you create each RCDS on separate volumes, disk controllers, and channels. In addition, your VSAM spheres should be on different volumes, disk controllers and channels from your RCDS. These precautions prevent a single hardware failure from making all three RCDSs unusable, or from making the RCDS and the VSAM sphere to be recovered, unusable. CICSVR regularly checks the consistency of data between the three data sets. If one or two of the RCDSs are temporarily lost (for example, the unavailability of a disk path) CICSVR automatically copies the data from the remaining RCDSs the next time the data set becomes available.

The RCDSs should be created and cataloged in a globally shared catalog. All systems in the sysplex need to have write access to all the RCDSs. You must have Global Resource Serialization (GRS) installed to access the RCDS from more than one MVS system.

Figure 6 on page 21 shows sample AMS commands to define the three RCDSs.
The space values of (200 30), (200 30), and (150 30) are a good starting point if you plan to use CICSVR VSAM batch logging. If you do not plan to use CICSVR VSAM batch logging, see "Creating and maintaining the RCDS" on page 42 if you use CICS TS or "Creating and maintaining your RCDS" on page 108 if you use CICS V4 to help you estimate the size of your RCDSs.

Notice that one of the space allocation values is different than the other two. When the smallest data set becomes full, CICSVR issues DWW1605I DWWCONn is full. This is an informational message to warn you that you need to increase the size of your RCDSs before they all become full. If all RCDSs become full, CICSVR will no longer run.

If one or more of the RCDSs become full or unusable, perform the following steps to repair the RCDS:

- Deallocate the unusable RCDS
- Repair the RCDS
  - RCDS full situation
  - RCDS damaged situation
- Activate the CICSVR server address space on all the systems in the sysplex.

The following sections describe each of the above steps.

**Deallocate the unusable RCDS:** To deallocate the unusable RCDS, you must terminate the CICSVR server on all the systems in the sysplex. Use the ROUTE "ALL,V SMS,CICSVR,TERMINATESERVER command to accomplish this. See "Terminate the CICSVR server" on page 23 for more information on this command.

**Note:** IBM intends to change the method of performing RCDS maintenance in the future.
**Repair the RCDS:** You may need to repair the RCDS because one or more of them has become full or become damaged. See [Maintaining the RCDS on page 44](#) if you use CICS TS or [Maintaining your RCDS on page 110](#) if you use CICS V4.

**Activate the CICSVR server address space:** Activate the CICSVR server address space on all the systems in the sysplex using the ROUTE *ALL,V SMS,CICSVR,ACTIVE* command. Since the CICSVR server address space has already been terminated, this command will take effect immediately. CICSVR will reallocate the new RCDs and repopulate them with the information from one of the existing RCDs automatically.

**Identifying preallocation errors**

If the DWWMSG or DWWCONn data sets were not preallocated correctly, the CICSVR server address space issues the following message:

```
DWW180E UNEXPECTED ERROR DURING CICSVR SERVER PROCESSING.
MODULE WHICH DETECTED THE ERROR: DWWIRCM
RETURN CODE (HEX): 00000024
REASON CODE (HEX): 711152E0
CALLED MODULE ERROR INFORMATION:
RETURN CODE (HEX): 00000024
REASON CODE (HEX): 71122412
```

*Figure 7. DWW180I message due to preallocation error or invalid prefix*

**Preallocating other CICSVR data sets**

You must preallocate the following data sets so dump and diagnostic information are available if you encounter a problem and need to contact your local IBM Support Center:

- **DWWDUMP.** This data set contains tracing and diagnostic information that is produced by individual CICSVR subroutines as requested by CICSVR.
- **DWWDMSG.** This data set contains the trace table of prologs and epilogs of the latest called modules.

For both of these data sets, you must specify the same hlq.sq1 as you did in CICSVR_DSNNAME_PREFIX(hlq.sq1) in the SYS1.PARMLIB IGDSMSxx member or use the default of DWW if you did not specify CICSVR_DSNNAME_PREFIX(hlq.sq1).

You must also use the system name as the lowest level qualifier.

*Figure 8 on page 23* shows an example of how to preallocate the DWWDUMP and DWWDMSG data sets on a system for use by the CICSVR server address space.
Attention: You also need to define the DWWDUMP, and DWWDMMSG data sets in the CICSVR job skeleton and when you allocate these data sets for the CICSVR dialog. IBM recommends:

- You use SYSOUT=* for your batch jobs
- You use a data set name that is different than the one you used for the CICSVR server address space when you allocate the data sets in the CICSVR job skeleton and for the CICSVR dialog. If you allocate the same data set name to the CICSVR dialog that is used with the CICSVR server for the DWWMSG, DWWDMMSG, and DWWDUMP ddnames, you could inadvertently overwrite data.

**Understanding the CICSVR dynamically created data sets**

CICSVR dynamically creates and deletes change accumulation output data sets when the CICSVR change accumulation batch jobs are run. CICSVR keeps track of the CA output data sets and uses them when performing a forward recovery if the RECOVER command APPLYCA keyword is specified.

**Using the DISPLAY command to display the CICSVR server status**

You can use the DISPLAY command to display the status of the CICSVR server on a single system or for all the systems in the sysplex. The following variations of the DISPLAY command are available for the CICSVR server address space:

- DISPLAY SMS,CICSVR
- DISPLAY SMS,CICSVR,ALL
- DISPLAY SMS,CICSVR,LOGSTREAMS(logstreamname)
- DISPLAY SMS,CICSVR,LOGSTREAMS(ALL)

**Displaying the CICSVR server status**

Use DISPLAY SMS,CICSVR to display the status of the CICSVR server on the system the command was issued.

If DISPLAY SMS,CICSVR is issued and the CICSVR server is not active, the following messages are displayed on the operator console:

```plaintext
//*******************************************************//
//* PREALLOCATE THE DWWDUMP AND DWWDMSG DATA SET *//*
//*******************************************************//
//* DWDUMP EXEC PGM=IEFBR14 *
//DWDUMP DD DSN=hlq.s1q.DWWDUMP.systemname,DISP=(NEW,CATLG), // SPACE=(CYL,(10,10)), // DCB=(RECFM=VBA,LRECL=84,BLKSIZE=3120)
//DWWDMSG DD DSN=hlq.s1q.DWWDMMSG.systemname,DISP=(NEW,CATLG), // SPACE=(CYL,(1,1)), // DCB=(RECFM=VBA,LRECL=84,BLKSIZE=3120)
//*
```

Figure 8. Preallocate the DWWDUMP and DWWDMSG data sets for the CICSVR server address space
If DISPLAY SMS,CICSVR,ALL is issued, data for all the systems connected to the sysplex is displayed on the operator console:

```
DWW020I DISPLAY SMS,CICSVR
DISPLAY SMS,CICSVR - SERVER STATUS
SYSNAME: SYSTEM1 UNAVAILABLE ASID: 0037 STEP: AS_Init_Started

DISPLAY SMS,CICSVR - JOB STATUS
NUMBER OF JOBS USING BATCH LOGGING:
SYSNAME: SYSTEM1 0

DATA SET NAMING CONVENTION IN USE:
SYSNAME: SYSTEM1 DWW
```

Figure 9. DISPLAY SMS,CICSVR results when the CICSVR server is not active

If DISPLAY SMS,CICSVR,ALL is issued, data for all the systems connected to the sysplex is displayed on the operator console:

```
DWW020I DISPLAY SMS,CICSVR
DISPLAY SMS,CICSVR - SERVER STATUS
SYSNAME: SYSTEM2 AVAILABLE ASID: 0032 STEP: CICSVR_Init_Complete
SYSNAME: SYSTEM1 AVAILABLE ASID: 0033 STEP: CICSVR_Init_Complete

DISPLAY SMS,CICSVR - JOB STATUS
NUMBER OF JOBS USING BATCH LOGGING:
SYSNAME: SYSTEM2 4
SYSNAME: SYSTEM1 4

DATA SET NAMING CONVENTION IN USE:
SYSNAME: SYSTEM2 DWW
SYSNAME: SYSTEM1 DWW
```

Figure 10. DISPLAY SMS,CICSVR,ALL results when the CICSVR server is active

### Displaying the log stream status

Use DISPLAY SMS,CICSVR,LOGSTREAMS(logstreamname) to display the status of the specified logstream on the system the command was issued. Alternatively, you can specify DISPLAY SMS,CICSVR,LOGSTREAMS(ALL) to display the status of all the log streams that are known to CICSVR.

```
DWW020I DISPLAY SMS,CICSVR
DISPLAY SMS,CICSVR - LOG STREAM CONNECTION STATUS
SYSNAME 0000000001111111111222222222333
IDENTIFIER 12345678901234567890123456789012
CICSVS.TTCICS2.FILELOG1 SS..............................
CICSVS.TTCICS2.FILELOG2 SS..............................
01 SYSNAME: SYSTEM1
02 SYSNAME: SYSTEM2
```

Figure 11. DISPLAY SMS,CICSVR,LOGSTREAMS(ALL) results when the CICSVR server is active

### Using the VARY command to activate or terminate the CICSVR server

You can use the VARY command to activate or terminate the CICSVR server on a single system. The following VARY commands are available for the CICSVR server address space:

- VARY SMS,CICSVR,ACTIVE
- VARY SMS,CICSVR,TERMINATESERVER
**Note:** The VARY commands listed above are not multi-system commands so you must issue “ROUTE “ALL,” with the VARY commands to route them to all the systems in the sysplex. These commands take effect immediately.

### Activate the CICSVR server

If the CICSVR server fails, it will automatically restart (up to a maximum of six times). If this limit is exceeded, the automatic restart mechanism is disabled and CICSVR terminates. When you have resolved the problem, you can restart the CICSVR server by using the VARY SMS,CICSVR,ACTIVE command. When the CICSVR server is restarted, the following message is displayed on the operator console:

```
DWW014I  CICSVR SERVER ADDRESS SPACE IS NOW ACTIVE.
```

*Figure 12. VARY SMS,CICSVR,ACTIVE results*

VARY SMS,CICSVR,ACTIVE is also needed to restart the CICSVR server after the VARY SMS,CICSVR,TERMINATESERVER is issued. Also, the CICSVR_INIT parameter in the SYS1.PARMLIB IGDSMSxx member must be set to YES for successful completion of the VARY SMS,CICSVR,ACTIVE command.

**Note:** If a failure occurs during the initialization of the CICSVR server address space, automatic restart is not performed.

### Terminate the CICSVR server

If you need to terminate the CICSVR server address space (for example, to perform maintenance on the RCDS), use the VARY SMS,CICSVR,TERMINATESERVER command. When the CICSVR server is terminated, the following messages are displayed on the operator console:

```
DWW172I  REQUEST TO TERMINATE THE CICSVR ADDRESS SPACE IS ACCEPTED:
CICSVR SERVER TERMINATION SCHEDULED.
DWW008I  CICSVR SERVER SUCCESSFULLY TERMINATED AT END OF MEMORY
```

*Figure 13. VARY SMS,CICSVR,TERMINATESERVER results*

### Authorizing userids to issue DFSMShsm commands

If you want to use DFSMShsm as your backup utility and the CICSVR server address space is enabled, you must request that your userid be authorized to issue DFSMShsm-authorized commands. Ask your system programmer to add the following AUTH command to the ARCCMDxx member: :

```
AUTH userid DATABASEAUTHORITY(USER)
```

where *userid* is the userid of the person running CICSVR. USER authority is needed for CICSVR change accumulation processing for data sets that have been backed up by DFSMShsm.

**Note:** The AUTH command only needs to be issued if the CICSVR server address space is enabled.

For more information on the DFSMShsm AUTH command, refer to [Z/OS DFSMShsm User Commands Reference Summary](https://www.ibm.com/support/knowledgecenter/SSSUGG/dfsmsdss/dfsmsdss_chap03.html).
Chapter 4. Setting up CICSVR VSAM batch logging

This chapter describes the tasks you need to do to set up CICSVR VSAM batch logging (hereafter referred to as batch logging). Batch logging provides forward recovery capability for VSAM data sets that are updated by batch applications. VSAM data set updates made by batch applications are logged to an MVS, OS/390, or z/OS log stream. Batch logging can log changes to VSAM data sets that are not currently in-use. Then, you can use CICSVR to forward recover the updates made by the batch applications.

CICSVR VSAM batch logging requires the following:

- OS/390 V2R10 with the necessary PTFs applied. See Software requirements for CICSVR on page xvi for details.
- CICSVR server address space must be activated. See Chapter 3. Activating the CICSVR server address space on page 17 for details.
- All VSAM data sets processed with batch logging must be SMS-managed data sets.

CICSVR VSAM batch logging calls the MVS system logger to write the log records to the MVS log stream. The MVS system logger is a component of MVS/ESA, OS/390, and z/OS, which provides a programming interface to access records on a log stream. For detailed information on the MVS system logger, refer to the following publications:

- z/OS MVS Programming: Assembler Services Guide
- z/OS MVS Programming: Assembler Services Reference ABE-HSP
- z/OS MVS Setting Up a Sysplex

The following topics are described in this section:

- Using CICSVR VSAM batch logging
- Enabling CICSVR VSAM batch logging
- Setting up the logging environment for CICSVR VSAM batch logging
- Detecting batch logging error conditions

Using CICSVR VSAM batch logging

You can use batch logging to log changes to VSAM data sets that are not accessed in record level sharing mode (RLS mode) when the VSAM data set is not in-use by CICS. CICSVR VSAM batch logging can only be used with non-RLS access, that is, if MACRF is set to NSR (non-shared resources), LSR (local shared resources) or GSR (global shared resources), and if MACRF is not set to RLS.

Note: RLS is a mode of access to a VSAM sphere that is interpreted at OPEN time; RLS is not an attribute of a sphere. RLS is selected by specifying the RLS JCL parameter or by specifying MACRF=RLS in the ACB.

Use CICSVR VSAM batch logging after you have brought down CICS or have issued a command to deallocate the files your batch jobs need to update. For example, you can use the following CICS command to disable (dealocate) PAYROLL so your batch jobs can update that file and CICSVR VSAM batch logging can log the changes:

CEMT SET FILE(PAYROLL) CLOSED DISABLED
The "Enabling CICSVR VSAM batch logging" section describes the different ways you can enable CICSVR VSAM batch logging.

Enabling CICSVR VSAM batch logging

CICSVR VSAM batch logging provides logging for non-RLS data sets. The VSAM data sets should be opened with the ACB setting MACRF=NSR or LSR or GSR and by not specifying the RLS JCL parameter.

There are five different ways that you can enable CICSVR VSAM batch logging:
- Using ISMF Data Class Define panel
- Using NaviQuest batch job to define a data class
- Using IDCAMS DEFINE CLUSTER
- Using IDCAMS ALTER
- Using the AMP JCL DD parameter

Using ISMF data class define panel

You can use the Interactive Storage Management Facility (ISMF) dialog interface to enable VSAM batch logging. On the DATA CLASS DEFINE panel, enter ‘R’ for REDO, in the FRlog input field and press enter. ‘R’ enables batch VSAM forward recovery logging to be performed. See Figure 14 for the DATA CLASS DEFINE panel.

If you specify REDO, you must specify a Logstream Id.

All VSAM data sets processed with batch logging must be SMS-managed data sets. FRLOG cannot be used with LINEAR, KEYRANGE, or temporary data sets.
Using NaviQuest batch job to define a data class

You can use NaviQuest to define a data class. See Figure 15 for a section of the NaviQuest sample batch job, ACBJBAD1. Set the FRLOG parameter to FRLOG(R) to enable CICSVR VSAM Batch Logging. If you set FRLOG(R), you must also specify a log stream ID in the LOGSTID parameter.

```
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSDUMP DD SYSOUT=* 
//SYSTSIN DD * 
PROFILE PREFIX(IBMUSER) 
ISPSTART CMD(ACBJBAD1 + 
DEFINE/ALTER + 
SCDS(TEST.CDS) + 
DCNAME() + 
DESCR() + 
RECURG() + 
. 
. 
REUSE() + 
INILOAD() + 
SPANONSP() + 
BWO() + 
LOG() + 
LOGSTID(CICSMVS.CICSVR.LGSTRM) + 
FRLOG(R) + 
SPCCONRL() + 
REDSPCUT() + 
) + 
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAx(999999) 
/*
```

Figure 15. NaviQuest sample batch job

All VSAM data sets processed with batch logging must be SMS-managed data sets. FRLOG cannot be used with LINEAR, KEYRANGE, or temporary data sets.

Refer to DFSMS/MVS NaviQuest User's Guide for more information on NaviQuest.

Using IDCAMS DEFINE CLUSTER

You can use the FRLOG(REDO) keyword of IDCAMS DEFINE to enable VSAM batch logging. If you specify REDO, you must specify LOGSTREAMID.

DEFINE CLUSTER

FRLOG(NONE | REDO)

Establishes whether CICSVR VSAM batch logging will be performed for your VSAM data sets. FRLOG is an optional keyword. If FRLOG is not specified or FRLOG(NONE) is specified, CICSVR VSAM batch logging will not occur.

NONE

Specifies that CICSVR VSAM batch logging will not be performed for your VSAM data sets. CICSVR will not write any forward recovery log records to the log specified by the LOGSTREAMID parameter.

REDO

Specifies that CICSVR VSAM batch logging will be performed for your VSAM data sets. CICSVR will write forward recovery log records to the log specified by the LOGSTREAMID parameter.
If you specify FRLOG(REDO), you must also specify the LOGSTREAMID parameter, unless it is already defined. LOGSTREAMID specifies the MVS log stream name to be used for forward recovery log records.

All VSAM data sets processed with batch logging must be SMS-managed data sets. FRLOG cannot be used with LINEAR, KEYRANGE, or temporary data sets.

### Using IDCAMS ALTER

You can use the FRLOG(REDO) keyword of IDCAMS ALTER to enable CICSVR VSAM batch logging. If you specify REDO, you must specify LOGSTREAMID.

**FRLOG(NONE | REDO)**

Establishes whether CICSVR VSAM batch logging will be performed for your VSAM data sets. FRLOG is an optional keyword. If FRLOG is **not** specified or FRLOG(NONE) is specified, CICSVR VSAM batch logging will not occur.

**NONE**

Specifies that CICSVR VSAM batch logging will not be performed for your VSAM data sets. CICSVR will not write any forward recovery log records to the log specified by the LOGSTREAMID parameter.

Use FRLOG(NONE) to disable CICSVR VSAM batch logging if the data set has been defined with FRLOG(REDO).

**REDO**

Specifies that CICSVR VSAM batch logging will be performed for your VSAM data sets. CICSVR will write forward recovery log records to the log specified by the LOGSTREAMID parameter.

If you specify FRLOG(REDO), you must also specify the LOGSTREAMID parameter, unless it is already defined. LOGSTREAMID specifies the MVS log stream name to be used for forward recovery log records.

All VSAM data sets processed with batch logging must be SMS-managed data sets. FRLOG cannot be used with LINEAR, KEYRANGE, or temporary data sets.

**Figure 16** shows how you can enable CICSVR VSAM batch logging.

```plaintext
//ALTER JOB ...
//STEP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
ALTER USER1.BASE1A FRLOG(REDO) -
LOGSTREAMID(CICSMVS.BASE1A)
/*
```

**Figure 16. Enabling CICSVR VSAM batch logging using IDCAMS ALTER and FRLOG**

FRLOG(REDO) enables the forward recovery logging capability for batch applications. Changes made to VSAM data sets will be logged to the log stream specified by the LOGSTREAMID.
Using the AMP JCL DD parameter to enable VSAM Batch Logging

You can use the FRLOG subparameter of the AMP JCL DD to enable or disable VSAM batch logging, instead of using IDCAMS ALTER. There is no default JCL value for FRLOG. If you specify REDO, you must specify LOGSTREAMID unless it is already defined.

Figure 17 shows how you can enable CICSVR VSAM batch logging using the AMP JCL DD.

```
//DS1 DD DSNAME=USER1.BASE1A,AMP=('FRLOG=REDO'),DISP=SHR
```

Figure 17. Enabling CICSVR VSAM batch logging using AMP JCL DD

Figure 18 shows how you can disable CICSVR VSAM batch logging using the AMP JCL DD.

```
//DS1 DD DSNAME=USER1.BASE1A,AMP=('FRLOG=NONE'),DISP=SHR
```

Figure 18. Disabling CICSVR VSAM batch logging using AMP JCL DD

Determining if VSAM batch logging is enabled

You can see if batch logging is enabled by using the ISMF DATA CLASS LIST panel. See Figure 19 for the DATA CLASS LIST panel.

```
Panel List Utilities Scroll Help
```

```
DGTLGP21 DATA CLASS LIST
Command === Scroll ===
Entries 1-9 of 9
Data Columns 37-40
CDS Name : Y421252.MYSCDS

Enter Line Operators below:

<table>
<thead>
<tr>
<th>LINE OPERATOR</th>
<th>DATACLASS NAME</th>
<th>SPC CONS</th>
<th>REDUCE</th>
<th>REC ACC</th>
<th>FRLOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) DC1</td>
<td>NO</td>
<td>--</td>
<td>-------</td>
<td>NONE</td>
<td>------</td>
</tr>
<tr>
<td>(2) DC1</td>
<td>NO</td>
<td>-- USER</td>
<td>REDO</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(3) DC3</td>
<td>NO</td>
<td>--</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(4) DC66</td>
<td>NO</td>
<td>-- USER</td>
<td>------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(5) DC777</td>
<td>NO</td>
<td>-- USER</td>
<td>------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(6) DDC1</td>
<td>NO</td>
<td>--</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(7) DDC2</td>
<td>NO</td>
<td>--</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(8) TDC1</td>
<td>NO</td>
<td>--</td>
<td>-------</td>
<td>REDO</td>
<td>------</td>
</tr>
</tbody>
</table>

Figure 19. Data Class List Panel

The FRLOG column indicates the state of VSAM batch logging for each data class name. The FRLOG field indicates if VSAM batch logging was specified. NONE or blank indicates that no logging will be done. REDO indicates that VSAM will call CICSVR to log changes to the forward recovery log stream specified by the LOGSTREAM ID.
Setting up the logging environment for CICSVR VSAM batch logging

All CICSVR VSAM batch logging is performed using the services that are provided by the MVS system logger. All CICSVR logs are written to MVS system logger log streams. CICSVR only supports forward recovery logs for CICSVR VSAM batch logging.

The following topics are described in this section:
- Understanding CICSVR and VSAM batch logging
- Planning log streams for forward recovery logs
- Authorizing access to MVS log streams
- Setting up the logging environment

Understanding CICSVR and VSAM batch logging

VSAM obtains the forward recovery log stream name from the ICF catalog and passes this information to the CICSVR server. The forward recovery log name is stored in the RCDS and it is used when a forward recovery is performed. CICSVR can create forward recovery logs and use them to recover a user’s VSAM data set.

CICSVR can create forward recovery logs in a single system environment or a multiple system sysplex environment connected through the coupling facility.

When the forward recovery log data is written through the coupling facility, the MVS system logger will automatically merge online output with the log stream by log stream name. It does not matter which system in the sysplex the log stream came from.

You can use the CICSVR ISPF dialog interface to set up retention periods for your log stream blocks, so the old information in the log stream (log tail) will be deleted when it is older than a specified number of days.

Deletion of the log stream blocks is handled by the CICSVR server address space. The CICSVR server address space checks the retention period daily and deletes the blocks that are older than the retention period. The information in the RCDS pertaining to those blocks is also deleted. Retention periods are also checked when log of logs scan is run with CICS TS.

Planning log streams for forward recovery logs

CICSVR performs the logging for VSAM data sets when you use FRLOG=REDO. Multiple data sets can write to one forward recovery log stream; you do not need to define a log stream for each forward-recoverable data set. Your decision as to how many log streams to define is a balance of transaction performance, rapid recovery, and the work involved in managing a large number of log streams.

The MVS logger merges all the forward recovery log records from the various CICSVR instances onto the shared forward recovery log. Some points to consider are:
- All data sets used by one batch job should use the same log stream (to reduce the number of log streams written to at syncpoint).
- Share a forward recovery log stream between data sets that:
  - Have similar security requirements
  - Have similar backup frequency
  - Are likely to need restoring in their entirety at the same time.
Log stream names should relate to the data sets. For example, PAYROLL.data_sets could be mapped to a forward recovery log named PAYROLL.FWDRECOV.PAYLOG.

Do not mix high update frequency data sets with low update frequency data sets because this causes a disproportionate amount of unwanted log data to be read during recovery of low frequency data sets.

Do not put all high update frequency data sets on a single log stream because you could exceed the throughput capacity of the log stream.

If you define too many data sets to a single log stream, you could experience frequent structure-full events when the log stream can not keep up with data flow.

Delete redundant data from log streams periodically so that the log streams do not become excessively large. Typically, for a forward recovery log, deletion of old data is related to the data backup frequency. For example, you might keep the four most recent generations of backup so that when you delete a redundant backup generation, you can also delete the corresponding redundant forward recovery log records. These are the records older than the redundant backup and they are no longer needed for forward recovery.

### Authorizing access to MVS log streams

You must define authorization to system logger resources in order for CICSVR to be able to access, read, and write its forward recovery log streams. This applies to DASD-only log streams and the coupling facility. You can use RACF®, or an equivalent security product to implement access.

### Setting up the logging environment

If you want to use CICSVR VSAM batch logging, you must set up your logging environment according to the type of logging you want: DASD-only, coupling facility logging, or a combination of both types of logging.

#### DASD-only logging

Use DASD-only log streams when you do not have access to or do not want to use a coupling facility for logging. DASD-only log streams can only be single-system in scope; only one system can connect to a DASD-only log stream.

#### Coupling facility logging

Use coupling facility logging if you have applications that run on multiple systems in the sysplex and require a single sysplex-wide log stream. A coupling facility is required to merge the data from multiple systems into a single log stream.

#### Combination of coupling facility logging and DASD-only logging

You can use a combination of coupling facility logging and DASD-only logging if you use a coupling facility. You can have applications that run on multiple systems in the sysplex that require the log records to be merged into a single log stream. You can also have applications that require separate DASD-only log streams for each system. The coupling facility manages both the coupling facility logging and the DASD-only logging in the sysplex.

To help you decide which type of log stream is right for you, refer to [z/OS MVS Setting Up a Sysplex](https://publib.boulder.ibm.com/infocenter/zos/v2r12/topic/com.ibm.zos.r12.govm.zos_concepts_1908/zosinspace.html). This book also contains detailed information on how to set up the coupling facility resource management policy (CFRM policy) and the LOGR couple data set (LOGR policy).
Detecting batch logging error conditions

If CICSVR detects a problem with the MVS system logger, CICSVR will attempt to trap the error and issue a 3999 condition code. The 3999 condition code indicates that no batch logging occurred for that sphere. Since no batch logging is performed, you must take a backup of the sphere to ensure recoverability.

[Appendix C. Diagnosing logging problems on page 281] discusses the different types of problems that might occur. It also show examples of conditional JCL that you can add to your batch jobs to detect the 3999 condition code from CICSVR VSAM batch logging and trigger a DFSMShsm backup or DFSMSdss dump.
## Part 2. CICSVR and CICS TS

### Chapter 5. Setting up CICSVR for CICS TS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up your CICSVR ISPF environment</td>
<td>37</td>
</tr>
<tr>
<td>Allocating data sets to your TSO session</td>
<td>37</td>
</tr>
<tr>
<td>Running the CICSVR ISPF dialog interface</td>
<td>38</td>
</tr>
<tr>
<td>Modifying an existing ISPF selection menu</td>
<td>39</td>
</tr>
<tr>
<td>Selecting the CICSVR main menu directly from TSO</td>
<td>39</td>
</tr>
<tr>
<td>Using the ISPF select service</td>
<td>40</td>
</tr>
<tr>
<td>Editing the CICSVR JCL skeleton</td>
<td>40</td>
</tr>
<tr>
<td>Setting up your CICSVR ISPF environment</td>
<td>41</td>
</tr>
<tr>
<td>Creating and maintaining the RCDS</td>
<td>42</td>
</tr>
<tr>
<td>Sizing the RCDS</td>
<td>42</td>
</tr>
<tr>
<td>Creating the RCDS</td>
<td>43</td>
</tr>
<tr>
<td>Allocating the RCDS</td>
<td>43</td>
</tr>
<tr>
<td>Maintaining the RCDS</td>
<td>44</td>
</tr>
<tr>
<td>Deciding which VSAM spheres to protect</td>
<td>46</td>
</tr>
<tr>
<td>Registering or deregistering your log of logs data sets</td>
<td>47</td>
</tr>
<tr>
<td>Registering your log of logs data sets</td>
<td>47</td>
</tr>
<tr>
<td>Deregistering your log of logs data sets</td>
<td>47</td>
</tr>
<tr>
<td>Setting up your retention periods for your log stream copies and log stream blocks</td>
<td>48</td>
</tr>
<tr>
<td>Retention periods for log stream copies</td>
<td>48</td>
</tr>
<tr>
<td>Retention periods for log stream blocks</td>
<td>48</td>
</tr>
<tr>
<td>Setting up the log of logs scan utility (scan)</td>
<td>49</td>
</tr>
<tr>
<td>Improving forward recovery performance using the scan utility</td>
<td>49</td>
</tr>
<tr>
<td>Coupling facility logging</td>
<td>49</td>
</tr>
<tr>
<td>DASD-only logging</td>
<td>50</td>
</tr>
<tr>
<td>Setting up the scan utility to run as a batch job</td>
<td>50</td>
</tr>
<tr>
<td>Understanding the log of logs scan utility report</td>
<td>50</td>
</tr>
<tr>
<td>Setting up CICSVR change accumulation</td>
<td>52</td>
</tr>
<tr>
<td>Step 1: Set up change accumulation batch jobs</td>
<td>52</td>
</tr>
<tr>
<td>Understanding the change accumulation utility report</td>
<td>55</td>
</tr>
<tr>
<td>Step 2: Use the change accumulation data sets in a forward recovery</td>
<td>59</td>
</tr>
<tr>
<td>Deciding if you want to use shadow copies</td>
<td>60</td>
</tr>
<tr>
<td>Preparing for shadow forward recovery</td>
<td>61</td>
</tr>
<tr>
<td>Replacing the original sphere with a shadow copy</td>
<td>62</td>
</tr>
</tbody>
</table>

### Chapter 6. Setting up your CICS TS environment

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding logging with CICS TS for z/OS and OS/390</td>
<td>65</td>
</tr>
<tr>
<td>CICS logging and journaling</td>
<td>65</td>
</tr>
<tr>
<td>Understanding MVS log streams</td>
<td>66</td>
</tr>
<tr>
<td>Defining system log streams</td>
<td>66</td>
</tr>
<tr>
<td>Defining forward recovery log streams</td>
<td>66</td>
</tr>
<tr>
<td>Defining the log of logs</td>
<td>67</td>
</tr>
<tr>
<td>Defining recovery attributes by CICS resource definition</td>
<td>67</td>
</tr>
<tr>
<td>Establishing procedures to backup and recover your VSAM spheres</td>
<td>69</td>
</tr>
<tr>
<td>Deciding how often to make backups</td>
<td>69</td>
</tr>
<tr>
<td>Making daily backups of your VSAM spheres</td>
<td>69</td>
</tr>
<tr>
<td>Making nondaily backups of your VSAM spheres</td>
<td>69</td>
</tr>
<tr>
<td>Deciding when to make additional backups</td>
<td>69</td>
</tr>
<tr>
<td>Deciding how to group your backups</td>
<td>69</td>
</tr>
<tr>
<td>Scheduling backups to minimize your CICS TS down time</td>
<td>70</td>
</tr>
<tr>
<td>Determining your backup and restore products</td>
<td>70</td>
</tr>
<tr>
<td>Using DFSMShsm as your backup utility</td>
<td>70</td>
</tr>
</tbody>
</table>
Chapter 5. Setting up CICSVR for CICS TS

This chapter describes the tasks you need to do to set up CICSVR for CICS TS.

The following sections are described:

- Set up your CICSVR ISPF environment
- Understand the recovery control data set
- Decide which VSAM spheres to protect
- Register or deregister your log of logs data sets
- Set up the retention periods for your log stream blocks
- Set up log of logs scan to run at regularly scheduled times
- Set up CICSVR change accumulation (CA)
- Decide if you want to use shadow copies

You can also use CICSVR VSAM batch logging when your CICS system is down. For more information, see "Chapter 4. Setting up CICSVR VSAM batch logging" on page 27.

Setting up your CICSVR ISPF environment

The best way to use CICSVR is through the CICSVR ISPF dialog interface. If you use DFSMShsm or DFSMSdss, the dialog interface automates your recovery process. The dialog interface builds and submits your forward recovery jobs. If a DFSMShsm or DFSMSdss backup exists for the VSAM sphere that you are recovering, CICSVR restores this for you. CICSVR builds the recovery jobs using information in the RCDS and data you have entered through the CICSVR panels and windows. You can also use the dialog interface to list information from the RCDS about your VSAM spheres and logs.

CICSVR dialogs run under ISPF, so you must set up an appropriate ISPF environment. If you are not familiar with ISPF dialogs, refer to z/OS ISPF Dialog Developer’s Guide and Reference or OS/390 ISPF Dialog Developer’s Guide and Reference.

To set up your ISPF environment, perform these steps:
1. Allocate ISPF data sets to the TSO session.
2. Run the CICSVR ISPF dialog interface.
3. Edit the CICSVR JCL skeleton.

Allocating data sets to your TSO session

Table 2 shows the data sets that you must allocate to the TSO session to run the CICSVR ISPF dialog interface.

<table>
<thead>
<tr>
<th>ddname</th>
<th>CICSVR</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWWCON1</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWCON2</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWCON3</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWMSG</td>
<td>CICSVR message data set</td>
<td>See Note 1.</td>
</tr>
<tr>
<td>DWWPRINT</td>
<td>CICSVR output data set</td>
<td>See Note 2.</td>
</tr>
</tbody>
</table>
Table 2. ISPF dialog data sets (continued)

<table>
<thead>
<tr>
<th>ddname</th>
<th>CICSVR</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWWLLIB</td>
<td>Load Library</td>
<td>See Note 4.</td>
</tr>
<tr>
<td>DWWSLIB</td>
<td>JCL skeleton library</td>
<td>See Note 4.</td>
</tr>
<tr>
<td>ISPFILE</td>
<td>File tailoring output and your JCL skeleton</td>
<td>Your existing ISPF data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See note 3.</td>
</tr>
<tr>
<td>ISPLLIB</td>
<td>Load library</td>
<td></td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>Message library</td>
<td></td>
</tr>
<tr>
<td>ISPLLIB</td>
<td>Panel library</td>
<td></td>
</tr>
<tr>
<td>ISPPROF</td>
<td>User-session defaults, read/write tables</td>
<td>Your existing ISPPROF data set.</td>
</tr>
<tr>
<td>ISPSLIB</td>
<td>JCL skeleton library</td>
<td>See Note 3.</td>
</tr>
<tr>
<td>ISPTLIB</td>
<td>Read tables (default)</td>
<td></td>
</tr>
<tr>
<td>DWWDUMP</td>
<td>CICSVR diagnostic data set</td>
<td>See Note 6</td>
</tr>
<tr>
<td>DWWDMSG</td>
<td>CICSVR prolog and epilog data set</td>
<td>See Note 6</td>
</tr>
</tbody>
</table>

Notes:

1. DWWMSG defines the output data set that contains the CICSVR messages. Use a different data set name than the one used on this system by the CICSVR server address space.

   The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.

2. DWWPRINT defines the output data set that contains the reports produced by CICSVR. Use a different data set name than the one used on this system by the CICSVR server address space.

3. Ensure that the library containing your own CICSVR JCL skeletons is allocated to both the ISPSLIB and ISPFILE DD statements.

4. Since CICSVR uses LMFIND to locate a member in ISPLLIB and ISPSLIB, only four data sets can be concatenated in these libraries. Since this may be a problem for some users, CICSVR supports alternate names: DWWLLIB for ISPLLIB and DWWSLIB for ISPSLIB. CICSVR will use DWWLLIB in place of ISPLLIB and DWWSLIB in place of ISPSLIB.

5. Do not use the RCDSs created by the Installation Verification Procedure (IVP). Instead, make sure the clean up job (DWWCLNUP) has been run to delete the IVP data sets. Then review Migrating your RCDS from CICSVR V2R3 to CICSVR V3R1 on page 14 for instructions, or Creating and maintaining the RCDS on page 42.

6. DWWDMSG and DWWDUMP define the output data sets that CICSVR writes diagnostic and abend information to. If these data sets are not allocated, this information will be unavailable.

Running the CICSVR ISPF dialog interface

This section discusses three different methods for invoking the CICSVR dialog:

- Modify an existing ISPF selection menu
- Select the CICSVR main menu directly from TSO
- Use the ISPF select service
Modifying an existing ISPF selection menu

You can invoke CICSVR simply by including CICSVR as an option on the existing primary option menu (ISR@PRIM) that is shipped in ISP@SSPPENDU or on another selection menu. Insert the statements that are marked on the right with an arrow (<=) in Figure 20.

```plaintext
)BODY
  :
  1 "" - "" - ""
  2 "" - "" - ""
  "" - "" - ""
  0 CICSVR - CICS VSAM Recovery <=
  "" - "" - ""
)PROC
  :
&ZSEL = TRANS(TRUNC(&ZCMD,'.'))
  1 , ....
  2 , ....
  , ....
  0 , 'PGM(DWPM) NEWAPPL(DW) PASSLIB' <=
  , ....
  ...
)END
```

Figure 20. Modifying the ISPF selection menu for CICSVR

Specify that you want to receive all write-to-programmer messages (including the message ID), at your terminal. This will help you to analyze any problems that might occur while setting up the CICSVR ISPF dialog interface. To do this, enter this TSO command:

```
PROFILE WTPMSG MSGID
```

Before you can invoke CICSVR, you must allocate the CICSVR data sets. You can allocate these data sets through the TSO logon procedure, or by executing a command list (CLIST) after TSO logon. See "Appendix D. Sample CLIST (DWWCLIST)" on page 287.

Selecting the CICSVR main menu directly from TSO

To select the CICSVR main menu directly from TSO, enter this TSO command:

```
ISPSTART PGM(DWPM) NEWAPPL(DW)
```

Remember that you must allocate the CICSVR ISPF dialog interface data sets before running this command. If you select the CICSVR main menu directly from TSO, you cannot split the screen to run two CICSVR sessions.

Using this method to invoke the dialog means that Figure 21 on page 40 is the first ISPF panel displayed:
Using the ISPF select service

Another method of starting the CICSVR ISPF dialog interface is to execute the SELECT command from a CLIST, or from a program. A sample CLIST (DWW.SDWWCNTL(DWWCLIST)) is provided on the product tape and is shown in "Appendix D. Sample CLIST (DWWCLIST)" on page 287. Modify DWWCLIST according to your site requirements. Refer to your ISPF manuals for more information about these methods.

Editing the CICSVR JCL skeleton

If you select option 5 from the main menu (Figure 21), this JCL skeleton secondary window appears (Figure 22 on page 41) and invokes the ISPF/PDF editor.
Here you can edit the CICSVR JCL skeleton information to conform to your organization's standards. Use F3 to leave the editor and return to the CICSVR main menu.

You must edit the JCL skeleton and modify the following information:

- The job card, TESTGFS&CJOBCHAR, (do not remove the &CJOBCHAR variable)
- The CICSVR load library, STEPLIB
- The CICSVR RCDS, DWWCON1, DWWCON2, and DWWCON3 DDs to what you have defined.

```
//+ PROPRIETARY V3 STATEMENT
//+ LICENSED MATERIALS - PROPERTY OF IBM
//+ "RESTRICTED MATERIALS OF IBM"
//+5655-H91
//(C) COPYRIGHT 1996,2001 IBM CORP.
//+ END PROPRIETARY V3 STATEMENT
//+
//+ ****************************************************************************
//+ Add the JOB statement to meet your system requirements.
//+
//+ Do not remove the &CJOBCHAR variable in the JOB name.
//+
//+ Change the DSN value in STEPLIB DD to the name of your CICSVR library.
//+
//+ Change RCDS DSN name in DWWCON1, DWWCON2, and DWWCON3 DDs to what you have defined.
//+
//+ ****************************************************************************
//+ TESTGFS&CJOBCHAR JOB (ACCOUNT), MSGLEVEL=(1,1), NOTIFY=USERID,
//+ MSGCLASS=X, CLASS=A, REGION=4M
//+ DWW PROC
//+ RECOVER EXEC PGM=DWWCO
//+ STEPLIB DD DSN=DWW.SDWWLOAD, DISP=SHR
//+ DWWMSG DD SYSOUT=* 
//+ DWWPRINT DD SYSOUT=* 
//+ DWWCON1 DD DSN=DWW.DWWCON1, DISP=SHR 
//+ DWWCON2 DD DSN=DWW.DWWCON2, DISP=SHR 
//+ DWWCON3 DD DSN=DWW.DWWCON3, DISP=SHR 
//+ PEND 
//+ END OF PROC
```

**Figure 22. JCL skeleton secondary window**

Here you can edit the CICSVR JCL skeleton information to conform to your organization’s standards. Use F3 to leave the editor and return to the CICSVR main menu.

You must edit the JCL skeleton and modify the following information:

- The job card, TESTGFS&CJOBCHAR, (do not remove the &CJOBCHAR variable)
- The CICSVR load library, STEPLIB
- The CICSVR RCDS, DWWCON1, DWWCON2, and DWWCON3

**Understanding the recovery control data set**

The CICSVR recovery control data set (RCDS) contains essential information for recovering your VSAM spheres. The RCDS contains information on the following:

- CICSVR ISPF dialog interface default values
- MVS log streams
- MVS log stream copies
- DFSMSShsm and DFSMSdss backups
- Change accumulation data sets
- CICSVR VSAM batch logging
- Shadows
This information is stored in three identical linear VSAM data sets. Three data sets are used to reduce the possibility of data loss. CICSVR regularly checks to make sure the three data sets are consistent.

The information in the RCDS is updated under the following conditions:

- Through the CICSVR ISPF dialog
- By using the log of logs scan, with CICS TS
- By using log stream copy, with CICS TS
- Automatically by CICSVR, with change accumulation
- Automatically by CICSVR, with VSAM batch logging
- Automatically by doing a shadow forward recovery
- Automatically when the CICSVR server address space is active

You can reduce the likelihood of your RCDS becoming full by deregistering old information from the RCDS. You can set up automatic deregistration for your log streams (log stream copies or the log-tail) using the CICSVR ISPF dialog interface. In addition, you can instantly deregister the following using the CICSVR dialog interface:

- Log streams
- SAM copies of a log
- Log of logs

Deregistration is triggered by running scan or automatically by the CICSVR server address space. Once deregistered, CICSVR automatically reuses the freed space in the RCDS; you never need to manually recover free space. For more information on deregistering, refer to CICSVR V3R1 User’s Guide and Reference.

Creating and maintaining the RCDS

The steps you need to take to create and maintain the RCDSs are slightly different depending on if the CICSVR server address space is enabled or not. If you have enabled the CICSVR server address space, see Creating and maintaining the RCDSs on page 20 for information related to the CICSVR server.

In general, you can use the following information for creating and maintaining the RCDSs. The following sections describe how to:

- Size the RCDS
- Create the RCDS
- Maintain the RCDS

Sizing the RCDS

Use this formula to estimate the size of your RCDS:

\[ \text{nvsam} \times \text{nopen} \times \text{ndays} \times 8192 \text{ bytes} \]

Where:

- \text{nvsam} is the number of VSAM data sets
- \text{nopen} is the average number of times a VSAM file is opened or closed per day
- \text{ndays} is the number of days of log information you want to keep

For example, if you have 100 VSAM data sets that are opened or closed 50 times each per day and the information is kept for 30 days, calculate the approximate size of the RCDS as follows:

\[ 100 \times 2 \times 30 \times 8192 \text{ bytes} = 49152000 \text{ bytes} \]
You can estimate the size of your RCDS as 49 megabytes.

IBM recommends that you make one of the recovery control data sets smaller than the other two. If the smaller one becomes full, CICSVR will issue messages warning you that your smaller RCDS is full, then will continue using the larger RCDSs. This warning notifies you that your RCDS needs maintenance and lets you take action before all three RCDSs are full and halts recovery or CICSVR VSAM batch logging. See "Maintaining the RCDS" on page 42 for more information.

Creating the RCDS
The three RCDSs are linear VSAM data sets. They contain identical copies of essential information CICSVR needs to recover your VSAM spheres. To reduce the possibility of losing all three copies of the RCDSs, IBM recommends that you create each RCDS on separate volumes, disk controllers, and channels. In addition, your VSAM spheres should be on different volumes, disk controllers and channels from your RCDSs. These precautions prevent a single hardware failure from making all three RCDSs unusable, or from making the RCDS and the VSAM sphere to be recovered unusable. CICSVR regularly checks the consistency of data between the three data sets. If one or two of the RCDSs are temporarily lost (for example, the unavailability of a disk path), CICSVR automatically copies the data from the remaining RCDSs the next time the other data sets are available.

Figure 23 shows sample AMS commands to define three RCDSs.

```
//ALLOC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  //SYSIN DD *
    DEFINE CLUSTER -
        (NAME('DWW.DWWCON1') -
        VOLUMES(TSOL01) -
        CYLINDERS(60 20) -
        LINEAR -
        SHR(3 3))

    DEFINE CLUSTER -
        (NAME('DWW.DWWCON2') -
        VOLUMES(TSO007) -
        CYLINDERS(60 20) -
        LINEAR -
        SHR(3 3))

    DEFINE CLUSTER -
        (NAME('DWW.DWWCON3') -
        VOLUMES(TSOL02) -
        CYLINDERS(30 20) -
        LINEAR -
        SHR(3 3))
/*
```

Figure 23. AMS Commands to Define the CICSVR RCDS

Allocating the RCDS
The ddnames for the RCDSs are DWWCON1, DWWCON2, and DWWCON3. Figure 24 provides an example of allocating the RCDS in the CICSVR recovery job.
This section describes how to repair your RCDS if one or more RCDSs become full or damaged.

If your RCDS is full: In this situation, you have received the DWW1605I DWWCONx is full informational message and you want to increase the size of your RCDSs before all the RCDSs become full. Figure 26 on page 45 and Figure 27 on page 46 show how you can:

- Create temporary VSAM data sets with more space than the original data sets.
- Use the AMS REPRO command to copy the old VSAM data sets to the temporary data sets.
- Check the results of the REPRO.
- Delete the original VSAM data sets.
- Rename the temporary VSAM data sets to the original names.

Figure 26 on page 45 shows the AMS commands to create new temporary data sets and how to copy the information in the old VSAM data sets to the temporary data sets.
Verify that all the AMS REPRO steps are successful before proceeding.

The example in Figure 27 on page 46 shows the AMS commands to delete the original data sets and rename the new data sets.

```plaintext
//DEFCLUS1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
  DEFINE CLUSTER - 
    (NAME('DWW.DWWTEMP1') -  
     VOLUMES(TSO001) -  
     CYLINDERS(100 30) -  
     LINEAR -  
     SHR(3 3))  
  DEFINE CLUSTER - 
    (NAME('DWW.DWWTEMP2') -  
     VOLUMES(TSO007) -  
     CYLINDERS(100 30) -  
     LINEAR -  
     SHR(3 3))  
  DEFINE CLUSTER - 
    (NAME('DWW.DWWTEMP3') -  
     VOLUMES(TSO002) -  
     CYLINDERS(75 30) -  
     LINEAR -  
     SHR(3 3))  
  REPRO - 
    INDATASET('DWW.DWWCON1') -  
    OUTDATASET('DWW.DWWTEMP1')  
  REPRO - 
    INDATASET('DWW.DWWCON2') -  
    OUTDATASET('DWW.DWWTEMP2')  
  REPRO - 
    INDATASET('DWW.DWWCON3') -  
    OUTDATASET('DWW.DWWTEMP3')  
/*

Figure 26. Increasing the size of the RCDS (steps 1 and 2)

Verify that all the AMS REPRO steps are successful before proceeding.

The example in Figure 27 on page 46 shows the AMS commands to delete the original data sets and rename the new data sets.

Chapter 5. Setting up CICSVR for CICS TS 45
Remember to rename the cluster and the data components of the data sets.

If your RCDS is damaged: Assume that your DWWCON3 RCDS is damaged and you want to delete it and create a new replacement RCDS. Figure 28 shows the AMS commands to delete the damaged RCDS and define a new one.

Deciding which VSAM spheres to protect

You will probably want to be able to forward recover all VSAM spheres that are defined as CICS files. Of course, data sets that are read-only or browse-only do not need forward recovery or backout, but you should still back them up regularly in case they become unusable.
Ensure that you define your VSAM base clusters and alternate index (AIX) data sets with SHAREOPTION 1 or SHAREOPTION 2. You can do this by using the SHAREOPTION parameter of the appropriate access method services (AMS) command, such as the DEFINE CLUSTER command or the ALTER command. SHAREOPTION 3 and SHAREOPTION 4 are not suitable for CICS, because CICS cannot prevent concurrent write access to the data sets by other CICS regions or by batch jobs.

If you must use SHAREOPTION 3 or SHAREOPTION 4 with multiple CICS regions sharing the same VSAM sphere, you must use unique ddnames across different systems. Unique ddnames ensures the correct sequencing and enables CICSVR to correctly process the log records for forward recovery. Refer to the SHAREOPTION description in z/OS DFSMS Access Method Services for more information about the risks involved with SHAREOPTION 3.

Registering or deregistering your log of logs data sets

All log of logs data sets must be registered in the RCDS for CICSVR to know about them. Use the CICSVR ISPF dialog interface to register or deregister your log of logs.

Registering your log of logs data sets

When you define a new log of logs, you must register it by selecting option 4, List of registered log of logs (CICS TS), from the CICSVR main menu panel. Then, use the Administrate pull-down from the CICSVR log of logs list panel and select option 1, Register. The CICSVR log of logs register panel is displayed as shown in Figure 29.

Deregistering your log of logs data sets

By deregistering deleted log of logs, you can help to keep the RCDS from becoming larger than necessary. You can delete the log of logs first and then deregister it or vice versa; the order in which this is performed does not matter. Select option 4, List of registered log of logs (CICS TS), from the CICSVR main menu panel. Select the log of logs that you want to deregister from the log of logs list panel, then use the Administrate pull-down and select option 2, Deregister. The CICSVR log of logs deregister panel is displayed as shown in Figure 30 on page 48.
Setting up your retention periods for your log stream copies and log stream blocks

IBM recommends that you set up retention periods for your log stream copies and log streams. Setting up retention periods will help you in the following ways:

- It prevents your RCDS from filling up with obsolete information.
- It helps you to manage your disk space by uncataloging and deleting obsolete copies of log streams.
- It helps you to manage the size of your log streams by performing the log tail delete function on the oldest information. Log stream blocks that are older than the retention period will be deleted.

Retention periods for log stream copies

When the log of logs scan utility is invoked, scan will check all log stream copies that are registered in the RCDS and it will deregister the copies that are older than the retention period. To specify the retention criteria for a log stream copy, select option 3 from the CICSVR main menu panel, List of log streams (CICS TS). Then, use the Administrate pull-down from the CICSVR log stream list panel and select option 2, Automatic deregister.

Retention periods for log stream blocks

When the log of logs scan utility is invoked, scan will check all the log streams that are registered in the RCDS and it will delete all the log stream blocks that are older than the retention period. To specify the retention criteria for a log stream block, select option 3 from the CICSVR main menu panel, List of log streams (CICS TS). Use the Administrate pull-down from the CICSVR log stream list panel and select option 2, Automatic deregister.
Figure 31 shows the CICSVR automatic log stream deregister secondary window. To enable automatic deregistration of log stream blocks from the RCDS, enter a value in the Retention period for blocks field.

If the CICSVR server address space is active, it will also check all log stream blocks that are registered in the RCDS and it will deregister the copies that are older than the retention period.

```
DWWPPSAD  CICSVR automatic log stream deregister
Command ===>

Specify the automatic deregister criteria for log stream blocks and log stream copies, then press ENTER. Or, leave all fields blank, then press ENTER to turn off the automatic deregister function.

Automatic deregister . . . . : ON
Retention period for blocks  30 (Number of days)
Retention period for copies  30 (Number of days)
Uncatalog and delete . . . . . . 1 (1=Yes, 2=No)
1=Help  F12=Cancel
```

Setting up the log of logs scan utility (scan)

The LOGOFLOGS scan utility scans the log of logs, gathers information, and updates the RCDS with this information. When you use the CICSVR ISPF dialog interface to perform a forward recovery or a complete recovery, scan runs automatically. If scan has not run recently, it may take a significant amount of time for it to finish.

Improving forward recovery performance using the scan utility

You can significantly improve the performance of your forward recoveries by setting up scan to run as a batch job that runs at regularly scheduled times. If your system is setup as part of a sysplex, you may need to run scan on more than one system, depending on whether you use coupling facility log streams or DASD-only log steams, or a combination of both types of log streams.

Coupling facility logging

If coupling facility logging is used on all the systems in the sysplex, then you only need to run scan on one system to update the RCDS. Scan will connect to all the log streams registered in the RCDS on all the systems it has access to using the coupling facility.

If both coupling facility logging and DASD-only logging are used on systems connected to the sysplex, the two types of logging must be handled separately. You only need to run scan on one of the systems using coupling facility logging to update the RCDS. You must run scan on each of the DASD-only logging systems to update the RCDS. Scan cannot connect to systems with DASD-only logging from another system in the sysplex on those systems.
If you want to access the RCDS from more than one MVS system, you must have Global Resource Serialization (GRS) installed.

**DASD-only logging**

If DASD-only logging is used, you must run scan to update the RCDS on each system. Each system is independent of the next system, so separate log of logs, RCDS, and log streams must be defined on each system. Scan cannot connect to systems with DASD-only logging from another system in the sysplex.

For more information on DASD-only logging, refer to z/OS MVS Setting Up a Sysplex or OS/390 MVS Setting Up a Sysplex.

### Setting up the scan utility to run as a batch job

Set up scan as a batch job that can be run at regularly scheduled times as shown in Figure 32. Run scan several times a day (for example, every four hours) on all the necessary systems to keep the RCDS current. If you have multiple log streams (for example, more than one system using DASD-only logging, or if you use both coupling facility logging and DASD-only logging) consider scheduling your batch scan jobs so that they run at different times to avoid contention (that is, system level enqueues).

```plaintext
//JOBSCN2 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M /* CICSVR PROGRAM */ STEP1 EXEC PGM=DWWAR /* LIBRARY CONTAINING CICSVR PROGRAMS */ STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR /* CICSVR RECOVERY CONTROL DATA SET NAMES */ DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR DWWIN DD * LOGOFLOGS SCAN
```

Figure 32. Sample scan job you can run at regularly scheduled times

The LOGOFLOGS command in Figure 32 tells CICSVR to scan all the log of logs that are registered in the RCDS. All the information needed for recovery is stored in the RCDS. In addition, if the retention period for any log stream blocks or log stream copies is met, the log stream blocks or copies are deregistered from the RCDS and deleted accordingly. Use the CICSVR dialog interface to set or change the retention period for log stream blocks or log stream copies.

Each time that scan is run, CICSVR begins at the point in which the last scan ended. For more information on scan see the LOGOFLOGS command in LOGOFLOGS—Process a log of logs on page 179.

### Understanding the log of logs scan utility report

A CICSVR log of logs scan utility report is produced every time the scan utility runs. Figure 33 on page 51 shows the log of logs scan report, followed by descriptions of the fields in the report.
Figure 33. Log of logs scan report

<table>
<thead>
<tr>
<th>Log of logs name</th>
<th>The name of the MVS log of logs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>First time GMT</td>
<td>The first time recorded on the log of logs in GMT format.</td>
</tr>
<tr>
<td>Last time GMT</td>
<td>The last time recorded on the log of logs in GMT format.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>First time LOCAL</td>
<td>The first time recorded on the log of logs in local time format.</td>
</tr>
<tr>
<td>Last time LOCAL</td>
<td>The last time recorded on the log of logs in local time format.</td>
</tr>
<tr>
<td>First block number</td>
<td>The first block number on the log of logs.</td>
</tr>
<tr>
<td>Last block number</td>
<td>The last block number on the log of logs.</td>
</tr>
<tr>
<td>VSAM data set name</td>
<td>The unique data set name for each tie-up record.</td>
</tr>
<tr>
<td>CICSID</td>
<td>The CICS APPLID.</td>
</tr>
<tr>
<td>FCT name</td>
<td>The ddname of the data set, as defined in the FCT.</td>
</tr>
<tr>
<td>Open date/time</td>
<td>The date and time the log of logs was opened.</td>
</tr>
<tr>
<td>Close date/time</td>
<td>The date and time the log of logs was closed.</td>
</tr>
<tr>
<td>MVS log stream name</td>
<td>The name of the MVS log stream.</td>
</tr>
</tbody>
</table>

**Information for a Forward Recovery:**

This information is provided if you request a recovery report from the LOGOFLOGS utility:

| Start time GMT         | The start time for the recovery of this data set, in GMT format. |
| Stop time GMT          | The stop time for the recovery of this data set, in GMT format. |
| MVS log streams needed | The MVS log streams needed for the recovery of this data set. |

The LOGOFLOGS utility might produce error or information messages in the DWWMSG file.

### Setting up CICSVR change accumulation

CICSVR CA reduces the time it takes to perform a forward recovery. CICSVR CA consolidates forward recovery log records into a CA data set. CICSVR uses the CA data set in conjunction with the forward recovery log to reduce the number of log records that CICSVR needs to apply to get the sphere back to the exact state before the data was lost.

If you use DFSMShsm, change accumulation is supported. If you use DFSMSdss, you must activate the CICSVR server address space to use change accumulation. See "Chapter 3. Activating the CICSVR server address space" on page 17 for more information.

CA is a two step process. The next two sections cover both of these steps.

### Step 1: Set up change accumulation batch jobs

The first step is to create batch jobs that define CA groups and identify the spheres in each group. Figure 34 on page 53 shows an example of the commands.
necessary to define a CA group and the sphere names associated with that group when a DFSMShsm or DFSMSdss backup is available.

```
//JOBCA1 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//* CICSVR PROGRAM
//STEP1 EXEC PGM=DDWWCA
//* LIBRARY CONTAINING CICSVR PROGRAMS
//STEPLIB DD DSN=DDWW.SDWWLOAD,DISP=SHR
//DDWMSG DD SYSOUT=*
//DDWPRINT DD SYSOUT=*
//DDWSORT DD SYSOUT=*
//* CICSVR RECOVERY CONTROL DATA SET NAMES
//DDWCON1 DD DSN=DDWW.DWWCON1,DISP=SHR
//DDWCON2 DD DSN=DDWW.DWWCON2,DISP=SHR
//DDWCON3 DD DSN=DDWW.DWWCON3,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//DDWIN DD *
CA -
   GROUP(MYCAGROUP1) -
   PREFIX(CA) -
   VOLUME(123456)
   UNIT(3390)
SPHERE -
   NAME(PROD.PAYROLL1)
SPHERE -
   NAME(PROD.PAYROLL2)
```

Figure 34. Change accumulation batch job using DFSMShsm or DFSMSdss backup

The CA command:

- Defines a CA group name, MYCAGROUP1
- Specifies that the high level qualifier, CA, is used for the dynamically created CA output data set
- Specifies the volume serial number for the CA output data set if it is not an SMS-managed data set
- Specifies the device type for the CA output data set if it is not an SMS-managed data set

The SPHERE command:

- Identifies the spheres in that group, PROD.PAYROLL1 and PROD.PAYROLL2

You should choose VSAM spheres that have the same backup requirements when you create your CA group. IBM recommends that you backup all the spheres in a CA group at the same time and then run the CA batch job.

When this job is run, CA calls DFSORT, or an equivalent product with equivalent support, to sort the forward recovery log stream records for PROD.PAYROLL1 and PROD.PAYROLL2. See "Appendix B. Diagnosing change accumulation and DFSORT problems" on page 279 for information on eliminating common errors.

When the records are sorted, CA consolidates the records by applying the log records for PROD.PAYROLL1 and PROD.PAYROLL2 to a CICSVR dynamically created CA data set.

Every time the CA job is run, the CA data set is updated with the information from the next forward recovery log range.
When the CA batch job is run again, CA gets the latest information about the backups. If a new backup has been taken, CA determines that the information in the CA data set for PROD.PAYROLL1 is no longer valid and does not use it. Instead, CA collects the log records from the forward recovery log for PROD.PAYROLL1 and the log records from the CA data set and the forward recovery log for PROD.PAYROLL2 and calls DFSORT to sort the records. When the records are sorted, CA consolidates the records by applying the log records for PROD.PAYROLL1 and PROD.PAYROLL2 to the CA data set.

It is important that you run the CA batch job immediately after a backup is taken for any of the spheres in the MYCAGROUP1 group so the CA information for that sphere is invalidated and will not be used by CICSVR if a recovery is requested. The CA batch job for MYCAGROUP1 should be run over and over again so that the CA data set is kept up-to-date. CA significantly speeds up forward recovery processing by consolidating the log stream records before a recovery is needed.

For information on diagnosing DFSORT problems see Appendix B. Diagnosing change accumulation and DFSORT problems" on page 279.

Figure 35 shows an example of the commands necessary to define a CA group and the sphere names associated with that group when a DFSMShsm or DFSMSdss backup is not available.

```
//JOBCA2 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//* CICSVR PROGRAM
//STEP1 EXEC PGM=DWWCA
//* LIBRARY CONTAINING CICSVR PROGRAMS
//STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR
//DWWMSG DD SYSOUT=*  
//DWWPRINT DD SYSOUT=*  
//DWWSORT DD SYSOUT=*  
//* CICSVR RECOVERY CONTROL DATA SET NAMES
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR
  //SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//DWWIN DD *
CA -
  GROUP(MYCAGROUP2) -
  VOLUME(123456) -
  UNIT(3490)
SPHERE -
  NAME(PROD.PAYROLL1) -
  BACKUPTIME(dateandtime,GMT)
SPHERE -
  NAME(PROD.PAYROLL2) -
  BACKUPTIME(dateandtime,GMT)
SPHERE -
  NAME(PROD.PAYROLL3) -
  BACKUPTIME(dateandtime,GMT)
SPHERE -
  NAME(PROD.PAYROLL4) -
  BACKUPTIME(dateandtime,GMT)
```

Figure 35. Change accumulation batch job with no DFSMShsm or DFSMSdss backup

The CA command defines a CA group name, MYCAGROUP2, and the SPHERE command identifies the spheres in that group, PROD.PAYROLL1,
PROD.PAYROLL2, PROD.PAYROLL3, and PROD.PAYROLL4. You must specify the date and time of the most recent backup for the sphere in the BACKUPTIME field. CA will only collect records for the VSAM sphere that occur after the backup was made. You can run this job multiple times for the VSAM spheres until a new backup is made. When a new backup is made for a VSAM sphere, you must update the BACKUPTIME field and run this job again. When you run this job, CA calls DFSORT to sort the forward recovery log stream records for PROD.PAYROLL1, PROD.PAYROLL2, PROD.PAYROLL3, and PROD.PAYROLL4. When the records are sorted, CA records are written to a CICSVR dynamically created CA data set on tape volume 123456.

Understanding the change accumulation utility report
A CICSVR CA utility report is produced every time the CA command is issued. Figure 36 on page 56 shows the CA utility report.
CICSVR - CHANGE ACCUMULATION UTILITY    DATE : 01/30    TIME : 17:50:36

CHANGE ACCUMULATION STATISTICS FOR CA GROUP

CA GROUP NAME : CICSMVS

CURRENT VSAM DATA SETS IN THIS CA GROUP  PREVIOUS VSAM DATA SETS IN THIS CA GROUP

<table>
<thead>
<tr>
<th>Current</th>
<th>Previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS1.PROD1.BASE</td>
<td>CICS1.PROD1.BASE</td>
</tr>
<tr>
<td>CICS1.PROD2.BASE</td>
<td>CICS1.PROD2.BASE</td>
</tr>
<tr>
<td>CICS1.PROD3.BASE</td>
<td></td>
</tr>
</tbody>
</table>

INPUT CA DATA SET STATISTICS

INPUT CA DATA SET NAME : A12345.B6789.C12345
FIRST RECORD TIMESTAMP, LOCAL : 2001.030 14.10.30
FIRST RECORD TIMESTAMP, GMT : 2001.030 16.10.30
LAST RECORD TIMESTAMP, LOCAL : 2001.030 14.10.35
LAST RECORD TIMESTAMP, GMT : 2001.030 16.10.35
NUMBER OF RECORDS READ : 20

VSAM DATA SETS STATISTICS FOR INPUT CA DATA SET OF THIS CA GROUP

<table>
<thead>
<tr>
<th>Base Cluster Name</th>
<th>Start Date and Time Local</th>
<th>Start Date and Time GMT</th>
<th>Stop Date and Time Local</th>
<th>Stop Date and Time GMT</th>
<th>Last Backup Time Local</th>
<th>Last Backup Time GMT</th>
</tr>
</thead>
</table>

CICSVR - CHANGE ACCUMULATION UTILITY    DATE : 01/30    TIME : 17:50:36

JOURNAL STATISTICS

JOURNAL NAME : CICSPROD.DFHJ02A
FIRST TIME LOCAL : 2001.030 14.15.30
LAST TIME LOCAL : 2001.030 14.15.35
NUMBER OF RECORDS READ : 20

JOURNAL STATISTICS

JOURNAL NAME : CICSPROD.DFHJ02B
FIRST TIME LOCAL : 2001.030 14.15.35
LAST TIME LOCAL : 2001.030 14.15.40
NUMBER OF RECORDS READ : 30

Figure 36. Change accumulation output, page 1 of 2

Figure 37 on page 57 shows CA output for an MVS log stream.
CA utility report fields description:
CA group name

The group name from the CA command. The CA Group name must be
unique and is used by CICSVR to identify the specified group of data sets in the RCDS. The group name is a maximum of 36 characters.

**Input CA data set name**
The name of the CA data set dynamically created during the previous CA utility run for this CA group.

**First record timestamp, local**
The first time recorded on the input CA data set (in the local time format).

**First record timestamp, GMT**
The first time recorded on the input CA data set (in the GMT format).

**Last record timestamp, local**
The last time recorded on the input CA data set (in the local time format).

**Last record timestamp, GMT**
The last time recorded on the input CA data set (in the GMT format).

**Number of records read**
The number of records read from the input CA data set.

**Base cluster name**
The base VSAM data set name from the input CA data set of this CA group.

**Start date and time local**
The first VSAM data set timestamp recorded on the input CA data set (in the local time format).

**Start date and time GMT**
The first VSAM data set timestamp recorded on the input CA data set (in the GMT format).

**Stop date and time local**
The last VSAM data set timestamp recorded on the input CA data set (in the local time format).

**Stop date and time GMT**
The last VSAM data set timestamp recorded on the input CA data set (in the GMT format).

**Last backup time local**
The VSAM data set backup time used when the input CA data set was created (in the local time format).

**Last backup time GMT**
The VSAM data set backup time used when the input CA data set was created (in the GMT format).

**Journal name**
The input journal name if CICS/ESA logs are used for this CA group.

**First time local**
The first time recorded on this journal (in the local time format only, GMT format is not applicable).

**Last time local**
The last time recorded on this journal (in the local time format only, GMT format is not applicable).

**Number of records read**
The number of records read from this journal.

**Logstream name**
The name of the MVS log stream used for CICS TS logging or VSAM batch logging.

**First time GMT**
The first time recorded on this log stream (in the GMT format).

**Last time GMT**
The last time recorded on this log stream (in the GMT format).

**First time local**
The first time recorded on this log stream (in the local time format).
Last time local
   The last time recorded on this log stream (in the local time format).
First block number
   The first block number on the log stream.
Last block number
   The last block number on the log stream.
Number of records read
   The number of records read from this log stream.
Output CA data set name
   The name of the CA data set dynamically created during the current CA utility run for this CA group.
Volume
   The volume serial number from the CA command. It must be specified if the output CA data set is not a SMS-managed data set.
Unit
   The device type from the CA command. It must be specified if the output CA data set is not a SMS-managed data set.
First record timestamp, local
   The first time recorded on the output CA data set (in the local time format).
First record timestamp, GMT
   The first time recorded on the output CA data set (in the GMT format).
Last record timestamp, local
   The last time recorded on the output CA data set (in the local time format).
Last record timestamp, GMT
   The last time recorded on the output CA data set (in the GMT format).
Number of records written
   The number of records written to the output CA data set.
Base cluster name
   The base VSAM data set name for the output CA data set of this CA group.
Start date and time local
   The first VSAM data set timestamp recorded on the output CA data set (in the local time format).
Start date and time GMT
   The first VSAM data set timestamp recorded on the output CA data set (in the GMT format).
Stop date and time local
   The last VSAM data set timestamp recorded on the output CA data set (in the local time format).
Stop date and time GMT
   The last VSAM data set timestamp recorded on the output CA data set (in the GMT format).
Last backup time local
   The VSAM data set backup time used when the output CA data set was created (in the local time format).
Last backup time GMT
   The VSAM data set backup time used when the output CA data set was created (in the GMT format).
Number of records written
   The number of records written to the output CA data set for this VSAM data set.

Step 2: Use the change accumulation data sets in a forward recovery

The second step occurs automatically when you use the CICSVR ISPF dialog interface to forward recover a sphere. When you use the ISPF dialog to forward recover a sphere, CICSVR automatically generates the APPLYCA keyword on the RECOVER command in the forward recovery job. APPLYCA tells CICSVR to apply
the records in the CA data set to the restored VSAM sphere, then apply the remaining log range from the forward recovery log, to pick up the most recent changes. If no CA data set exists, CICSVR ignores the APPLYCA keyword.

Using the CA data set can significantly improve the performance of your forward recovery runs. If there is a specific reason you do not want to use the CA data set for a forward recovery run, you can manually delete the APPLYCA lines from the forward recovery job. If the APPLYCA keywords are deleted, CICSVR will not use the CA data set and it will only apply the records from the forward recovery log.

Deciding if you want to use shadow copies

You can reduce the time it takes to forward recover a VSAM file by creating a shadow copy of it. With a shadow copy, you:

- Create a copy of the VSAM sphere.
- Regularly run a CICSVR shadow forward recovery job to update that copy with all the log entries up to the current time.
If the shadow forward recovery job is run often, the shadow copy of the VSAM sphere is kept up-to-date. If the VSAM sphere is damaged, only a few updates need to be made to the shadow copy to get it to the same state as the original VSAM sphere.

Shadow forward recovery is intended for users who are willing to maintain a copy of critical data sets to reduce the forward recovery time.

Shadow copies are also useful at remote recovery sites when using products such as Remote Recovery Data Facility (RRDF).

The following topics are discussed in this section:
- Preparing for shadow forward recovery
- Replacing the original sphere with a shadow copy

**Preparing for shadow forward recovery**

If you want to create a shadow copy of your VSAM sphere, you need to:
- Use IDCAMS to create a new, empty VSAM sphere.
- Copy the original VSAM sphere into the new VSAM sphere. The new VSAM sphere will be your shadow copy.
- Set up a CICSVR shadow forward recovery job to regularly update the shadow copy.

You can set up a shadow forward recovery job to update a shadow copy of your VSAM file. Figure 39 on page 62 shows an example of using LOGOFLOGS SCAN to update the RCDS with the latest information from the log of logs and the forward recovery job to update the shadow copy. You should run this job regularly so that the shadow is kept as up-to-date as possible.
Replacing the original sphere with a shadow copy

When a problem occurs, you can replace the original sphere with the updated shadow copy. Using the shadow copy can reduce the time it takes to perform a forward recovery because most of the log records have already been applied.

The basic steps to recover from a shadow copy are as follows:

- Run shadow forward recovery one more time to be sure that the shadow has all updates applied. If any change accumulation data sets are available, they will be used in addition to the forward recovery log.
- Run IDCAMS SHCDS with the FRSETRR keyword to mark the original sphere as being under maintenance. This makes it unavailable and also allows the following unbind operation to succeed.
- Run IDCAMS SHCDS with the FRUNBIND keyword to unbind any retained locks against the original sphere. This enables SMS/VSAM to preserve the locks ready for re-binding later to the recovered sphere.
- Run IDCAMS DELETE to delete the original sphere and define an empty sphere with the same name.
- Run IDCAMS SHCDS with the FRSETRR keyword to mark the new empty sphere as being under maintenance. This makes it unavailable while the REPRO from the shadow copy is in progress. This is also necessary to allow the later re-bind operation to succeed.

Figure 39. Sample scan and shadow forward recovery job
- Run IDCAMS REPRO to copy the shadow copy to the empty sphere, hereafter called the recovered sphere.
- Run IDCAMS BLDINDEX to rebuild the AIXs for the recovered sphere. This step is only needed if the shadow copy was created with non-reusable AIXs. It can be skipped if the shadow copy was created with reusable AIXs.
- Run IDCAMS SHCDS with the FRBIND keyword to re-bind to the recovered sphere all the retained locks that were unbound from the original sphere.
- Run IDCAMS SHCDS with the FRRESETRR keyword after the re-bind to reset the maintenance flag and enable the recovered sphere for use.
Chapter 6. Setting up your CICS TS environment

This chapter describes the tasks you will need to do to set up your CICS TS environment for CICSVR. If you were using CICSVR V2R3, many or all of the tasks may have already been done.

The following sections describe what you need to do to set up your CICS TS environment:

- Understand logging with CICS TS for OS/390 and z/OS
- Establish procedures to backup and recover your VSAM spheres
- Determine your backup and restore products
- Understand nonrecoverable versus recoverable
- Make the CICS System Definition (CSD) Data Set recoverable
- Decide on your naming conventions.

Understanding logging with CICS TS for z/OS and OS/390

All CICS system logging and journaling is controlled by the CICS log manager, which uses MVS system logger log streams to store its output.

CICS logging and journaling

CICS logging and journaling can be divided into these broad types of activity:

*System Logging*: CICS maintains a system log to support transaction backout for recoverable resources. System logging is implemented by CICS TS automatically, but you can define the log stream as DUMMY to inhibit the function. If you specify TYPE(DUMMY) for the system log, CICS TS runs without any transaction backout facilities and without any restart capability.

*Forward Recovery Logging*: CICS supports forward recovery logs for VSAM data sets. A forward recovery log is a general log stream managed by the MVS system logger. CICS writes the after images of changes made to a data set to a forward recovery log. Forward recovery logging is not automatic. You must specify that you want this facility for your files and also define the forward recovery log streams. You must specify this information in the ICF catalog if you want RLS mode; if you do not want RLS mode, you can specify this information in either the file resource definition or ICF catalog. For information on specifying file definitions, see "Defining recovery attributes by CICS resource definition" on page 67.

*Autojournaling*: CICS supports autojournaling of file control data and terminal control messages. Autojournaling is generally used for audit trails. CICSVR does not support forward recovery of a CICS TS VSAM data set using a CICS autojournal.

*User journaling*: CICS supports programming interfaces to let CICS applications write user-defined records to user journals, which are held on general log streams. CICSVR does not support forward recovery of a CICS TS VSAM data set using a user journal.

*Note*: Autojournaling and user journals play no part in CICS recovery.
The CICS log manager writes the data associated with these logging and journaling activities to MVS log streams, which the log manager controls as two types of log streams:

**System log streams**
These are used by the CICS log manager and the CICS recovery manager for unit of work (UOW) recovery purposes. Each system log is unique to a CICS region and cannot be merged with any other system log.

**General log streams**
These are used by the CICS log manager for all other types of logging and journaling.

**Understanding MVS log streams**
Coupling facility log streams reside in the coupling facility and in either a data space in the IXGLOGR address space or in staging data sets. DASD-only log streams reside in a data space in the IXGLOGR address space and in staging data sets.

All log streams that are needed by CICS must be defined to the MVS system logger before CICS can use them. You can either define log streams explicitly, or you can let CICS create them dynamically when they are first used. To enable CICS to create log streams dynamically, you first define model log streams to the MVS system logger. To define explicit log streams and model log streams, use the MVS IXCMIAPU utility.

For information about the coupling facility and defining log structures, refer to **z/OS MVS Setting Up a Sysplex**. For information about defining coupling facility log structures and MVS log streams for use by CICS Transaction Server, refer to **CICS System Definition Guide**.

**Defining system log streams**
You must define a system log if you want to preserve data integrity in the event of unit of work failures and CICS failures. A system log is mandatory for the following situations:

- The backout of recoverable resources changed by failed units of work.
- Cold starts, to enable CICS to restore units of work that were shunted at the time of the shutdown.
- Warm restarts, to enable CICS to restore the region to its state at the previous shutdown, including units of work that were shunted at the time of the shutdown.
- Emergency restarts, to enable CICS to perform the following tasks:
  - Restore the region to its state at the previous shutdown, including units of work that were shunted at the time of the shutdown.
  - Recover units of work that were in-flight at the time of the CICS failure and perform backout of recoverable resources that were updated before the failure.

CICS log manager connects to its log stream automatically during system initialization, unless it is defined as TYPE(DUMMY) in a CICS JOURNALMODEL resource definition.

**Defining forward recovery log streams**
VSAM data sets that are defined as recoverable files must have associated forward-recovery logs. In the event of physical failure or corruption, CICSVR can reapply the updates.
Specify forward recovery attributes in either the Integrated Catalog Facility (ICF) or in the CICS file resource definitions. For more information on specifying recovery attributes, see "Defining recovery attributes for RLS and non RLS files" on page 85.

**Note:** Define a data set as recoverable if you want forward recovery logging. Neither CICS nor VSAM provides any support for forward recovery logging for a nonrecoverable data set.

### Defining the log of logs

The log of logs is an MVS log stream created by CICS TS that contains records that are written each time a file is opened or closed. At file-open time, a tie-up record is written that identifies the following names:

- File
- Underlying VSAM data set
- Forward recovery log stream
- CICS region that performed the file open

At file-close time, a tie-up record is written that identifies the following names:

- File
- CICS region that performed the file close

The log of logs helps CICSVR maintain an index of log data sets. CICSVR scans the log of logs and saves information needed for recovery in the RCDS.

You should define a log-of-logs log stream that is shared among all CICS regions in the Parallel Sysplex®. This must be explicitly defined, because CICS does not support model log streams for dynamic creation of this log stream. You should also define a journal model resource definition that references the log-of-logs log stream.

For more information about logging in CICS Transaction Server, refer to CICS System Definition Guide.

### Defining recovery attributes by CICS resource definition

To use CICSVR to recover your data, you must specify the correct recovery attributes when you define your files to CICS. You can use the file resource definition if you do not plan to open the VSAM data set in RLS mode (by specifying RLS=NO as a system initialization parameter). If you use RLS mode (by specifying RLS=YES as a system initialization parameter), see "Understanding VSAM RLS" on page 85 for information on how to define recovery attributes for VSAM data set accessed in RLS mode.

**RDO Format**

CEDA DEFINE FILE(name) GROUP(groupname)

.  .
DSNAME(data set name)

.  .
RECOVERY(NONE|BACKOUTONLY|ALL)
FWDRECOVLOG(NO|1–99)
BACKUPTYPE(STATIC|DYNAMIC)

.
RECOVERY(NONE | BACKOUTONLY | ALL)

Defines the recovery requirements for the file. For CICS TS, this parameter is valid only for files defined with RLSACCESS(NO). For files that are accessed in RLS mode, you must specify the recovery parameters with the data set definition in the ICF catalog. For more information about RLS recovery parameters, refer to CICS Transaction Server for OS/390 Release Guide.

NONE

Causes CICS not to write before-images or after-images. CICSVR cannot forward recover or back out such a file. Do not specify the FWDRECOVLOG option if you specify RECOVERY(NONE).

BACKOUTONLY

Causes CICS to record images for backout only. CICSVR cannot forward recover such a file. If you specify this option, do not specify the FWDRECOVLOG option.

ALL

Requests before-images (for backout) to be recorded on the system log, and after-images (for forward recovery) to be recorded on the log specified in the FWDRECOVLOG parameter. You can specify the system log (log 01), or other log (logs 02–99) for forward-recovery logging. Before-images are always recorded on the system log.

Records written to the FWDRECOVLOG are totally independent of the automatic logging options that might be set.

RECOVERY(ALL), with the FWDRECOVLOG parameter, provides a way of separating the requirements of CICSVR from automatic logging. More information, which is not available with automatic logging, is recorded using FWDRECOVLOG. The forward-recovery log will contain images that are not compatible with previous releases of CICS. RECOVERY(ALL) and the use of the FWDRECOVLOG parameter are the options you should use. They cause CICS to record images for forward recovery and backout.

Note: For CICSVR purposes, use the RECOVERY(ALL) option for all VSAM data set types.

FWDRECOVLOG(NO | 1–99)

Specifies the log ID that the after-images for forward recovery are written to. CICS uses this value only if RECOVERY(ALL) is specified.

Specify the same recovery options and the same forward-recovery log ID for all CICS files that point to the same VSAM sphere. Whenever CICS opens a file for update, it checks that all other files that have been opened for update for the VSAM sphere have the same recovery options and forward-recovery log ID. Refer to CICS Recovery and Restart Guide for more information about what happens if CICS finds a mismatch.

BACKUPTYPE(STATIC | DYNAMIC)

Defines the eligibility of files for backup-while-open processing.

STATIC

Defines the file as not being eligible for backup-while-open processing.

DYNAMIC

Causes the file to be eligible for backup-while-open processing.

You can set other options of the RDO entry for whatever you require. Refer to CICS Resource Definition Guide for more information.
Establishing procedures to backup and recover your VSAM spheres

It is very important that you establish procedures to backup your VSAM spheres that you need to protect on a regular basis. CICSVR uses the backup copy of your VSAM sphere as the starting point of the forward recovery.

Deciding how often to make backups

Set up a backup strategy for your VSAM spheres. The sections that follow discuss daily versus nondaily backups.

Making daily backups of your VSAM spheres

If you make daily backups of your VSAM spheres, then if one of them is damaged, CICSVR can easily recreate it using the backup. Since the backup is recent, CICSVR only needs to apply the changes that were made that day to bring the VSAM sphere back to the exact state before the data was lost. You can speed up forward recovery processing even more by using CICSVR change accumulation. See “Setting up CICSVR change accumulation” on page 52 for more information on change accumulation.

Making nondaily backups of your VSAM spheres

If you do not make daily backups of your VSAM spheres, CICSVR can still easily recreate it using the backup. CICSVR needs to apply more changes, perhaps several days’ worth of transactions, to bring the VSAM sphere back to the exact state before the data was lost. You can significantly speed up forward recovery processing for nondaily backups by using CICSVR change accumulation. See “Setting up CICSVR change accumulation” on page 52 for more information on change accumulation.

Deciding when to make additional backups

Even though you make regular backups, there may be other times that you need to make a backup of your VSAM spheres. Here are several examples of when you should make a new backup of your VSAM sphere:

- If you change the physical characteristics of a VSAM sphere, such as the control interval (CI) size, or the length or location of a key field.
- If you successfully recover your VSAM sphere.
- If you are using CICSVR VSAM batch logging and your batch job receives a 3999 condition code.

Deciding how to group your backups

If you use CICSVR CA, IBM recommends that you create CA groups containing VSAM spheres that have the same backup requirements.

- First, schedule backups for all the VSAM spheres in a CA group at the same time.
- Second, when all the backups have completed successfully, rerun the CA batch job for that CA group. See “Setting up CICSVR change accumulation” on page 52 for more information.

Use this process for all the CA groups you have defined. This process prevents CICSVR from reading all the records in the CA data set and not applying any of them because they occurred prior to the backup.
Scheduling backups to minimize your CICS TS down time

Your CICS TS online system may be running 24 hours a day or it may only be running during certain hours. You should schedule backups so that they do not affect your online CICS TS users.

If you have CICS applications that depend on data sets being open for update over a long period of time, deciding when to schedule backups can be tricky. Normally, you cannot make a backup of the data set while the data set is open. Thus, if a failure occurs that requires forward recovery, all updates that have been made to the data set since it was opened must be recovered. This means that you must keep all forward recovery logs that have been produced since the data set was opened. A heavily used data set that has been open for update for several days or weeks may need much forward recovery.

The backup-while-open (BWO) facility, together with other system facilities and products, allows you to make a backup copy of a VSAM data set while it remains open for update. Then, only the updates that have been made since the last backup copy was taken need be recovered. This could considerably reduce the amount of forward recovery that is needed. For more information on taking a backup copy of a VSAM data set while it is open for update, see “Understanding the backup-while-open (BWO) facility” on page 80.

Determining your backup and restore products

You can use any utility to backup your VSAM spheres but there are significant benefits to using the IBM DFSMShsm or DFSMSdss storage management products instead of another backup product. The benefits are discussed in the following sections.

Using DFSMShsm as your backup utility

If your VSAM spheres are managed by DFSMS/MVS, the best option is to use DFSMShsm as your backup utility.

CICSVR provides complete data set forward recovery automation if DFSMShsm backups are used. When you perform a complete recovery or a forward recovery, CICSVR displays the latest DFSMShsm backup for your sphere. If you do not want to use the latest backup, you can get a list of all the DFSMShsm logical backups for that sphere and select the backup that you want from this list.

The CICSVR server address space is not required to use DFSMShsm. But, if it is activated (for example, so you can use the CICSVR VSAM Batch Logging feature), you must perform some additional tasks to set up DFSMShsm. See “Chapter 3. Activating the CICSVR server address space” on page 17 for information on how to setup DFSMShsm when you activate the CICSVR server address space.

DFSMShsm automates the backup of your VSAM spheres using aggregate backup and recovery support (ABARS). When the NetView File Transfer Program is installed on any two nodes, it can be used to transfer copies of aggregate backup files by running a batch job or using NetView File Transfer Program input panels. For information on performing disaster backup, see z/OS DFSMShsm Storage Administration Guide.
Using DFSMSdss as your backup utility

If you cannot use DFSMShsm, then the next best option is to use DFSMSdss if you have activated the CICSVR server address space. The CICSVR server address space is required if you want to use DFSMSdss as your backup utility.

CICSVR provides complete data set forward recovery automation if the DFSMSdss COPY or DUMP function is used to backup your VSAM spheres. When you perform a complete recovery or a forward recovery, CICSVR displays the latest DFSMSdss backup for your sphere. If you do not want to use the latest backup, you can get a list of all the DFSMSdss backups (logical copies, logical dumps) for that sphere and select the backup that you want from the list.

You must run a special DFSMSdss COPY or DUMP job to tell DFSMSdss to create backups for use by CICSVR. DFSMSdss notifies the CICSVR server address space every time a CICSVR backup is made. CICSVR stores the backup information in the RCDS. This enables CICSVR to know about DFSMSdss backups.

DFSMSdss does not provide an automated method for backing up your VSAM spheres so you need to create batch jobs that can be regularly submitted with a production planning system such as OPC/ESA. For information on using the DFSMSdss DUMP or COPY function, refer to z/OS DFSMSdss Storage Administration Guide or z/OS DFSMSdss Storage Administration Guide.

The next two sections describe how to use DFSMSdss COPY or DFSMSdss DUMP to tell DFSMSdss to create special backups for CICSVR.

Using DFSMSdss COPY

You can use the DFSMSdss COPY command to make copies of VSAM data sets for use by CICSVR. Figure 40 shows how to use the CICSVRBACKUP and the RENAME(**,CICSVR.**) keywords to notify DFSMSdss that the copies are for use by CICSVR.

```
//JOB6 JOB accounting information,REGION=nrrrK
//STEP1 EXEC PGM=ADDRSSU
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
COPY DATASET(  
   INCLUDE(USER1.**) /* FILTER ON DS/ST LEV USER1 */  
   OUTDYNAM((339001),(339002),(339003)) /* DYNAM ALLOC VOLS */  
   RENAMEU(**,CICSVR.**)  
   CICSVRBACKUP
) /*
```

Figure 40. DFSMSdss data set copy for use with CICSVR

This example is different than an ordinary DFSMSdss COPY job. The INCLUDE and the OUTDYNAM keywords operate like the normal DFSMSdss keywords causing all data sets with the high-level qualifier USER1 that are in the standard order of search to be copied to the target volumes that are labeled 339001, 339002, and 339003. Since CICSVRBACKUP is specified, RENAMEU(**,CICSVR.**) does not cause the INCLUDE data sets to be renamed with the CICSVR high level qualifier. Instead, CICSVR provides DFSMSdss with a new name for each copy using the naming convention prefix.DSDSNAMExThhmmss, where:

- The prefix is the CICSVR_DSNAMxPREFX defined in the IGDSMSxx PARMLIB member.
The yyyy is the year and ddd is the Julian day.
The hh is the hours, mm is the minutes, and sss is the seconds and tenths of a second.

You do not need to remember the names of the backup copies that DFSMSdss creates. Instead, when you use the CICSVR dialog, you only need to know the name of the original VSAM sphere and you can select the desired DFSMSdss backup copy from a list of backups. To view all the backup copies for a VSAM sphere from the CICSVR dialog:
- Select a VSAM sphere that you have created a DFSMSdss copy for from the CICSVR VSAM sphere list.
- Press F5 for forward recovery only.
- Put the cursor in the backup time field and press F4 to list all the backups for the selected VSAM sphere.

All the backup copies of the spheres will be listed, including the DFSMSdss backups.

Note: The RENAMEU keyword must be specified as "RENAMENU(**,CICSVR,**)
for DFSMSdss to create backup copies for CICSVR.

Using DFSMSdss DUMP
If you want to use DFSMSdss DUMP, you must create a batch job and set it up so that its regularly submitted with a production planning system such as OPC/ESA. Your output data set name must be unique each time the job is run so that you can maintain multiple backup copies of your VSAM data sets. Three methods of providing a unique output data set name are discussed:
- Using a batch REXX procedure to modify and submit the JCL
- Using a started task
- Using a generation data group (GDG)

Using a batch REXX procedure to generate a unique output data set name:
Figure 41 shows a sample of how to put date and time in the JCL to generate unique names for the output data set each time the job is run.

```
//JOB7 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//DSS EXEC PGM=ADRDSSU
//DD1 DD UNIT=3590,VOL=SER=TAPE04,
//   LABEL=(1,SL),DISP=(NEW,CATLG),
//   DSN=BACKUP.A&DATE.A&TIME
//SYSPRINT DD SYSOUT=*-*
//SYSIN DD *-
   DUMP OUTDD(DD1) -
   DS(INCL(USER1.DATASET1)) -
   CICSVRBACKUP
/*
```

Figure 41. Sample JCL to put date and time in the JCL and execute DFSMSdss DUMP

Figure 42 on page 73 and Figure 43 on page 73 show two samples of how to invoke a batch REXX procedure to modify and submit the JCL in Figure 41.

Figure 42 on page 73 calls the REXX procedure in Figure 44 on page 74 to replace the &DATE and &TIME values in Figure 41 with the real date and time values. You
will need to customize the input and output DD names, library, and member names in the REXX procedure.

```
//JOB8   JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//TSO    EXEC PGM=IKJEFT01,PARM='JCL1'
//SYSPROC DD DSN=USER1.CLIST,DISP=SHR
//SYSTSIN DD DUMMY
```

*Figure 42. Sample JCL to start REXX procedure, JCL1 (input/output files in the proc)*

*Figure 43* calls the REXX procedure in *Figure 45 on page 76* to replace the &DATE and &TIME values in *Figure 41 on page 72* with the real date and time values. This is similar to *Figure 43*, but the input and output DD names, library, and member names are specified in the JCL instead of the REXX procedure. You will need to customize the input and output DD names, library, and member names in the JCL.

```
//JOB9   JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//TSO    EXEC PGM=IKJEFT01,PARM='JCL2'
//SYSPROC DD DSN=USER1.CLIST,DISP=SHR
//DDI    DD DSN=USER1.CNTL(DFDSS),DISP=SHR
//DDO    DD DSN=USER1.CNTL(TEMP),DISP=SHR
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN DD DUMMY
```

*Figure 43. Sample JCL to start REXX procedure, JCL2 (input/output files in the JCL)*

For both samples, the REXX procedure must be placed in the dataset allocated to SYSPROC.

Here is the REXX procedure, JCL1 (input and output files in the procedure):
/*REXX: **************************************************************/
/* */
/* This REXX procedure will get system date and time and replace */
/* &DATE and &TIME in the input member with current date and time */
/* and write all changed and unchanged lines to the output member. */
/* */
/* Input is taken from the dataset specified in variable dsni and */
/* member specified in inmem. */
/* */
/* Output is written to the dataset specified in variable dsno and */
/* member specified in outmem. */
/* */
/* After conversion the output is submitted. */
/* */
/* The indd and outdd names are used to allocate the datasets to */
/* ddnames. They are freed in the end of this procedure. */
/* */
/* Customize the names below to get the correct input and output. */
/* */
/* ******************************************************************/
/* DATASETNAMES */
/* ******************************************************************/
/* dsni = "USER1.CNTL" /* INPUT LIBRARY */
/* dsno = "USER1.CNTL" /* OUTPUT LIBRARY */
/* indd = 'DDI' /* input ddname */
/* outdd = 'DDO' /* output ddname */
/* ******************************************************************/
/* MEMBERS */
/* ******************************************************************/
/* inmem = 'DFDSS' /* Input JCL */
/* outmem = 'TEMP' /* Output JCL */
/* ******************************************************************/
/* Allocate datasets */
/* ******************************************************************/
"ALLOC DSN(" || dsni || "," || inmem || ")" FILE(" || indd || ")"
"ALLOC DSN(" || dsno || "," || outmem || ")" FILE(" || outdd || ")"
/* Read all lines */
/* ******************************************************************/
"EXECIO * DISKR" indd "1 (FINIS STEM IN.)"

Figure 44. Sample REXX procedure JCL1 for dynamic allocation (Part 1 of 2)
Here is the REXX procedure, JCL2 (input and output files in the JCL):

```rexx
/* Get system date and time. */
/* Date is saved as yymmdd */
/* Time is saved as hhmms */

date = date('S')
date = substr(date,3)

time = time('N')
time = substr(time,1,2) || substr(time,4,2) || substr(time,7,2)

/* Check all lines for &DATE or &TIME and replace them with the */
/* saved values in date and time. */

o = 0 /* init */
do i = 1 to in.o /* do for all input */
o = o + 1 /* one more line for output */
out.o = in.i /* copy to output */

ind = pos('&DATE',out.o) /* any date? */
do while ind > 0 /* do while date found */
out.o = insert(' ',out.o,ind,1) /* insert a blank since */
out.o = overlay(date,out.o,ind) /* overlay is one longer */
ind = pos('&DATE',out.o) /* any more? */
end /* */

ind = pos('&TIME',out.o) /* any time? */
do while ind > 0 /* do while time found */
out.o = insert(' ',out.o,ind,1) /* insert a blank since */
out.o = overlay(time,out.o,ind) /* overlay is one longer */
ind = pos('&TIME',out.o) /* any more? */
end /* */
end /* */

/* Write all lines to output dataset */

"EXECIO = DISKW" outdd "(FINIS STEM OUT.)"

/* Free the allocated datasets */

"FREE FILE("indd | "")"
"FREE FILE("outdd | "")"

/* Submit the changed job and return */

"SUBMIT = dsno | "("outmem | ")""
return
```

Figure 44. Sample REXX procedure JCL1 for dynamic allocation (Part 2 of 2)
/* This REXX procedure will get system date and time and replace &DATE and &TIME in the input member with current date and time and write all changed and unchanged lines to the output member.*/

/* Input is taken from the dataset specified in the JCL for the ddname given in variable indd, in this example DDI*/

/* Output is written to the dataset specified in the JCL for the ddname given in variable outdd, in this example DDO*/

/* After conversion the output is submitted and the member used for that is in variable outmem, in this example TEMP*/

/* Customize the names below to get the correct input and output.*/

outmem = 'TEMP' /* output member given in JCL */
indd = 'DDI' /* input ddname */
outdd = 'DDO' /* output ddname */

/* Read all lines */

"EXECIO * DISKR" indd "1 (FINIS STEM IN.)"

/* Get system date and time. */
/* Date is saved as yymmdd */
/* Time is saved as hhmmss */

date = date('S')
date = substr(date,3)

time = time('N')
time = substr(time,1,2) || substr(time,4,2) || substr(time,7,2)

Figure 45. Sample REXX procedure JCL2 for dynamic allocation (Part 1 of 2)
The job submitted by the above procedures looks like:

```plaintext
/*Check all lines for DATE or TIME and replace them with the saved values in date and time.*/

o = 0 /* init */
do i = 1 to in.o /* do for all input */
o = o + 1 /* one more line for output */
out.o = in.i /* copy to output */

ind = pos('DATE',out.o) /* any date? */
do while ind > 0 /* do while date found */
  out.o = insert(' ',out.o,ind,1) /* insert a blank since */
  out.o = overlay(date,out.o,ind) /* overlay is one longer */
  ind = pos('DATE',out.o) /* any more? */
end /* */

ind = pos('TIME',out.o) /* any time? */
do while ind > 0 /* do while time found */
  out.o = insert(' ',out.o,ind,1) /* insert a blank since */
  out.o = overlay(time,out.o,ind) /* overlay is one longer */
  ind = pos('TIME',out.o) /* any more? */
end /* */

end

/* Write all lines to output dataset */

"EXECIO * DISKW" outdd "(FINIS STEM OUT.)"

/* Get datasetname for output */
x = LISTDSI(outdd "FILE")

/* Submit the changed job and return */

"SUBMIT '" || sysdsname || "(" || outmem || ")'"
return
```

Figure 45. Sample REXX procedure JCL2 for dynamic allocation (Part 2 of 2)
Using a started task to generate unique output data set names: The job could also be run as a started task with this JCL.

```plaintext
//JOB10 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//DSS EXEC PGM=ADDRSSU
//DD1 DD UNIT=3480, VOL=SER=TAPE04,
//   LABEL=(1,SL), DISP=(NEW,CATLG),
//   DSN=BACKUP.A001103.A141213
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
// DUMP OUTDD(DD1) -
// DS(INCL(USER1.DATASET1)) -
// CICSVRBACKUP
/*
```

Figure 46. Job submitted by REXX procedure JCL1 and JCL2

Using a GDG to generate unique output data set names: The user can also define a GDG for each dump job and specify the generation data sets (GDS) as the output dump data set name. The GDG must specify scratch and 255 as the maximum number of data sets.

```plaintext
//DFDSS PROC
//DSS EXEC PGM=ADDRSSU
//DD1 DD UNIT=3480, VOL=SER=TAPE04,
//   LABEL=(1,SL), DISP=(NEW,CATLG),
//   DSN=BACKUP.A&LYYMMDD.A&LHHMMSS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DSN=SYS1.SYSIN(DFDUMP),DISP=SHR
```

Figure 47. Sample JCL to start a task

JES2 will replace the &LYYMMDD and &LHHMMSS with local date and time. Note that this will only work if the job is started as a started task.

```plaintext
//JOB11 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//STEP1 EXEC PGM=IDCAMS
//GDGMOD DD DSNNAME=GDG01,DISP=(,KEEP),
//   SPACE=(TRK,(0)),UNIT=DISK, VOL=SER=VSER03,
//   DCB=(RECFM=FB,BLKSIZE=7892,LRECL=100)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD */
/*
```

Figure 48. Sample JCL to define a GDG

```plaintext
DEFINE GENERATIONDATAGROUP -
   (NAME(GDG01) -
    EMPTY -
    SCRATCH -
    LIMIT(255) )
/*
```

Figure 49 on page 79 shows a sample of how to create a GDG entry.
Note: CICSVR will not support more than 255 GDS data sets for a GDG because non-SMS GDG data sets beyond the 255th data set are not cataloged. CICSVR requires that all associated data sets be cataloged.

Understanding nonrecoverable versus recoverable resources

With VSAM file recovery, there are two different types of resources, ones that are not recoverable and ones that are recoverable.

**Nonrecoverable Data Sets:** Nonrecoverable data sets are those for which updates are committed immediately, and therefore cannot be backed out. These data sets can be read and updated by both CICS and batch jobs but neither CICS nor VSAM provides any support for forward recovery logging for a nonrecoverable data set.

**Recoverable Data Sets:** A recoverable data set is one that participates in syncpoint activity and is eligible for changes to be backed out in the event of a transaction failure. When several changes are made to a recoverable data set from within a unit-of-work (UOW), these changes are either all committed, or all backed out, together.

The ICF catalog recovery option allows you to specify the following requirements:

- No recovery is required; the sphere is not recoverable.
- Backout is required; the sphere is recoverable.
- Backout and forward recovery are required; the sphere is recoverable.

If you also require a forward recovery capability for recoverable data sets, you must specify in the ICF catalog the name of the MVS log stream that CICS uses as the forward recovery log. This log stream name must match the name of a log stream that is defined to the MVS system logger.

If you specify forward recovery for a data set (LOG=ALL) VSAM may not check that you have specified the required forward recovery MVS log stream name. However, CICS performs this check when it opens a file against a data set that has an ICF catalog entry that specifies forward recovery. CICS fails the open request if the ICF catalog forward recovery attributes are inconsistent.

The section that follows focuses on recoverable data sets, that is, data sets that can be recovered using CICSVR. The following topics are described:

- Backup-while-open (BWO)
- VSAM RLS
- CICS System Definition data

---

//JOB12 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//DSS EXEC PGM=ADRDSSU
//GDGDD1 DD DSNAME=GDG01(+1),DISP=(NEW,CATLG),
//   UNIT=3480,VOL=SER=TAPE04,
//   LABEL=(1,SL)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
   DUMP OUTDD(GDGDD1) - DS(INCL(USER1.DATASET1)) -
   CICSVRBACKUP
*/

Figure 49. Sample JCL to use the GDG
Understanding the backup-while-open (BWO) facility

Many CICS applications depend on their data sets being open for update over a long period of time. Normally, you cannot make a backup of the data set while the data set is open. Thus, if a failure occurs that requires forward recovery, all updates that have been made to the data set since it was opened must be recovered. This means that you must keep all forward recovery logs that have been produced since the data set was opened. A heavily used data set that has been open for update for several days or weeks may require many log records to be applied during forward recovery.

The backup-while-open (BWO) function allows DFSMSdss to make a backup when applications are running in continuous operation while the data set is open for update with full data integrity of copied data. This is only feasible for CICS VSAM file control data sets for which CICS creates forward recovery logs. Only the updates that have been made since the last backup copy was taken need to be recovered. This could significantly reduce the amount of forward recovery that is needed. Long running transactions, automated teller machines, and continuously available applications require the database to be running when the backup is being taken.

DFSMSdss is an optional feature of z/OS and is the primary data mover. When used with supporting hardware, DFSMSdss also provides concurrent copy capability. Concurrent copy lets you copy or backup data while that data is being used; even when control-area and control interval splits and data set additions (new extents or add-to-end) are occurring for VSAM key sequenced data sets.

Eligible data sets for backup-while-open processing

You can use the BWO facility for the following data sets:

- Data sets that reside on SMS-managed storage and that use an ICF catalog.
- VSAM data sets that are accessed by CICS file control, and for the CICS system definition (CSD) data set. ESDS, KSDS, RRDS, and VRRDS are supported. ESDS and KSDS are supported with and without alternate indexes (AIXs).

Backup-while-open is supported at the VSAM sphere level; thus, you cannot make backup-while-open copies of some sphere components and not others. The first data set that is opened for update against a VSAM base cluster sets the backup-while-open eligibility for the sphere. This includes base clusters that are accessed through a VSAM path key. For example, if the first data set is eligible for backup-while-open processing, CICS fails the file-open operation for a following data set that is opened for update against that cluster and that is not defined as eligible for backup-while-open processing.

You can make BWO volume backups if all data sets that are open for update on the volume are eligible for BWO. The backup-while-open facility does not support physical dumps.

For detailed information on BWO, refer to CICS Recovery and Restart Guide.

Recovering VSAM spheres with nonreusable AIXs

If your VSAM sphere was restored from a copy made using the backup-while-open facility, ensure that no nonreusable AIXs are in the upgrade set. CICSVR checks the upgrade set of data sets that are restored from backup-while-open copies and issues this message if it finds AIXs.
Forward-recovery request for data set dddd is rejected because it is a backup-while-open copy containing alternate indexes in the upgrade set.

To forward recover such a data set, after the restore, use the access method services (AMS) ALTER or DELETE command to remove or delete the AIXs from the upgrade set. After forward recovery has completed successfully, you can re-create the upgrade set by rebuilding the AIXs using the AMS BLDINDEX command.

**Operation of the BWO facility**

To use the backup-while-open facility with CICSVR, define DFSMSdss as the method DFSMShsm uses to move data. You can specify this using the DFSMShsm SETSYS command:

```
SETSYS DATAMOVER(DSS)
```

When DFSMSdss performs a logical dump, it examines the backup-while-open flags in the integrated catalog facility (ICF) catalog and reports on the success or failure of the backup. For example, if a VSAM control interval (CI) or control area (CA) split is in progress at the start of a backup, the backup is not made. Also, if a split occurs during a backup-while-open backup, or a split is still in progress after a backup, the backup is logically discarded by DFSMShsm. If you are using DFSMSdss without DFSMShsm, examine the reports that DFSMSdss produces to find out if the logical dump has been successful. But, if you use the concurrent copy function with BWO, CI or CA splits will not affect the contents of the copy that is being made. See [Using the concurrent copy function with CICSVR](#) on page 85 for more information.

Because the process of making backups can lead to increased I/O activity, consider making backups during periods of low activity. This would avoid an effect on the response time of the online application and reduce the probability of CI or CA splits.

For more information about DFSMSdss refer to [z/OS DFSMSdss Storage Administration Guide](https://www.ibm.com). For more information about DFSMShsm, refer to [z/OS DFSMShsm Storage Administration Reference](https://www.ibm.com).

The master terminal operator (MTO) must be aware of any new environment. Before the availability of the backup-while-open facility, to backup a VSAM sphere, the MTO would quiesce all transactions that would update the data set, notify the users, and close and deallocate the necessary files. These procedures are not needed with the backup-while-open facility.

With the backup-while-open facility, the MTO receives a message if an application attempts to open a file that is flagged as back level. This check occurs regardless of whether you specify the BACKUPTYPE option DYNAMIC or STATIC for the file. A back-level data set is one that has been:

* Identified by the ICF catalog as corrupt.

If this situation occurs, the following messages are issued, and the MTO must restore and recover the VSAM sphere to ensure data integrity:

DFHFC5801A  
* applid File OPEN has failed for VSAM data set. The BWO values in the ICF catalog indicate that the data set needs to be restored and forward recovered. Data set 'dsname'.

DFHFC5806  
* applid File OPEN failed. DFHFCAT returned an error response from a BWO action on a VSAM data set. File 'filename' data set 'dsname'.
- Restored from a backup copy but not recovered.
- Forward recovered, but the forward-recovery operation has not completed successfully.

In either of these cases, the following messages are issued, and the MTO must decide if the backup copy had been restored but not recovered, or if recovery has failed:

DFHFC5802A
applid File OPEN has failed for VSAM data set. The BWO values in the ICF catalog indicate that data set needs to be restored and forward recovered. Data set ‘dsname’.

DFHFC5806

If recovery was not run, you can run CICSVR to recover the VSAM data set. If the messages are issued because of the failure of CICSVR recovery, the MTO must restore and recover the associated VSAM sphere to ensure data integrity.

If an application attempts to open a file that is eligible for backup-while-open processing in the ICF catalog, but is not defined as such in the CICS file definition, the following actions occur:
- The ICF catalog is set to show that the data set is not eligible for backup-while-open processing.
- The recovery point is updated to the time held in the MVS system clock (see “BWO and the recovery point time” on page 84).
- This warning message is issued at the MTO console:

DFHFC5809
applid File OPEN warning. BACKUPTYPE attributes conflict with BWO values defined in ICF catalog. BWO values have been updated. File ‘filename’ data set ‘dsname’.

- The MTO can use this command to ask whether the first file to be opened against a data set name was indicated as eligible for backup-while-open processing:
  
  CEMT INQUIRE DSN

For more information, refer to CICS Supplied Transactions.

Note: When you are recovering a restored data set that was backed up using the backup-while-open facility, record not found and duplicate record messages are not reported, but statistics indicating these conditions will appear in the report. You can analyze and act on these messages if you have provided an error exit for this purpose. The CICSVR error exit is invoked for all types of VSAM errors.

How you request BWO
There are two ways in which files can be defined as eligible for BWO:
- By specifying the BWO option for the data set using access method services
- By specifying the BWO option on the CICS file resource definition in the CSD

Specifying BWO using access method services: If your data sets are accessed in RLS mode, the BWO option must be defined in the ICF catalog. To define BWO in the ICF catalog, use the BWO parameter on the access method services DEFINE CLUSTER statement. You can specify the BWO parameter as follows:

TYPECICS The data set is eligible for BWO in CICS.
NO
The data set is not eligible for BWO.

TYPEIMS
The data set is eligible for BWO in IMS, but CICS treats this as NO.

TYPEOTHER
The data set is eligible for BWO, but CICS treats this as NO.

If you specify BWO(TYPECICS), you must also specify LOG(ALL) and a forward recovery log stream name, LOGSTREAMID(logstream_name).

If you omit the BWO parameter from the DEFINE statement, by default it is UNDEFINED in the ICF catalog, and the BWO attribute from the CICS file resource definition is used.

BWO(TYPECICS) is the equivalent of BACKUPTYPE(DYNAMIC) in a CICS file resource definition. All other values, including UNDEFINED, are treated by CICS as the equivalent of BACKUPTYPE(STATIC) in a CICS file resource definition. For simplicity, the CICS terms BACKUPTYPE(DYNAMIC) and BACKUPTYPE(STATIC) are used unless it is necessary to specifically mention the access method services BWO parameters.

BWO for CSD: The BWO options for the CSD are taken from the ICF catalog if they are defined there, and the system initialization parameters (CDSBKUP, CSDRECOV, and CSDFRLOG) are ignored.

Specifying BWO on CICS file resource definitions: You define a file as eligible for BWO with the BACKUPTYPE attribute on a CICS file resource definition in the CSD.

If you specify the BACKUPTYPE(DYNAMIC) option, the file is defined as eligible for BWO when the data set is opened. This definition applies only to the base cluster in a VSAM sphere. If a VSAM data set is accessed through one or more AIX, the path definitions for the AIXs must also be defined as eligible for BWO processing. You must also specify the RECOVERY(ALL) and the FWDRECOVLOG(nn) options to request forward recovery support.

BACKUPTYPE(STATIC), the default, defines a file as not eligible for BWO. In this case, if DFSMShsm is to back up a data set, all CICS files currently open for update against that data set must be closed before the backup can start.

All files that are opened against the same VSAM base cluster must have the same BACKUPTYPE value. That value is established by the first file opened against the cluster; it is stored in the CICS data set name block (DSNB) for the cluster. If the value for a subsequent file does not match, the file-open operation fails.

BWO for CSD: To use BWO for the CSD file, specify the CDSBKUP=DYNAMIC system initialization parameter. Also specify CSDRECOV=ALL and CSDFRLOG=nn to request forward recovery support.

BWO and systems administration
The systems administrator must decide which VSAM user data sets are eligible for BWO, and then set up the appropriate operating procedures for taking the BWO backup copies and for forward recovery. These procedures should include:

- How to forward recover a data set by using the BWO backup copy, the forward recovery log, and CICSVR to bring the data set to a point of consistency. Users must not have access to the file during the recovery process.
- How to forward recover a data set that may have been damaged while allocated to CICS. This operation may require backout of partially committed units of work during CICS emergency restart, after forward recovery has been done.
The procedures are simpler when using BWO than when not, because:

- Backups can be taken more frequently, so there are fewer forward recovery logs to manage. This also reduces the amount of processing that is required to forward recover the data set.
- The point from which forward recovery should start is recorded in the ICF catalog. CICSVR uses this value to automate this part of the forward recovery process. This recovery point is saved with the backup copy and subsequently replaced in the ICF catalog when the backup copy is restored.
- During data set restore and forward recovery processing, CICS does not allow files to be opened for the same data set.

**BWO data set security**
CICS must have RACF ALTER authority for all data sets that are defined as BACKUPTYPE(DYNAMIC), because CICS needs to update the BWO attributes in the ICF catalog. The authority must apply either to the data set or to the ICF catalog in which the data set is cataloged. For information on defining RACF ALTER authority, refer to *CICS RACF Security Guide*.

**BWO and the recovery point time**
To recover a VSAM sphere, CICSVR must have the following information:
- The data set to associate with each record on the logs
- The point to start recovery from

Each data set after-image record on the log is associated with a file name but many files might be associated with the same data set. When a file is opened, the association between the file and the data set is recorded on the log by a tie-up record (TUR). For backups made without the backup-while-open facility, CICSVR uses this TUR to apply the log records to the correct data sets.

For backups made using the backup-while-open facility, CICSVR need not process a log from file-open time. Here, TURs for all open files are regularly written on the log during activity-keypoint processing. To reduce the number of TURs if the activity keypoint frequency is high, CICS ensures that there is at least a 30-minute separation between sets of TURs on the log. The recovery point is a time that can be converted to a position on a forward-recovery log. It is also the point when CICSVR forward recovery should start for VSAM data sets eligible for backup-while-open processing. Recovery of the data set requires only the records that are written after that position. Thus, CICSVR can ignore all previous records.

The recovery point is stored in the ICF catalog. It is set when the first file is opened for update against the data set, and it is updated during activity-keypoint processing and when the file is closed.

The recovery point is not the time of the current keypoint, because there might still be some uncommitted log records that have not been forced. Instead, it is the time of the start of the last keypoint that wrote a complete set of TURs and that completed earlier than the oldest uncommitted write to a forward-recovery log.

**Notes:**
1. Only one new recovery point is calculated during an activity keypoint. It is used for all data sets that are open for update and eligible for backup-while-open processing. A long-running task that updates a data set that uses the backup-while-open facility affects the forward recovery that is needed for all data sets.
2. If you disable activity keypointing in your system (by specifying AKPFREQ=0 in your SIT), backup-while-open support is seriously affected because no more
TURs are written and the recovery point is not updated after backup-while-open. Forward recovery of a data set that is eligible for backup-while-open processing must occur from the time that the data set was first opened for update.

CICSVR extracts the recovery point time of the restored backup from the RCDS or ICF catalog. It is not necessary or recommended to specify a STARTTIME keyword in your recovery run.

Using the concurrent copy function with CICSVR

The concurrent copy function provides concurrent and unrestricted access to CICS VSAM data sets during the backup process. The concurrent copy function provides a point-in-time copy of your CICS data sets while maintaining full update access to the VSAM spheres being copied, once the concurrent copy operation is initialized. This function lets you offer continuous availability of your CICS VSAM spheres.

Understanding VSAM RLS

Record level sharing (RLS) is a VSAM function provided by DFSMS/MVS Version 1 Release 3 that is exploited by the CICS Transaction Server. RLS lets VSAM data sets be shared, with update capability between many applications in many CICS regions. With VSAM RLS, CICS regions that share VSAM data sets can reside in one or more MVS images within an MVS parallel sysplex. Whether or not the data set is accessed as an RLS data set is determined at OPEN time by the ACB macro setting or by a JCL parameter. To enable RLS, use the MACRF=RLS ACB macro setting.

CICSVR can recover VSAM data sets accessed in RLS-mode; however, CICSVR VSAM batch logging cannot perform logging for VSAM data sets that are accessed in RLS mode. See Chapter 4, Setting up CICSVR VSAM batch logging on page 27 for more information.

Defining recovery attributes for RLS and non RLS files

This section describes how to define the recovery attributes for files managed by CICS file control.

You specify recovery options, including forward recovery, either in the integrated catalog facility (ICF) catalog or in the CICS file resource definition, as follows:

1. If your VSAM data sets are accessed by CICS in RLS mode, define the recovery attributes in the ICF catalog. The recovery options on the CICS file resource definitions (RECOVERY, FWDRECOVLOG, and BACKUPTYPE) are ignored if the file definition specifies RLS mode.

2. If your VSAM data sets are accessed by CICS in non-RLS mode, define the recovery attributes in either the file resource definition or the ICF catalog. If you use the ICF catalog to define attributes for data sets accessed in non-RLS mode, the ICF catalog entry recovery attributes override the CICS file resource definition.

**Note:** CICS uses the ICF catalog definitions only when you specify RLS=YES as a system initialization parameter. If you specify RLS=NO, recovery attributes are always taken from the CICS file definition.

**VSAM files accessed in RLS mode:** If you specify file definitions that open a data set in RLS mode, you must specify the recovery options in the ICF catalog. The recovery options on the CICS file resource definitions (RECOVERY, FWDRECOVLOG, and BACKUPTYPE) are ignored if the file definition specifies RLS access.
The VSAM parameters LOG and LOGSTREAMID on the access methods services DEFINE CLUSTER and ALTER commands, determine recoverability for the entire sphere. Locating these recovery attributes in the ICF catalog enforces the same attributes, for all CICS regions in the sysplex, for all the files opened against a given sphere.

**LOG((NONE | UNDO| ALL))**

Specifies the type of recovery required for the VSAM sphere. Specify the LOG parameter for data sets that are to be used by CICS in RLS mode:

- **NONE**
  The sphere is not recoverable.

- **UNDO**
  The sphere is recoverable. CICS must maintain system log records for backout purposes.

- **ALL**
  The sphere is recoverable for both backout and forward recovery. CICS must maintain system log records (as for UNDO) and forward recovery log records. If you specify LOG(ALL), also specify LOGSTREAMID to indicate the name of the forward recovery log. You should use LOG(ALL) for files that you want to recover using CICSVR.

**Note:** Forward recovery support is available for recoverable files only—you cannot have forward recovery without backout recovery.

**LOGSTREAMID(log_stream_name)**

Specifies the name of the MVS log stream to be used for forward recovery log records when LOG(ALL) is defined. Note that IDCAMS does not check for the presence of the LOGSTREAMID during DEFINE processing. CICS checks for an MVS log stream name when it attempts to open a data set in RLS mode. The open fails if the log stream is not defined and cannot be created dynamically.

**BWO(TYPECICS | TYPEIMS | TYPEOTHER | NO)**

TYPECICS specifies that CICS BWO support is required. This is the equivalent of BACKUPTYPE(DYNAMIC) in the CSD. The other values refer to non-CICS support.

If you omit the LOG parameter when you define your VSAM data sets, recovery is assumed to be UNDEFINED, and the data set cannot be opened in RLS mode. You can also set the UNDEFINED status explicitly by specifying NULLIFY(LOG).

For information about these ICF parameters, refer to [z/OS DFSMS Access Method Services](https://www.ibm.com/support/knowledgecenter/SST46U_2.2.0/com.ibm.cluster.dfsmsdf5.doc/vsdms(VSAD).html).

**Inquiring on recovery attributes:** You can use CEMT, or EXEC CICS, INQUIRE FILE, and INQUIRE DSNNAME commands to determine the recovery options that are specified for files and data sets. The INQUIRE FILE command shows the options from the CICS file definition until the first file for the data set is opened. If the options are obtained from the ICF catalog when the first file is opened, the ICF catalog values are returned. The INQUIRE DSNNAME commands returns values from the VSAM base cluster block (BCB). However, because base cluster block (BCB) recovery values are not set until the first open, if you issue an INQUIRE DSNNAME commands before the first file is opened, CICS returns NOTAPPLIC for RECOVSTATUS.
**VSAM files accessed in non-RLS mode:** Because the VSAM files are accessed in non-RLS mode, define recovery attributes in the file resource definition. You can specify support for both forward and backward recovery for VSAM files using the RECOVERY and FWDRECOVLOG options. You define the type of data set backup that you want using the BACKUPTYPE parameter.

**RDO Format**

CEDA DEFINE FILE(name) GROUP(groupname)

  DSNAME(data set name)

  RECOVERY(NONE | BACKOUTONLY | ALL)
  FWDRECOVLOG(NO | 1–99)
  BACKUPTYPE(STATIC | DYNAMIC)

**RECOVERY(NONE | BACKOUTONLY | ALL)**

Defines the recovery requirements for the file. For CICS TS, this parameter is valid only for files defined with RLSACCESS(NO). For files that are accessed in RLS mode, you must specify the recovery parameters with the data set definition in the ICF catalog. For more information about RLS recovery parameters, refer to *CICS Transaction Server for OS/390 Release Guide*.

- **NONE**
  Causes CICS not to write before-images or after-images. CICSVR cannot forward recover or back out such a file. Do not specify the FWDRECOVLOG option if you specify RECOVERY(NONE).

- **BACKOUTONLY**
  Causes CICS to record images for backout only. CICSVR cannot forward recover such a file. If you specify this option, do not specify the FWDRECOVLOG option.

- **ALL**
  Requests before-images (for backout) to be recorded on the system log, and after-images (for forward recovery) to be recorded on the log specified in the FWDRECOVLOG parameter. You can specify the system log (log 01), or other log (logs 02–99) for forward-recovery logging. Before-images are always recorded on the system log.

  Records written to the FWDRECOVLOG are totally independent of the automatic logging options that might be set.

  RECOVERY(ALL), with the FWDRECOVLOG parameter, provides a way of separating the requirements of CICSVR from automatic logging. More information, which is not available with automatic logging, is recorded using FWDRECOVLOG. The forward-recovery log will contain images that are not compatible with previous releases of CICS. RECOVERY(ALL) and the use of the FWDRECOVLOG parameter are the options you should use. They cause CICS to record images for forward recovery and backout.

  **Note:** For CICSVR purposes, use the RECOVERY(ALL) option for all VSAM data set types.

**FWDRECOVLOG(NO | 1–99)**

Specifies the log ID that the after-images for forward recovery are written to. CICS uses this value only if RECOVERY(ALL) is specified.
Specify the same recovery options and the same forward-recovery log ID for all CICS files that point to the same VSAM sphere. Whenever CICS opens a file for update, it checks that all other files that have been opened for update for the VSAM sphere have the same recovery options and forward-recovery log ID. Refer to CICS Recovery and Restart Guide for more information about what happens if CICS finds a mismatch.

**BACKUPTYPE(STATIC | DYNAMIC)**

Defines the eligibility of files for backup-while-open processing.

- **STATIC**
  - Defines the file as not being eligible for backup-while-open processing.

- **DYNAMIC**
  - Causes the file to be eligible for backup-while-open processing.

You can set other options of the RDO entry for whatever you require. Refer to CICS Resource Definition Guide for more information.

**RLS locks**

For files opened in RLS mode, VSAM maintains a single central lock structure using the lock-assist mechanism of the MVS coupling facility. This central lock structure provides sysplex-wide locking at a record level—control interval (CI) locking.

The locks for files accessed in non-RLS mode, the scope of which is limited to a single CICS region, are CI locks. However, the CICS enqueue domain also holds a record lock for the record accessed within the CI.

For coupling facility data tables that are updated under the locking model, the coupling facility data table server stores the lock with its record in the CFDT. As in the case of RLS locks, storing the lock with its record in the coupling facility list structure that holds the coupling facility data table ensures sysplex-wide locking at record level.

For both RLS and non-RLS recoverable files, CICS releases all locks on completion of a unit of work. For recoverable coupling facility data tables, the locks are released on completion of a unit of work by the CFDT server.

**Active and retained states for locks:** CICS supports active and retained states for locks.

When a lock is first acquired, it is an active lock. It remains an active lock until successful completion of the unit of work, until it is released, or until it is converted into a retained lock if the unit of work fails or for a CICS or SMSVSAM failure:

- **If a unit of work fails,** RLS VSAM or the CICS enqueue domain continues to hold the record locks that were owned by the failed unit of work for recoverable data sets, but converted into retained locks. Retaining locks ensures that data integrity for those records is maintained until the unit of work is completed.
- **If a CICS region fails,** locks are converted into retained locks to ensure that data integrity is maintained while CICS is being restarted.
- **If an SMSVSAM server fails,** locks are converted into retained locks (with the conversion being carried out by the other servers in the sysplex, or by the first server to restart if all servers have failed). This means that a unit of work (UOW) that held active RLS locks will hold retained RLS locks following the failure of an SMSVSAM server.
Converting active locks into retained locks not only protects data integrity. It also ensures that new requests for locks owned by the failed unit of work do not wait, but instead are rejected with the LOCKED response.

**Forward recovery log failure**
If a general log that is being used by CICS as a forward recovery log (or for automatic logging) fails, CICS ensures that any files that were using that log are closed, and that appropriate messages are issued to warn operators.

For example, if there is a hardware failure that affects a forward recovery log, CICS prevents access to all the files for all the data sets that are using that log. How CICS prevents access to the files depends on whether they are open in RLS or non-RLS mode:

**RLS access-mode files**
CICS uses its RLS quiesce mechanism to initiate the closure actions across the sysplex.

**Non-RLS access-mode files**
A CICS region detecting the failure closes only its own open ACBs. It also marks the data set as unavailable.

The associated messages recommend that a new backup of the affected data sets should be made. This is because the failure of the forward recovery log means that it cannot be used if forward recovery becomes necessary. New backup copies of the actual data sets, which at this stage are unaffected by the failure, ensure that the failed logs are redundant.

When backup copies have been made, processing can resume with different forward recovery logs.

**Making the CICS system definition (CSD) data set recoverable**

The CSD VSAM data set is unique in that it is a CICS system data set that is managed by CICS file control. If you want to forward recover the CSD data set, you must specify the recovery attributes in the system initialization table (SIT).

```plaintext
CSDFRLOG=(NO|1–99)
CSDRECOV=(NONE|ALL|BACKOUTONLY)
CSDBKUP=(STATIC|DYNAMIC)
```

Refer to the *CICS System Definition Guide* for more information on defining the CSD.

**Deciding on your naming conventions**

Give unique names to the backup and log data sets so that you can easily distinguish them from the operational data sets.

If you have no suitable naming convention, create one for all future backup and log data sets. MVS data set names can be 1–44 characters, divided into qualifiers of up to 8 characters. Here are some examples:

**clusternamex.BUDate.Ttime**
For backup data sets of VSAM base cluster clustername.

**logstreamname.COPY1.DDate.Ttime**
For the first copy of MVS log stream logstreamname. The date and time
might not be the time that archiving occurs; instead, they could be the time
that CICS opened the log that is being archived.

Here are examples of data set names that follow these naming conventions:

PAYROLL.BASE.BU.D01159.T235205  A backup data set

PAYROLL.LOGSTREAM.COPY1.D01200.T232145  A copy of an MVS log stream
Chapter 7. Using CICSVR with CICS TS at your disaster recovery site

This chapter describes the tasks you need to do to prepare your primary site and your remote site for disaster recovery using CICS TS.

Preparing your primary site

The sections that follow describe what you need to do at your primary site to prepare for remote site disaster recovery. Important information should be copied from your primary site and sent to the remote site.

The following topics are described:
- Making backups of your VSAM spheres
- Running your change accumulation batch jobs to update the RCDS
- Running scan to update the RCDS
- Using the RCDS utility to export your RCDS to the remote site
- Sending your shadow copies to the remote site
- Using the log stream copy utility to copy your log streams

Making backups of your VSAM spheres

In “Chapter 6. Setting up your CICS TS environment” on page 63, you learned about the importance of establishing a backup and recovery strategy, as well as using DFSMShsm, DFSMSdss, or another product for backing up your VSAM spheres. If you are setting up your remote disaster recovery site, you need to make sure backup copies of your VSAM spheres are made and regularly sent to your remote site.

If you use DFSMShsm, the recommended method of disaster backup is to use aggregate backup and recovery support (ABARS). Refer to z/OS DFSMShsm Storage Administration Guide for more information on ABARS.

If you use DFSMSdss, the recommended method of disaster backup is to use the logical data set DUMP command and filter on the data set name. Logical data set dump processing allows you to back up data sets and to restore them to unlike devices. Refer to z/OS DFSMSdss Storage Administration Guide for more information on DFSMSdss and disaster recovery.

If you use another product for backing up your data sets, refer to the appropriate publication for more information on disaster recovery.

Regularly send the backup copies of your VSAM spheres to your remote site.

Running your change accumulation batch jobs to update the RCDS

In “Chapter 5. Setting up CICSVR for CICS TS” on page 37, you learned about setting up CA batch jobs and running the appropriate job immediately after a backup is taken for any of the spheres in the CA group. Running the CA batch job for VSAM spheres that have been backed up ensures that the RCDS is kept up-to-date with the latest backup and CA information. See Figure 34 on page 53 for a sample change accumulation batch job.
Running scan to update the RCDS

In ["Chapter 5. Setting up CICSVR for CICS TS" on page 37](#) you learned about the log of logs scan utility.

If you are setting up your remote disaster recovery site, you need to make sure the scan job is set up and is submitted on a regular basis so your RCDS is kept up-to-date. See [Figure 32 on page 50](#) for a sample scan job you can run at regularly scheduled times. Then, you need to use the CICSVR RCDS utility to export your RCDS to the remote site. This topic is discussed in the next section.

Using the RCDS utility to export your RCDS to the remote site

Because CICSVR saves important information needed for recovery in the RCDS, you need to capture this information regularly and send it to your remote site. You must use the CICSVR RCDS utility to copy the information in the RCDS. The RCDS utility copies pertinent information from the RCDS linear VSAM data set to a sequential access method (SAM) data set. You should use a production planning system, like OPC/ESA, to submit the export job and send the SAM RCDS to your remote recovery site at regularly scheduled times.

Use the RCDS EXPORT command to tell CICSVR to copy pertinent information in the RCDS to the SAM data set that is specified on the DWWCOPY1 DD statement. [Figure 50](#) shows a sample RCDS EXPORT job to create a SAM data set that can be exported to your remote site.

```bash
//RCDSEX1 JOB ACCOUNTING INFORMATION,REGION=OM 01
//EXPORT EXEC PGM=DWWGJCDS 02
//STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR 03
//DWWCOPY1 DD DSN=USER1.DWWCOPYR, 04
// UNIT=3590,VOL=SER=TAPE04, 05
// LABEL=(1,SL),DISP=(NEW,KEEP) 06
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR 07
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR 08
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR 09
//DWWMSG DD SYSOUT=* 10
//DWWPRINT DD SYSOUT=* 11
//DWWIN DD * 12
//      RCDS EXPORT 13
/*
```

**Figure 50. Sample RCDS EXPORT job to copy the RCDS for the remote recovery site**

Here is a description of each of the numbered statements:

**Line** | **Explanation**
---|---
01 | JOB statement defines a CICSVR job
02 | EXEC statement defines a CICSVR job step. Specifies that program DWWGJCDS is to be run. The prefix DWW always refers to a component of CICSVR.
03 | STEPLIB DD statement. Specifies the name of the CICSVR load library, for example, DWW.SDWWLOAD, used for this job step.
04–06 | DWWCOPY1 DD statement. Specifies the SAM data set, USER1.DWWCOPYR, in which the EXPORTed RCDS information will be placed. USER1.DWWCOPYR is allocated on a standard label 3590 volume TAPE04 and kept.
07–09 DWWCON1–DWWCON3 defines the three recovery control data sets. If you are running the CICSVR server, you must use the hlq.slq that was defined in the IGDSMSxx PARMLIB member for these data sets. Pertinent information from these data sets will be copied into the data set that is specified on the DWWCOPY1 DD statement.

Note: CICSVR uses three identical RCDS data sets to provide fault tolerance and improved reliability.

10 DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a SYSOUT data set.

The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.

11 DWWPRINT defines the output data set that contains the reports produced by CICSVR. This is usually defined as a SYSOUT data set.

12 DWWIN DD statement. Specifies the data set that contains the CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream (as shown).

13 RCDS EXPORT command. Specifies that all the necessary RCDS information will be copied to the USER1.DWWCOPYR data set.

Sending your shadow copies to the remote site

In "Chapter 5. Setting up CICSVR for CICS TS" on page 37, you learned about creating shadow copies of critical VSAM files to reduce the time it takes to forward recover them. If you have created shadow copies of any of your VSAM spheres, you should send the shadow copies to your remote recovery site at regularly scheduled times.

Using the log stream copy utility to copy your log streams

VSAM sphere update activity is recorded in the MVS log stream so you should capture this information regularly and send it to your remote site. You must use the CICSVR log stream copy utility to copy your MVS log stream to a sequential access method (SAM) data set; you cannot use IDCAMS or CICSVR archive to copy an MVS log stream. You should use a production planning system, like OPC/ESA, to submit the log stream copy job and send the SAM log stream copy data sets to your remote recovery site at regularly scheduled times.

Use the LOGSTREAMCOPY command to tell CICSVR to copy an MVS log stream to the SAM data sets specified on the DWWCOPYx DD statement, where x can be 1 to 9. Figure 51 on page 94 shows a sample LOGSTREAMCOPY job to make one copy of the MVS log stream.
Here is a description of each of the numbered statements:

1. The program to be run is DWWLC. The prefix DWW always refers to a component of CICSVR.

2. This statement supplies the name of the CICSVR load library.

3. DWWLOAD is optional and defines the alternate load library to STEPLIB, once DWWLC has been loaded.

4. This is the DWWCOPY1 DD statement. It specifies the SAM data set USER1.DWWCOPYL, in which the log stream copy will be placed. USER1.DWWCOPYL is allocated on a standard label 3590 volume TAPE04 and kept.

5. DWWCON1–DWWCON3 defines the three recovery control data sets. If you are running the CICSVR server, you must use the hlq.slq that was defined in the IGDSMSxx PARMLIB member for these data sets.

6. DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a SYSOUT data set.

The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.

7. DWWPRINT defines the output data set that contains the reports produced by CICSVR. This is usually defined as a SYSOUT data set.

8. DWWIN defines the data set that contains the CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream.

9. The log stream copy utility processes an MVS log stream called CICSVR.MVSLOG. The log records produced by CICS and by CICSVR (for VSAM batch logging) are copied from CICSVR.MVSLOG. One copy of the log is requested; the log records are copied and added to the end of the DWWCOPY1 data set.

Once set up, this job can be run over and over again to update the copy of the log stream.
Figure 52 shows the log stream copy report, followed by descriptions of the fields in the report.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS log stream name</td>
<td>The name of the MVS log stream.</td>
</tr>
<tr>
<td>First time GMT</td>
<td>The first time recorded on the MVS log stream in GMT format.</td>
</tr>
<tr>
<td>Last time GMT</td>
<td>The last time recorded on the MVS log stream in GMT format.</td>
</tr>
<tr>
<td>First time LOCAL</td>
<td>The first time recorded on the MVS log stream in local time format.</td>
</tr>
<tr>
<td>Last time LOCAL</td>
<td>The last time recorded on the MVS log stream in local time format.</td>
</tr>
<tr>
<td>First block number</td>
<td>The first block number on the MVS log stream.</td>
</tr>
<tr>
<td>Last block number</td>
<td>The last block number on the MVS log stream.</td>
</tr>
<tr>
<td>Number of blocks read</td>
<td>The number of blocks read from the MVS log stream.</td>
</tr>
<tr>
<td>Number of CICS blocks read</td>
<td>The number of CICS-specific blocks read from the MVS log stream.</td>
</tr>
<tr>
<td>Number of blocks copied</td>
<td>The number of blocks copied to the sam copy of the MVS log stream.</td>
</tr>
<tr>
<td>Type of copy</td>
<td>The type of copy requested (CICSVR ONLY or ALL).</td>
</tr>
<tr>
<td>Output log name(s)</td>
<td>The name of the SAM copy data set.</td>
</tr>
</tbody>
</table>

CICSVR - MVS LOG STREAM COPY UTILITY

DATE : 01/06/07  TIME : 12:51:13  PAGE : 1

STATISTICS FROM THE MVS LOG STREAM COPY UTILITY

MVS LOG STREAM NAME : CICSVR1.MVSLOG
FIRST TIME GMT : 01.159 12:39:09
LAST TIME GMT : 01.159 12:39:44
FIRST TIME LOCAL : 01.159 13:39:09
LAST TIME LOCAL : 01.159 13:39:44
FIRST BLOCK NUMBER : 5965196
LAST BLOCK NUMBER : 11890088
NUMBER OF BLOCKS READ : 148
NUMBER OF CICS BLOCKS : 148
NUMBER OF BLOCKS COPIED: 148
TYPE OF COPY : ALL
OUTPUT LOG NAME(S) : CICSVR1.MVSLOG.COPY

Figure 52. Log stream copy report
On rare occasions, you may want to make a copy of a portion of a log stream. You can do this using the STARTTIME and STOPTIME keywords or the STARTBLKID and STOPBLKID keywords.

With the STARTTIME and STOPTIME keywords, CICSVR copies records from the time specified on the STARTTIME keyword up to the time specified on the STOPTIME keyword. Figure 53 shows a sample LOGSTREAMCOPY command with STARTTIME and STOPTIME.

```
LOGSTREAMCOPY NAME(CICSVR1.MVSLOG) -
SELECT(ALL) -
COPIES(3) -
STARTTIME(96159/07:30:00,GMT) -
STOPTIME(96249/07:30:00,GMT) -
MOD
```

Figure 53. Sample LOGSTREAMCOPY command with STARTTIME and STOPTIME

The LOGSTREAMCOPY command in Figure 53 tells CICSVR to make three copies of the MVS log stream. These copies will be made to the data sets that are specified on the ddnames DWWCOPY1, DWWCOPY2, and DWWCOPY3. Records will be copied to the end of the data sets that are specified in the DWWCOPYn ddnames. All records are copied from the MVS log stream between the start and the stop time that are specified in the command. You should only use this method if you need a one-time-copy for test data. You should not use it for copying consecutive portions of the log because the time values are not precise enough to prevent gaps in the log.

**Attention:** With the STARTBLKID and STOPBLKID keywords, CICSVR copies records from the STARTBLKID to the STOPBLKID. Use the CICS DFHJUP program to find out the MVS block identifier you can use as the STARTBLKID and the STOPBLKID. This method is not easy to use and is not recommended. If you require more information on DFHJUP, refer to *CICS Operations and Utilities Guide*.

**Format of the log copy**

Log copy starts and ends with writing a special record. *Table 3* describes the layout of first record.

*Table 3. Layout of the first log copy record.*

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant (104)</td>
<td>fixed (32)</td>
<td>Length of the rest of this record</td>
</tr>
<tr>
<td>constant (‘&gt;DWW’)</td>
<td>char (4)</td>
<td>Type of record</td>
</tr>
<tr>
<td>constant (1)</td>
<td>fixed (31)</td>
<td>Version number</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
<tr>
<td>blkid</td>
<td>char (8)</td>
<td>Blockid of the first block read</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
<tr>
<td>gmttime</td>
<td>char (8)</td>
<td>GMT time of the first block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>localtime</td>
<td>char (8)</td>
<td>Local time of the first block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
Table 3. Layout of the first log copy record. (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logname</td>
<td>char (26)</td>
<td>MVS log name</td>
</tr>
<tr>
<td>*</td>
<td>char (22)</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Table 4 describes the layout of last record.

Table 4. Layout of last record

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant (104)</td>
<td>fixed (32)</td>
<td>Length of the rest of this record</td>
</tr>
<tr>
<td>constant ('DWW')</td>
<td>char (4)</td>
<td>Type of record</td>
</tr>
<tr>
<td>constant (1)</td>
<td>fixed (31)</td>
<td>Version number</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
<tr>
<td>blkid</td>
<td>char (8)</td>
<td>Blockid of the last block read</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
<tr>
<td>gmttime</td>
<td>char (8)</td>
<td>GMT time of the last block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>localtime</td>
<td>char (8)</td>
<td>Local time of the last block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>*</td>
<td>char (8)</td>
<td>Reserved</td>
</tr>
<tr>
<td>logname</td>
<td>char (26)</td>
<td>MVS log name</td>
</tr>
<tr>
<td>*</td>
<td>char (22)</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Table 5 describes the layout of all other records.

Table 5. Layout of all other records

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>fixed (32)</td>
<td>Length of the prefix(24) + length of the MVS log block read, i.e. length of the rest of this record</td>
</tr>
<tr>
<td>blkid</td>
<td>char (8)</td>
<td>Block ID of the MVS block read</td>
</tr>
<tr>
<td>gmttime</td>
<td>char (8)</td>
<td>GMT time of the MVS block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>localtime</td>
<td>char (8)</td>
<td>Local time of the last block read, time-of-day (TOD) format</td>
</tr>
<tr>
<td>block</td>
<td>char (*)</td>
<td>The MVS log block read</td>
</tr>
</tbody>
</table>

Note: All of the above records are written to a SAM file using the following spanned format:
Table 6 describes the spanned format for records using a SAM file.

**Table 6. Spanned format for records using a SAM file.**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>fixed (16)</td>
<td>Total length of this block, including this field</td>
</tr>
<tr>
<td>bb</td>
<td>fixed (16)</td>
<td>Block indicator</td>
</tr>
<tr>
<td>record</td>
<td>char (*)</td>
<td>A whole or part of one of the above records</td>
</tr>
</tbody>
</table>

The block indicator can be one of the following:
- **00**: The whole record is contained in this block
- **01**: First part of the record is contained in this block
- **02**: Last part of the record is contained in this block
- **03**: A middle part of the record is contained in this block

One of the reasons for this is that an MVS block can be up to 64K, but the largest SAM block that can be written is 32K.

### Preparing your remote site

The sections that follow describe what you need to do at your remote site to prepare for remote site disaster recovery. Important information sent from your primary site should be incorporated into the remote site. The following topics are described:

- Installing CICSVR at your Remote Disaster Recovery Site
- Saving the backups of your VSAM spheres
- Saving the log stream copies
- Saving the SAM RCDS
- Saving the shadow copies

### Installing CICSVR at your remote disaster recovery site

You must install CICSVR V3R1 at your remote disaster recovery site and, at a minimum, allocate an RCDS before you can use the CICSVR RCDS utility. See the [CICSVR V3R1 Program Directory](#) for CICSVR installation instructions. See "Creating and maintaining the RCDS" on page 42 for information on how to allocate an RCDS.

### Saving the backups of your VSAM spheres

The primary site will regularly send the remote recovery site backup copies of your VSAM spheres. Save several generations of backups to ensure that you can fully recover the lost data if an error occurs with one or more of the backup copies. In most cases, you only need to restore the latest backup copy when recovering a VSAM sphere.

If DFSMSdss made the backup using ABARS, install DFSMSdss at the disaster recovery site and use DFSMSdss to restore the backup. Refer to [z/OS DFSMSdss Storage Administration Guide](#) for more information on ABARS.

If you used DFSMSdss to backup your data sets, install DFSMSdss at the disaster recovery site. Therefore, CICSVR can restore the backups during the recovery process.
If you used another product for backing up your data sets, refer to the appropriate publication for more information on how to restore the backup.

**Saving the log stream copies**

The primary site should regularly send the remote recovery site SAM copies of the MVS log streams. Save several generations of log stream copies to ensure that you can fully recover the lost data if an error occurs with one or more of the backup copies. CICSRVR can use log stream copies in place of MVS log streams to forward recover a VSAM sphere.

**Saving the SAM RCDS**

The primary site should regularly send the remote site SAM copies of its RCDS. Save several generations of the SAM RCDS copies at the remote site. Do not IMPORT the copy into the remote site’s RCDS until you need to perform a recovery. When you run the RCDS IMPORT utility at the remote site, the information in the SAM copy from the primary site is merged with the information in the remote site’s RCDS which is a linear data set.

If the remote site’s RCDS becomes too large, you will need to delete it, redefine it, and then run IMPORT to fill it with the latest information from the primary site. See the "Creating and maintaining the RCDS" on page 42 for information on creating a new RCDS.

See Figure 54 for a sample RCDS IMPORT job that updates the remote recovery site’s RCDS with the information from the primary site. RCDS IMPORT will only work with SAM RCDS data sets that were EXPORTed using RCDS EXPORT.

```plaintext
//RCDSIM JOB (ACCT), 'USER', CLASS=A, NOTIFY=, MSGLEVEL=(1,1) 01
//IMPORT EXEC PGM=DWWGJCDS 02
//STEPLIB DD DSN=DWW.SDWWLOAD, DISP=SHR 03
//DWWCOPY1 DD DSN=USER1.DWWCOPYR, UNIT=3590, VOL=SER=TAPE 04
//DWWCON1 DD DISP=SHR, DSN=DWW.DWWCON1 05
//DWWCON2 DD DISP=SHR, DSN=DWW.DWWCON2 06
//DWWCON3 DD DISP=SHR, DSN=DWW.DWWCON3 07
//DWWMSG DD SYSOUT=* 08
//DWWPRINT DD SYSOUT=* 09
//DWIN DD * 10

RCDS IMPORT 11

/*

Chapter 7. Using CICSRVR with CICS TS at your disaster recovery site 99

Figure 54. Sample RCDS IMPORT job to update the remote recovery site’s RCDS

Here is a description of each of the numbered statements:

**Line**   **Explanation**

01   JOB statement defines a CICSRVR job.

02   EXEC statement defines a CICSRVR job step. Specifies that the program DWWGJCDS is to be run. The prefix DWW always refers to a component of CICSRVR.

03   STEPLIB DD statement. Specifies the name of the CICSRVR load library, for example, DWW.SDWWLOAD, used for this job step.

04–05   This is the DWWCOPY1 DD statement. It specifies the SAM data set, USER1.DWWCOPYR, from which the IMPORTed RCDS information will be
extracted and placed into the RCDS data sets specified on the
DWWCON1–DWWCON3 DD statements. USER1.DWWCOPYR is on a
standard label 3590 volume TAPE04.

06–08 DWWCON1–DWWCON3 defines the three recovery control data sets. If
you are running the CICSVR server, you must use the hlq.s1q that was
defined in the IGDSMSxx PARMLIB member for these data sets.

Note: CICSVR uses three identical RCDS data sets to provide fault
tolerance and improved reliability.

09 DWWMSG defines the output data set that contains the CICSVR
messages. This is usually defined as a SYSOUT data set.
The DCB parameters for this data set are RECFM=FBA and LRECL=133.
The block size can be provided on the DWWMSG DD statement and must
be a multiple of 133. The default is 27930.

10 DWWPRINT defines the output data set that contains the reports produced
by CICSVR. This is usually defined as a SYSOUT data set.

11 This is the DWWIN DD statement. It specifies the data set that contains the
CICSVR commands. You can either specify a sequential data set with
80-byte, fixed-length records, or include the CICSVR commands in-stream
(as shown).

12 This is the RCDS IMPORT command. It specifies that all the RCDS
information will be copied from the USER1.DWWCOPYR data set to the
RCDS data sets, specified on DWWCON1–DWWCON3 DD statements.

Saving the shadow copies

The primary site may be regularly sending the remote recovery site shadow copies
of its critical VSAM files. Save the shadow copies. If the original VSAM sphere is
damaged, you can use the shadow copy instead of the backup to forward recover
the VSAM sphere. [Replacing the original sphere with a shadow copy on page 62]
for instructions on how to recover from a shadow copy.
# Part 3. CICSVR and CICS V4

## Chapter 8. Setting up CICSVR for CICS V4

<table>
<thead>
<tr>
<th>Setting up your CICSVR ISPF environment</th>
<th>103</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocating data sets to your TSO session</td>
<td>103</td>
</tr>
<tr>
<td>Running the CICSVR ISPF dialog interface</td>
<td>104</td>
</tr>
<tr>
<td>Modifying an existing ISPF selection menu</td>
<td>104</td>
</tr>
<tr>
<td>Selecting the CICSVR main menu directly from TSO</td>
<td>105</td>
</tr>
<tr>
<td>Using the ISPF select service</td>
<td>106</td>
</tr>
<tr>
<td>Editing the CICSVR JCL skeleton</td>
<td>106</td>
</tr>
</tbody>
</table>

**Understanding the recovery control data set (RCDS)**

<table>
<thead>
<tr>
<th>Creating and maintaining your RCDS</th>
<th>108</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizing your RCDS</td>
<td>108</td>
</tr>
<tr>
<td>Creating your RCDS</td>
<td>109</td>
</tr>
<tr>
<td>Allocating the RCDS</td>
<td>109</td>
</tr>
<tr>
<td>Maintaining your RCDS</td>
<td>110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deciding which VSAM spheres to protect</th>
<th>112</th>
</tr>
</thead>
</table>

**Setting up your retention periods for your logs**

| Retention periods for archived logs | 113 |

**Setting up automatic archiving with CICS V4**

<table>
<thead>
<tr>
<th>Setting up the journal archive job skeleton</th>
<th>114</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the archive utility report</td>
<td>116</td>
</tr>
</tbody>
</table>

| Setting up the journal archive job skeleton with shadow forward recovery | 119 |

## Chapter 9. Setting up your CICS V4 environment

<table>
<thead>
<tr>
<th>Setting up CICS V4 for CICSVR</th>
<th>133</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Files</td>
<td>133</td>
</tr>
<tr>
<td>Defining Journals</td>
<td>134</td>
</tr>
<tr>
<td>Setting up journal archive data sets</td>
<td>135</td>
</tr>
<tr>
<td>Testing the automatic archiving setup</td>
<td>135</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Establishing procedures to backup and recover your VSAM spheres</th>
<th>135</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deciding how often to make backups</td>
<td>135</td>
</tr>
<tr>
<td>Making daily backups of your VSAM spheres</td>
<td>135</td>
</tr>
<tr>
<td>Making nondaily backups of your VSAM spheres</td>
<td>135</td>
</tr>
<tr>
<td>Deciding when to make additional backups</td>
<td>135</td>
</tr>
<tr>
<td>Deciding how to group your backups</td>
<td>136</td>
</tr>
<tr>
<td>Scheduling backups to minimize your CICS down time</td>
<td>136</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Determining your backup and restore products</th>
<th>136</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using DFSMSHsm as your backup utility</td>
<td>136</td>
</tr>
<tr>
<td>Using DFSMSdss as your backup utility</td>
<td>137</td>
</tr>
<tr>
<td>Using DFSMSdss COPY</td>
<td>137</td>
</tr>
<tr>
<td>Using DFSMSdss DUMP</td>
<td>138</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Understanding the backup-while-open (BWO) facility</th>
<th>145</th>
</tr>
</thead>
<tbody>
<tr>
<td>What data sets can use BWO</td>
<td>146</td>
</tr>
<tr>
<td>AIX</td>
<td>146</td>
</tr>
<tr>
<td>How you request BWO</td>
<td>146</td>
</tr>
<tr>
<td>BWO for CDS</td>
<td>147</td>
</tr>
<tr>
<td>BWO and Systems Administration</td>
<td>147</td>
</tr>
</tbody>
</table>
Chapter 8. Setting up CICSVR for CICS V4

This chapter describes the tasks you need to do to set up CICSVR for CICS V4. It includes the following sections:

- Set up your CICSVR ISPF environment
- Understand the recovery control data set
- Review CICSVR V2R3 to CICSVR V3R1 migration considerations
- Decide which VSAM spheres to protect
- Set up automatic archiving with CICS V4
- Set up CICSVR change accumulation
- Decide if you want to use shadow copies

You can also use CICSVR VSAM batch logging to log your batch changes to VSAM data sets that are not accessed in RLS mode. Then, you can use CICSVR to forward recover the updates made by the batch applications. For more information, see "Chapter 4. Setting up CICSVR VSAM batch logging" on page 27.

Setting up your CICSVR ISPF environment

The best way to use CICSVR is through the dialog interface. CICSVR dialogs run through ISPF, so you must set up an appropriate ISPF environment. If you are not familiar with ISPF dialogs, refer to z/OS ISPF Dialog Developer's Guide and Reference.

To set up your ISPF environment, perform these steps:
1. Allocate ISPF data sets to the TSO session.
2. Run the CICSVR ISPF dialog interface.
3. Edit the CICSVR JCL skeleton.

Allocating data sets to your TSO session

Table 7 shows the data sets that you must allocate to the TSO session to run the CICSVR ISPF dialog interface.

Table 7. ISPF dialog data sets

<table>
<thead>
<tr>
<th>ddname</th>
<th>CICSVR</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWWCON1</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWCON2</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWCON3</td>
<td>CICSVR recovery control data set</td>
<td>See Note 5.</td>
</tr>
<tr>
<td>DWWMSG</td>
<td>CICSVR message library</td>
<td>See Note 1.</td>
</tr>
<tr>
<td>DWWPRINT</td>
<td>CICSVR output library</td>
<td>See Note 2.</td>
</tr>
<tr>
<td>DWLLLlib</td>
<td>Load Library</td>
<td>See Note 4.</td>
</tr>
<tr>
<td>DWWSLIB</td>
<td>JCL skeleton library</td>
<td>See Note 4.</td>
</tr>
<tr>
<td>ISFILE</td>
<td>File tailoring output and your JCL skeleton</td>
<td>Your existing ISPF data set.</td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>Load library</td>
<td>See note 3.</td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>Message library</td>
<td></td>
</tr>
<tr>
<td>ISPPLIB</td>
<td>Panel library</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. ISPF dialog data sets (continued)

<table>
<thead>
<tr>
<th>ddname</th>
<th>CICSVR</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPPROF</td>
<td>User-session defaults, read/write tables</td>
<td>Your existing ISPPROF data set.</td>
</tr>
<tr>
<td>ISPSLIB</td>
<td>JCL skeleton library</td>
<td>See Note 3.</td>
</tr>
<tr>
<td>ISPTLIB</td>
<td>Read tables (default)</td>
<td></td>
</tr>
<tr>
<td>DWWDUMP</td>
<td>CICSVR diagnostic data set</td>
<td>See Note 6</td>
</tr>
<tr>
<td>DWWDMSG</td>
<td>CICSVR prolog and epilog data set</td>
<td>See Note 6</td>
</tr>
</tbody>
</table>

Notes:
1. DWWMDD defines the output data set that contains the CICSVR messages. Use a different data set name than the one used on this system by the CICSVR server address space.
   The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.
2. DWPPRINT is the output data set that contains the reports produced by CICSVR. Use a different data set name than the one used on this system by the CICSVR server address space.
3. Ensure that the library containing your own CICSVR JCL skeletons is allocated to both the ISPSLIB and ISPFILE DD statements.
4. Since CICSVR uses LMFIND to locate a member in ISPLLIB and ISPSLIB, only four data sets can be concatenated in these libraries. Since this may be a problem for some users, CICSVR supports alternate names: DWWDLLIB for ISPLLIB and DWWSLLIB for ISPSLIB. CICSVR will use DWWDLLIB in place of ISPLLIB and DWWSLLIB in place of ISPSLIB.
5. Do not use the RCDSs created by the Installation Verification Procedure (IVP). Instead, make sure the cleanup job has been run (DWWCLNUP) to delete the IVP data sets. Then review Migrating your RCDS from CICSVR V2R3 to CICSVR V3R1 on page 14 for instructions, or Creating and maintaining the RCDS on page 42
6. DWWDMSG and DWWDUMP define the output data sets that CICSVR writes diagnostic and abend information to. If these data sets are not allocated, this information will be unavailable.

Running the CICSVR ISPF dialog interface

This section discusses three different methods for invoking the CICSVR dialog:

- Modify an existing ISPF selection menu
- Select the CICSVR main menu directly from TSO
- Use the ISPF select service

Modifying an existing ISPF selection menu

You can invoke CICSVR simply by including CICSVR as an option on your existing Primary Option menu, or on another selection menu. Insert the statements that are marked on the right with an arrow (====) in Figure 55 on page 105.
Specify that you want to receive all write-to-programmer messages (including the message ID), at your terminal. This will help you to analyze any problems that might occur while setting up the CICSVR ISPF dialog interface. To do this, enter this TSO command:

```
PROFILE WTPMSG MSGID
```  

Before you can invoke CICSVR, you must allocate the CICSVR data sets. You can allocate these data sets through the TSO logon procedure, or by executing a command list (CLIST) after TSO logon. See "Appendix D. Sample CLIST (DWWCLIST)" on page 287 for a sample CLIST.

**Selecting the CICSVR main menu directly from TSO**

To select the CICSVR main menu directly from TSO, enter this TSO command:

```
ISPSTART PGM(DWWPM) NEWAPPL(DWW)
```  

Remember that you must allocate the CICSVR ISPF dialog interface data sets before running this command. If you select the CICSVR main menu directly from TSO, you cannot split the screen to run two CICSVR sessions.

Using this method to invoke the dialog means that Figure 56 on page 106 is the first ISPF panel displayed:

```
0 CICSVR - CICS VSAM Recovery  
```
Using the ISPF select service
Another method of starting the CICSVR ISPF dialog interface is to execute the SELECT command from a CLIST, or from a program. A sample CLIST (DWW.SDWWCNTL(DWWCLIST)) is provided on the product tape and is shown in "Appendix D. Sample CLIST (DWWCLIST)" on page 287. Modify DWWCLIST according to your site requirements. Refer to your ISPF manuals for more information about these methods.

Editing the CICSVR JCL skeleton
If you select option 5 from the main menu (Figure 56), this JCL skeleton secondary window appears (Figure 57 on page 107) and invokes the ISPF/PDF editor.
Here you can edit the CICSVR JCL skeleton information to conform to your organization’s standards. Use F3 to leave the editor and return to the CICSVR main menu.

You must edit the JCL skeleton and modify the following information:
- TESTGFS&CJOBCHAR, the job card (do not remove the &CJOBCHAR variable)
- STEPLIB, the CICSVR load library
- DWWCON1, DWWCON2, and DWWCON3, the CICSVR RCDSs

<table>
<thead>
<tr>
<th>Understanding the recovery control data set (RCDS)</th>
</tr>
</thead>
</table>

The CICSVR RCDS contains essential information for recovering your VSAM spheres. The RCDS contains the following information:
- CICSVR ISPF dialog interface default values
- Archived logs
- DFSMSHsm and DFSMSdss backups
- Change accumulation data sets
- CICSVR VSAM batch logging
- Shadows

This information is stored in three identical linear VSAM data sets. Three data sets are used to reduce the possibility of data loss. CICSVR regularly checks to make sure the three data sets are consistent.
The information in the RCDS is updated:

- Through the CICSVR ISPF dialog
- By using archive with CICS V4
- Automatically by CICSVR, with change accumulation
- Automatically by CICSVR, with VSAM batch logging
- Automatically by doing a shadow forward recovery
- Automatically when the CICSVR server address space is active

You can reduce the likelihood of your RCDS becoming full by deregistering old information from the RCDS. You can set up automatic deregistration for your archived logs using the CICSVR ISPF dialog interface. In addition, you can instantly deregister an archived log using the CICSVR dialog interface.

Deregistration is triggered by running scan or automatically by the CICSVR server address space. Once deregistered, CICSVR automatically reuses the freed space in the RCDS; you never need to manually recover free space. For more information on deregistering, refer to *CICSVR V3R1 User’s Guide and Reference*.

### Creating and maintaining your RCDS

The steps you need to take to create and maintain the RCDSs are slightly different depending on if the CICSVR server address space is enabled or not. If you have enabled the CICSVR server address space, see [Creating and maintaining the RCDSs](#) on page 20 for information related to the CICSVR server.

In general, you can use the following information for creating and maintaining the RCDSs. The following sections describe how to:

- Size your RCDS
- Create your RCDS
- Maintain your RCDS

### Sizing your RCDS

Use this formula to estimate the size of your RCDS:

\[
\text{RCDS size} = n\text{vsam} \times n\text{switch} \times n\text{days} \times 8192 \text{ bytes}
\]

Where:

- \( n\text{vsam} \) is the number of VSAM data sets
- \( n\text{switch} \) is the average number of times the logs are switched in a day
- \( n\text{days} \) is the number of days archived log information is kept

For example, if you have an average of 100 VSAM data sets with update activity whose CICS logs are switched once a day and the information is kept for 30 days, the approximate size of the RCDS is calculated as follows:

\[
100 \times 1 \times 30 \times 8192 = 24,576,000 \text{ bytes}
\]

You can estimate the size of your RCDS as 24 megabytes.

IBM recommends that you make one of the recovery control data sets slightly smaller than the other two. If the smaller one becomes full, CICSVR will issue messages warning you that your smaller RCDS is full, then will continue using the larger RCDSs. This warning notifies you that your RCDS needs maintenance and lets you take action before all three RCDSs are full and halts recovery or CICSVR VSAM batch logging. See [Maintaining your RCDS](#) on page 110 for more information.
Creating your RCDS

The three RCDSs are linear VSAM data sets. They contain identical copies of essential information CICSVR needs to recover your VSAM spheres. To reduce the possibility of losing all three copies of the RCDSs, IBM recommends that you create each RCDS on separate volumes, disk controllers, and channels. In addition, your VSAM spheres should be on different volumes, disk controllers and channels from your RCDSs. These precautions prevent a single hardware failure from making all three RCDSs unusable, or from making the RCDS and the VSAM sphere to be recovered unusable. CICSVR regularly checks the consistency of data between the three data sets. If one or two of the RCDSs are temporarily lost (for example, the unavailability of a disk path), CICSVR automatically copies the data from the remaining RCDS the next time the other data sets are available.

Figure 58 shows sample AMS commands to define three RCDSs.

```
//ALLOC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DEFINE CLUSTER -  
  (NAME('DWW.DWWCON1')  
   VOLUMES(TSOL01)  
   CYLINDERS(60 20)  
   LINEAR  
   SHR(3 3))  

DEFINE CLUSTER -  
  (NAME('DWW.DWWCON2')  
   VOLUMES(TSO007)  
   CYLINDERS(60 20)  
   LINEAR  
   SHR(3 3))  

DEFINE CLUSTER -  
  (NAME('DWW.DWWCON3')  
   VOLUMES(TSOL02)  
   CYLINDERS(30 20)  
   LINEAR  
   SHR(3 3))  
/*
```

Figure 58. AMS Commands to Define the CICSVR RCDS

Allocating the RCDS

The ddnames for the RCDSs are DWWCON1, DWWCON2, and DWWCON3. Figure 59 provides an example of allocating the RCDS in the CICSVR recovery job.

```
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR  
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR  
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR
```

Figure 59. Allocating the RCDS for the CICSVR run

Figure 60 on page 110 provides an example of allocating the RCDS to the CICSVR ISPF dialog interface:
Maintaining your RCDS
This section describes how to repair your RCDS if one or more RCDSs become full or damaged.

If your RCDS is full: In this situation, you have received the DWW1605I DWWCONx is full informational message and you want to increase the size of your RCDSs before all the RCDSs become full. Figure 61 on page 111 and Figure 62 on page 112 show how you can:
• Create temporary VSAM data sets with more space than the original data sets.
• Use the AMS REPRO command to copy the old VSAM data sets to the temporary data sets.
• Check the results of the REPRO.
• Delete the original VSAM data sets.
• Rename the temporary VSAM data sets to the original names.

Figure 61 on page 111 shows the AMS commands to create new temporary data sets and how to copy the information in the old VSAM data sets to the temporary data sets.

```
ALLOC FI(DWWCON1) DA('DWW.DWWCON1') SHR REUSE
ALLOC FI(DWWCON2) DA('DWW.DWWCON2') SHR REUSE
ALLOC FI(DWWCON3) DA('DWW.DWWCON3') SHR REUSE
```

Figure 60. Allocating the RCDS to the CICSVR ISPF Dialog Interface
Verify that all the AMS REPRO steps are successful before proceeding.

The example in Figure 62 on page 112 shows the AMS commands to delete the original data sets and rename the new data sets.

```bash
//DEFCLUS1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSSIN DD *
DEFINE CLUSTER -
   (NAME(DWW.DWWTEMP1) -
    VOLUMES(TSOL01) -
    CYLINDERS(100 30) -
    LINEAR -
    SHR(3 3))

DEFINE CLUSTER -
    (NAME('DWW.DWWTEMP2') -
     VOLUMES(TSO007) -
     CYLINDERS(100 30) -
     LINEAR -
     SHR(3 3))

DEFINE CLUSTER -
    (NAME('DWW.DWWTEMP3') -
     VOLUMES(TSOL02) -
     CYLINDERS(75 30) -
     LINEAR -
     SHR(3 3))

REPRO -
   INDATASET('DWW.DWWCON1') -
   OUTDATASET('DWW.DWWTEMP1')

REPRO -
   INDATASET('DWW.DWWCON2') -
   OUTDATASET('DWW.DWWTEMP2')

REPRO -
   INDATASET('DWW.DWWCON3') -
   OUTDATASET('DWW.DWWTEMP3')
/
```

Figure 61. Increasing the size of the RCDS (steps 1 and 2)
Remember to rename the cluster and the data components of the data sets.

**If your RCDS is damaged:** Assume that your DWWCON3 RCDS is damaged and you want to delete it and create a new replacement RCDS. Figure 63 shows the AMS commands to delete the damaged RCDS and define a new one.

---

Deciding which VSAM spheres to protect

You will probably want to be able to forward recover all VSAM spheres that are defined as CICS files. Of course, data sets that are read-only or browse-only do not need forward recovery or backout, but you should still back them up regularly in case they become unusable.
Ensure that you define your VSAM base clusters and alternate index (AIX) data sets with SHAREOPTION 1 or SHAREOPTION 2. You can do this by using the SHAREOPTION parameter of the appropriate access method services (AMS) command, such as the DEFINE CLUSTER command or the ALTER command. SHAREOPTION 3 and SHAREOPTION 4 are not suitable for CICS, because CICS cannot prevent concurrent write access to the data sets by other CICS regions or by batch jobs.

If you must use SHAREOPTION 3 or SHAREOPTION 4 with multiple CICS regions sharing the same VSAM sphere, you must use unique ddnames across different systems. Unique ddnames ensures the correct sequencing and enables CICSVR to correctly process the log records for backout or forward recovery. Refer to the SHAREOPTION description in z/OS DFSMS Access Method Services for more information about the risks involved with SHAREOPTION 3.

Setting up your retention periods for your logs

IBM recommends that you set up retention periods for your logs. Setting up retention periods will help you in the following ways:

- It prevents your RCDS from filling up with obsolete information.
- It helps you to manage your disk space by uncataloging and deleting obsolete archived logs.

Retention periods for archived logs

When the archive utility invoked, archive will check all the logs that are registered in the RCDS and it will deregister the logs that are older than the retention period. To specify the retention criteria for one or more logs, select option 2 from the CICSVR main menu panel, List of logs (CICS V4). Then, use the Administrate pull-down from the CICSVR log list panel and select option 2, Automatic deregister. If you set the Retention period and the Uncatalog and delete fields, the logs will be uncataloged and deleted accordingly when archive is run (if AUTODEREG is set or defaulted to YES) and the retention period has passed.

If the CICSVR server address space is active, it will also check all logs that are registered in the RCDS and it will deregister the ones that are older than the retention period.

The CICSVR automatic log deregister secondary window is displayed as shown in Figure 64 on page 114.
To ensure you have recovery capability for your VSAM spheres, set up automatic archiving with CICS V4. Automatic archiving ensures that log data sets are not overwritten before the contents have been archived for recovery purposes. Two disk logs must be used with automatic archiving. When the first log becomes full, CICS switches to the second log and continues to write log data there. When the log is switched, CICS automatically submits an archive job to make a copy of the first log. If a copy is not made of the first log, the log information will be lost when the second log data set becomes full and CICS automatically switches back to the first log data set.

Figure 64. Automatic log deregister secondary window

Setting up automatic archiving with CICS V4

To ensure you have recovery capability for your VSAM spheres, set up automatic archiving with CICS V4. Automatic archiving ensures that log data sets are not overwritten before the contents have been archived for recovery purposes. Two disk logs must be used with automatic archiving. When the first log becomes full, CICS switches to the second log and continues to write log data there. When the log is switched, CICS automatically submits an archive job to make a copy of the first log. If a copy is not made of the first log, the log information will be lost when the second log data set becomes full and CICS automatically switches back to the first log data set.

Setting up the journal archive job skeleton

Figure 66 on page 115 shows the DFH$ARCH sample archive job skeleton that is shipped with CICS.
You can replace the IEFBR14 step with an archive step. Figure 66 shows a sample archive step that you could add to the DFH$ARCH member.

Here is a description of each of the numbered statements:

1. Archive the log to 1 data set and produce a detailed recovery report.

The DFH$ARCH member can be submitted by the CICS automatic archive job submission program when the disk log is switched.
For more information on the CICS V4 options to set for automatic archiving, see "Chapter 9. Setting up your CICS V4 environment" on page 133.

Understanding the archive utility report
If you would like a CICSVR archive utility report produced each time archive is run, specify the RECOVERYREPORT(YES) keyword in the ARCHIVE command, as shown in Figure 66 on page 115.

Figure 67 on page 117 shows the archive utility report.
ARCHIVE STATISTICS FOR A CICS/ESA LOG

This section shows general information about the CICS V4 log.

CICSID

The CICS APPLID from the ARCHIVE command.
MVS ID
The MVS ID from the ARCHIVE command.

Log ID
The log ID from the journal-label record.

CICS version
The CICS version from the journal-label record.

First time
The first time recorded on the journal-label record.

Last time
The last time recorded on the journal-label record.

First sequence number
The first sequence number on the journal-label record. This field does not appear for CICS/MVS logs.

Last sequence number
The last sequence number on the journal-label record. This field does not appear for CICS/MVS logs.

Input log name
The input log name from the DWWARC1 ddname in the archive JCL.

Output log name(s)
The output log names from the DWWCOPYn ddname in the archive JCL.

2 VSAM DATA SET STATISTICS
This section shows the names of the VSAM data sets that were open during the time period this journal was written. When a VSAM data set was opened, a tie-up record was written. Tie-up records relate the VSAM data set name and FCTNAME (DD name).

VSAM data set name
The unique data set name for each tie-up record.

Date/time of first reference
The date and time of the tie-up record for this data set.

Date/time of last reference
The date and time of the last after-image for this data set.

3 TIE-UP STATISTICS
This section relates the VSAM data set name and FCTNAME.

FCT name
The ddname of the data set, as defined in the FCT.

Date/time of tie-up record
The date and time of the tie-up record for this data set.

FCT format
The record format of this data set, as defined in the FCT.

Base data set name
The base data set name.

Path data set name
The path name for this data set.
4 REDO FILE ID WITHOUT TIE-UPS
This section shows that REDO records (after images) were written to this journal, but the tie-up information (both FCTNAME and VSAM data set name) is not available on this journal. Only the REDO records and the FCTNAME are available. CICSVR must also have the VSAM data set name to recover the data set. So, earlier journals are needed to recover the VSAM sphere associated with this FCTNAME. Review the TIE-UP statistics sections of earlier Archive reports. If the FCTNAME in question is listed (i.e. BASE1C), then that journal contains the tie-up information for that VSAM sphere. If CICSVR has all the tie-up information for the VSAM sphere and the REDO records, it can recover the VSAM sphere.

FCT name
The ddname of the data set, as defined in the FCT.

Date/time of first reference
The date and time of the first reference for this FCT name.

Date/time of last reference
The date and time of the last after-image for this FCT name.

5 BACKOUT FAILING LOG RECORDS FOUND
This section shows spheres in which CICS has updated the VSAM sphere but could not commit all of the changes successfully (uncommitted changes exist). The CICSVR BACKOUT function must be performed to get the VSAM sphere back to a consistent state (note CICSVR BACKOUT is only available for CICS V4). Use the CICSVR panels to construct a BACKOUT job to backout all uncommitted changes. CICSVR will read the system log backward until it has backed out all uncommitted changes, or until it finds the start of the earliest system log provided.

Base data set name
The base data set name.

Date/time of last BOFL record
The date and time of the last backout failing log record for this data set.

Record types missing
The missing record type.

Setting up the journal archive job skeleton with shadow forward recovery

Figure 68 on page 120 shows the DFH$ARCH sample archive job skeleton with CICSVR statements that can be submitted by the CICS automatic archive job submission program when the disk log is switched. Three forward recoveries are performed to update the shadow copies of the data sets to keep the shadow copy up-to-date.
Figure 68. Sample journal archive job skeleton with updates to a shadow copy (Part 1 of 2)
Setting up CICSVR change accumulation

CICSVR CA reduces the time it takes to perform a forward recovery. CICSVR CA consolidates forward recovery log records into a CA data set. CICSVR uses the CA data set in conjunction with the forward recovery log to reduce the number of log records that CICSVR needs to apply to get the sphere back to the exact state before the data was lost.

If you use DFSMShsm, change accumulation is supported. If you use DFSMSdss, you must activate the CICSVR server address space to use change accumulation. See “Chapter 3. Activating the CICSVR server address space” on page 17 for more information.

CA is a two step process. The next two sections cover both of these steps.

Step 1: Set up change accumulation batch jobs

The first step is to create batch jobs that define CA groups and identify the spheres in each group. Figure 69 on page 122 shows an example of the commands necessary to define a CA group and the sphere names associated with that group when a DFSMShsm or DFSMSdss backup is available.
The CA command:

- Defines a CA group name, MYCAGROUP1
- Specifies that the high level qualifier, CA, is used for the dynamically created CA output data set
- Specifies the volume serial number for the CA output data set if it is not an SMS-managed data set
- Specifies the device type for the CA output data set if it is not an SMS-managed data set

The SPHERE command:

- Identifies the spheres in that group, PROD.PAYROLL1 and PROD.PAYROLL2

You should choose VSAM spheres that have the same backup requirements when you create your CA group. IBM recommends that you backup all the spheres in a CA group at the same time and then run the CA batch job.

When this job is run, CA calls DFSORT, or an equivalent product with equivalent support, to sort the forward recovery log stream records for PROD.PAYROLL1 and PROD.PAYROLL2. CA saves the latest update for each record using the sorted log records by applying the log records for PROD.PAYROLL1 and PROD.PAYROLL2 to the CICSVR dynamically created CA data set.

Every time the CA job is run, the CA data set is updated with the information from the next forward recovery log range.

When the CA batch job is run again, CA gets the latest information about the backups. If a new backup has been taken, CA determines that the information in

---

1. See Appendix B, Diagnosing change accumulation and DFSORT problems on page 273 for information on eliminating common errors.
the CA data set for PROD.PAYROLL1 is no longer valid and does not use it. Instead, CA collects the log records from the forward recovery log for PROD.PAYROLL1 and the log records from the CA data set and the forward recovery log for PROD.PAYROLL2 and calls DFSORT to sort the records. When the records are sorted, CA consolidates the records by applying the log records for PROD.PAYROLL1 and PROD.PAYROLL2 to the CA data set.

It is important that you run the CA batch job immediately after a backup is taken for any of the spheres in the MYCAGROUP1 group so the CA information for that sphere is invalidated and will not be used by CICSVR if a recovery is requested. The CA batch job for MYCAGROUP1 should be run over and over again so that the CA data set is kept up-to-date. CA significantly speeds up forward recovery processing by consolidating the log stream records before a recovery is needed.

For information on diagnosing DFSORT problems see "Appendix B. Diagnosing change accumulation and DFSORT problems" on page 279.

Figure 70 shows an example of the commands necessary to define a CA group and the sphere names associated with that group when a DFSMShsm or DFSMSdss backup is not available.

```
//JOBCA2 JOB MSGCLASS=X,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//* CICSVR PROGRAM
//STEP1 EXEC PGM=DWWCA
//* LIBRARY CONTAINING CICSVR PROGRAMS
//STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR
//DWWMSG DD SYSOUT=* 
//DWWPRINT DD SYSOUT=* 
//DWWSORT DD SYSOUT=* 
//* CICSVR RECOVERY CONTROL DATA SET NAMES
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR 
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR 
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100,25)) 
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100,25)) 
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100,25)) 
//DWWIN DD *
CA -
  GROUP(MYCAGROUP2) - 
  VOLUME(123456) - 
  UNIT(3490) 
SPHERE -
  NAME(PROD.PAYROLL1) - 
  BACKUPTIME(dateandtime,GMT) 
SPHERE -
  NAME(PROD.PAYROLL2) - 
  BACKUPTIME(dateandtime,GMT) 
SPHERE -
  NAME(PROD.PAYROLL3) - 
  BACKUPTIME(dateandtime,GMT) 
SPHERE -
  NAME(PROD.PAYROLL4) - 
  BACKUPTIME(dateandtime,GMT) 
```

Figure 70. Change accumulation batch job with no DFSMShsm or DFSMSdss backup

The CA command defines a CA group name, MYCAGROUP2, and the SPHERE command identifies the spheres in that group, PROD.PAYROLL1, PROD.PAYROLL2, PROD.PAYROLL3, and PROD.PAYROLL4. You must specify the date and time of the most recent backup for the sphere in the BACKUPTIME field.
CA will only collect records for the VSAM sphere that occur after the backup was made. You can run this job multiple times for the VSAM spheres until a new backup is made. When a new backup is made for a VSAM sphere, you must update the BACKUPTIME field and run this job again. When you run this job, CA calls DFSORT to sort the forward recovery log stream records for PROD.PAYROLL1, PROD.PAYROLL2, PROD.PAYROLL3, and PROD.PAYROLL4. When the records are sorted, CA records are written to a CICSVR dynamically created CA data set on tape volume 123456.

**Note:** If CA is processing log records from a CICS V4 log and it detects a CICS backout failure, CA stops for this data set. The CA records for this data set cannot be used but all other VSAM spheres in this CA group that do not have backout failed records can be processed normally by CA.

**Understanding the change accumulation utility report**

A CICSVR CA utility report is produced every time the CA command is issued. **Figure 71 on page 125** shows the CA utility report.
Figure 72 on page 126 shows change accumulation output for an MVS log stream:
### LOGSTREAM STATISTICS

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSTREAM NAME</td>
<td>CICSVR1.MVSLOG.LOL1</td>
</tr>
<tr>
<td>FIRST TIME GMT</td>
<td>2001.030 16.15.30</td>
</tr>
<tr>
<td>LAST TIME GMT</td>
<td>2001.030 16.15.35</td>
</tr>
<tr>
<td>FIRST TIME LOCAL</td>
<td>2001.030 14.15.30</td>
</tr>
<tr>
<td>LAST TIME LOCAL</td>
<td>2001.030 14.15.35</td>
</tr>
<tr>
<td>FIRST BLOCK NUMBER</td>
<td>737</td>
</tr>
<tr>
<td>LAST BLOCK NUMBER</td>
<td>994</td>
</tr>
<tr>
<td>NUMBER OF RECORDS READ</td>
<td>20</td>
</tr>
</tbody>
</table>

**CICSVR - CHANGE ACCUMULATION UTILITY**  
**DATE:** 2001/01/30  **TIME:** 17:50:36

### OUTPUT CA DATA SET STATISTICS

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT CA DATA SET NAME</td>
<td>A12345.B6789.C12345</td>
</tr>
<tr>
<td>VOLUME</td>
<td>DEFGHJ</td>
</tr>
<tr>
<td>UNIT</td>
<td>2401</td>
</tr>
<tr>
<td>FIRST RECORD TIMESTAMP, LOCAL</td>
<td>2001.030 14.10.35</td>
</tr>
<tr>
<td>FIRST RECORD TIMESTAMP, GMT</td>
<td>2001.030 16.10.35</td>
</tr>
<tr>
<td>LAST RECORD TIMESTAMP, LOCAL</td>
<td>2001.030 14.10.40</td>
</tr>
<tr>
<td>LAST RECORD TIMESTAMP, GMT</td>
<td>2001.030 16.10.40</td>
</tr>
<tr>
<td>NUMBER OF RECORDS WRITTEN</td>
<td>277</td>
</tr>
</tbody>
</table>

#### VSAM DATA SETS STATISTICS FOR OUTPUT CA DATA SET OF THIS CA GROUP

- **BASE CLUSTER NAME:** CICS1.PROD1.BASE  
  **START DATE AND TIME LOCAL:** 2001.030 14:19:23  
  **STOP DATE AND TIME LOCAL:** 2001.030 16:23:05  
  **LAST BACKUP TIME LOCAL:** 2001.030 13:23:05  
  **NUMBER OF RECORDS WRITTEN:** 120

- **BASE CLUSTER NAME:** CICS1.PROD2.BASE  
  **START DATE AND TIME LOCAL:** 2001.030 14:21:37  
  **STOP DATE AND TIME LOCAL:** 2001.030 17:23:27  
  **LAST BACKUP TIME LOCAL:** 2001.030 12:23:27  
  **NUMBER OF RECORDS WRITTEN:** 39

- **BASE CLUSTER NAME:** CICS1.PROD3.BASE  
  **START DATE AND TIME LOCAL:** 2001.030 14:44:02  
  **STOP DATE AND TIME LOCAL:** 2001.030 16:44:02  
  **LAST BACKUP TIME LOCAL:** 2001.030 18:23:05  
  **NUMBER OF RECORDS WRITTEN:** 98

---

**Figure 72. Change accumulation output, page 3 of 3**

**CA utility report fields description:**

**CA group name**

The group name from the CA command. The CA Group name must be
unique and is used by CICSVR to identify the specified group of data sets in the RCDS. The group name is a maximum of 36 characters.

**Input CA data set name**
The name of the CA data set dynamically created during the previous CA utility run for this CA group.

**First record timestamp, local**
The first time recorded on the input CA data set (in the local time format).

**First record timestamp, GMT**
The first time recorded on the input CA data set (in the GMT format).

**Last record timestamp, local**
The last time recorded on the input CA data set (in the local time format).

**Last record timestamp, GMT**
The last time recorded on the input CA data set (in the GMT format).

**Number of records read**
The number of records read from the input CA data set.

**Base cluster name**
The base VSAM data set name from the input CA data set of this CA group.

**Start date and time local**
The first VSAM data set timestamp recorded on the input CA data set (in the local time format).

**Start date and time GMT**
The first VSAM data set timestamp recorded on the input CA data set (in the GMT format).

**Stop date and time local**
The last VSAM data set timestamp recorded on the input CA data set (in the local time format).

**Stop date and time GMT**
The last VSAM data set timestamp recorded on the input CA data set (in the GMT format).

**Last backup time local**
The VSAM data set backup time used when the input CA data set was created (in the local time format).

**Last backup time GMT**
The VSAM data set backup time used when the input CA data set was created (in the GMT format).

**Journal name**
The input journal name if CICS/ESA logs are used for this CA group.

**First time local**
The first time recorded on this journal (in the local time format only, GMT format is not applicable).

**Last time local**
The last time recorded on this journal (in the local time format only, GMT format is not applicable).

**Number of records read**
The number of records read from this journal.

**Logstream name**
The name of the MVS log stream used for CICS TS logging or VSAM batch logging.

**First time GMT**
The first time recorded on this log stream (in the GMT format).

**Last time GMT**
The last time recorded on this log stream (in the GMT format).

**First time local**
The first time recorded on this log stream (in the local time format).
**Last time local**
The last time recorded on this log stream (in the local time format).

**First block number**
The first block number on the log stream.

**Last block number**
The last block number on the log stream.

**Number of records read**
The number of records read from this log stream.

**Output CA data set name**
The name of the CA data set dynamically created during the current CA utility run for this CA group.

**Volume**
The volume serial number from the CA command. It must be specified if the output CA data set is not a DFSMS-managed data set.

**Unit**
The device type from the CA command. It must be specified if the output CA data set is not a DFSMS-managed data set.

**First record timestamp, local**
The first time recorded on the output CA data set (in the local time format).

**First record timestamp, GMT**
The first time recorded on the output CA data set (in the GMT format).

**Last record timestamp, local**
The last time recorded on the output CA data set (in the local time format).

**Last record timestamp, GMT**
The last time recorded on the output CA data set (in the GMT format).

**Number of records written**
The number of records written to the output CA data set.

**Base cluster name**
The base VSAM data set name for the output CA data set of this CA group.

**Start date and time local**
The first VSAM data set timestamp recorded on the output CA data set (in the local time format).

**Start date and time GMT**
The first VSAM data set timestamp recorded on the output CA data set (in the GMT format).

**Stop date and time local**
The last VSAM data set timestamp recorded on the output CA data set (in the local time format).

**Stop date and time GMT**
The last VSAM data set timestamp recorded on the output CA data set (in the GMT format).

**Last backup time local**
The VSAM data set backup time used when the output CA data set was created (in the local time format).

**Last backup time GMT**
The VSAM data set backup time used when the output CA data set was created (in the GMT format).

**Number of records written**
The number of records written to the output CA data set for this VSAM data set.

---

**Step 2: Use the change accumulation data sets in a forward recovery**

The second step occurs automatically when you use the CICSVR ISPF dialog interface to forward recover a sphere. When you use the ISPF dialog to forward recover a sphere, CICSVR automatically generates the APPLYCA keyword on the RECOVER command in the forward recovery job. APPLYCA tells CICSVR to apply
the records in the CA data set to the restored VSAM sphere, then apply the remaining log range from the forward recovery log, to pick up the most recent changes. If no CA data set exists, CICSVR ignores the APPLYCA keyword.

Using the CA data set can significantly improve the performance of your forward recovery runs. If there is a specific reason you do not want to use the CA data set for a forward recovery run, you can manually delete the APPLYCA lines from the forward recovery job. If the APPLYCA keywords are deleted, CICSVR will not use the CA data set and it will only apply the records from the forward recovery log.

Deciding if you want to use shadow copies

You can reduce the time it takes to forward recover a VSAM file by creating a shadow copy of it. With a shadow copy, you:

- Create a copy of the VSAM sphere.
- Regularly run a CICSVR shadow forward recovery job to update that copy with all the log entries up to the current time.

Figure 73. Sample ISPF dialog generated forward recovery job with APPLYCA
If the shadow forward recovery job is run often, the shadow copy of the VSAM sphere is kept up-to-date. If the VSAM sphere is damaged, only a few updates need to be made to the shadow copy to get it to the same state as the original VSAM sphere.

Shadow forward recovery is intended for users who are willing to maintain a copy of critical data sets to reduce the forward recovery time.

Shadow copies are also useful at remote recovery sites when using products such as Remote Recovery Data Facility (RRDF).

The following topics are discussed in this section:
- Preparing for shadow forward recovery
- Replacing the original sphere with a shadow copy

Preparing for shadow forward recovery
If you want to create a shadow copy of your VSAM sphere, you need to:
- Use IDCAMS to create a new, empty VSAM sphere.
- Copy the original VSAM sphere into the new VSAM sphere. The new VSAM sphere will be your shadow copy.
- Set up a CICSVR shadow forward recovery job to regularly update the shadow copy. With CICS V4, you do this as part of the journal archive job skeleton. See Figure 68 on page 120 for a sample archive job skeleton with updates to a shadow copy.

Replacing the original sphere with a shadow copy
When a problem occurs, you can replace the original sphere with the updated shadow copy. Using the shadow copy can reduce the time it takes to perform a forward recovery because most of the log records have already been applied.

The basic steps to recover from a shadow copy are as follows:
- Run shadow forward recovery one more time to be sure that the shadow has all updates applied. If any change accumulation data sets are available, they will be used in addition to the forward recovery log.
- Run IDCAMS SHCDS with the FRSETRR keyword to mark the original sphere as being under maintenance. This makes it unavailable and also allows the following unbind operation to succeed.
- Run IDCAMS SHCDS with the FRUNBIND keyword to unbind any retained locks against the original sphere. This enables SMS/VSAM to preserve the locks ready for re-binding later to the recovered sphere.
- Run IDCAMS DELETE to delete the original sphere and define an empty sphere with the same name.
- Run IDCAMS SHCDS with the FRSETRR keyword to mark the new empty sphere as being under maintenance. This makes it unavailable while the REPRO from the shadow copy is in progress. This is also necessary to allow the later re-bind operation to succeed.
- Run IDCAMS REPRO to copy the shadow copy to the empty sphere, hereafter called the recovered sphere.
- Run IDCAMS BLDINDEX to rebuild the AIXs for the recovered sphere. This step is only needed if the shadow copy was created with non-reusable AIXs. It can be skipped if the shadow copy was created with reusable AIXs.
- Run IDCAMS SHCDS with the FRBIND keyword to re-bind to the recovered sphere all the retained locks that were unbound from the original sphere.
- Run IDCAMS SHCDS with the FRRESETRR keyword after the re-bind to reset the maintenance flag and enable the recovered sphere for use.
Chapter 9. Setting up your CICS V4 environment

This chapter describes the tasks you will need to do to set up your CICS V4 environment for CICSVR. If you were using CICSVR V2R3, many or all of the tasks may have already been done.

The following sections describe what you need to do:
- Set up CICS V4 for CICSVR
- Establish procedures to backup and recover your VSAM spheres
- Determine your backup and restore products
- Understand the backup while open (BWO) facility
- Make your CICS System Definition (CSD) data set recoverable
- Decide on your naming conventions

Setting up CICS V4 for CICSVR

You must first define your CICS V4 files with the correct recovery attributes before you can use CICSVR. This section describes how to define the recovery characteristics of files using the CEDA transaction of the DFHFCT macro. You do not need to change your existing CICS V4 applications to use CICSVR.

Defining Files

With the CEDA DEFINE FILE command, you can specify support for both forward recovery and backout. The necessary parameters are RECOVERY and FWDRECOVLOG. A CEDA command to support a batch backout and forward recovery utility is shown in Figure 74:

CEDA DEFINE FILE(name) GROUP(groupname)
  DSNAME(CICS1.PRODZ.BASE) 1
  JOURNAL(NO) 2
  RECOVERY(ALL) 3
  FWDRECOVLOG(number) 4

Figure 74. Recovery options on the CEDA DEFINE FILE command

1. DSNAME=CICS1.PRODZ.BASE tells CICS to perform these operations (forward recovery logging, in this case) for this data set.
2. JOURNAL(NO) turns off autojournaling. CICSVR cannot perform forward recovery using an autojournal.
3. RECOVERY(ALL) tells CICS to log before images to the system log and after images to the journal specified in the FWDRECOVLOG parameter.
4. FWDRECOVLOG=number tells CICS to log after images to the journal associated with that number.

Notes:
1. RECOVERY(ALL), plus FWDRECOVLOG, provides forward recovery support for VSAM files. Forward recovery support supplied by RECOVERY(ALL) and FWDRECOVLOG is totally independent of any automatic journaling options set.
2. CICS V4 provides auto journaling to allow you to record transactions for audit purposes. Neither CICS V4 nor CICSVR can ensure that a forward recovery from an auto journal will be correct. If you would like to be able to perform forward recoveries, use the RECOVERY parameter to request forward recovery journaling.

For information about defining files, refer to CICS Resource Definition Guide.

Defining Journals

The journal control table (JCT) describes the system log and user journals to CICS V4. The JCT contains control information and operating system control blocks for each journal. Each journal referred to during CICS V4 execution must have a JCT entry as generated by the DFHJCT TYPE=ENTRY macro.

A recovery run is faster if you select a user journal for the after-image instead of selecting the system log. With a user journal, CICSVR has fewer records to search in order to find those belonging to the damaged VSAM sphere.

Specify the characteristics of all journals that you want to use, by defining a journal control table entry for each journal. The JCT for the system log and user journal are described in Figure 75.

DFHJCT TYPE=ENTRY,  
JFILEID=2,  
JOUROPT=AUTOARCH,  
ARCHJCL=DWWARCH,  
JTYPE=DISK2,  
BUFSIZE=nnnnn

Figure 75. Automatic archiving with two disk data set options on the JCT

1. JFILEID=2 tells CICS that after images will be logged to user journal number 2.

2. JOUROPT=AUTOARCH tells CICS to invoke automatic journal archiving for the journal specified by the JFILEID operand. When AUTOARCH is specified, JTYPE=DISK2 must also be specified.

3. ARCHJCL=DWWARCH is the name of the member that will be submitted by automatic archiving when the journal file is closed. DWWARCH is the CICSVR supplied journal archive job skeleton.

4. JTYPE=DISK2 tells CICS to journal to two disk data sets, alternately. JTYPE must be DISK2 for automatic archiving (when JOUROPT=AUTOARCH).

5. BUFSIZE=nnnnn is the number of bytes for each buffer that will be used for journal I/O operations. The buffer must be large enough to hold the largest request made through journal control.

These are the options you should use for your journal definitions for automatic archiving. For a description of all the options available for the JCT, refer to CICS Resource Definition Guide.
Setting up journal archive data sets

To set up automatic journal archiving, refer to CICS System Definition Guide, section "Journal archive data sets" for information on the following topics:
1. Defining the journal archive control data set (JACD)
2. Defining a partitioned data set (JPDS) for the skeletal JCL
3. Job control statements for CICS execution

Testing the automatic archiving setup

After automatic journal archiving has been setup, you should verify that CICSVR archive is called automatically by CICS V4. Perform several CICS V4 transactions and use the CEMT SET command to force the journal to be switched. For example, the following CEMT command will close the currently active journal (02) and switch to the alternate journal file.

CEMT SET JOURNAL (02) ADVANCE

When the journal is switched, CICSVR archive should be called automatically. When CICSVR archive is complete, review the archive report and verify that it contains the name of the VSAM file on which you performed transactions. It should also contain a return code of zero.

Establishing procedures to backup and recover your VSAM spheres

It is very important that you establish procedures to backup your VSAM spheres that you need to protect on a regular basis. CICSVR uses the backup copy of your VSAM sphere as the starting point of the forward recovery.

Deciding how often to make backups

Set up a backup strategy for your VSAM spheres. The sections that follow discuss daily versus non-daily backups.

Making daily backups of your VSAM spheres
If you make daily backups of your VSAM spheres, then if one of them is damaged, CICSVR can easily recreate it using the backup. Since the backup is recent, CICSVR only needs to apply the changes that were made that day to bring the VSAM sphere back to the exact state before the data was lost. You can speed up forward recovery processing even more by using CICSVR change accumulation. See [Setting up CICSVR change accumulation on page 121] for more information on change accumulation.

Making nondaily backups of your VSAM spheres
If you do not make daily backups of your VSAM spheres, then CICSVR can still easily recreate it using the backup. CICSVR needs to apply more changes, perhaps several days or even weeks worth of transactions, to bring the VSAM sphere back to the exact state before the data was lost. You can significantly speed up forward recovery processing for nondaily backups by using CICSVR change accumulation. See [Setting up CICSVR change accumulation on page 121] for more information on change accumulation.

Deciding when to make additional backups
Even though you make regular backups, there may be other times that you need to make a backup of your VSAM spheres. Here are several examples of when you should make a new backup of your VSAM sphere:
If you change the physical characteristics of a VSAM sphere, such as the control interval (CI) size, or the length or location of a key field.

If you successfully recover your VSAM sphere.

Deciding how to group your backups

If you use CICSVR change accumulation, IBM recommends that you create change accumulation groups containing VSAM spheres that have the same backup requirements:

• First, schedule backups for all VSAM spheres in a change accumulation group at the same time.

• Second, when all the backups have completed successfully, rerun the change accumulation batch job for that change accumulation group. See "Setting up CICSVR change accumulation" on page 121 for more information.

Use this process for all the change accumulation groups you have defined. This process prevents CICSVR from reading all the records in the change accumulation data set and not applying any of them because they occurred prior to the backup.

Scheduling backups to minimize your CICS down time

Your CICS online system may be running 24 hours a day or it may only be running during certain hours. You should schedule backups so that they do not affect your online CICS users. If you have VSAM data sets that must remain online and open for updates for extended periods, you can back up your VSAM data sets while they are open for update with the backup-while-open (BWO) facility. For more information on taking a backup copy of a VSAM data set while it is open for update, see "Understanding the backup-while-open (BWO) facility" on page 145.

Determining your backup and restore products

You can use any utility to backup your VSAM spheres but there are significant benefits to using IBM’s DFSMSHsm or DFSMSdss storage management products instead of just any backup product. The benefits are discussed in the following sections.

Using DFSMSHsm as your backup utility

If your VSAM spheres are managed by DFSMS/MVS, the best option is to use DFSMSHsm as your backup utility.

CICSVR provides complete data set forward recovery automation if DFSMSHsm backups are used. When you perform a complete recovery or a forward recovery, CICSVR displays the latest DFSMSHsm backup for your sphere. If you do not want to use the latest backup, you can get a list of all the DFSMSHsm logical backups for that sphere and select the backup that you want from this list.

The CICSVR server address space is not required to use DFSMSHsm. But, if it is activated (for example, so you can use the CICSVR VSAM Batch Logging feature), you must perform some additional tasks to set up DFSMSHsm. See "Chapter 3: Activating the CICSVR server address space" on page 17 for information on how to setup DFSMSHsm when you activate the CICSVR server address space.

DFMSHsm automates the backup of your VSAM spheres using aggregate backup and recovery support (ABARS). When the NetView File Transfer Program is installed on any two nodes, it can be used to transfer copies of aggregate backup
files by running a batch job or using NetView File Transfer Program input panels. For information on performing disaster backup, see z/OS DFSMSHsm Storage Administration Guide.

Using DFSMSdss as your backup utility

If you cannot use DFSMSHsm, then the next best option is to use DFSMSdss if you have activated the CICSVR server address space. The CICSVR server address space is required if you want to use DFSMSdss as your backup utility.

CICSVR provides complete data set forward recovery automation if the DFSMSdss COPY or DUMP function is used to backup your VSAM spheres. When you perform a complete recovery or a forward recovery, CICSVR displays the latest DFSMSdss backup for your sphere. If you do not want to use the latest backup, you can get a list of all the DFSMSdss backups (logical copies, logical dumps) for that sphere and select the backup that you want from the list.

You must run a special DFSMSdss COPY or DUMP job to tell DFSMSdss to create backups for use by CICSVR. DFSMSdss notifies the CICSVR server address space every time a CICSVR backup is made. CICSVR stores the backup information in the RCDS. This enables CICSVR to know about DFSMSdss backups.

DFSMSdss does not provide an automated method for backing up your VSAM spheres so you need to create batch jobs which can be regularly submitted with a production planning system such as OPC/ESA. For information on using DFSMSdss DUMP or COPY, refer to z/OS DFSMSdss Storage Administration Guide.

The next two sections describe how to use DFSMSdss COPY or DFSMSdss DUMP to tell DFSMSdss to create special backups for CICSVR.

Using DFSMSdss COPY

You can use the DFSMSdss COPY command to make copies of VSAM data sets for use by CICSVR. Figure 76 shows how to use the CICSVRBACKUP and the RENAMEU(**,CICSVR.**) keywords to notify DFSMSdss that the copies are for use by CICSVR.

```
//JOB6 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
COPY DATASET( - 
  INCLUDE(USER1.**) /* FILTER ON DS W/1ST LEV Q USER1 */ -
  OUTDYNAM((339001),(339002),(339003)) /* DYNAM ALLOC VOLS */ -
  RENAMEU(**,CICSVR.**) -
  CICSVRBACKUP
*/
```

Figure 76. DFSMSdss data set copy for use with CICSVR

This example is different than an ordinary DFSMSdss COPY job. The INCLUDE and the OUTDYNAM keywords operate like the normal DFSMSdss keywords causing all data sets with the high-level qualifier USER1 that are in the standard order of search to be copied to the target volumes that are labeled 339001, 339002, and 339003. Since CICSVRBACKUP is specified, RENAMEU(**,CICSVR.**) does not cause the INCLUDE data sets to be renamed with the CICSVR high level
qualifier. Instead, CICSVR provides DFSMSdss with a new name for each copy using the naming convention prefix.DSDSNAME.Dyyyyddd.Thhmmsss, where:

- The prefix is the CICSVR_DSNAME_PREFIX defined in the IGDSMSxx PARMLIB member.
- The yyyy is the year and ddd is the Julian day.
- The hh is the hours, mm is the minutes, and sss is the seconds and the tenth of a second.

You do not need to remember the names of the backup copies that DFSMSdss creates. Instead, when you use the CICSVR dialog, you only need to know the name of the original VSAM sphere and you can select the desired DFSMSdss backup copy from a list of backups. To view all the backup copies for a VSAM sphere from the CICSVR dialog:

- Select a VSAM sphere that you have created a DFSMSdss copy for from the CICSVR VSAM sphere list.
- Press F5 for forward recovery only.
- Put the cursor in the backup time field and press F4 to list all the backups for the selected VSAM sphere.

All the backup copies of the spheres will be listed, including the DFSMSdss backups.

**Note:** The RENAMEU keyword must be specified as "RENAMEU("**,CICSVR.**")" for DFSMSdss to create backup copies for CICSVR.

**Using DFSMSdss DUMP**

If you want to use DFSMSdss DUMP, you must create a batch job and set it up so that it's regularly submitted with a production planning system such as OPC/ESA. The tricky part is that your output data set name must be unique each time the job is run so you can maintain multiple backup copies of your VSAM data sets. Three methods of providing a unique output data set name are discussed:

- Using a batch REXX procedure to modify and submit the JCL
- Using a started task
- Using a generation data group (GDG)

**Using a batch REXX procedure to generate a unique output data set name:**

Figure 77 shows a sample of how to put date and time in the JCL to generate unique names for the output data set each time the job is run.

```plaintext
//JOB1 JOB ACCOUNTING INFORMATION,REGION=0M
//DSS EXEC PGM=ADRDSSU
//DD1 DD UNIT=3480,VOL=SER=TAPE04,
//    LABEL=(1,SL),DISP=(NEW,CATLG),
//    DSN=BACKUP.A&DATE.A&TIME
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
    DUMP OUTDD(DD1) - 
    DS(INCL(USER1.DATASET1)) - 
    CICSVRBACKUP
/*
```

Figure 77. Sample JCL to put date and time in the JCL and execute DFSMSdss DUMP
Figure 78 and Figure 79 show two samples of how to invoke a batch REXX procedure to modify and submit the JCL in Figure 77 on page 138.

Figure 78 calls the REXX procedure in Figure 80 on page 140 to replace the &DATE and &TIME values in Figure 77 on page 138 with the real date and time values. You will need to customize the input and output DD names, library, and member names in the REXX procedure.

Figure 79 calls the REXX procedure in Figure 80 on page 140 to replace the &DATE and &TIME values in Figure 77 on page 138 with the real date and time values. This is similar to Figure 79, but the input and output DD names, library, and member names are specified in the JCL instead of the REXX procedure. You will need to customize the input and output DD names, library, and member names in the JCL.

For both samples, the REXX procedure must be placed in the data set allocated to SYSPROC.
Figure 80 shows the REXX procedure, JCL1 (input and output files in the procedure):

```rexx
/*REXX: ***************************************************************/
/* */
/* This REXX procedure will get system date and time and replace */
/* &DATE and &TIME in the input member with current date and time */
/* and write all changed and unchanged lines to the output member. */
/* */
/* Input is taken from the dataset specified in variable dsni and */
/* member specified in inmem. */
/* */
/* Output is written to the dataset specified in variable dsno and */
/* member specified in outmem. */
/* */
/* The indd and outdd names are used to allocate the datasets to */
/* ddnames. They are freed in the end of this procedure. */
/* */
/* Customize the names below to get the correct input and output. */
/* */
/* *********************************************************************/
/* DATASETNAMES */
/* *********************************************************************/
dsni = "USER1.CNTL" /* INPUT LIBRARY */
dsno = "USER1.CNTL" /* OUTPUT LIBRARY */
indd = 'DDI' /* input ddname */
outdd = 'DDO' /* output ddname */
/* */
/* MEMBERS */
/* */
inmem = 'DFDSS' /* Input JCL */
outmem = 'TEMP' /* Output JCL */
/* */
/* Allocate datasets */
/* *********************************************************************/
"ALLOC DSN('' || dsni || "(" || inmem || ")") FILE('' || indd || ")"
"ALLOC DSN('' || dsno || "(" || outmem || ")") FILE('' || outdd || ")"
/* Read all lines */
/* *********************************************************************/
"EXECIO * DISKR" indd "1 (FINIS STEM IN.)"
```

Figure 80. Sample REXX procedure JCL1 for dynamic allocation (Part 1 of 2)
/* Get system date and time. */
/* Date is saved as yymmdd */
/* Time is saved as hhmms */

```rexx
date = date('S')
date = substr(date,3)

time = time('N')
time = substr(time,1,2) || substr(time,4,2) || substr(time,7,2)
```

/* Check all lines for &DATE or &TIME and replace them with the saved values in date and time. */

```rexx
o = 0 /* init */
do i = 1 to in.o /* do for all input */
o = o + 1 /* one more line for output */
out.o = in.i /* copy to output */

ind = pos('&DATE',out.o) /* any date? */
do while ind > 0 /* do while date found */
out.o = insert(' ',out.o,ind,1) /* insert a blank since */
out.o = overlay(date,out.o,ind) /* overlay is one longer */
ind = pos('&DATE',out.o) /* any more? */
end /* */

ind = pos('&TIME',out.o) /* any time? */
do while ind > 0 /* do while time found */
out.o = insert(' ',out.o,ind,1) /* insert a blank since */
out.o = overlay(time,out.o,ind) /* overlay is one longer */
ind = pos('&TIME',out.o) /* any more? */
end /* */

end /* */

/* Write all lines to output dataset */

```
"EXECIO * DISKW" outdd "(FINIS STEM OUT.)"
```

/* Free the allocated datasets */

```
"FREE FILE(" indd | "")"
"FREE FILE(" outdd | "")"
```

/* Submit the changed job and return */

```
"SUBMIT " dsno | "(" outmem | "")"
return
```

Figure 80. Sample REXX procedure JCL1 for dynamic allocation (Part 2 of 2)
Figure 81 shows the REXX procedure, JCL2 (input and output files in the JCL):

```rexx
/*REXX: ***************************************************************/
/* */
/* This REXX procedure will get system date and time and replace */
/* &DATE and &TIME in the input member with current date and time */
/* and write all changed and unchanged lines to the output member. */
/* */
/* Input is taken from the dataset specified in the JCL for the */
/* ddbname given in variable indd, in this example DDI */
/* */
/* Output is written to the dataset specified in the JCL for the */
/* ddbname given in variable outdd, in this example DDO */
/* After conversion the output is submitted and the member used */
/* for that is in variable outmem, in this example TEMP */
/* */
/* Customize the names below to get the correct input and output. */
/* */
/* DD NAMES and member */
/* */
outmem = 'TEMP' /* output member given in JCL */
indd = 'DDI' /* input ddbname */
outdd = 'DDO' /* output ddbname */

/**********************************************************************/
/* Read all lines */
/**********************************************************************/
"EXECIO * DISKR" indd "1 (FINIS STEM IN.)"

/**********************************************************************/
/* Get system date and time. */
/* Date is saved as yymmdd */
/* Time is saved as hhmmss */
/**********************************************************************/
date = date('S')
date = substr(date,3)

time = time('N')
time = substr(time,1,2) || substr(time,4,2) || substr(time,7,2)
```

Figure 81. Sample REXX procedure JCL2 for dynamic allocation (Part 1 of 3)
Get system date and time.
/* Date is saved as yymmd 
** Time is saved as hhmmss */

```rexx
date = date('S')
date = substr(date,3)

time = time('N')
time = substr(time,1,2) || substr(time,4,2) || substr(time,7,2)
```

Check all lines for &DATE or &TIME and replace them with the saved values in date and time.

```rexx
o = 0 /* init */
do i = 1 to in.o /* do for all input */
o = o + 1 /* one more line for output */
out.o = in.i /* copy to output */

ind = pos('&DATE',out.o) /* any date? */
do while ind > 0 /* do while date found */
  out.o = insert(' ',out.o,ind,1) /* insert a blank since */
  out.o = overlay(date,out.o,ind) /* overlay is one longer */
  ind = pos('&DATE',out.o) /* any more? */
end /* */

ind = pos('&TIME',out.o) /* any time? */
do while ind > 0 /* do while time found */
  out.o = insert(' ',out.o,ind,1) /* insert a blank since */
  out.o = overlay(time,out.o,ind) /* overlay is one longer */
  ind = pos('&TIME',out.o) /* any more? */
end /* */
```

Figure 81. Sample REXX procedure JCL2 for dynamic allocation (Part 2 of 3)
Figure 82 shows the job that is submitted by the above procedures.

```perl
/* Write all lines to output dataset */
EXECIO * DISKW" outdd "(FINIS STEM OUT.)"

/* Free the allocated datasets */
FREE FILE(" indd ")
FREE FILE(" outdd ")

/* Submit the changed job and return */
SUBMIT '" dsno "(" outmem ")"'
return

Figure 81. Sample REXX procedure JCL2 for dynamic allocation (Part 3 of 3)
```

Using a started task to generate unique output data set names: The job as shown in Figure 83 could also be run as a started task with this JCL.

```perl
//JOB3 JOB ACCOUNTING INFORMATION,REGION=0M
//DSS EXEC PGM=ADROSSU
//DD1 DD UNIT=3480, VOL=SER=TAPE04, 
// LABEL=(1,SL), DISP=(NEW,CATLG), 
// DSN=BACKUP.A&LYYMMDD.A&LHHMMSS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD * 
DUMP OUTDD(DD1) - 
DS(INCL(USER1.DATASET1)) - 
CICSVRBACKUP 
/*
```

Figure 82. Job submitted by REXX procedure JCL1 and JCL2

JES2 will replace the &LYYMMDD and &LHHMMSS with local date and time. Note that this will only work if the job is started as a started task.
Using a GDG to generate unique output data set names: The user can also define a GDG for each dump job and specify the generation data sets (GDS) as the output dump data set name. The GDG must specify scratch and 255 as the maximum number of data sets.

**Figure 84** shows a sample of how to define a GDG.

```jcl
//DEFGDG1 JOB ACCOUNTING INFORMATION,REGION=0M
//STEP1 EXEC PGM=IDCAMS
//GDGMOD DD DSNAME=GDG01,DISP=(KEEP),
//     SPACE=(TRK,(0)),UNIT=DISK,VOL=SER=VSER03,
//     DCB=(RECFM=FB,BLKSIZE=7892,LRECL=100)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DEFINE GENERATIONDATAGROUP -
   (NAME(GDG01) -
    EMPTY -
    SCRATCH -
    LIMIT(255))
/*
```

**Figure 84. Sample JCL to define a GDG**

**Figure 85** shows a sample of how to create a GDG entry.

```jcl
//DEFGDG2 JOB ACCOUNTING INFORMATION,REGION=0M
//DSS EXEC PGM=ADDRSSU
//GDGDD1 DD DSNAME=GDG01(+1),DISP=(NEW,CATLG),
//     UNIT=3480,VOL=SER=TAPE04,
//     LABEL=(1,SL)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
DUMP OUTDD(GDGDD1) -
   DS(INCL(USER1.DATASET1)) -
   CICSVRBACKUP
/*
```

**Figure 85. Sample JCL to use the GDG**

**Note:** CICSVR will not support more than 255 GDS data sets for a GDG. This is because for a non-SMS GDG, generation data sets beyond the 255th data set are not cataloged. CICSVR requires that all associated data sets be cataloged.

**Understanding the backup-while-open (BWO) facility**

Many CICS applications depend on their data sets being open for update over a long period of time. Normally, you cannot make a backup of the data set while the data set is open. Thus, if a failure occurs that requires forward recovery, all updates that have been made to the data set since it was opened must be recovered. This means that you must keep all forward recovery logs that have been produced since the data set was opened. A heavily used data set that has been open for update for several days or weeks may require many log records to be applied during forward recovery.
The backup-while-open (BWO) function allows DFSMSdss to make a backup when applications are running in continuous operation while the data set is open for update with full data integrity of copied data. This is only feasible for CICS VSAM file control data sets for which CICS creates forward recovery logs. Only the updates that have been made since the last backup copy was taken need to be recovered. This could considerably reduce the amount of forward recovery that is needed. Long running transactions, automated teller machines, and continuously available applications require the database to be up and running when the backup is being taken.

**What data sets can use BWO**

You can use the BWO facility only for:

- Data sets that reside on DFSMS/MVS-managed storage and that use an Integrated Catalog Facility (ICF) catalog.
- VSAM data sets that are accessed by CICS V4 file control, and for the CICS system definition (CSD) data set. ESDS, KSDS, and RRDS are supported. ESDS and KSDS are supported with and without alternate indexes (AIXs).

Backup-while-open is supported at the VSAM sphere level; thus, you cannot make backup-while-open copies of some sphere components and not others. The first data set opened for update against a VSAM base cluster sets the backup-while-open eligibility for the sphere. This includes base clusters that are accessed through a VSAM path key. For example, if the first data set is eligible for backup-while-open processing, CICS V4 fails the file-open operation for a following data set that is opened for update against that cluster and that is not defined as eligible for backup-while-open processing.

You can create BWO volume backups if all data sets that are open for update on the volume are eligible for BWO. The backup-while-open facility does not support physical dumps.

**AIX**

CICS normally uses a base key or a path key to access data in a VSAM base cluster data set. It is also possible, but not normal, for CICS to access AIX records by specifying the AIX name as the data set name. If an AIX data set is used in this way, you cannot define the AIX as eligible for BWO. Instead, the AIX adopts the BWO characteristics already defined for the VSAM sphere.

**How you request BWO**

You define a file as eligible for BWO with the BACKUPTYPE attribute on the CEDA DEFINE FILE command. You cannot define the backup type with the FCT macro.

If you specify BACKUPTYPE(DYNAMIC), the file is defined as eligible for BWO when the data set is opened. This definition applies only to the base cluster in a VSAM sphere. If a VSAM data set is accessed through one or more AIX, the path definitions for the AIXs must also be defined as eligible for BWO processing. You must also specify RECOVERY(ALL) and FWDRECOVLOG(nn) to request forward recovery support.

BACKUPTYPE(STATIC), the default, defines a file as not eligible for BWO. In this case, if DFSMSHsm is to back up a data set, all CICS files currently open for update against that data set must be closed before the backup can start.

All files opened against the same VSAM base cluster must have the same BACKUPTYPE value. That value is established by the first file opened against the
cluster; it is stored in the CICS data set name block (DSNB) for the cluster. If the value for a subsequent file does not match, the file-open operation fails.

**BWO for CDS**

To use BWO for the CSD file, specify the CSDBKUP=DYNAMIC system initialization parameter. You must also specify CSDRECOV=ALL and CSDFRLOG=nn to request forward recovery support.

**BWO and Systems Administration**

The systems administrator must decide which VSAM user data sets are eligible for BWO, and then set up the appropriate operating procedures for taking the BWO backup copies and for forward recovery. These procedures should include:

- How to forward recover a data set by using the BWO backup copy, the forward recovery log, and CICSVR to bring the data set to a point of consistency. Users must not have access to the file during the recovery process.
- How to forward recover a data set that may have been damaged while allocated to CICS. This operation may require backout of partially committed units of work during CICS emergency restart, after forward recovery has been done.

The procedures are simpler when using BWO than when not, because:

- Backups can be taken more frequently, so there are fewer forward recovery logs to manage. This also reduces the amount of processing that is required to forward recover the data set.
- The point from which forward recovery should start is recorded in the ICF catalog. The CICSVR uses this value to automate this part of the forward recovery process. This recovery point is saved with the backup copy and subsequently replaced in the ICF catalog when the backup copy is restored.
- During data set restore and forward recovery processing, CICS does not allow files to be opened for the same data set.

**BWO Data Set Security**

CICS must have RACF ALTER authority for all data sets that are defined as BACKUPTYPE(DYNAMIC), because CICS needs to update the BWO attributes in the ICF catalog. The authority must apply either to the data set or to the ICF catalog in which the data set is cataloged. For information on defining RACF ALTER authority, refer to *CICS RACF Security Guide*.

**BWO and the Recovery Point Time**

To recover a VSAM sphere, CICSVR must know:

- The data set to associate with each record on the logs
- The point to start recovery from

Each data set after-image record on the log is associated with a file name. But many files might be associated with the same data set. When a file is opened, the association between the file and the data set is recorded on the log by a tie-up record (TUR). For backups made without the backup-while-open facility, CICSVR uses this TUR to apply the log records to the correct data sets.

For backups made using the backup-while-open facility, CICSVR need not process a log from file-open time. Here, TURs for all open files are regularly written on the log during activity-keypoint processing. To reduce the number of TURs if the activity keypoint frequency is high, CICS V4 ensures that there is at least a 30-minute separation between sets of TURs on the log. The recovery point is a time that can be converted to a position on a forward-recovery log. It is also the point when CICSVR forward recovery should start for VSAM data sets eligible for...
backup-while-open processing. Recovery of the data set requires only the records that are written after that position. Thus, CICSVR can ignore all previous records.

The recovery point is stored in the ICF catalog. It is set when the first file is opened for update against the data set, and it is updated during activity-keypoint processing and when the file is closed.

The recovery point is not the time of the current keypoint, because there might still be some uncommitted log records that have not been forced. Instead, it is the time of the start of the last keypoint that wrote a complete set of TURs and that completed earlier than the oldest uncommitted write to a forward-recovery log.

Notes:
1. Only one new recovery point is calculated during an activity keypoint. It is used for all data sets that are open for update and eligible for backup-while-open processing. Thus, a long-running task that updates a data set that uses the backup-while-open facility will affect the forward recovery needed for all data sets.
2. If you disable activity keypointing in your system (by specifying AKPFREQ=0 in your SIT), backup-while-open support will be seriously affected. That is because, after the file-open operation, no more TURs are written and the recovery point is not updated. Thus, forward recovery of a data set that is eligible for backup-while-open processing must occur from the time that the data set was first opened for update.

CICSVR will extract the recovery point time of the restored backup from the RCDS or ICF catalog. It is not necessary or recommended to specify a STARTTIME keyword in your recovery run.

Recovering VSAM spheres with AIXs

Before you can forward recover a data set that was restored from a copy made using BWO, ensure that no AIXs are in the update set. CICSVR checks the upgrade set of data sets restored from BWO copies and issues a message if AIXs are found.

To forward recover such a data set, after the restore, use the AMS ALTER or DELETE command to remove or delete the AIXs from the upgrade set. After forward recovery has completed successfully, you can re-create the upgrade set by rebuilding the AIXs using the AMS BLDINDX command.

BWO and concurrent copy

Concurrent copy improves BWO processing by eliminating the invalidation of a BWO dump because of updates to the data set. The following is a comparison of various kinds of dumps you can ask for.

- Normal dump. Use of the data set must be quiesced so that serialization is obtained, the data set is dumped, and serialization is released. The data set cannot be used for the entire time.
- Concurrent copy dump. Use of the data set must be quiesced so that serialization is obtained, concurrent copy utilization is completed within a very short time (compared with the actual time to dump the data set), serialization is released, and the data set is dumped. The data set can be used after concurrent copy initialization is complete.
• BWO dump. Serialization is attempted but is not required, and the data set is dumped. If it is eligible for BWO, the data set is dumped without serialization and can remain in use for the entire time, but the dump can be invalidated by update activity to the data set.

• BWO dump using concurrent copy. Serialization is attempted but is not required, concurrent copy initialization is completed, and the data set is dumped. If it is eligible for BWO, the data set is dumped without serialization and can remain in use for the entire time, and updates that occur do not cause the dump to be invalidated.

To use concurrent copy, specify the CONCURRENT keyword when you use DFSMSHsm to dump BWO data sets.

**BWO and backups**

• Backups made without the BWO function require CICS to close VSAM files. Concurrent copy without BWO requires CICS to close the VSAM files when the concurrent copy session is being established in the control unit, which may require: a CICS shutdown to prevent:
  – In-flight transaction failures
  – Disruption of DB2 and/or IMS DL/I data access if the transactions map to data residing in DB2 or DL/I
  – Disruption of related applications.

These are some of the major limitations of backups made without BWO and concurrent copy.

• The BWO function allows backups to be made by DFSMSdss when applications are running in a continuous 7x24 or 5x24 operation while the data set is open for update with full data integrity of copied data. This is feasible only for CICS VSAM file control data sets for which CICS creates forward-recovery logs. Long-running transactions, automated teller machines, and 7x24 or 5x24 applications require the database to be up and running when the backup is being taken.

• The concurrent copy function used along with BWO by DFSMSdss allows backups to be made with integrity even when control-area and control-interval splits and data set additions (new extents or add-to-end) are occurring for VSAM key sequenced data sets.

**Program requirements**

CICS/ESA 3.3 (program number 5685–083) or higher with APAR PN48447 or CICS/ESA 4.1 is required for the definition of the VSAM files to be backed up with the BWO and concurrent copy functions of DFSMS/MVS 1.2.

**Hardware requirements**

The concurrent copy function is supported by the IBM 3990 model 3 with the extended platform and the IBM 3990 Model 6 control units.

---

**Making the CICS System Definition (CSD) data set recoverable**

The CSD VSAM data set is unique in that it is a CICS system data set that is managed by CICS file control. If you want to forward recovery the CSD data set, you must specify the recovery attributes in the system initialization table (SIT).

```
CSDFRLOG=(NO|1–99)
CSDRECOV=(NONE|ALL|BACKOUTONLY)
CSDBKUP=(STATIC|DYNAMIC)
```
Refer to the CICS System Definition Guide for more information on defining the CSD.

Deciding on your naming conventions

Give unique names to the backup and log data sets so that you can easily distinguish them from the operational data sets.

If you have no suitable naming convention, create one for all future backup and log data sets. MVS data set names can be 1–44 characters, divided into qualifiers of up to 8 characters. Here are some examples:

**clusternamex.BU.Ddate.Ttime**
For backup data sets of VSAM base cluster *clustername*.

**LOGnn.COPY1.Ddate.Ttime**
For the first copy of an archived log journal ID *nn*. The date and time might not be the time that archiving occurs; instead, they could be the time that CICS opened the log that is being archived.

Here are examples of data set names that follow these naming conventions:

- PAYROLL.BASE.BU.D01159.T235205 A backup data set
- LOG01.COPY1.D01159.T232145 An archived log

You can create these names automatically, by providing the date and time a log was opened, to a job control language (JCL) substitution variable for use with the automatic archiving function of CICS V4.
Chapter 10. Using CICSVR with CICS V4 at your disaster recovery site

This chapter describes the tasks you need to do to prepare your primary site and your remote site for disaster recovery using CICS V4.

Preparing your primary site

The sections that follow describe what you need to do at your primary site to prepare for remote site disaster recovery. Important information should be copied from your primary site and sent to the remote site.

The following topics are described:

• Making backups of your VSAM spheres
• Running your change accumulation batch jobs to update the RCDS
• Issuing the CEMT command to switch the log
• Using the RCDS utility to export your RCDS to the remote site
• Sending your shadow copies to the remote site
• Using the archive utility to copy your logs

Making backups of your VSAM spheres

In “Chapter 8. Setting up CICSVR for CICS V4” on page 103, you learned about the importance of establishing a backup and recovery strategy, as well as using DFSMShsm, DFSMSdss, or another product for backing up your VSAM spheres. If you are setting up your remote disaster recovery site, you need to make sure backup copies of your VSAM spheres are made and regularly sent to your remote site.

If you use DFSMShsm, the recommended method of disaster backup is to use aggregate backup and recovery support (ABARS). Refer to z/OS DFSMShsm Storage Administration Guide for more information on ABARS.

If you use DFSMSdss, the recommended method of disaster backup is to use the logical data set DUMP command and filter on the data set name. Logical data set dump processing allows you to back up data sets and to restore them to unlike devices. Refer to z/OS DFSMSdss Storage Administration Guide for more information on DFSMSdss and disaster recovery.

If you use another product for backing up your data sets, refer to the appropriate publication for more information on disaster recovery.

You should regularly send the backup copies of your VSAM spheres to your remote site.

Running your change accumulation batch jobs to update the RCDS

In “Chapter 8. Setting up CICSVR for CICS V4” on page 103, you learned about setting up CA batch jobs and running the appropriate job immediately after a backup is taken for any of the spheres in the CA group. Running the CA batch job for VSAM spheres that have been backed up ensures that the RCDS is kept up-to-date with the latest backup and CA information. See Figure 69 on page 122 for a sample change accumulation batch job.
Issuing the CEMT command to switch the log

In "Chapter 8. Setting up CICSVR for CICS V4" on page 103 and "Chapter 9. Setting up your CICS V4 environment" on page 133, you learned about setting up automatic journal archiving. If you are setting up your remote disaster recovery site, you need to make sure the RCDS contains the latest information before you send it to the remote recovery site.

The following CEMT command will close the currently active journal (02) and switch to the alternate journal file:

```
CEMT SET JOURNAL (02) ADVANCE
```

When the journal is switched, CICSVR archive is called automatically and the RCDS is updated with the latest information.

Using the RCDS utility to export your RCDS to the remote site

Because CICSVR saves important information needed for recovery in the RCDS, you need to capture this information regularly and send it to your remote site. You must use the CICSVR RCDS utility to copy the information in the RCDS. The RCDS utility copies pertinent information from the RCDS linear VSAM data set to a sequential access method (SAM) data set. You should use a production planning system, like OPC/ESA, to submit the export job and send the SAM RCDS to your remote recovery site at regularly scheduled times.

Use the RCDS EXPORT command to tell CICSVR to copy pertinent information in the RCDS to the SAM data set specified on the DWWCOPY1 DD statement. Figure 86 shows a sample RCDS EXPORT job to create a SAM data set that can be exported to your remote site.

```
//RCDSEX1 JOB ACCOUNTING INFORMATION,REGION=0M 01
//EXPORT EXEC PGM=DWWGJCDS 02
//STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR 03
//DWWCOPY1 DD DSN=USER1.DWWCOPY, 04
// UNIT=3590,VOL=SER=TAPE04, 05
// LABEL=(1,SL),DISP=(NEW,KEEP) 06
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR 07
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR 08
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR 09
//DWWMSG DD SYSOUT=** 10
//DWWPRINT DD SYSOUT=** 11
//DWWIN DD * 12
//RCDS EXPORT 13
/*
/*
```

**Figure 86. Sample RCDS EXPORT job to copy the RCDS for the remote recovery site**

Here is a description of each of the numbered statements:

<table>
<thead>
<tr>
<th>Line</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>The JOB statement defines a CICSVR job</td>
</tr>
<tr>
<td>02</td>
<td>The EXEC statement defines a CICSVR job step. Specifies the program DWWGJCDS is to be run. The prefix DWW always refers to a component of CICSVR.</td>
</tr>
<tr>
<td>03</td>
<td>This is the STEPLIB DD statement. It specifies the name of the CICSVR load library, for example, DWW.SDWWLOAD, used for this job step.</td>
</tr>
</tbody>
</table>
This is the DWWCOPY1 DD statement. It specifies the SAM data set, USER1.DWWCOPY, in which the EXPORTed RCDS information will be placed. USER1.DWWCOPY is allocated on a Standard label 3590 volume TAPE04 and kept.

DWWCON1–DWWCON3 defines the three recovery control data sets. If you are running the CICSVR server, you must use the hlq.slq that was defined in the IGDSMSxx PARMLIB member for these data sets. Pertinent information from these data sets will be copied into the data set specified on the DWWCOPY1 DD statement.

Note: CICSVR uses three identical RCDS data sets to provide fault tolerance and improved reliability.

DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a SYSOUT data set. The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.

DWWPRINT defines the output data set that contains the reports produced by CICSVR. This is usually defined as a SYSOUT data set.

DWWIN DD statement. Specifies the data set that contains the CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream (as shown).

RCDS EXPORT command. Specifies that all the necessary RCDS information will be copied to the USER1.DWWCOPY data set.

Sending your shadow copies to the remote site

In "Chapter 8. Setting up CICSVR for CICS V4" on page 103, you learned about creating shadow copies of critical VSAM files to reduce the time it takes to forward recover them. If you have created shadow copies of any of your VSAM spheres, you should send the shadow copies to your remote recovery site at regularly scheduled times.

Using the archive utility to copy your logs

In "Setting up automatic archiving with CICS V4" on page 114, you learned how to set up an archive job skeleton that could be submitted by the CICS automatic job submission program. If you have a remote recovery site set up, you need to make an extra tape copy of the log to send to the remote recovery site. You can easily create an archive job to make a second copy. A sample archive job is shown in Figure 87 on page 154.
Here is a description of each of the numbered statements:

1. The program to be run is DWWAR. The prefix DWW always refers to a component of CICSVR.
2. Supplies the name of the CICSVR load library.
3. DWWLOAD is optional and defines the alternate load library to STEPLIB, once DWWAR has been loaded.
4. DWWARC1 defines the input log to be archived.
5. DWWCON1–DWWCON3 defines the three recovery control data sets. If you are running the CICSVR server, you must use the hlq.siq that was defined in the IGDSMSxx PARMLIB member for these data sets.
6. DWWCOPY1—DWWCOPY2 defines two output data sets that the input log will be copied to.
7. DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a SYSOUT data set.
   The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.
8. DWWPRINT defines the output data set that contains the reports produced by the CICSVR archive utility. This is usually defined as a SYSOUT data set.
9. DWWIN defines the data set that contains the CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream.
10. The archive utility processes a log with an ID of 3, produced by the CICS system with an APPLID of PRODCICS (provided as a parameter on the EXEC statement).

**Using TSO COPY to copy a QSAM log**

CICSVR archive can use a copy of a QSAM log created using TSO COPY if you use the “NONUM” keyword to avoid adding seven bytes of X’Fs to the beginning of the record. To avoid the addition of the X’Fs, specify the “NONUM” keyword. For example, specify:

```plaintext
//XMMERRYA JOB ACCOUNTING INFORMATION,REGION=OM
//COMMAND EXEC PGM=DWWAR,PARM='CICSID(PRODCICS)'
//STEP2LIB DD DSN=DWW.SDWWLOAD,DISP=SHR
//DWWLOAD DD DSN=DWW.OPTIONAL.LOAD,DISP=SHR
//DWWARC1 DD DSN=PRODCICS.DFHJ03A,DISP=SHR
//DWWCON1 DD DSN=DWW.DWWCON1,DISP=SHR
//DWWCON2 DD DSN=DWW.DWWCON2,DISP=SHR
//DWWCON3 DD DSN=DWW.DWWCON3,DISP=SHR
//DWWCOPY1 DD DSN=PRODCICS.LOG03.D96159.T141530.COPY1,DISP=SHR
//DWWCOPY2 DD DSN=PRODCICS.LOG03.D96159.T141530.COPY2,DISP=SHR
//DWWMSG DD SYSOUT=* 7
//DWWPRINT DD SYSOUT=* 8
//DWWIN DD * 9
   ARCHIVE - 10
      JOURNALID(03) -
         COPIES(2)
/*
//
```
Preparing your remote site

The sections that follow describe what you need to do at your remote site to prepare for remote site disaster recovery. Important information sent from your primary site should be incorporated into the remote site. The following topics are described:

- Installing CICSVR at your remote disaster recovery site
- Saving the backups of your VSAM spheres
- Saving the log copies
- Saving the SAM RCDS
- Saving the shadow copies

Installing CICSVR at your remote disaster recovery site

You must install CICSVR V3R1 at your remote disaster recovery site and allocate an RCDS before you can use the CICSVR RCDS utility. See the CICSVR V3R1 Program Directory for CICSVR installation instructions. See “Creating and maintaining your RCDS” on page 108 for information on how to allocate an RCDS.

Saving the backups of your VSAM spheres

The primary site will regularly send the remote recovery site backup copies of your VSAM spheres. Save several generations of backups to ensure that you can fully recover the lost data if an error occurs with one or more of the backup copies. In most cases, you only need to restore the latest backup copy when recovering a VSAM sphere.

If DFSMSdfss made the backup using ABARS, install DFSMSdfss at the disaster recovery site and use DFSMSdfss to restore the backup. Refer to z/OS DFSMSdfss Storage Administration Guide for more information on ABARS.

If you used DFSMSdss to backup your data sets, install DFSMSdss at the disaster recovery site. Therefore, CICSVR can restore the backups during the recovery process.

If you used another product for backing up your data sets, refer to the appropriate publication for more information on how to restore the backup.

Saving the log copies

The primary site should regularly send the remote recovery site SAM copies of the logs. Save enough generations of log copies to ensure that you can fully recover the lost data if an error occurs with one or more of the backup copies. CICSVR can use log copies in place of logs to forward recover a VSAM sphere.

Saving the SAM RCDS

The primary site should regularly send the remote site SAM copies of its RCDS. Save several generations of the SAM RCDS copies at the remote site. Do not IMPORT the copy into the remote site’s RCDS until you need to perform a recovery. When you run the RCDS IMPORT utility at the remote site, the information in the SAM copy from the primary site is merged with the information in the remote site’s RCDS which is a linear data set.
If the remote site’s RCDS becomes too large, you will need to delete it, redefine it, and then run IMPORT to fill it with the latest information from the primary site. See “Creating and maintaining your RCDS” on page 108 for information on creating a new RCDS.

See Figure 88 for a sample RCDS IMPORT job that updates the remote recovery site’s RCDS with the information from the primary site. RCDS IMPORT will only work with SAM RCDS data sets that were EXPORTed using RCDS EXPORT.

```
//RCDSIM JOB (ACCT),’USER’,CLASS=A,NOTIFY=,MSGLEVEL=(1,1) 01
//IMPORT EXEC PGM=DWWGJCDS 02
//STEPLIB DD DSN=DWW.SDWWLOAD,DISP=SHR 03
//DWWCOPY1 DD DSN=USER1.DWWCOPY,DISP=SHR 04
//DWWCON1 DD DISP=SHR,DSN=DWW.DWWCON1 05
//DWWCON2 DD DISP=SHR,DSN=DWW.DWWCON2 06
//DWWCON3 DD DISP=SHR,DSN=DWW.DWWCON3 07
//DWWMSG DD SYSOUT** 08
//DWWPRINT DD SYSOUT** 09
//DWWIN DD * 10
   RCDS IMPORT 11
/*

Figure 88. Sample RCDS IMPORT job to update the remote recovery site's RCDS
```

Here is a description of each of the numbered statements:

**Line** | **Explanation**
---|---
01 | The JOB statement defines a CICSVR job.
02 | The EXEC statement defines a CICSVR job step. It specifies the program DWWGJCDS is to be run. The prefix DWW always refers to a component of CICSVR.
03 | This is the STEPLIB DD statement. It specifies the name of the CICSVR load library, for example, DWW.SDWWLOAD, used for this job step.
04 | This is the DWWCOPY1 DD statement. It specifies the SAM data set, for example, USER1.DWWCOPY, from which the imported RCDS information will be extracted and placed into the RCDS data sets specified on DWWCON1–DWWCON3 DD statements.
05–07 | DWWCON1–DWWCON3 defines the three recovery control data sets. If you are running the CICSVR server, you must use the hlq.slig that was defined in the IGDSMSxx PARMLIB member for these data sets.

**Note:** CICSVR uses three identical RCDS data sets to provide fault tolerance and improved reliability.
08 | DWWMSG defines the output data set that contains the CICSVR messages. This is usually defined as a SYSOUT data set.

The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. The default is 27930.
09 | DWWPRINT defines the output data set that contains the reports produced by CICSVR. This is usually defined as a SYSOUT data set.
10 | This is the DWWIN DD statement. It specifies the data set that contains the
CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream (as shown).

This is the RCDS IMPORT command. It specifies that all the RCDS information will be copied from the USER1.DWWCOPY data set to the RCDS data sets, specified on DWWCON1–DWWCON3 DD statements.

Saving the shadow copies

The primary site may be regularly sending the remote recovery site shadow copies of its critical VSAM files. Save the shadow copies. If the original VSAM sphere is damaged, you can use the shadow copy instead of the backup to forward recover the VSAM sphere. See "Replacing the original sphere with a shadow copy" on page 130 for instructions on how to recover from a shadow copy.
Part 4. CICSVR Command Reference

Chapter 11. Command reference
Command rules ........................................... 161
How to read the syntax diagrams ...................... 161
Synonyms for all commands and keywords ............ 163
CICSVR program names ................................... 165

Chapter 12. CICSVR commands used with CICS TS.
ALLOCATE—Allocate a log .................................. 168
BLDVRP—Build the VSAM resource pool .............. 171
CA—Process all the log records ......................... 173
DEFEXIT—Define CICSVR exit names for the log stream copy utility ........... 175
DEFEXIT—Define CICSVR exit names for forward recovery .................. 177
LOGOFLOGS—Process a log of logs .................... 179
LOGSTREAMCOPY—Copy an MVS log stream ............ 181
MVSLOG—Specify an MVS log stream .................... 186
RCDS—Create a RCDS copy .............................. 188
RECOVER—Forward recover a VSAM sphere ............ 190
SPHERE—Define spheres .................................. 200
VSAMSTART—Start passing AMS commands to VSAM ........... 202
VSAMEND—Stop passing AMS commands to VSAM ........ 203

Chapter 13. CICSVR commands used with CICS V4.
ALLOCATE—Allocate a log .................................. 206
ARCHIVE—Archive and register a log ............... 209
BACKOUT—VSAM sphere backout ....................... 214
BLDVRP—Build the VSAM resource pool ............ 218
CA—Process all the log records ......................... 220
DEFEXIT—Define CICSVR exit names for the archive utility ........ 222
DEFEXIT—Define CICSVR exit names for forward recovery and backout 224
FCTCOMP—pass FCT record-format information to CICSVR (use with CICS V2). 227
NOFORCE—Do not force archive to stop after an error ........ 230
RCDS—Create a RCDS copy .............................. 231
RECOVER—Forward recover a VSAM sphere ............ 233
RELATE—Relate a path to the base cluster ............ 243
SEQCHKL—Define sequence check criteria for log records .......... 245
SEQCHKR—Define sequence check criteria for log records .......... 248
SPHERE—Define spheres .................................. 251
VSAMSTART—Start passing AMS commands to VSAM ........... 253
VSAMEND—Stop passing AMS commands to VSAM ........ 254
Chapter 11. Command reference

This chapter describes the CICSVR command rules and how to read the syntax diagrams.

Command rules

As shown here, CICSVR commands are made up of the command name, one or more keywords, and one or more values for the keywords:

```
keyword
RECOVER SPHERE(PAYROLL.BASE)
```

Commands can be in the CICSVR job stream or in a data set, as defined by the DWWIN DD statement, and must be entered between columns 1 and 72. You can enter the commands in free format. There are no special column rules to follow, but only blanks and comments can precede the command name. The constituents of the command must be separated by one or more blanks or comments, or a combination of blanks and comments. Each command must start on a new line.

Values must be enclosed in parentheses. Between values, you can have a comma, blanks, or comments. If a value contains a special character, enclose the value in single quotes.

You can insert comments between the parts of a command, and between values. Comments must be enclosed by /* and */ and must not be nested. That is, a comment cannot contain another comment.

The continuation character is a hyphen (-). It must be in column 1–72, after the last character on a line.

How to read the syntax diagrams

Throughout this chapter, syntax is described using diagrams such as the one shown here, which describes the RECOVER command:

```
RECOVER SPHERE(sphere name)
```

© Copyright IBM Corp. 1991, 2001
The symbols used in syntax diagrams have this meaning:

- The statement begins here.
- The statement is continued on the next line.
- The statement is continued from a previous line.
- The statement ends here.

Read the syntax diagrams from left to right and from top to bottom, following the path of the line.

These are the conventions used in the diagrams:

- Required items appear on the horizontal line (main path):
  
- Optional items appear below the main path:
  
- If you can choose from two or more items, they appear vertically, in a stack.
If you must choose one of the items, one item of the stack appears on the main path.

```plaintext
STATEMENT ▶ required-choice-1 ▶ required-choice-2 ▶
```

If choosing one of the items is optional, the entire stack appears below the main path.

```plaintext
STATEMENT ▶ optional-choice-1 ▶ optional-choice-2 ▶
```

- Keywords that are above the main line are default keywords.

```plaintext
STATEMENT ▶ default-choice-1 ▶ other-choice-2 ▶
```

- An arrow returning to the left above the item shows an item that you can repeat. Required items appear on the main line and optional items appear below the main line.

```plaintext
STATEMENT ▶ repeatable-item ▶
```

A repeat arrow shows that you can make more than one choice from the stacked items, or repeat a single item. If a separator is required between items, it is shown as a comma in the repeat arrow.

- Keywords appear in uppercase (for example, `STATEMENT`).
- Parentheses and commas must be entered as part of the command syntax as shown.

Syonyms for all commands and keywords

<table>
<thead>
<tr>
<th>Command name</th>
<th>Synonyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCATE</td>
<td>ALLOC</td>
<td>Allocate a journal.</td>
</tr>
<tr>
<td>BACKOUT</td>
<td>BO</td>
<td>Back out a sphere.</td>
</tr>
<tr>
<td>BLDVRP</td>
<td>BVRP</td>
<td>Set VSAM BLDVRP values.</td>
</tr>
<tr>
<td>DEFEXIT</td>
<td>X, DX, EXIT, EXITS</td>
<td>Define CICSVR exits.</td>
</tr>
<tr>
<td>FCTCOMP</td>
<td>FCTC, FCTCOMPLEMENT</td>
<td>Pass FCT record-format information.</td>
</tr>
<tr>
<td>RECOVER</td>
<td>RO, RECO, RECONLY, RECOV</td>
<td>Forward recover a sphere.</td>
</tr>
</tbody>
</table>
### Table 8. Command Synonyms (continued)

<table>
<thead>
<tr>
<th>Command name</th>
<th>Synonyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATE</td>
<td>REL</td>
<td>Relate a path to the base cluster.</td>
</tr>
<tr>
<td>SEQCHKL</td>
<td>SEQL, LDS</td>
<td>Define sequence check for log data sets.</td>
</tr>
<tr>
<td>SEQCHKR</td>
<td>SEQR, LREC</td>
<td>Define sequence check for log records.</td>
</tr>
</tbody>
</table>

Table 9 is a summary of all the synonyms that you can use rather than the keyword names and values shown in the command descriptions.

### Table 9. Keyword and Keyword Value Synonyms

<table>
<thead>
<tr>
<th>Keyword name or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1K</td>
<td>B1024</td>
</tr>
<tr>
<td>B2K</td>
<td>B2048</td>
</tr>
<tr>
<td>B4K</td>
<td>B4096</td>
</tr>
<tr>
<td>B8K</td>
<td>B8192</td>
</tr>
<tr>
<td>B12K</td>
<td>B12288</td>
</tr>
<tr>
<td>B16K</td>
<td>B16384</td>
</tr>
<tr>
<td>B20K</td>
<td>B20480</td>
</tr>
<tr>
<td>B24K</td>
<td>B24576</td>
</tr>
<tr>
<td>B28K</td>
<td>B28672</td>
</tr>
<tr>
<td>B32K</td>
<td>B32768</td>
</tr>
<tr>
<td>CICSID</td>
<td>CID</td>
</tr>
<tr>
<td>CATALOG</td>
<td>CAT, CTLG</td>
</tr>
<tr>
<td>ERROR</td>
<td>ERR</td>
</tr>
<tr>
<td>ESDSDELETE</td>
<td>ESDSDEL, EDEL</td>
</tr>
<tr>
<td>EXPORT</td>
<td>EX</td>
</tr>
<tr>
<td>FCTNAME</td>
<td>FCT</td>
</tr>
<tr>
<td>FIXED</td>
<td>FXD</td>
</tr>
<tr>
<td>GAPINSEQUENCE</td>
<td>GAPINSEQ, GAP</td>
</tr>
<tr>
<td>IMPORT</td>
<td>IM</td>
</tr>
<tr>
<td>JOURNALID</td>
<td>JID</td>
</tr>
<tr>
<td>LOG</td>
<td>JOURNAL, JNL</td>
</tr>
<tr>
<td>NEWPATH</td>
<td>NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
<tr>
<td>NEWSPHERE</td>
<td>NEWSPH, NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
<tr>
<td>OUTOFSEQUENCE</td>
<td>OUTOFSEQ, OUT</td>
</tr>
<tr>
<td>PATH</td>
<td>DATASETNAME, DSNAME, DSN</td>
</tr>
<tr>
<td>PREAPPLY</td>
<td>PREAPP, PRE</td>
</tr>
<tr>
<td>RECFORM</td>
<td>RECFM, RECORDFORMAT</td>
</tr>
</tbody>
</table>
Table 9. Keyword and Keyword Value
Synonyms (continued)

<table>
<thead>
<tr>
<th>Keyword name or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESETSEQUENCE</td>
<td>RESETSEQ, RESET</td>
</tr>
<tr>
<td>SPHERE</td>
<td>SPH, DATASETNAME, DSNNAME, DSN</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>STARTDATETIME, STARTDATE, START</td>
</tr>
<tr>
<td>STOPTIME</td>
<td>STOPDATETIME, STOPDATE, STOP</td>
</tr>
<tr>
<td>TERMINATION</td>
<td>TERM</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>VAR</td>
</tr>
<tr>
<td>VERSION</td>
<td>VER</td>
</tr>
<tr>
<td>VOLUME</td>
<td>VOL</td>
</tr>
</tbody>
</table>

CICSVR program names

You must use the specific CICSVR program name when you call CICSVR to perform a certain function. In most cases, use the CICSVR ISPF dialog interface so that CICSVR generates the correct CICSVR program name when it generates the JCL for the job. See Table 10 for the CICSVR program names for the various CICSVR functions. The commands apply to both CICSVR V4 and CICS TS, unless otherwise noted.

Table 10. CICSVR program names

<table>
<thead>
<tr>
<th>CICSVR program name</th>
<th>Associated command name or utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWWAR</td>
<td>ARCHIVE command (CICS V4 only)</td>
</tr>
<tr>
<td></td>
<td>LOGOFLOGS command (CICS TS only)</td>
</tr>
<tr>
<td>DWWCO</td>
<td>RECOVER command</td>
</tr>
<tr>
<td></td>
<td>BACKOUT command (CICS V4 only)</td>
</tr>
<tr>
<td></td>
<td>LOGOFLOGS command (CICS TS only)</td>
</tr>
<tr>
<td>DWWCA</td>
<td>CA command</td>
</tr>
<tr>
<td>DWWLC</td>
<td>LOGSTREAMCOPY command</td>
</tr>
<tr>
<td>DWWGJCDS</td>
<td>RCDS command</td>
</tr>
<tr>
<td>DWWMIW</td>
<td>Migration Utility</td>
</tr>
</tbody>
</table>
Chapter 12. CICSVR commands used with CICS TS

This chapter describes the following CICSVR functions and commands.

Table 11. CICSVR commands used with CICS TS

<table>
<thead>
<tr>
<th>Function</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Recovery</td>
<td>ALLOCATE</td>
</tr>
<tr>
<td></td>
<td>BLDVRP</td>
</tr>
<tr>
<td></td>
<td>DEFEEXIT</td>
</tr>
<tr>
<td></td>
<td>MVSLOG</td>
</tr>
<tr>
<td></td>
<td>RECOVER</td>
</tr>
<tr>
<td></td>
<td>VSAMSTART</td>
</tr>
<tr>
<td></td>
<td>VSAMEND</td>
</tr>
<tr>
<td>Log stream copy</td>
<td>DEFEEXIT</td>
</tr>
<tr>
<td></td>
<td>LOGSTREAMCOPY</td>
</tr>
<tr>
<td></td>
<td>MVSLOG</td>
</tr>
<tr>
<td>Log of logs scan</td>
<td>LOGOFLOGS</td>
</tr>
<tr>
<td>Change accumulation</td>
<td>CA</td>
</tr>
<tr>
<td></td>
<td>SPHERE</td>
</tr>
<tr>
<td>Export or import the RCDS</td>
<td>RCDS</td>
</tr>
</tbody>
</table>
ALLOCATE—Allocate a log

Use the ALLOCATE command to identify the CICS log or log stream copy that you want CICSVR to read. Do not use ALLOCATE for MVS log streams. For MVS log streams, use the MVSLOG command.

Format

ALLOCATE LOG(dsn1,dsn2,...) VOLUME(vol1,vol2,...)
UNIT(unit1,unit2,...)

Keywords

LOG(dsn1,dsn2,...)
Specifies the logs in ascending time order, with the earliest specified first. The LOG keyword is required.

dsn1,dsn2,...
Specifies the log data set name, which is 1–44 characters. You can specify one or more logs.

VOLUME(vol1,vol2,...)
Specifies the first volume serial number that the log resides on. The VOLUME keyword is required only if at least one of the logs specified is not cataloged. If specifying more than one log, where some are cataloged and some are not, use an asterisk (*) to denote the volume for the cataloged logs.

vol1,vol2,...
Specifies the volume field, which is either 1–6 characters, or an asterisk (*) to denote the volume for the cataloged logs. You can specify one or more volumes that correspond to the number of logs you have specified.

UNIT(unit1,unit2,...)
Specifies the device type that the log resides on. The UNIT keyword is required only if at least one of the logs specified is not cataloged. If specifying several logs, where some are cataloged and some are not, use an asterisk (*) to denote the unit for the cataloged logs.

unit1,unit2,...
Specifies the unit name, which is 1–8 characters, use an asterisk (*) to denote the volume for the uncataloged logs. You can specify one or more unit types, corresponding to the number of logs you have specified.

Usage Notes

Consider the following information when you use the ALLOCATE command:

- You can specify these types of logs to CICSVR:
  - CICSVR SAM copies of MVS log streams
  - CICS V4 logs
- You can specify the log in a DWWLOG DD instead of specifying the ALLOCATE statement.
You cannot specify MVS log streams using the ALLOCATE command. You must use the MVSLOG command (see page "MVSLOG—Specify an MVS log stream" on page 186).

You cannot use CICS Transaction Server logs for backout. CICS Transaction Server provides online backout failure support.

If you use the ALLOCATE command to tell CICSVR to read a log stream copy, you must also specify the MVSLOG command to tell CICSVR it is a SAM copy of an MVS log stream.

You must know which backup to restore and which logs to use for your CICSVR run. If you use DFSMShsm or DFSMSdss, CICSVR displays the latest backup or lets you choose from a list of valid backups. If you do not use DFSMShsm or DFSMSdss, you can get this information from the scan reports.

Base and path updates must be recorded on the same sequence of logs for a given sphere.

You have the option of allocating your logs through dynamic allocation (with the ALLOCATE command) or through ddnames in the CICSVR JCL.

If you use the ALLOCATE command to specify your logs, keep each uncataloged archived log on a single volume.

All logs that are needed for CICSVR recovery must be provided in a single step.

**Attention**

Do not split the logs needed for recovery into several CICSVR steps. This might cause loss of data integrity without warning.

Only one ALLOCATE command is permitted per CICSVR step.

**Examples**

There is a positional correspondence between the ALLOCATE keyword values. That is, if three log data set names are supplied, there must be three VOLUME and UNIT keywords (if one or more of the logs are uncataloged).

```
ALLOCATE LOG(CICSA.ARCH1 CICSA.NARCH1 CICSA.ARCH2) - VOLUME(* * 123456) - UNIT(* * TAPE)
```

Here is an explanation of each of the ALLOCATE commands:

1. The ALLOCATE command describes three logs.
2. The VOLUME command describes the first two logs as being cataloged; the third log is uncataloged and resides on volume serial number 123456.
3. The UNIT command describes the first two logs as being cataloged; the third log is uncataloged and resides on tape.

**Synonyms**

Table 12 provides the ALLOCATE commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCATE</td>
<td>ALLOC</td>
</tr>
<tr>
<td>LOG</td>
<td>JOURNAL, JNL</td>
</tr>
</tbody>
</table>
Table 12. ALLOCATE synonyms (continued)

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLUME</td>
<td>VOL</td>
</tr>
</tbody>
</table>
BLDVRP—Build the VSAM resource pool

Use the BLDVRP command to request CICSVR to issue the build VSAM resource pool (BLDVRP) VSAM macro for a forward-recovery run. This causes VSAM to generate a local shared resource (LSR) pool, which is a set of index and data buffers that are to be shared by all VSAM files participating in the CICSVR run.

You can improve the performance of your CICSVR run by using the BLDVRP command to specify the size and number of VSAM buffers.

Format

```
  BLDVRP
  -B512(number of buffers)
  -B1K(number of buffers)
  -B2K(number of buffers)
  -B4K(number of buffers)
  -B8K(number of buffers)
  -B12K(number of buffers)
  -B16K(number of buffers)
  -B20K(number of buffers)
  -B24K(number of buffers)
  -B28K(number of buffers)
  -B32K(number of buffers)
```

Keywords

```
B512(number of buffers)...B32K(number of buffers)
```

Defines the size and number of buffers in each buffer pool within the VSAM resource pool. A VSAM file uses the buffer pool whose buffer size exactly matches the file’s CI size, or if this CI size is not available, the buffer pool with the next–larger buffer size. You can only specify each buffer pool once.

```
number of buffers
```

Specifies the number of buffers to be defined for each buffer pool. The number of buffers must be in the range 3–65 535.

Usage Notes

Consider the following information when you use the BLDVRP command:

- Use the BLDVRP command to improve the performance of VSAM for a CICSVR run.

- The LSR is divided into subpools, each representing a different control interval size. Before reading the data and index control intervals, the buffers are scanned to see whether the request can be satisfied from the LSR pool. The BLDVRP command improves the performance of VSAM by reducing I/Os to disk.

- You can specify only one BLDVRP per CICSVR step.

- The BLDVRP command is optional. If you specify BLDVRP, it will apply to the entire CICSVR step.

- Use the AMS LISTCAT command to determine the CI size of the data and index components of the VSAM data set.
**Attention**

Familiarize yourself with the technique of specifying local shared resources before using the BLDVRP command in your CICSVR run; otherwise, you could inadvertently degrade performance. The more buffers you allocate, the greater the number of data and index control intervals that can be held in virtual storage. Over allocation can lead to severe paging on your system and can also increase the CPU time because of long buffer searches. With some investigation, you can find an optimal buffer size.

**Examples**

```
BLDVRP B4K(50) B2K(200)
```

This BLDVRP command defines 50 buffers of 4K and 200 buffers of 2K in the VSAM resource pool.

**Synonyms**

Table 13 provides the BLDVRP commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDVRP</td>
<td>BVRP</td>
</tr>
<tr>
<td>B1K</td>
<td>B1024</td>
</tr>
<tr>
<td>B2K</td>
<td>B2048</td>
</tr>
<tr>
<td>B4K</td>
<td>B4096</td>
</tr>
<tr>
<td>B8K</td>
<td>B8192</td>
</tr>
<tr>
<td>B12K</td>
<td>B12288</td>
</tr>
<tr>
<td>B16K</td>
<td>B16384</td>
</tr>
<tr>
<td>B20K</td>
<td>B20480</td>
</tr>
<tr>
<td>B24K</td>
<td>B24576</td>
</tr>
<tr>
<td>B28K</td>
<td>B28672</td>
</tr>
<tr>
<td>B32K</td>
<td>B32768</td>
</tr>
</tbody>
</table>
CA—Process all the log records

Use the CA command to define an existing change accumulation group and create or update the change accumulation data set for that group. Use an automated timer process, such as OPC, to regularly submit your change accumulation job.

Format

```
CA GROUP(group name) PREFIX(prefix name) VOLUME(volume) UNIT(unit)
```

Keywords

**GROUP**(group name)
Identifies the name of the change accumulation group. The GROUP keyword is required. Only one GROUP keyword is allowed for each CA control statement.

*group name*
Defines the name of the change accumulation group. This name is 1 - 36 characters.

**PREFIX**(prefix name)
Identifies the name of the high level qualifier that will be used for the dynamically created change accumulation output data set. PREFIX is optional. If specified, it overrides any value specified to the CICSVR server address space. Only one PREFIX keyword is allowed for each CA control statement.

*prefix name*
Defines the name of the high level qualifier that will be used for the dynamically created change accumulation output data set. This name is 1 to 8 characters.

**VOLUME**(volume)
Specifies the volume serial number that the change accumulation data set resides on. The VOLUME keyword is required if the output data set is not a SMS-managed data set. The data set name generated by CICSVR change accumulation will start with the prefix specified in the user prefix.

*volume*
Specifies the volume field, which is 1 - 6 characters.

**UNIT**(unit)
Specifies the device type that the change accumulation data set resides on. The UNIT keyword is required if the output data set is not a SMS-managed data set. The data set name generated by CICSVR change accumulation will start with the prefix specified in the user prefix.

*unit*
Specifies the unit name, which is 1 - 8 characters.

Usage Notes

Consider these points when you use the CA command:
• If the CA command is specified, at least one SPHERE command must be specified in the same job.
• You can specify only one CA command per CICSVR change accumulation group run.
• Use CICSVR change accumulation for CICS forward recovery logs; CICS V4 or TS autojournals are not supported.
DEFEXIT—Define CICSVR exit names for the log stream copy utility

Use the DEFEXIT command to request CICSVR archive or the log stream copy utility to call one or both of the exits.

Format

```
DEFEXIT PRECOPY(prename)
TERMINATION(terminename)
```

Keywords

PRECOPY(prename)

Specifies that the precopy exit should be used. This exit gains control before every log record is written to the output data sets. You can then use the copy of this record for any purpose. You cannot change the original log record. The PRECOPY keyword is optional; only one PRECOPY keyword is allowed for each DEFEXIT control statement.

**prename**

Defines the name of the load module for the precopy exit. This name is 1–8 characters.

TERMINATION(terminename)

Specifies that the termination exit should be used, and gives the name of the exit program that CICSVR should call. This exit is called when CICSVR archive or the log stream copy utility is about to terminate normally. The TERMINATION keyword is optional; only one TERMINATION keyword is allowed for each DEFEXIT control statement.

**terminename**

Defines the name of the load module for the termination exit. This name is 1–8 characters.

Usage Notes

Consider these points when you use the DEFEXIT command:

- This command is optional.
- Do not include a DEFEXIT command automatically in every archive or log stream copy run. Write an exit program for a specific purpose, usually for a specific situation that you have investigated.
- The exit programs must reside in a data set that is defined in the JOBLIB, STEPLIB, or DWWLOAD ddname.
- You can only specify one DEFEXIT command per CICSVR archive or log stream copy run.

Examples

```
DEFEXIT PRECOPY(PRECOPX) 1
DEFEXIT TERMINATION(TERMX) 2
```

Here is an explanation of each of these DEFEXIT commands:
In this CICSVR run, the precopy exit will be given control before every log record is copied to the output data sets.

In this CICSVR run, the termination exit will be given control when CICSVR is about to terminate.

Synonyms

Table 14 provides DEFEXIT commands or keywords along with acceptable synonyms to use in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFEXIT</td>
<td>X, DX, EXIT, EXITS</td>
</tr>
<tr>
<td>PRECOPY</td>
<td>COPY</td>
</tr>
<tr>
<td>TERMINATION</td>
<td>TERM</td>
</tr>
</tbody>
</table>
DEFEXIT—Define CICSVR exit names for forward recovery

Use the DEFEXIT command to request CICSVR to call one or more of the exits.

Format

```
DEFEXIT
   PREAPPLY(preapply exit name,NEW | OLD)
   ERROR(error exit name)
   TERMINATION(termination exit name)
```

Keywords

**PREAPPLY**(*preapply exit name*,**NEW | OLD*)

Specifies that the preapply exit should be used and gives the name of the exit program that CICSVR should call. Before every update that is made to the VSAM data set, CICSVR passes the log record and the corresponding data set record (unless the update is an add) to the exit program. The exit program, which can change the log record, must return an action code telling CICSVR which action to take. The PREAPPLY keyword is optional, and only one PREAPPLY keyword is allowed for each DEFEXIT command.

*preapply exit name*,**NEW | OLD**

Defines the name of the load module for the preapply exit. This name is 1–8 characters.

The NEW and OLD keywords are optional; if neither are specified, the default preapply exit type of **NEW** is used. If you are using CICS Transaction Server logs, specify **NEW** or use the default.

**ERROR**(*error exit name*)

Specifies that the error exit should be used, and gives the name of the exit program that CICSVR should call. When an I/O error occurs, CICSVR passes an error code and relevant error information to the exit program. The exit program must return an action code telling CICSVR which action to take. The ERROR keyword is optional, and only one ERROR keyword is allowed for each DEFEXIT command. If CICSVR meets a serious error and no error exit is provided, CICSVR issues the following message and terminates:

```
DWW0212S
Preceding I/O error has forced termination.
```

*error exit name*

Defines the name of the load module for the error exit. This name is 1–8 characters.

**TERMINATION**(*termination exit name*)

Specifies that the termination exit should be used, and gives the name of the exit program that CICSVR should call. When CICSVR is about to end normally, CICSVR passes the completion code to the exit program. The exit program can then change the completion code and return an action code telling CICSVR what action to take. The TERMINATION keyword is optional; only one TERMINATION keyword is allowed for each DEFEXIT command.

*termination exit name*

Defines the name of the load module for the termination exit. This name is 1–8 characters.
Usage Notes

Consider the following when using the DEFEXIT command:

- This command is optional.
- You can only specify one DEFEXIT command per CICSVR step.
- Do not include a DEFEXIT command automatically in every recovery run.
- Write an error exit program that is appropriate for that situation.
- The exit programs must reside in a data set that is defined in the JOBLIB, STEPLIB, or DWWLOAD ddname. You can also keep the exit programs in your CICSVR load library. If CICSVR cannot find the exit program, the following message is issued and CICSVR ends:

  DWW0206S
  The exit load module xxxx cannot be found.

Examples

Here is an explanation of each of these DEFEXIT commands:

1. The PREAPPLY keyword specifies the name of the preapply exit program, PREAPP. CICSVR passes the log record and the file to PREAPP. PREAPP can modify the record if needed, update the log, and return an action code back to CICSVR.

2. The ERROR keyword specifies the name of the error exit program, ERRX. CICSVR calls ERRX when an I/O error occurs and passes it information about the error. If there are logical errors on a VSAM sphere, ERRX may be able to correct them and then pass an appropriate action code back to CICSVR.

The TERMINATION keyword specifies the name of the termination program, TERMX. CICSVR calls TERMX just before CICSVR terminates. If, in the same run, an I/O error occurs while CICSVR is reading one of the logs or processing a VSAM file, TERMX may be able to recover, change the completion code accordingly, and return an action code back to CICSVR.

Synonyms

Table 15 provides the DEFEXIT commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFEXIT</td>
<td>X, DX, EXIT, EXITS</td>
</tr>
<tr>
<td>ERROR</td>
<td>ERR</td>
</tr>
<tr>
<td>ESDSDELETE</td>
<td>ESDSDEL, EDEL</td>
</tr>
<tr>
<td>PREAPPLY</td>
<td>PREAPP, PRE</td>
</tr>
<tr>
<td>TERMINATION</td>
<td>TERM</td>
</tr>
</tbody>
</table>
LOGOFLOGS—Process a log of logs

Use the LOGOFLOGS command to scan a CICS TS log of logs and to store essential recovery information in the RCDS. See “Defining the log of logs” on page 67 for a description of the information in a log of logs.

Format

```
LOGOFLOGS
  SCAN
    AUTODEREG(YES | NO)
    RECOVERYREPORT(NO | YES)
```

Keywords

SCAN
Scans all log of logs that are registered in the RCDS and updates the RCDS with the information from those log of logs.

AUTODEREG(YES | NO)
Specifies if the automatic deregistration function should be invoked. The automatic deregistration parameters must be set in the CICSVR dialog if you want to invoke automatic deregistration. RCDS entries are older than the retention period, they should be deleted.

YES
Specifies that automatic deregister processing is to be invoked. The default is YES.

NO
Specifies that automatic deregister processing should not be invoked.

The AUTODEREG(YES | NO) keyword is optional. If it is not specified, automatic deregistration processing is performed.

Note: Automatic deregistration will be performed only if:
- AUTODEREG is set or defaulted to YES in your scan run.
- Automatic deregister criteria has been set by using dialog option 3, List of log streams (CICS TS) and then selecting option 2, Automatic deregister, from the Administrate pull-down.

RECOVERYREPORT(NO | YES)
Specifies if the recovery report should be printed.

NO
Specifies no report is required.

YES
Specifies that a recovery report is produced for all the spheres on all the log of logs that are also registered in the RCDS. The report contains all the changes to the spheres since the last LOGOFLOGS SCAN was run.

The RECOVERYREPORT keyword is optional. If it is not specified, the report is suppressed.

Usage Notes

Consider these points when using the LOGOFLOGS command:
The LOGOFLOGS command is started by the CICSVR module DWWAR. Each time that scan is run, CICSVR begins at the point in which the last scan ended.

Never run scan at your disaster recovery site unless your disaster recovery site has been converted to your primary site and CICS is actively running and recording open and close information in the log of logs.

You can also process a log of logs using the ISPF dialog interface. For more information about processing the log of logs using the ISPF dialog interface, refer to CICSVR V3R1 User’s Guide and Reference.

For more information about log of logs, see “Defining the log of logs” on page 67.

To keep your RCDS up-to-date, set up a batch log of logs scan job so that it is submitted regularly with a production planning system, such as OPC/ESA. See “Setting up the log of logs scan utility (scan)” on page 49 for more information on log of logs scan.

You must register a log of logs in the RCDS the first time it is used by CICSVR. You can register (or deregister) log of logs entries in the RCDS through the ISPF dialog interface.

If the log of logs is switched to another log stream, ensure that you have completed the scan on all the records in the previous log stream before you change the scan to run on the new log stream. If the scan is not completed on the previous log stream, log record changes may be skipped on VSAM data set forward recovery.

Examples

Here is an explanation of each of these LOGOFLOGS commands:

1. Scan all the log of logs that are registered in the RCDS.
2. Scan all the log of logs registered in the RCDS. Since AUTODEREG is set to NO, log of logs that meet the retention criteria are not deregistered from the RCDS. A recovery report is produced for all the log of logs that are scanned and registered in the RCDS.

Synonyms

Table 16 provides LOGOFLOGS commands or keywords along with acceptable synonyms to use in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command, keyword or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGOFLOGS</td>
<td>LOL</td>
</tr>
<tr>
<td>RECOVERYREPORT</td>
<td>RECREP</td>
</tr>
<tr>
<td>AUTODEREG</td>
<td>DEREG</td>
</tr>
</tbody>
</table>
LOGSTREAMCOPY—Copy an MVS log stream

Use the LOGSTREAMCOPY command to copy an MVS log stream to a sequential access method (SAM) data set. Information about the copies is stored in the RCDS.

Format

```
LOGSTREAMCOPYNAME(MVS log stream)
  SELECT(CICS|CICSVR|ALL)
  COPIES(number of copies)
  STARTTIME(start date and time, GMT)
  STARTBLKID(start block)
  STOPTIME(stop date and time, GMT)
  STOPBLKID(stop block)
```

Keywords

**NAME(MVS log stream)**
Specifies the name of the MVS log stream.

*MVS log stream*
Is the name of the MVS log stream. The name is 1–26 characters.

**SELECT(CICS|CICSVR|ALL)**
Specifies the records that you want to copy. You may specify one of the following: (CICS), (CICSVR), or (ALL).

*CICS*
Specifies records produced by CICS and CICSVR batch logging are to be copied from the MVS log stream. CICS is the default value.

*CICSVR*
Specifies records produced by CICS and CICSVR batch logging are to be copied from the MVS log stream.

*ALL*
Specifies all records are to be copied from the MVS log stream.

**COPIES(1 | number of copies)**
Specifies if the MVS log stream is to be copied to another data set.

*number of copies*
Specifies the number of copies to be made from the input MVS log stream. The COPIES keyword is optional; if it is not specified, the default value of 1 will be used.
If COPIES(0) is specified, no copy will be made, and a report is produced (see Figure 33 on page 51).

If the number of copies specified is greater than zero, the MVS log stream is copied to the data sets named on the corresponding ddnames, DWWCOPYyn, where n is a number, 1–9, that you specify on the COPIES keyword.

**STARTTIME** *(start date and time, LOCAL | GMT)*

Identifies the earliest timestamp that you want to copy the MVS log stream from. CICSVR ignores all records for this MVS log stream whose timestamp is earlier than the time specified in the STARTTIME keyword. CICSVR copies records from this place in the log. Only one STARTTIME keyword is allowed for each LOGSTREAMCOPY control statement. You need only specify STARTTIME the first time you run the log stream copy utility. Following log stream copy runs will continue from the point on the MVS log stream after the last copy. If you do not specify STARTTIME or STARTBLKID, the MVS log stream is copied from the beginning of the log, or after the point of the last copy.

*start date and time*

Must be in the format *yy/ddd/hh/mm/ss*, where:

- *yy* is the last two digits of the year (00–99)
- *ddd* is the day of the year (001–366)
- *hh* is the hour of the day (00–23)
- *mm* is the number of minutes (00–59)
- *ss* is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
STARTTIME(01.159/22:23:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify `STARTTIME(01.159)`, CICSVR will assume that the time segment is 00:00:00.

CICSVR interprets year values (*yy*) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**LOCAL**

Specifies that the date and time are in local format. LOCAL is the default value.

**GMT**

Specifies that the date and time are in Greenwich mean time (GMT) format.

**STARTBLKID** *(start block)*

Identifies the block number from which the records are to be copied. CICSVR ignores all records for this MVS log stream before the block number specified in the STARTBLKID keyword. Only one STARTBLKID keyword is allowed for each LOGSTREAMCOPY control statement. If you do not specify STARTBLKID or STARTTIME, the MVS log stream is copied from the beginning of the log.

*start block*

Is the block number on the MVS log stream. You must specify the exact block ID.

**STOPTIME** *(stop date and time, LOCAL | GMT)*

Identifies the latest timestamp that you want to copy the MVS log stream from. CICSVR ignores all records for this MVS log stream whose timestamp is later
than the time specified in the STOPTIME keyword. CICSVR copies records up
to this place in the log. Only one STOPTIME keyword is allowed for each
LOGSTREAMCOPY control statement. If you do not specify STOPTIME or
STOPBLKID, the copy ends at job run time.

stop date and time
Must be in the format $yy/dd/mm/hh/mm/ss$, where:
  $yy$ is the last two digits of the year (00–99)
  $dd$ is the day of the year (001–366)
  $hh$ is the hour of the day (00–23)
  $mm$ is the number of minutes (00–59)
  $ss$ is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You
also can omit the separator character. For example:
STOPTIME(01.159/00:30:00)

You cannot substitute commas, blanks, and so on, for the time values, but
you can omit values from the right. For example, you can specify day
01159, time 23:59:59, as follows:
STOPTIME(01.159)

If you specify this, CICSVR will assume that the day is 01365 and the time
is 23:59:59:
STOPTIME(01)

CICSVR assumes that these values are for day 01159, and time 16:59:59:
STOPTIME(0115916)

CICSVR interprets year values ($yy$) in the range 00–85 to be years

LOCAL
Specifies that the date and time are in local format. LOCAL is the default
value.

GMT
Specifies that the date and time are in GMT format.

STOPBLKID(stop block)
Identifies the block number at which CICSVR is to stop copying records.
CICSVR ignores all records for this MVS log stream after the block number
specified in the STOPBLKID keyword. Only one STOPBLKID keyword is
allowed for each LOGSTREAMCOPY control statement. If you do not specify
STOPBLKID or STOPTIME, the copy ends at job run time.

stop block
Is the block number on the MVS log stream. You must specify the exact
block ID.

MOD
Indicates that records are to be added to the end of the data set specified in the
DWWCOPY$n$ ddname. Use the MOD keyword if you specified DISP=MOD for
any of the DWWCOPY$n$ ddnames.

Usage Notes
Consider these points when using the LOGSTREAMCOPY command:
In general, log stream copies are made for use at a disaster recovery site. For more information on what is needed at a disaster recovery site, see "Chapter 7. Using CICSVR with CICS TS at your disaster recovery site" on page 91.

After the first log stream copy run, subsequent runs will automatically begin copying after the point on the MVS log stream that the last copy finished.

The log stream copy utility is started by the CICSVR module, DWWLC (see Figure 51 on page 94).

You can have only one LOGSTREAMCOPY command in a single CICSVR job step.

LOGSTREAMCOPY keywords can be coded in any sequence.

You can copy the entire MVS log stream or just the records produced by CICS and CICSVR batch logging.

LOGSTREAMCOPY copies the MVS log stream to a maximum of nine data sets and stores information about the copies in the RCDS.

You invoke the log stream copy utility by running the required JCL, and providing the statements you need as parameters on the DWWIN DD statement.

You define the data sets to which the MVS log stream is to be copied on the DWWCOPY1—DWWCOPY9 DD statements. See Figure 51 on page 94 for an example of the JCL to run the log stream copy utility.

Before the log stream copy utility copies the MVS log stream record to the output data set a precopy exit can be called. This exit lets you inspect and act on a copy of the record before it is written to the output data set. This exit must be defined during the LOGSTREAMCOPY run by the DEFEXIT command.

Before the log stream copy utility terminates, a termination exit can be called. This exit lets you terminate actions that were performed.

For more information on using LOGSTREAMCOPY, see "Using the log stream copy utility to copy your log streams" on page 93.

Use the MOD keyword if you specified DISP=MOD for any of the DWWCOPYn ddnames.

Only the MVS log stream copied to DWWCOPY1 is registered in the RCDS.

Examples

The following LOGSTREAMCOPY command tells CICSVR to make three copies of the MVS log stream. These copies will be made to the data sets specified on the ddnames DWWCOPY1, DWWCOPY2, and DWWCOPY3. Records will be copied to the end of the data sets specified in the DWWCOPYn ddnames. Records are copied from the beginning of the MVS log stream or the first record after the end of the previous LOGSTREAMCOPY and stop at current time of this job or when the end of the log stream is reached. CICSVR will remember where the log copy stopped; so this same job could be run over and over to continuously copy the MVS log stream. This is the best way to do a log copy.

LOGSTREAMCOPY NAME(CICSVR1.MVSLOG) -
SELECT(ALL) -
COPIES(3) -
MOD

The following LOGSTREAMCOPY command tells CICSVR to make two copies of the MVS log stream. These copies will be made to the data sets specified on the ddnames DWWCOPY1, DWWCOPY2. Records will be copied to the end of the data sets specified in the DWWCOPYn ddnames. Records are copied from the MVS log stream at the start time specified and stop at the current time of this job or when the end of this log stream is reached. CICSVR will remember where the log
copy stopped Because a start time was specified, you must update the STARTTIME when a new backup is taken.

LOGSTREAMCOPY NAME(CICSVR1.MVSLOG) -
SELECT(ALL) -
COPIES(2) -
STARTTIME(01159/07:30:00,GMT) -
MOD

The following LOGSTREAMCOPY command tells CICSVR to make three copies of the MVS log stream. These copies will be made to the data sets specified on the ddnames DWWCOPY1, DWWCOPY2, and DWWCOPY3. Records will be copied to the end of the data sets specified in the DWWCOPYn ddnames. Records are copied from the beginning of the MVS log stream or the first record after the end of the previous LOGSTREAMCOPY and stop at the specified stop time. CICSVR will remember where the log copy stopped. This could be used to control the size of the output.

LOGSTREAMCOPY NAME(CICSVR1.MVSLOG) -
SELECT(ALL) -
COPIES(3) -
STOPTIME(01249/07:30:00,GMT) -
MOD

The following is LOGSTREAMCOPY command tells CICSVR to make three copies of the MVS log stream. These copies will be made to the data sets specified on the ddnames DWWCOPY1, DWWCOPY2, and DWWCOPY3. Records will be copied to the end of the data sets specified in the DWWCOPYn ddnames. All records are copied from the MVS log stream between the start and the stop time specified in the command. CICSVR will remember where the log copy stopped. This is not a recommended method to copy the log unless this is a one time copy to produce test data. Do not use this method to make consecutive copies because the time values are not precise enough to prevent gaps.

LOGSTREAMCOPY NAME(CICSVR1.MVSLOG) -
SELECT(ALL) -
COPIES(3) -
STARTTIME(01159/07:30:00,GMT) -
STOPTIME(01249/07:30:00,GMT) -
MOD

Synonyms

Table 17 provides LOGSTREAMCOPY commands or keywords along with acceptable synonyms to use in place of the commands or keywords.

Table 17. LOGSTREAMCOPY Synonyms

<table>
<thead>
<tr>
<th>Command, keyword or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSTREAMCOPY</td>
<td>LOGCOPY</td>
</tr>
<tr>
<td>SELECT</td>
<td>SEL</td>
</tr>
<tr>
<td>COPIES</td>
<td>CPS</td>
</tr>
<tr>
<td>STARTBLKID</td>
<td>STARTBLK</td>
</tr>
<tr>
<td>STOPBLKID</td>
<td>STOPBLK</td>
</tr>
</tbody>
</table>
MVSLOG—Specify an MVS log stream

Use the MVSLOG command to pass CICSVR the name of the MVS log stream that contains all the records for the data sets to be recovered in this step. Do not use MVSLOG for CICS logs or log stream copies. For both of these, use the ALLOCATE command.

Format

```
MVSLOG  
  CATALOG 
  NAME(log stream name) 
  COPY 
```

Keywords

**CATALOG**

Specifies that the MVS log stream name should be obtained from the MVS catalog. All data sets to be recovered in one step must have the same MVS log stream name. The CATALOG keyword is the default.

**NAME(log stream name)**

Identifies the data set name of the MVS log stream for this recovery step.

- **log stream name**
  
  Specifies the name of the MVS log stream. The name is 1–26 characters. The NAME keyword is optional.

**COPY**

Indicates that the logs specified as input by the CICSVR ALLOCATE command, or the logs specified by the DWWLOG ddname, are QSAM copies of MVS log streams. The COPY keyword is optional.

Usage Notes

Consider the following when using the MVSLOG command:

- You can only specify one MVSLOG command in a single recovery step. The MVS log stream you specify must contain all relevant records for all data sets to be recovered in this step.
- If you switch between a CICS log and an MVS log stream, you must process these different log types in separate CICSVR job steps.
- You can specify an active MVS log stream or a QSAM copy of an MVS log stream in the MVSLOG command. If you specify a QSAM copy, the data set name of the copy must be specified in either the CICSVR ALLOCATE command, or by the DWWLOG ddname.

Examples

```
MVSLOG  CATALOG  
MVSLOG  NAME(SYSA.MVSLOG1)  
MVSLOG  COPY 
```

Here is an explanation of each of these MVSLOG commands:

1. The name of the MVS log stream for this CICSVR step must be obtained from the MVS catalog.
The name of the MVS log stream for this CICSVR recovery step is SYSA.MVSLOG1.

The MVS log stream specified for this CICSVR step is a QSAM copy of an MVS log stream. The name of the QSAM copy is specified in either the CICSVR ALLOCATE command, or by the DWWLOG ddname.

### Synonyms

Table 18 provides the MVSLOG commands or keywords along with acceptable synonyms that can be used in place of the commands or keyword.

**Table 18. MVSLOG synonyms**

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG</td>
<td>CAT, CTLG</td>
</tr>
</tbody>
</table>
RCDS—Create a RCDS copy

Use the RCDS command to create a copy of the RCDS that can be sent (exported) to a remote site and used to update (import) the RCDS at the remote site. This will keep the remote site RCDS in synchronization with the RCDS at the primary site. Only RCDS records that are significant to the remote site are used to update the remote site’s RCDS.

Format

```
  RCDS EXPORT
    IMPORT
```

Keywords

**EXPORT**

Specifies that the information, which is stored in the RCDS, is copied into the SAM data set specified on the DWWCOPY1 DD statement.

**IMPORT**

Specifies that the information, which was previously exported, is loaded from the SAM data set specified on the DWWCOPY1 DD statement into the RCDS.

Usage Notes

Consider these points when you use the RCDS command:

- To use the RCDS utility, execute the CICSVR module DWWGJCDS.
- You can only have one RCDS command in a single job step.
- You invoke the RCDS utility by running the required JCL.
- You must specify the SAM data set that will contain the output of the EXPORT or the input for the IMPORT on the DWWCOPY1 DD STATEMENT.
- Before you use the RCDS utility on a remote site, you need to install CICSVR V3R1 and you must allocate an RCDS.

Examples

```
RCDS EXPORT
1
RCDS IMPORT
2
```

Here is an explanation of each of these RCDS commands:

1. This RCDS command tells CICSVR to copy all the necessary RCDS information to the SAM data set specified on the DWWCOPY1 DD statement. The SAM data set can be sent to a remote recovery site.

2. This RCDS command tells CICSVR to import the information, stored in the SAM data set specified on the DWWCOPY1 DD statement, into the RCDS. The information in the SAM data set must have been created using the RCDS EXPORT command.

Synonyms

Table 19 on page 189 provides RCDS commands or keywords along with acceptable synonyms to use in place of the commands or keywords.
### Table 19. RCDS Synonyms

<table>
<thead>
<tr>
<th>Command, keyword or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT</td>
<td>EX</td>
</tr>
<tr>
<td>IMPORT</td>
<td>IM</td>
</tr>
</tbody>
</table>
RECOVER—Forward recover a VSAM sphere

Use the RECOVER command to tell CICSVR to forward recover a VSAM sphere.

IBM recommends that you use the CICSVR dialog interface to generate the CICSVR jobs and not code the CICSVR statements directly. The CICSVR dialog interface performs the following tasks:

- Runs the log of logs scan to get the latest information about the logs and updates the CICSVR RCDS with this information.
- Restores the DFSMShsm or DFSMSdss backup automatically.
- Generates the necessary CICSVR statements automatically.

Format

```
RECOVER SPHERE(sphere name)
  ONLY
  FIRST
  INTERMEDIATE
  LAST

STARTTIME(start date and time, GMT)

STOPTIME(stop date and time, GMT)

STARTTOD(start time of day, GMT)

STOPTOD(stop time of day, GMT)

NEWSPHERE(updated sphere name)

SHADOW APPLY REDO STARTAT(DSNAME)

VERSION(DFSMShsm version number)

VOLUME(volume)

UNIT(unit)

HSMDATE(date)
```
Keywords

**SPHERE(sphere name)**

Specifies the VSAM sphere to be forward recovered. It is the name as recorded on the log. Only one SPHERE keyword is allowed for each RECOVER control statement. The SPHERE keyword is required.

*sphere name*

Specifies the MVS data set name of the VSAM sphere to be forward recovered. This name is 1–44 characters.

**ONLY | FIRST | INTERMEDIATE | LAST**

Specifies the relative order of the steps in a job. These keywords are mutually exclusive. One of these keywords must be specified on a forward recovery command when the NEWSPHERE keyword has not be specified. A forward recovery job is composed of one or more steps. CICSVR must know the relative forward recovery step that is executing. If there is only one forward recovery step then specify ONLY. If there are two forward recovery steps, then specify FIRST on the first step and specify LAST on the last step. When there are more than two steps, specify INTERMEDIATE on all the intermediate forward recovery steps (the steps that are not the first or the last). This is handled automatically for you through the CICSVR panels.

**STARTTIME(start date and time,LOCAL | GMT)**

Specifies the earliest timestamp of the after-images that you want to recover the VSAM sphere from. CICSVR ignores all after-images on the logs until it finds an after-image for this VSAM sphere, whose timestamp is the same as, or later than, the time specified in the STARTTIME keyword. CICSVR applies all after-images for this VSAM sphere, from this place in the log, if a tie-up record (TUR) has been encountered on the log. Only one STARTTIME keyword is allowed for each RECOVER control statement. If you do not specify STARTTIME or STARTTOD, recovery will start at the first after-image encountered on the logs, after the first TUR.

For VSAM data sets restored from a backup-while-open copy, STARTTIME will be the recovery point time. When you use the CICSVR dialog interface CICSVR will automatically retrieve the recovery point time from the integrated catalog facility (ICF) catalog, so you do not need to specify STARTTIME in this situation.

*start date and time*

Must be in the format *yy/ddd/hh/mm/ss*, where:

- *yy*  Is the last two digits of the year (00–99)
- *ddd*  Is the day of the year (001–366)
- *hh*  Is the hour of the day (00–23)
- *mm*  Is the number of minutes (00–59)
- *ss*  Is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

STARTTIME(01.159/22:23:00)

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify STARTTIME(01.159), CICSVR will assume that the time segment is 00:00:00.
CICSvr interprets year values (yy) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**LOCAL**

Specifies that the date and time are in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**

Specifies that the date and time are in Greenwich mean time (GMT) format. You can specify GMT only if you are using MVS log streams.

**STOPTIME**(stop date and time, LOCAL | GMT)

Specifies the latest timestamp of the after-images that you want to recover the VSAM sphere from. CICSvr ignores all after-images on the logs after it finds an after-image for this VSAM sphere, whose timestamp is later than the time specified in the STOPTIME keyword. CICSvr applies all after-images for this VSAM sphere up to this place in the log. Only one STOPTIME keyword is allowed for each RECOVER control statement. If STOPTIME or STOPTOD is not specified, recovery will stop after the last supplied log.

*stop date and time*

Must be in the format **yy/ddd/hh/mm/ss**, where:

- **yy** is the last two digits of the year (00–99)
- **ddd** is the day of the year (001–366)
- **hh** is the hour of the day (00–23)
- **mm** is the number of minutes (00–59)
- **ss** is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
STOPTIME(01.159/00:30:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, you can specify day 01159, time 23:59:59, as follows:

```
STOPTIME(01.159)
```

If you specify this, CICSvr will assume that the day is 01365 and the time is 23:59:59:

```
STOPTIME(01)
```

CICSvr assumes that these values are for day 01159, and time 16:59:59:

```
STOPTIME(0115916)
```

CICSvr interprets year values (yy) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**LOCAL**

Specifies that the date and time are in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**

Specifies that the date and time are in GMT format. You can specify GMT only if you are using MVS log streams.

**STARTTOD**(start time of day, LOCAL | GMT)

Identifies the earliest time, in hexadecimal time-of-day (TOD) format, of the
after-images that you want to recover the VSAM sphere from. CICSVR ignores
all after-images on the logs until it finds an after-image for this VSAM sphere,
whose TOD value is the same as, or later than, the time specified in the
STARTTOD keyword. CICSVR applies all after-images for this VSAM sphere,
from this place in the log, if a tie-up record (TUR) has been encountered on the
log. Only one STARTTOD keyword is allowed for each RECOVER control
statement. If you do not specify STARTTOD or STARTTIME, recovery will start
at the first after-image encountered on the logs, after the first TUR.

Use the STARTTOD keyword only if the STARTTIME keyword value is not
precise enough for your recovery stop time. The STARTTOD keyword has no
synonyms.

start time of day
   Must be 16 hex characters. Here is an example of a STARTTOD keyword:
   STARTTOD(AC47C0403792C101)

LOCAL
   Specifies that TOD value is in local format. You can specify LOCAL if you
   are using MVS log streams or CICS V4 logs. LOCAL is the default value.

GMT
   Specifies that the TOD value is in GMT format.

STOPTOD(start time of day,LOCAL | GMT)
Identifies the latest time, in hexadecimal time-of-day (TOD) format, of the
after-images that you want to recover the VSAM sphere from. CICSVR ignores
all after-images on the logs after it finds an after-image for this VSAM sphere,
whose TOD value is later than the time specified in the STOPTOD keyword.
CICSVR applies all after-images for this VSAM sphere up to this place in the
log. Only one STOPTOD keyword is allowed for each RECOVER control
statement. If STOPTOD or STOPTIME is not specified, recovery will stop after
the last supplied log.

Use the STOPTOD keyword only if the STOPTIME keyword value is not precise
enough for your recovery stop time. The STOPTOD keyword has no synonyms.

stop time of day
   Must be 16 hex characters. Here is an example of a STOPTOD keyword:
   STOPTOD(AC47C0403792C101)

LOCAL
   Specifies that TOD value is in local format. You can specify LOCAL if you
   are using MVS log streams or CICS V4 logs. LOCAL is the default value.

GMT
   Specifies that the TOD value is in GMT format.

NEWSPHERE(updated sphere name)
Specifies the new MVS data set name of the VSAM sphere to be recovered.
This name is 1–44 characters.

This keyword is optional, except when SHADOW is specified and it is then
required because a shadow is another copy of the data set. If this keyword is
omitted, the data set name specified by the SPHERE keyword is used.

Identifies the VSAM sphere to be recovered when the data set name differs
from the one recorded on the log. The NEWSPHERE keyword lets you recover
a VSAM sphere without having to first rename or delete the original.

If you require CICSVR to recover a copy of the VSAM sphere (for example a
restored backup copy), specify the data set name of the copy data set.
If you require CICSVR to recover the VSAM sphere itself, do not specify this keyword, or specify the same data set as in the SPHERE keyword.

This keyword is optional. If it is omitted, the data set name specified by the SPHERE keyword will be used.

*updated sphere name*

Specifies the MVS data set name of the VSAM sphere to be recovered.
This name is 1–44 characters.

**SHADOW**

The SHADOW keyword is optional and has no synonyms.

The concept is that the first time (run 1) that you run a shadow forward recovery, the backup is restored and some amount of the forward recovery log is applied. Subsequent shadow forward recovery runs (2 - n) apply more of the forward recovery log with the intent to keep the shadow copy as current as possible.

CICSVR saves the status of the shadow forward recovery in the RCDS, so the subsequent shadow forward recovery runs (2 - n) should not specify a start and stop time and should be run frequently, so as to minimize the log records to be applied.

When a problem occurs and a real forward recovery is required, the CICSVR shadow can be copied to replace the user data set.

**APPLYCA**

The APPLYCA keyword is optional and has no synonyms.

APPLYCA indicates to CICSVR that there is change accumulation data that is relative to this data set that needs to be applied during forward recovery processing. If there is no change accumulation data set or CICSVR detects an error with the CA data set, the APPLYCA keyword is ignored.

**STARTAT(REDO | DSNAME)**

Specifies where on the log the recovery will begin.

**REDO**

Specifies all after-images found after the time specified in the STARTTIME keyword will be used in this recovery, provided a TUR has been found on the log for this VSAM sphere.

**DSNAME**

Specifies the first TUR found on the log after the time specified in the STARTTIME keyword will be used. All after-images after this TUR and before the time specified in the STOPTIME keyword will be used in the recovery. So any after-images found after the time specified in the STARTTIME keyword, and before the TUR for this VSAM sphere will be ignored.

The STARTAT keyword is optional; if it is not specified, the default of REDO will be used.

If you use the STARTAT keyword without the STARTTIME keyword, STARTAT is ignored.

The STARTAT keyword has no synonyms.

**Note:** When a file is opened, the association between the file name and the data set name is recorded on the log by a TUR.
**HSMDATE**(*date*)
Identifies the date of the DFSMShsm full volume dump which is to be used to restore the VSAM sphere. For more information about the DFSMShsm restore process, refer to [z/OS DFSMShsm Storage Administration Guide](https://www.ibm.com/support/docview.ws?rs=1808&context=COM%20series%20name%3A%20IBM%20z/OS%20DFSMShsm%20Storage%20Administration%20Guide%20Context%20Page&docname=COM%20series%20name%3A%20IBM%20z/OS%20DFSMShsm%20Storage%20Administration%20Guide).

*date*
Specifies the DFSMShsm full volume dump date. The date must be in the format **yyddd**.

**VERSION**(*DFSMShsm version number*)

*DFSMShsm version number*
Identifies the version number of the DFSMShsm backup that is to be restored. The version number can be 1–999.

**VOLUME**(*volume*)
Specifies the volume serial number where the DFSMShsm backup of the VSAM sphere will be restored to. This keyword can only be used if you have specified VERSION. If you specify VOLUME you must also specify the UNIT keyword. For more information about the DFSMShsm restore process, refer to [z/OS DFSMShsm Storage Administration Guide](https://www.ibm.com/support/docview.ws?rs=1808&context=COM%20series%20name%3A%20IBM%20z/OS%20DFSMShsm%20Storage%20Administration%20Guide).

*volume*
Identifies the volume which is to be used for the DFSMShsm restore of the VSAM sphere. The volume is 1–6 characters.

**UNIT**(*unit*)
Specifies the unit name where the DFSMShsm backup of the VSAM sphere will be restored to. This keyword can only be used if you have specified VERSION. If you specify UNIT you must also specify the VOLUME keyword. For more information about the DFSMShsm restore process, refer to [z/OS DFSMShsm Storage Administration Guide](https://www.ibm.com/support/docview.ws?rs=1808&context=COM%20series%20name%3A%20IBM%20z/OS%20DFSMShsm%20Storage%20Administration%20Guide).

*unit*
Identifies the unit name which is to be used for the DFSMShsm restore of the VSAM sphere

**Usage Notes**

Consider these points when you use the RECOVER command:

- You can use multiple RECOVER commands in a single CICSVR run.
- It is assumed that the VSAM sphere to be recovered has been restored and is in a usable state. The CICSVR dialog interface recalls DFSMShsm or DFSMSdss backups for you and generates the appropriate RECOVER command and keywords.
- Only use the STARTTOD and STOPTOD keywords if the STARTTIME and STOPTIME values are not precise enough for your recovery run. You can print the MVS log stream to get the TOD values that you need for the recovery.
- RECOVER processing sequentially applies all changes that are found on the specified logs. It uses information in the dsname records to decide which after-images to apply. These records contain the file names of every data set that was opened for the base cluster or path. RECOVER applies the after-images to the VSAM sphere as follows:
  - Records that were updated are overwritten by their after-images.
  - Records that were added are added using their after-images.
  - Records that were deleted are deleted.
• You should specify the FCTCOMP command in your recovery run if you are using CICS V2 logs and recovering a file that is defined in the CICS FCT as being fixed record format.

• If the forward recovery run fails after starting to update the VSAM sphere, you must restore a new copy of the VSAM sphere before rerunning CICSVR.

• The base cluster specified in NEWSPHERE must have the same data set characteristics as the base cluster specified in SPHERE, regarding:
  – Control interval size
  – Maximum record size
  – Key length and offset
  – Data set format (that is, RRDS, VRRDS, KSDS, or ESDS)

### Examples

```
RECOVER  -  
   ONLY  -  
   VERSION(01)  -  
   APPLYCA  -  
   SPHERE(PAYROLL.MONTH5.BASE)
```

*Figure 89. RECOVER with DFSMShsm backup*

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

Because no STARTTIME or STARTTOD keyword is specified, recovery starts after the first TUR at the first after-image that is encountered on the logs for that VSAM sphere.

Because no STOPTIME or STOPTOD keywords are specified, recovery stops when all logs have been exhausted.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

The VERSION keyword tells CICSVR to restore the VSAM sphere from DFSMShsm backup Version 1.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

```
RECOVER  -  
   ONLY  -  
   STARTTIME(01.001/11:59:59)  -  
   STOPTIME(01.002/11:59:59)  -  
   APPLYCA  -  
   STARTAT(OSNAME)  -  
   SPHERE(CA.BASE02)
```

*Figure 90. RECOVER with no DFSMShsm or DFSMSdss backup*

Forward recovery processing is requested for the VSAM sphere that is named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.
Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery. The STARTTTIME value must be prior to the initial tie-up record in the forward recovery log.

The STARTTIME keyword tells CICSVR to start the recovery on 01.001 at 11:59:59.

The STOPTIME keyword tells CICSVR to stop the recovery on 01.002 at 11:59:59.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

The STARTAT keyword tells CICSVR to start the recovery at the first TUR after the time specified that is in the STARTTIME keyword.

```
RECOVER
  ONLY
  STOPTIME(01.002/00:09:10)
  APPLYCA
  FORMAT(OLD)
  SPHERE(CA.BASE02)
```

Figure 91. RECOVER with no DFSMShsm or DFSMSdss backup and CICS V2 log

Forward recovery processing is requested for the VSAM sphere that is named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

Because no STARTTTIME or STARTTOD keywords are specified, recovery starts at the first after-image met on the logs for that VSAM sphere, after the first TUR.

The STOPTIME keyword tells CICSVR to stop the recovery on date 01.002 at 00:09:10.

Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

The FORMAT keyword tells CICSVR that the CICS log is in the CICS V2 format.

```
RECOVER
  ONLY
  STARTTOD(B5E727FC28544000)
  VERSION(1)
  VOLUME(TSO001)
  UNIT(3390)
  APPLYCA
  SPHERE(CA.BASE02)
```

Figure 92. RECOVER with DFSMShsm backup and STARTTOD
Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

The STARTTOD keyword tells CICSVR to begin the recovery at the value specified by the hex TOD STARTTOD keyword. This value is in the default, local format.

Because no STOPTIME or STOPTOD keywords are specified, recovery stops when all logs have been exhausted.

The VERSION keyword tells CICSVR to restore the VSAM sphere from DFSMSHsm backup Version 1.

The VOLUME and UNIT keywords tell CICSVR that the VSAM sphere will be restored to volume TSO001 on a 3390 disk.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

```plaintext
RECOVER -
SHADOW -
NEWSPHERE(PAYROLL.MONTH5.SHADOW) -
SPHERE(PAYROLL.MONTH5.BASE)
```

*Figure 93. RECOVER with SHADOW copy*

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The SHADOW keyword tells CICSVR that the data set specified with NEWSPHERE will be a shadow copy of the data set specified on the SPHERE keyword. You can use a shadow copy as a replacement to the user’s VSAM sphere.

The NEWSPHERE keyword tells CICSVR that the VSAM sphere to be recovered has a different data set name than the one recorded on the log.

*Note:* This RECOVER command should be run at regularly scheduled intervals so that the shadow copy is up-to-date with the user’s VSAM sphere.

**Synonyms**

Table 20 provides the RECOVER commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

*Table 20. RECOVER synonyms*

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVER</td>
<td>RO, RECO, RECONLY, RECOV</td>
</tr>
<tr>
<td>NEWSPHERE</td>
<td>NE wspH, NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
<tr>
<td>SPHERE</td>
<td>SPH, DATASETNAME, DSNAME, DSN</td>
</tr>
</tbody>
</table>
Table 20. RECOVER synonyms (continued)

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTTIME</td>
<td>STARTDATETIME,</td>
</tr>
<tr>
<td></td>
<td>STARTDATE, START</td>
</tr>
<tr>
<td>STOPTIME</td>
<td>STOPDATETIME,</td>
</tr>
<tr>
<td></td>
<td>STOPDATE, STOP</td>
</tr>
<tr>
<td>VERSION</td>
<td>VER</td>
</tr>
</tbody>
</table>
SPHERE—Define spheres

Use the SPHERE command to define which spheres are in the CA group.

Format

```
SPHERE NAME(sphere name),
BACKUPTIME(date and time,GMT)
```

Keywords

**NAME(sphere name)**
Identifies the name of the sphere in the change accumulation group. Recovery information is collected in the change accumulation data set for each sphere that is listed. At least one SPHERE keyword is required for each CA control statement.

*sphere name*
Defines the name of the sphere. This name is 1–44 characters.

**BACKUPTIME(date and time,LOCAL|GMT)**
Identifies the date and time of the backup. This parameter is optional and is only specified when DFSMSshsm or DFSMSdss is not being used for your data set backups. If this parameter is specified, you must update this date and time each time you create a new data set backup.

*date and time*
Must be in the format *yyyy/ddd/hh/mm/ss*, where:
- *yyyy* is the four digit year number (2001)
- *ddd* is the day of the year (001–366)
- *hh* is the hour of the day (00–23)
- *mm* is the number of minutes (00–59)
- *ss* is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
BACKUPTIME(2001.159/:23:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify `STARTTIME(2001.159)`, CICSVR will assume that the time segment is `00:00:00`.

**LOCAL**
Specifies that the date and time are in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**
Specifies that the date and time are in Greenwich mean time (GMT) format. You can specify GMT only if you are using MVS log streams.

Usage Notes

Consider the following when you use the SPHERE command:

- This command is required if the CA command is specified
• You can specify up to 50 SPHERE commands for each CA command
• Do not specify duplicate sphere names for a CA group
VSAMSTART—Start passing AMS commands to VSAM

Use the VSAMSTART command to tell CICSVR to start passing AMS commands to VSAM after a VSAM sphere has been recovered.

The VSAMSTART command has no synonyms.

Format

```
VSAMSTART DSN(data set name)
```

Keywords

**DSN(data set name)**

Specifies the data set for which recovery must be successfully completed before the AMS commands are passed to VSAM. The DSN keyword is required. You can only specify this keyword once per VSAMSTART command.

**data set name**

Specifies the data set name, which is 1–44 characters.

Usage Notes

Consider the following when using the VSAMSTART command:

- The DSN keyword must be on the same line as the VSAMSTART command.
- You can use the VSAMSTART/VSAMEND combination to rebuild AIXs after recovering a VSAM sphere that was backed up using BWO.

Examples

```
VSAMSTART DSN(TEST.PAYROLL.BASE)
  BLDINDEX -
  INDATASET(TEST.PAYROLL.BASE) -
  OUTDATASET(TEST.PAYROLL.PATH)
VSAMEND DSN(TEST.PAYROLL.BASE)
```

This pair of VSAMSTART and VSAMEND commands tells CICSVR to pass the AMS BLDINDEX command to VSAM when the recovery of TEST.PAYROLL.BASE has successfully completed. The AIX for TEST.PAYROLL.BASE will be built by VSAM when CICSVR has recovered TEST.PAYROLL.BASE.

For details of the VSAMEND command, see "VSAMEND—Stop passing AMS commands to VSAM" on page 203.
VSAMEND—Stop passing AMS commands to VSAM

Use the VSAMEND command to tell CICSVR to stop passing AMS commands to VSAM.

The VSAMEND command has no synonyms.

Format

`VSAMEND DSN(data set name)`

Keywords

`DSN(data set name)`

Specifies the data set in the corresponding VSAMSTART command. The DSN keyword is required. You can only specify this keyword once per VSAMEND command.

data set name

Specifies the data set name, which is 1–44 characters.

Usage Notes

Consider the following when using the VSAMEND command:

- The DSN keyword must be on the same line as the VSAMEND command.
- You can use the VSAMSTART/VSAMEND combination to rebuild AIXs after recovering a VSAM sphere that was backed up using BWO.

Examples

```
VSAMSTART  DSN(TEST.PAYROLL.BASE)
    BLDINDEX -
    INDATASET(TEST.PAYROLL.BASE) -
    OUTDATASET(TEST.PAYROLL.BASE) -
VSAMEND    DSN(TEST.PAYROLL.BASE)
```

This pair of VSAMSTART and VSAMEND commands tells CICSVR to pass the AMS BLDINDEX command to VSAM when the recovery of TEST.PAYROLL.BASE has successfully completed. The AIX for TEST.PAYROLL.BASE will be built by VSAM when CICSVR has recovered TEST.PAYROLL.BASE.

For details of the VSAMSTART command, see "VSAMSTART—Start passing AMS commands to VSAM" on page 202.
This chapter describes the following CICSVR functions and commands.

**Table 21. CICSVR commands used with CICS V4**

<table>
<thead>
<tr>
<th>Function</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive utility</td>
<td>ARCHIVE&lt;br&gt;DEFEXIT&lt;br&gt;FCTCOMP&lt;br&gt;NOFORCE</td>
</tr>
<tr>
<td>Forward recovery and backout</td>
<td>ALLOCATE&lt;br&gt;BACKOUT&lt;br&gt;BLDVRP&lt;br&gt;DEFEXIT&lt;br&gt;FCTCOMP (used with CICS V2)&lt;br&gt;RECOVER&lt;br&gt;RELATE&lt;br&gt;SEQCHKL&lt;br&gt;SEQCHKR&lt;br&gt;VSAMSTART&lt;br&gt;VSAMEND</td>
</tr>
<tr>
<td>Change accumulation</td>
<td>CA&lt;br&gt;SPHERE</td>
</tr>
<tr>
<td>Export or import the RCDS</td>
<td>RCDS</td>
</tr>
</tbody>
</table>
ALLOCATE—Allocate a log

Use the ALLOCATE command to identify the CICS log or log stream copy that you want CICSVR to read. Do not use ALLOCATE for MVS log streams. For MVS log streams, use the MVSLOG command.

Format

```
ALLOCATE LOG(dsn1,dsn2,...)  
VOLUME(vol1,vol2,...)  
UNIT(unit1,unit2,...)
```

Keywords

**LOG(dsn1,dsn2,...)**

Specifies the logs in ascending time order, with the earliest specified first. The LOG keyword is required.

`dsn1,dsn2,...`

Specifies the log data set name, which is 1–44 characters. You can specify one or more logs.

**VOLUME(vol1,vol2,...)**

Specifies the first volume serial number that the log resides on. The VOLUME keyword is required only if at least one of the logs specified is not cataloged. If specifying more than one log, where some are cataloged and some are not, use an asterisk (*) to denote the volume for the cataloged logs.

`vol1,vol2,...`

Specifies the volume field, which is either 1–6 characters, or an asterisk (*) to denote the volume for the cataloged logs. You can specify one or more volumes that correspond to the number of logs you have specified.

**UNIT(unit1,unit2,...)**

Specifies the device type that the log resides on. The UNIT keyword is required only if at least one of the logs specified is not cataloged. If specifying several logs, where some are cataloged and some are not, use an asterisk (*) to denote the unit for the cataloged logs.

`unit1,unit2,...`

Specifies the unit name, which is 1–8 characters or use an asterisk (*) to denote the volume for the uncataloged logs. You can specify one or more unit types, corresponding to the number of logs you have specified.

Usage Notes

Consider the following information when you use the ALLOCATE command:

- You can specify these types of logs to CICSVR:
  - CICSVR SAM copies of MVS log streams
  - CICS V4 logs
- You can specify the log in a DWWLOG DD instead of specifying the ALLOCATE statement.
• You cannot specify MVS log streams using the ALLOCATE command. You must use the MVSLOG command (see page "MVSLOG—Specify an MVS log stream" on page 186).

• You cannot use CICS Transaction Server logs for backout. CICS Transaction Server provides online backout failure support.

• If you use the ALLOCATE command to tell CICSVR to read a log stream copy, you must also specify the MVSLOG command to tell CICSVR it is a SAM copy of an MVS log stream.

• You must know which backup to restore and which logs to use for your CICSVR run. If you use DFSMShsm or DFSMSdss, CICSVR displays the latest backup or lets you choose from a list of valid backups. If you do not use DFSMShsm or DFSMSdss, you can obtain this information from the CICSVR archive reports if you use CICS V4 or from the scan reports if you use CICS TS.

• Base and path updates must be recorded on the same sequence of logs for a given sphere.

• Attention: If you use CICS V4 you can run forward recovery using disk logs that are currently active in CICS, but you must ensure that these logs are not overwritten while they are being processed by CICSVR. This is not recommended. Instead, use 2 disk logs and set up CICS V4 to automatically call CICSVR archive to copy the disk log. You can process active logs while the CICS system is inactive.

• You have the option of allocating your logs through dynamic allocation (with the ALLOCATE command) or through ddnames in the CICSVR JCL. The JCL allocation differs for the BACKOUT and RECOVER commands.

• If you use the ALLOCATE command to specify your logs, keep each uncataloged archived log on a single volume.

• All logs that are needed for CICSVR recovery or backout must be provided in a single step.

Attention
Do not split the logs needed for recovery into several CICSVR steps. This might cause loss of data integrity without warning.

• Only one ALLOCATE command is permitted per CICSVR step.

Examples

There is a positional correspondence between the ALLOCATE keyword values. That is, if three log data set names are supplied, there must be three VOLUME and UNIT keywords (if one or more of the logs are uncataloged).

ALLOCATE LOG(CICSA.ARCH1 CICSA.NARCH1 CICSA.ARCH2) - 1
    VOLUME(* * 123456) - 2
    UNIT(* * TAPE) - 3

Here is an explanation of each of the ALLOCATE commands:

1 The ALLOCATE command describes three logs.
2 The VOLUME command describes the first two logs as being cataloged; the third log is uncataloged and resides on volume serial number 123456.
3 The UNIT command describes the first two logs as being cataloged; the third log is uncataloged and resides on tape.

Chapter 13. CICSVR commands used with CICS V4 207
Table 22 provides the ALLOCATE commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

Table 22. ALLOCATE synonyms

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCATE</td>
<td>ALLOC</td>
</tr>
<tr>
<td>LOG</td>
<td>JOURNAL, JNL</td>
</tr>
<tr>
<td>VOLUME</td>
<td>VOL</td>
</tr>
</tbody>
</table>
ARCHIVE—Archive and register a log

Archive scans the records in a CICS V4 log, registers the log in the RCDS, and stores a summary of those records in the RCDS. It can also, optionally, copy the log to a maximum of nine data sets. You can produce a report that contains the information that you need if you must manually construct a CICSVR forward-recovery or backout job.

With CICS V4, IBM recommends you set up a CICSVR archive job in the CICS JPDS, so that the job is automatically submitted when the log is switched. The archive keywords entered on the PARM parameter can be coded as symbolic values and set by CICS V4 when the job is run. See Figure 66 on page 115 for more information about setting up the Journal Archive Job Skeleton.

Format

```
ARCHIVE—CICSID(APPLID)
    JOURNALID(journal-label record value | log ID)
    MVSID(blank field | MVS system name)
    COPIES(number of copies)
    AUTODEREG(NO | YES)
    RECOVERYREPORT(NO | YES)
    NO MOD
```

Keywords

CICSID(APPLID)

Specifies the name of the CICS system that produced this log. Specify one CICSID keyword for each ARCHIVE command. The CICSID keyword is required if it is not provided in the PARM parameter of the EXEC statement in the JCL.

APPLID

Specifies the 1–8 character application identifier (APPLID) of the CICS system that produced this log.

JOURNALID(journal-label record value | log ID)

Shows the log identification (ID) number of the CICS log.

log ID

Specifies 01 (for the system log), or 02–99 (for other logs). The JOURNALID keyword is optional; if it is not specified, CICSVR uses the ID number from the log as the default value. The JOURNALID keyword can be provided on the PARM parameter of the EXEC statement in the JCL.

MVSID(blank field | MVS system name)

Identifies the MVS system where CICS is running.

MVS system name

Specifies the name you want to assign to the MVS system. The MVSID keyword is optional; if it is not specified, CICSVR uses a blank field as the
default value. The MVSID keyword can be provided on the PARM parameter of the EXEC statement in the JCL.

**COPYES**(*1 | number of copies*)

Specifies if the log is to be copied to another data set.

*number of copies*

Specifies the number of copies to be made from the input log. The COPIES keyword is optional; if it is not specified, the default value of 1 is used.

- If COPIES(0) is specified, the input log will be registered in the RCDS and no copy will be made.
- If the number of copies specified is greater than zero, the log is copied to the data sets named on the corresponding ddnames, DWWCOPYn, where n is a number, 1–9, that is specified on the COPIES keyword. Only the log copied to DWWCOPY1 is registered in the RCDS.

**AUTODEREG**(*YES | NO*)

Specifies if the automatic deregistration function should be invoked. The automatic deregistration parameters must be set in the CICSVR dialog if you want to invoke automatic deregistration.

*YES*

Indicates that automatic deregistration processing is to be invoked according to the dialog setting. The default is YES.

*NO*

Indicates that automatic deregister processing should not be invoked.

The AUTODEREG keyword is optional; if it is not specified, the default is to run the automatic deregister function. The AUTODEREG keyword can be provided on the PARM parameter of the EXEC statement in the JCL.

**RECOVERYREPORT**(*NO | YES*)

Specifies if a recovery report should be printed.

*NO*

Specifies no report is required.

*YES*

Specifies that a recovery report is produced for all the spheres that are on the journal specified by JOURNALID. A recovery report is useful if you do not use DFSMSHsm or DFSMSdss and you want to find out which logs are on the specified journal and need to be recovered.

The RECOVERYREPORT keyword is optional; if it is not specified, the default is to suppress the recovery report. The RECOVERYREPORT keyword can be provided on the PARM parameter of the EXEC statement in the JCL.
MOD

Indicates that records are to be added to the end of the data set specified in the DWWCOPYn ddname. Use the MOD keyword if you specified DISP=MOD for any of the DWWCOPYn ddnames.

Usage Notes

Consider these points when you use the ARCHIVE command:

- The ARCHIVE command is started by the CICSVR module, DWWAR.
- Archive keywords can be coded in any sequence.
- You invoke the archive utility by running the required JCL, and providing the statements you need as either parameters on the DWWIN DD statement, or as input to a PARM parameter on the EXEC statement.

These archive command keywords, or their synonyms, can be entered on the PARM statement:
- CICSID
- JOURNALID
- MVSID
- AUTODEREG
- RECOVERYREPORT

If you code the same keyword as input to the DWWIN DD statement and on the PARM parameter, the value on the PARM parameter is used. You define the log that you want to archive on the DWWARC1 DD statement, and the data sets to which the log is to be copied, on the DWWCOPY1—DWWCOPY9 DD statements. See Figure 87 on page 154 for a sample archive job.

- Before the archive utility copies the log record to the output data set, a precopy exit can be called. This exit lets you inspect and act on a copy of the record. This exit must be defined during the archive run by the DEFEXIT command.
- Before the archive utility terminates, a termination exit can be called. This exit lets you terminate actions that were performed by the precopy exit.
- The CICSVR archive utility scans the log, detecting control errors, such as gaps in sequence numbers, or changes of log ID. If you specify RECOVERYREPORT(YES), the archive utility analyzes the data that was stored earlier in the RCDS. It compares this information with that found on the log, to decide if logs are missing, or if this was a repeat of a previous archive run. Messages are issued reporting errors found during the analysis.

For more information on the archive utility see "Using the archive utility to copy your logs" on page 153.
Examples

The archive utility will process the system log from the CICS system with an APPLID of PRODCICS, on the MVS system called MVS1. Three copies of the log have been requested. These copies will be made to the data sets specified on the ddnames DWWCOPY1, DWWCOPY2, and DWWCOPY3. Records will be copied to the end of the data sets specified in the DWWCOPYn ddnames. The RCDS is updated with the latest information from the log.

The archive utility will process a log from the CICS system with an APPLID of TESTCICS. The log ID is taken from the log, and no copy of the log will be made. Archive will produce a recovery report. The RCDS is updated with the latest information from the log.

The archive utility will process the log with an ID of 02, from the CICS system with an APPLID of PRODCICS, on the MVS system called MVS2. The automatic deregister function will not be triggered and no recovery report will be produced. This archive run will take less time to run, because AUTODEREG(NO) and RECOVERYREPORT(NO) have been specified. The RCDS is updated with the latest information from the log.

Note: The CICSVR commands DEFEXIT, FCTCOMP, and NOFORCE can also be specified in an archive run. For examples of these commands, see "DEFEXIT—Define CICSVR exit names for forward recovery and backout" on page 224, "FCTCOMP—pass FCT record-format information to CICSVR (use with CICS V2)" on page 227, and "NOFORCE—Do not force archive to stop after an error" on page 230.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHIVE</td>
<td>AR</td>
</tr>
<tr>
<td>AUTODEREG</td>
<td>DEREG</td>
</tr>
<tr>
<td>CICSID</td>
<td>CID</td>
</tr>
<tr>
<td>JOURNALID</td>
<td>JID</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Synonyms</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
</tr>
<tr>
<td>RECOVERYREPORT</td>
<td>RECREP</td>
</tr>
</tbody>
</table>
BACKOUT—VSAM sphere backout

Use the BACKOUT command to request CICSVR to back out uncommitted units of work (LUW) on a VSAM sphere that was updated by CICS V4. Backout uses the system log which contains before images of the VSAM sphere. CICSVR backout is only supported for CICS V4.

When CICS backs out changes, all updates to a VSAM sphere are backed out to the previous synchronization point (sync point). This ensures that the logical integrity of the VSAM sphere is maintained, by allowing only complete logical units of work to affect the VSAM sphere.

If CICS fails in the attempt to backout these uncommitted changes, you can use the CICSVR backout function to back them out. CICSVR backout removes the partially completed transactions by reading the before-images from the system log backward and applies them to the VSAM sphere. When the partially completed transactions are backed out, CICS can use the recreated VSAM sphere.

Format

\[ \text{BACKOUT} \text{SPHERE} \text{sphere name} \text{STARTTIME} \text{start date and time} \text{STOPTIME} \text{stop date and time} \text{NEWSPHERE} \text{updated sphere name} \]

Keywords

\text{SPHERE} \text{sphere name}

Specifies the VSAM sphere to be backed out. It is the MVS data set name as recorded in the system log. Only one SPHERE keyword is allowed for each BACKOUT command. This keyword is required.

\text{sphere name}

Specifies the MVS data set name of the VSAM sphere to be backed out. The name is 1–44 characters.

\text{STARTTIME} \text{start date and time}

Specifies the earliest timestamp of the before-images to be applied during backout of the VSAM sphere. CICSVR will not process before-images that have timestamps earlier than the date and time specified in the STARTTIME keyword.

The STARTTIME keyword provides extra checking to ensure that the correct system logs are used. For example, STARTTIME can be specified as the startup time of the CICS system on which uncommitted data was written to a VSAM sphere.

The STARTTIME keyword lets you use the same command keywords for backout and recovery. If you run backout after a recovery run, you can specify the same STARTTIME keyword as in the recovery job.

This keyword is optional. If STARTTIME is not specified, CICSVR reads the system logs backward until it has backed out all uncommitted changes, or until it finds the start of the earliest system log that you provided. If CICSVR finds
the start of the earliest system log and still has not completed the backout, a message requesting the earlier logs is displayed on the MVS operator console (see Usage notes).

Only one STARTTIME keyword is allowed for each BACKOUT control statement.

**start date and time**

Must be in the format `yy/ddd/hh/mm/ss`, where:

- `yy` is the last two digits of the year (00–99)
- `ddd` is the day of the year (001–366)
- `hh` is the hour of the day (00–23)
- `mm` is the number of minutes (00–59)
- `ss` is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
STARTTIME(01.159/22:23:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify `STARTTIME(01.159)`, CICSVR will assume that the time segment is 00:00:00.

CICSVR interprets year values (`yy`) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**STOPTIME (stop date and time)**

Specifies the latest timestamp of the before-images that you want backed out from the VSAM sphere. CICSVR ignores all before-images on the system log that were logged after STOPTIME. Only one STOPTIME keyword is allowed for each BACKOUT command. If STOPTIME is not specified, the backout will use all relevant before-images after STARTTIME.

**stop date and time**

Must be in the format `yy/ddd/hh/mm/ss`, where:

- `yy` is the last two digits of the year (00–99)
- `ddd` is the day of the year (001–366)
- `hh` is the hour of the day (00–23)
- `mm` is the number of minutes (00–59)
- `ss` is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
STOPTIME(01.159/00:30:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, you can specify day 01159, time 23:59:59, as follows:

```
STOPTIME(01.159)
```

If you specify this, CICSVR will assume that the day is 01365 and the time is 23:59:59:

```
STOPTIME(01)
```

CICSVR assumes that these values are for day 01159, and time 16:59:59:

```
STOPTIME(0115916)
```
CICSVR interprets year values (yy) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**NEWSPHERE(updated sphere name)**

Specifies the VSAM sphere to be backed out when the data set name differs from the one recorded on the system log. The NEWSPHERE keyword lets you back out a VSAM sphere copy without having to rename or delete the original first.

If you require CICSVR to back out a copy of the VSAM sphere (for example a restored backup copy), specify the data set name of the copy data set.

If you require CICSVR to back out a VSAM sphere itself, do not specify the NEWSPHERE keyword.

This keyword is optional. If it is omitted, the data set name specified in the SPHERE keyword will be used.

**updated sphere name**

Specifies the MVS data set name of the VSAM sphere to be backed out. The name is 1–44 characters.

### Usage Notes

Consider the following information when using the BACKOUT command:

- BACKOUT is only supported with CICS V4.
- BACKOUT cannot be used with MVS log streams.
- CICS Transaction Server logs cannot be used for backout. CICS Transaction Server provides online backout failure support.
- You can use multiple BACKOUT commands in a single CICSVR run.
- Backout only accesses the base cluster. BACKOUT applies to the following base cluster:
  - All changes that were made directly to the base cluster
  - All changes that were made to the base cluster using paths in the VSAM sphere in which the are base cluster belonged

If alternate index data sets exist, ensure that these are correctly backed out. If the complete upgrade set exists during backout, the alternate indexes will be updated by VSAM while CICSVR updates the base cluster. If only the base cluster exists during backout, create empty alternate indexes and build them using the access method services (AMS) BLDINDEX command, after the backout run.

- BACKOUT reads the system log backward, starting with the last record on the log specified. It selects the before-images to be applied, using information provided in backout failing log records (BOFLGRECs). It applies the before-images to the VSAM base cluster as follows:
  - Records that were updated are overwritten by their before-images.
  - Records that were deleted are inserted (that is, the before-images are inserted).
  - Records that were added are deleted (ESDSs are handled differently).

For deletion of records that were added to ESDSs, CICSVR uses the ESDS delete exit, if this exit is provided. You can code an ESDS delete exit that adds a marked-for-deletion code to these records for this purpose.

- If the backout run fails after starting to update the VSAM sphere, you must restore a new copy of the VSAM sphere before rerunning CICSVR.
• The base cluster you specify in NEWSPHERE must have the same data set characteristics as the logged base cluster you specify in SPHERE, for:
  – Control interval size
  – Maximum record size
  – Key length and offset
  – Data set format (that is, RRDS, KSDS, or ESDS)
• If the BACKOUT function has processed all the logs provided, and has not completed the backout process, the following message is issued on the MVS operator console:

DWW0500A  
Additional logs needed to finish backout. Enter name of a log or STOP.
• If this message appears, enter a reply on the MVS operator console. Specify the cataloged data set name of the log before the earliest log specified in the backout run.
You will also get this message if the data set that you are trying to process does not need to be backed out.
• Once BACKOUT has successfully completed, you can make the data set available to CICS.

Examples

```
BACKOUT SPHERE(PAYROLL.BASE) - 1
STARTTIME(01159/23:00:00) - 2
NEWSPHERE(PAYROLL.BASE.RESTORE) 3
```

Here is an explanation of each of the BACKOUT commands:

1. The BACKOUT command tells CICSVR that backout processing is required. The SPHERE keyword specifies the name of the VSAM sphere to be backed out as it appears on the log.
2. The STARTTIME keyword requests backout to go back to day 159 in year 2001 at 23:00:00.
3. The NEWSPHERE keyword tells CICSVR that the VSAM sphere to be backed out has a different data set name than the one recorded on the system log.

Synonyms

Table 24 provides the BACKOUT commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKOUT</td>
<td>BO</td>
</tr>
<tr>
<td>NEWSPHERE</td>
<td>NEPSH, NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
<tr>
<td>SPHERE</td>
<td>SPH, DATASETNAME, DSNAME, DSN</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>STARTDATETIME, STARTDATE, START</td>
</tr>
<tr>
<td>STOPTIME</td>
<td>STOPDATETIME, STOPDATE, STOP</td>
</tr>
</tbody>
</table>
BLDVRP—Build the VSAM resource pool

Use the BLDVRP command to request CICSVR to issue the build VSAM resource pool (BLDVRP) VSAM macro for a forward-recovery or backout run. This causes VSAM to generate a local shared resource (LSR) pool, which is a set of index and data buffers that are to be shared by all VSAM files participating in the CICSVR run.

You can improve the performance of your CICSVR run by using the BLDVRP command to specify the size and number of VSAM buffers.

Format

```
BLDVRP
   B512(number of buffers)
   B1K(number of buffers)
   B2K(number of buffers)
   B4K(number of buffers)
   B8K(number of buffers)
   B12K(number of buffers)
   B16K(number of buffers)
   B20K(number of buffers)
   B24K(number of buffers)
   B28K(number of buffers)
   B32K(number of buffers)
```

Keywords

`B512(number of buffers)...B32K(number of buffers)`

Defines the size and number of buffers in each buffer pool within the VSAM resource pool. You can only specify each buffer pool once. A VSAM file uses the buffer pool whose buffer size exactly matches the file’s CI size, or if this CI size is not available, the buffer pool with the next–larger buffer size.

`number of buffers`

Specifies the number of buffers to be defined for each buffer pool. The number of buffers must be in the range 3–65,535.

Usage Notes

Consider the following information when you use the BLDVRP command:

- Use the BLDVRP command to improve the performance of VSAM for a CICSVR run.
- The LSR is divided into subpools, each representing a different control interval size. Before reading the data and index control intervals, the buffers are scanned to see whether the request can be satisfied from the LSR pool. The BLDVRP command improves the performance of VSAM by reducing I/Os to disk.
- You can specify only one BLDVRP per CICSVR run.
- The BLDVRP command is optional. If you specify BLDVRP, it will apply to the entire CICSVR run.
- Use the AMS LISTCAT command to determine the CI size of the data and index components of the VSAM data set.
Attention

Familiarize yourself with the technique of specifying local shared resources before using the BLDVRP command in your CICSVR run; otherwise, you could inadvertently degrade performance. The more buffers you allocate, the greater the number of data and index control intervals that can be held in virtual storage. Over allocation can lead to severe paging on your system and can also increase the CPU time because of long buffer searches. With some investigation, you can find an optimal buffer size.

Examples

BLDVRP B4K(50) B2K(200)

This BLDVRP command defines 50 buffers of 4K and 200 buffers of 2K in the VSAM resource pool.

Synonyms

Table 25 provides the BLDVRP commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

Table 25. BLDVRP synonyms

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDVRP</td>
<td>BVRP</td>
</tr>
<tr>
<td>B1K</td>
<td>B1024</td>
</tr>
<tr>
<td>B2K</td>
<td>B2048</td>
</tr>
<tr>
<td>B4K</td>
<td>B4096</td>
</tr>
<tr>
<td>B8K</td>
<td>B8192</td>
</tr>
<tr>
<td>B12K</td>
<td>B12288</td>
</tr>
<tr>
<td>B16K</td>
<td>B16384</td>
</tr>
<tr>
<td>B20K</td>
<td>B20480</td>
</tr>
<tr>
<td>B24K</td>
<td>B24576</td>
</tr>
<tr>
<td>B28K</td>
<td>B28672</td>
</tr>
<tr>
<td>B32K</td>
<td>B32768</td>
</tr>
</tbody>
</table>
CA—Process all the log records

Use the CA command to define an existing change accumulation group and create or update the change accumulation data set for that group. Use an automated timer process, such as OPC, to regularly submit your change accumulation job.

Format

```
CA GROUP(group name) PREFIX(prefix name) VOLUME(volume) UNIT(unit)
```

Keywords

GROUP(group name)
Identifies the name of the change accumulation group. The GROUP keyword is required. Only one GROUP keyword is allowed for each CA control statement.

group name
Defines the name of the change accumulation group. This name is 1 - 36 characters.

PREFIX(prefix name)
Identifies the name of the high level qualifier that will be used for the dynamically created change accumulation output data set. PREFIX is optional. If specified, it overrides any value specified to the CICSVR server address space. Only one PREFIX keyword is allowed for each CA control statement.

prefix name
Defines the name of the high level qualifier that will be used for the dynamically created change accumulation output data set. This name is 1 to 8 characters.

VOLUME(volume)
Specifies the volume serial number that the change accumulation data set resides on. The VOLUME keyword is required if the output data set is not a SMS-managed data set. The data set name generated by CICSVR change accumulation will start with the prefix specified in the user prefix.

volume
Specifies the volume field, which is 1 - 6 characters.

UNIT(unit)
Specifies the device type that the change accumulation data set resides on. The UNIT keyword is required if the output data set is not a SMS-managed data set. The data set name generated by CICSVR change accumulation will start with the prefix specified in the user prefix.

unit
Specifies the unit name, which is 1 - 8 characters.

Usage Notes

Consider these points when you use the CA command:
If the CA command is specified, at least one SPHERE command must be specified in the same job.

You can specify only one CA command per CICSVR change accumulation group run.

Use CICSVR change accumulation for CICS forward recovery logs; CICS V4 or TS autojournals are not supported.
DEFEXIT—Define CICSVR exit names for the archive utility

Use the DEFEXIT command to request CICSVR archive or the log stream copy utility to call one or both of the exits.

Format

```
DEFEXIT
  PRECOPY(precopy exit name)
  TERMINATION(termination exit name)
```

Keywords

**PRECOPY(precopy exit name)**

Specifies that the precopy exit should be used. This exit gains control before every log record is written to the output data sets. You can then use the copy of this record for any purpose. You cannot change the original log record. The PRECOPY keyword is optional; only one PRECOPY keyword is allowed for each DEFEXIT control statement.

*precopy exit name*

Defines the name of the load module for the precopy exit. This name is 1–8 characters.

**TERMINATION(termination exit name)**

Specifies that the termination exit should be used, and gives the name of the exit program that CICSVR should call. This exit is called when CICSVR archive or the log stream copy utility is about to terminate normally. The TERMINATION keyword is optional; only one TERMINATION keyword is allowed for each DEFEXIT control statement.

*termination exit name*

Defines the name of the load module for the termination exit. This name is 1–8 characters.

Usage Notes

Consider these points when you use the DEFEXIT command:

- This command is optional.
- Do not include a DEFEXIT command automatically in every archive or log stream copy run. Write an exit program for a specific purpose, usually for a specific situation that you have investigated.
- The exit programs must reside in a data set that is defined in the JOBLIB, STEPLIB, or DWWLOAD ddname.
- You can only specify one DEFEXIT command per CICSVR archive or log stream copy run.

Examples

```
DEFEXIT PRECOPY(PRECOPX)  1
DEFEXIT TERMINATION(TERMX)  2
```

Here is an explanation of each of these DEFEXIT commands:
In this CICSVR run, the precopy exit will be given control before every log record is copied to the output data sets.

In this CICSVR run, the termination exit will be given control when CICSVR is about to terminate.

**Synonyms**

Table 26 provides DEFEXIT commands or keywords along with acceptable synonyms to use in place of the commands or keywords.

*Table 26. DEFEXIT Synonyms*

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFEXIT</td>
<td>X, DX, EXIT, EXITS</td>
</tr>
<tr>
<td>PRECOPY</td>
<td>COPY</td>
</tr>
<tr>
<td>TERMINATION</td>
<td>TERM</td>
</tr>
</tbody>
</table>
**DEFEXIT—Define CICSVR exit names for forward recovery and backout**

Use the DEFEXIT command to request CICSVR to call one or more of the exits.

**Format**

```
DEFEXIT
  PREAPPLY(preapply exit name,NEW | OLD)
  ERROR(error exit name)
  ESDSDELETE(esds delete exit name)
  TERMINATION(termination exit name)
```

**Keywords**

**PREAPPLY**(preapply exit name,NEW | OLD)

Specifies that the preapply exit should be used and gives the name of the exit program that CICSVR should call. Before every update that is made to the VSAM data set, CICSVR passes the log record and the corresponding data set record (unless the update is an add) to the exit program. The exit program, which can change the log record, must return an action code telling CICSVR which action to take. The PREAPPLY keyword is optional, and only one PREAPPLY keyword is allowed for each DEFEXIT command.

*preapply exit name,NEW | OLD*

Defines the name of the load module for the preapply exit. This name is 1–8 characters.

The NEW and OLD keywords are optional; if neither are specified, the default preapply exit type of NEW is used. If you are using CICS Transaction Server logs, specify NEW or use the default.

**ERROR**(error exit name)

Specifies that the error exit should be used, and gives the name of the exit program that CICSVR should call. When an I/O error occurs, CICSVR passes an error code and relevant error information to the exit program. The exit program must return an action code telling CICSVR which action to take. The ERROR keyword is optional, and only one ERROR keyword is allowed for each DEFEXIT command. If CICSVR meets a serious error and no error exit is provided, CICSVR issues the following message and terminates:

*DWW0212S*

Preceding I/O error has forced termination.

*error exit name*

Defines the name of the load module for the error exit. This name is 1–8 characters.

**ESDSDELETE**(esds delete exit name)

Specifies that the ESDS delete exit should be used and gives the name of the exit program that CICSVR calls. If a record must be backed out for an ESDS, CICSVR calls the ESDS delete exit program. The exit program updates the record with the correct marked-for-deletion code. It updates the log record and returns an action code that tells CICSVR which action to take. The ESDSDELETE keyword is optional; only one ESDSDELETE keyword is allowed.
for each DEFEXIT command. If CICSVR encounters a serious error and no error exit is provided, CICSVR issues the following message and terminates:

**DWW0505S**
Delete of a record on an ESDS is requested. The data set name is **dddd**.
Processing terminates.

An ESDS delete exit can only be used with the CICSVR backout function.

**esds delete exit name**
Defines the name of the load module for the ESDS delete exit. This name is 1–8 characters.

**TERMINATION**(*termination exit name*)
Specifies that the termination exit should be used, and gives the name of the exit program that CICSVR should call. When CICSVR is about to end normally, CICSVR passes the completion code to the exit program. The exit program can then change the completion code and return an action code telling CICSVR what action to take. The TERMINATION keyword is optional; only one TERMINATION keyword is allowed for each DEFEXIT command.

**termination exit name**
Defines the name of the load module for the termination exit. This name is 1–8 characters.

**Usage Notes**

Consider the following when using the DEFEXIT command:

- This command is optional.
- You can only specify one DEFEXIT command per CICSVR step.
- Do not include a DEFEXIT command automatically in every recovery run. The only situation where an exit is **required** is to backout an ESDS.

Write an exit program for a specific purpose, usually for a specific situation that you have investigated. For example, if I/O errors occur during a recovery run, you might want to write an error exit program that is appropriate for that situation and rerun CICSVR.

- The exit programs must reside in a data set that is defined in the JOBLIB, STEPLIB, or DWWLOAD ddname. You can also keep the exit programs in your CICSVR load library. If CICSVR cannot find the exit program, the following message is issued and CICSVR ends:

**DWW0206S**
The exit load module xxxx cannot be found.

**Examples**

Here is an explanation of each of these DEFEXIT commands:

1. The PREAPPLY keyword specifies the name of the preapply exit program, PREAPP. CICSVR passes the log record and the file to PREAPP. PREAPP can modify the record if needed, update the log, and return an action code back to CICSVR.

2. The ESDSDELETE keyword specifies the name of the ESDS delete exit
program, ESDSX. CICSVR calls ESDSX if a record must be backed out (deleted) from an ESDS. ESDSX can insert a marked-for-deletion code in the record, update the log, and return an action code back to CICSVR.

The ERROR keyword specifies the name of the error exit program, ERRX. CICSVR calls ERRX when an I/O error occurs and passes it information about the error. If there are logical errors on a VSAM sphere, ERRX may be able to correct them and then pass an appropriate action code back to CICSVR.

The TERMINATION keyword specifies the name of the termination program, TERMX. CICSVR calls TERMX just before CICSVR terminates. If, in the same run, an I/O error occurs while CICSVR is reading one of the logs or processing a VSAM file, TERMX may be able to recover, change the completion code accordingly, and return an action code back to CICSVR.

### Synonyms

Table 27 provides the DEFEXIT commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFEXIT</td>
<td>X, DX, EXIT, EXITS</td>
</tr>
<tr>
<td>ERROR</td>
<td>ERR</td>
</tr>
<tr>
<td>ESDSDELETE</td>
<td>ESDSDEL, EDEL</td>
</tr>
<tr>
<td>PREAPPLY</td>
<td>PREAPP, PRE</td>
</tr>
<tr>
<td>TERMINATION</td>
<td>TERM</td>
</tr>
</tbody>
</table>
FCTCOMP—pass FCT record-format information to CICSVR (use with CICS V2)

Use the FCTCOMP command to pass CICSVR or the CICSVR archive utility the record format of the records containing the after-images of the VSAM sphere. This information is available from the CICS file control table (FCT). CICSVR cannot access the information from the CICS FCT, so in some cases, you need to provide this information to CICSVR. The FCTCOMP command is not needed for logs created by CICS V3 or V4; it is only needed for logs created by CICS V2 or earlier.

It is not necessary to use the FCTCOMP command unless you need to override the CICSVR determined value. See Usage Notes on page 223 for an explanation of when you need to override the CICSVR determined value.

Format

```
FCTCOMP—FCTNAME(name)
  | RECFORM(VARIABLE | FIXED)
  | STARTTIME(start date and time)
```

Keywords

**FCTNAME(name)**

Specifies the name of the FCT entry containing the information on the sphere to be recovered. Only one FCTNAME keyword is allowed for each FCTCOMP command.

*name*

Is the file name specified in the FILE parameter of the FCT or RDO definition. This name is 1–8 characters.

**RECFORM(VARIABLE | FIXED)**

Specifies the record format of the records containing the after-images of the VSAM sphere. You can specify either VARIABLE or FIXED. The RECFORM keyword is optional; if it is not specified, the default is VARIABLE. If the default is used, CICSVR looks at the length field of the record. If CICSVR determines the length is invalid, the record format of FIXED is used.

**STARTTIME(start date and time)**

Specifies the start date or date-and-time that the record format took effect. If STARTTIME is not specified, the record format specified in the RECFORM keyword of the FCTCOMP command applies to all records that are updated through this FCT entry. Only one STARTTIME keyword is allowed for each FCTCOMP control statement.

*start date and time*

Must be in the format *yy/ddd/hh/mm/ss*, where:

*yy*  Is the last two digits of the year (00–99)
*ddd*  Is the day of the year (001–366)
*hh*  Is the hour of the day (00–23)
*mm*  Is the number of minutes (00–59)
*ss*  Is the number of seconds (00–59)
You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

\texttt{STARTTIME(01.159/22:23:00)}

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify \texttt{STARTTIME(01.159)}, CICSVR will assume that the time segment is 00:00:00.

CICSVR interprets year values \((yy)\) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

\section*{Usage Notes}

Consider the following when using the FCTCOMP command:

- You may need to override the CICSVR determined FCTCOMP RECFORM value if all of the following statements are true:
  - A CICS V2 log is used
  - The record contains a 4-byte length field in bytes 1-4
  - The RECFORM parameter specified in the CICS DFHFCT TYPE=FILE macro is not the default value of VARIABLE
- FCTCOMP is only needed for CICS V2 logs. If you accidentally specify it when you are using logs created by later versions of CICS, the command is ignored.
- CICSVR does not have access to the CICS V2 FCT, so it does not know if the after-images for a VSAM sphere have been created with fixed- or variable-length records. If FCTCOMP is not specified, CICSVR assumes that the after-images are in variable format and issues the following message:

\texttt{DWW0619I}

Logging for ddname \texttt{ddn} is assumed to be in variable format on CICS/MVS logs in the absence of any FCTCOMP commands.

- If you use a combination of fixed- and variable-length formats, always specify FCTCOMP. This eliminates confusion over the record lengths used for a particular recovery run.
- If you always define your CICS V2 VSAM files with an FCT definition of \texttt{RECFORM=(VARIABLE,BLOCKED)}, you do not need to use the FCTCOMP command.

\section*{Examples}

\begin{verbatim}
FCTCOMP FCTNAME(PAYROLL) -
RECFORM(FIXED) -
STARTTIME(01159/07:30:00)
FCTCOMP FCTNAME(PAYROLL)
\end{verbatim}

Here is an explanation of each of the FCTCOMP commands:

\begin{itemize}
  \item \textbf{1} The FCTCOMP command tells CICSVR that the FCT record-format information is to be provided.
    The FCTNAME keyword associates this FCTCOMP command with the CICS file name PAYROLL.
  \item \textbf{2} The RECFORM keyword specifies that PAYROLL was opened and logging with fixed-length records.
\end{itemize}
The STARTTIME keyword states that the record format took effect after date 01159 at 07:30.

The FCTCOMP statement tells CICSVR that FCT record-format information is to be provided.

The FCTNAME keyword associates this FCTCOMP command with the CICS file name PAYROLL. No other FCTCOMP keywords are specified, so CICSVR assumes that the default record format of VARIABLE applies to all records that are updated through this FCT entry. If this example of the FCTCOMP command was used in a CICSVR run, it would serve no purpose other than to suppress the following message:

DWW0619I
Logging for ddname ddn is assumed to be in variable format on CICS/MVS logs in the absence of any FCTCOMP commands.

**Synonyms**

Table 28 provides the FCTCOMP commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

<table>
<thead>
<tr>
<th>Command, keyword or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCTCOMP</td>
<td>FCTC, FCTCOMPLEMENT</td>
</tr>
<tr>
<td>FCTNAME</td>
<td>FCT</td>
</tr>
<tr>
<td>FIXED</td>
<td>FIX, FXD</td>
</tr>
<tr>
<td>RECFORM</td>
<td>RECFM, RECORDFORMAT</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>STARTDATETIME, STARTDATE, START</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>VAR</td>
</tr>
</tbody>
</table>
NOFORCE—Do not force archive to stop after an error

Use the NOFORCE command if you want the CICSVR archive utility to produce a message if it encounters I/O errors or sequence errors on the log, and to try to continue after a sequence error.

The NOFORCE command has no keywords or synonyms.

Format

```
NOFORCE
```

Usage Notes

Consider these points when using the NOFORCE command:

- With NOFORCE, if the CICSVR archive utility encounters errors, an error message is issued, and the log is flagged on the RCDS as being in error.
- The NOFORCE command applies to all log I/O and sequence errors in the archive run.
- If you do not specify NOFORCE, and CICSVR archive detects I/O errors or sequence errors, this is treated as an end-of-file situation. CICSVR archive stops and prints the reports.

**Attention**

Do not automatically include the NOFORCE command in your archive JCL.

If the archive utility detects certain errors on the logs and you specify the NOFORCE command, the archive utility will continue after flagging the log in the RCDS as an error. So the CICSVR archive utility might process log records that were already archived.
RCDS—Create a RCDS copy

Use the RCDS command to create a RCDS copy that can be sent (exported) to a remote site and used to update (import) the RCDS at the remote site. This will keep the remote site RCDS in synchronization with the RCDS at the primary site. Only RCDS records that are significant to the remote site are used to update the remote site’s RCDS.

Format

RCDS EXPORT
RCDS IMPORT

Keywords

EXPORT
Specifies that the information, which is stored in the RCDS, is copied into the SAM data set specified on the DWWCOPY1 DD statement.

IMPORT
Specifies that the information, which was previously exported, is loaded from the SAM data set specified on the DWWCOPY1 DD statement into the RCDS.

Usage Notes

Consider these points when you use the RCDS command:

• To use the RCDS utility, execute the CICSVR module DWWGJCDS.
• You can only have one RCDS command in a single job step.
• You invoke the RCDS utility by running the required JCL.
• You must specify the SAM data set that will contain the output of the EXPORT or the input for the IMPORT on the DWWCOPY1 DD STATEMENT.
• Before you use the RCDS utility on a remote site, you need to install CICSVR V3R1 and you must allocate an RCDS.

Examples

RCDS EXPORT 1
RCDS IMPORT 2

Here is an explanation of each of these RCDS commands:

1 This RCDS command tells CICSVR to copy all the necessary RCDS information to the SAM data set specified on the DWWCOPY1 DD statement. The SAM data set can be sent to a remote recovery site.

2 This RCDS command tells CICSVR to import the information, stored in the SAM data set specified on the DWWCOPY1 DD statement, into the RCDS. The information in the SAM data set must have been created using the RCDS EXPORT command.

Synonyms

Table 29 on page 232 provides RCDS commands or keywords along with acceptable synonyms to use in place of the commands or keywords.
### Table 29. RCDS Synonyms

<table>
<thead>
<tr>
<th>Command, keyword or value</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT</td>
<td>EX</td>
</tr>
<tr>
<td>IMPORT</td>
<td>IM</td>
</tr>
</tbody>
</table>
**RECOVER—Forward recover a VSAM sphere**

Use the RECOVER command to tell CICSVR to forward recover a VSAM sphere.

IBM recommends that you use the CICSVR dialog interface to generate the CICSVR jobs and not code the CICSVR statements directly. The CICSVR dialog interface performs the following tasks:

- Runs the log of logs scan to get the latest information about the logs and updates the CICSVR RCDS with this information.
- Restores the DFSMShsm or DFSMSdss backup automatically.
- Generates the necessary CICSVR statements automatically.

**Format**

```
RECOVER SPHERE(sphere name),ONLY,FIRST,INTERMEDIATE,LAST

STARTTIME(start date and time,GMT)

STOPTIME(stop date and time,GMT)

STARTTOD(start time of day,GMT)

STOPTOD(stop time of day,GMT)

NEWSPHERE(updated sphere name)

SHADOW APPLYCA FORMAT(NEW OLD) STARTAT(DSNAME)

VERSION(DFSMShsm version number)

HSM DATE(date)

VOLUME(volume)

UNIT(unit)
```
Keywords

**SPHERE**(*sphere name*)

Specifies the VSAM sphere to be forward recovered. It is the name as recorded on the log. Only one SPHERE keyword is allowed for each RECOVER control statement. The SPHERE keyword is required.

*sphere name*

Specifies the MVS data set name of the VSAM sphere to be forward recovered. This name is 1–44 characters.

**ONLY | FIRST | INTERMEDIATE | LAST**

Specifies the relative order of the steps in a job. These keywords are mutually exclusive. One of these keywords must be specified on a forward recovery command when the NEWSPHERE keyword has not be specified. A forward recovery job is composed of one or more steps. CICSVR must know the relative forward recovery step that is executing. If there is only one forward recovery step then specify ONLY. If there are two forward recovery steps, then specify FIRST on the first step and specify LAST on the last step. When there are more than two steps, specify INTERMEDIATE on all the intermediate forward recovery steps (the steps that are not the first or the last). This is handled automatically for you through the CICSVR panels.

**STARTTIME**(*start date and time*, LOCAL | GMT)

Specifies the earliest timestamp of the after-images that you want to recover the VSAM sphere from. CICSVR ignores all after-images on the logs until it finds an after-image for this VSAM sphere, whose timestamp is the same as, or later than, the time specified in the STARTTIME keyword. CICSVR applies all after-images for this VSAM sphere, from this place in the log, if a tie-up record (TUR) has been encountered on the log. Only one STARTTIME keyword is allowed for each RECOVER control statement. If you do not specify STARTTIME or STARTTOD, recovery will start at the first after-image encountered on the logs, after the first TUR.

For VSAM data sets restored from a backup-while-open copy, STARTTIME will be the recovery point time. When you use the CICSVR dialog interface CICSVR will automatically retrieve the recovery point time from the integrated catalog facility (ICF) catalog, so you do not need to specify STARTTIME in this situation.

*start date and time*

Must be in the format *yy/ddd/hh/mm/ss*, where:

- *yy* is the last two digits of the year (00–99)
- *ddd* is the day of the year (001–366)
- *hh* is the hour of the day (00–23)
- *mm* is the number of minutes (00–59)
- *ss* is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

STARTTIME(01.159/22:23:00)

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify STARTTIME(01.159), CICSVR will assume that the time segment is 00:00:00.
CICSVR interprets year values (yy) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**LOCAL**

Specifies that the date and time are in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**

Specifies that the date and time are in Greenwich mean time (GMT) format. You can specify GMT only if you are using MVS log streams.

**STOPTIME** *(stop date and time, LOCAL | GMT)*

Specifies the latest timestamp of the after-images that you want to recover the VSAM sphere from. CICSVR ignores all after-images on the logs after it finds an after-image for this VSAM sphere, whose timestamp is later than the time specified in the STOPTIME keyword. CICSVR applies all after-images for this VSAM sphere up to this place in the log. Only one STOPTIME keyword is allowed for each RECOVER control statement. If STOPTIME or STOPTOD is not specified, recovery will stop after the last supplied log.

*stop date and time*

Must be in the format `yy/ddd/hh/mm/ss`, where:
- **yy** is the last two digits of the year (00–99)
- **ddd** is the day of the year (001–366)
- **hh** is the hour of the day (00–23)
- **mm** is the number of minutes (00–59)
- **ss** is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
STOPTIME(01.159/00:30:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, you can specify day 01159, time 23:59:59, as follows:

```
STOPTIME(01.159)
```

If you specify this, CICSVR will assume that the day is 01365 and the time is 23:59:59:

```
STOPTIME(01)
```

CICSVR assumes that these values are for day 01159, and time 16:59:59:

```
STOPTIME(0115916)
```

CICSVR interprets year values (yy) in the range 00–85 to be years 2000–2085, and year values in the range 86–99 to be years 1986–1999.

**LOCAL**

Specifies that the date and time are in local format. LOCAL is the default value.

**GMT**

Specifies that the date and time are in GMT format. You can specify GMT only if you are using MVS log streams.

**STARTTOD** *(start time of day, LOCAL | GMT)*

Identifies the earliest time, in hexadecimal time-of-day (TOD) format, of the
after-images that you want to recover the VSAM sphere from. CICSVR ignores all after-images on the logs until it finds an after-image for this VSAM sphere, whose TOD value is the same as, or later than, the time specified in the STARTTOD keyword. CICSVR applies all after-images for this VSAM sphere, from this place in the log, if a tie-up record (TUR) has been encountered on the log. Only one STARTTOD keyword is allowed for each RECOVER control statement. If you do not specify STARTTOD or STARTTIME, recovery will start at the first after-image encountered on the logs, after the first TUR.

Use the STARTTOD keyword only if the STARTTIME keyword value is not precise enough for your recovery stop time. The STARTTOD keyword has no synonyms.

**start time of day**

Must be 16 hex characters. Here is an example of a STARTTOD keyword:

```
STARTTOD(AC47C0403792C101)
```

**LOCAL**

Specifies that TOD value is in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**

Specifies that the TOD value is in GMT format.

**STOPTOD**(start time of day, LOCAL | GMT)

Identifies the latest time, in hexadecimal time-of-day (TOD) format, of the after-images that you want to recover the VSAM sphere from. CICSVR ignores all after-images on the logs after it finds an after-image for this VSAM sphere, whose TOD value is later than the time specified in the STOPTOD keyword. CICSVR applies all after-images for this VSAM sphere up to this place in the log. Only one STOPTOD keyword is allowed for each RECOVER control statement. If STOPTOD or STOPTIME is not specified, recovery will stop after the last supplied log.

Use the STOPTOD keyword only if the STOPTIME keyword value is not precise enough for your recovery stop time. The STOPTOD keyword has no synonyms.

**stop time of day**

Must be 16 hex characters. Here is an example of a STOPTOD keyword:

```
STOPTOD(AC47C0403792C101)
```

**LOCAL**

Specifies that TOD value is in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

**GMT**

Specifies that the TOD value is in GMT format.

**NEWSPHERE**(updated sphere name)

Specifies the new MVS data set name of the VSAM sphere to be recovered. This name is 1–44 characters.

This keyword is optional, except when SHADOW is specified and it is then required because a shadow is another copy of the data set. If this keyword is omitted, the data set name specified by the SPHERE keyword is used.

Identifies the VSAM sphere to be recovered when the data set name differs from the one recorded on the log. The NEWSPHERE keyword lets you recover a VSAM sphere without having to first rename or delete the original.

If you require CICSVR to recover a copy of the VSAM sphere (for example a restored backup copy), specify the data set name of the copy data set.
If you require CICSVR to recover the VSAM sphere itself, do not specify this keyword, or specify the same data set as in the SPHERE keyword.

This keyword is optional. If it is omitted, the data set name specified by the SPHERE keyword will be used.

updated sphere name
   Specifies the MVS data set name of the VSAM sphere to be recovered.
   This name is 1–44 characters.

SHADOW
   The SHADOW keyword is optional and has no synonyms.
   The concept is that the first time (run 1) that you run a shadow forward recovery, the backup is restored and some amount of the forward recovery log is applied. Subsequent shadow forward recovery runs (2 - n) apply more of the forward recovery log with the intent to keep the shadow copy as current as possible.
   CICSVR saves the status of the shadow forward recovery in the RCDS, so the subsequent shadow forward recovery runs (2 - n) should not specify a start and stop time and should be run frequently, so as to minimize the log records to be applied.
   When a problem occurs and a real forward recovery is required, the CICSVR shadow can be copied to replace the user data set.

APPLYCA
   The APPLYCA keyword is optional and has no synonyms.
   APPLYCA indicates to CICSVR that there is change accumulation data that is relative to this data set that needs to be applied during forward recovery processing. If there is no change accumulation data set or CICSVR detects an error with the CA data set, the APPLYCA keyword is ignored.

FORMAT(NEW | OLD)
   Specifies the logging format of the forward recovery log. The default value (FORMAT=NEW) is the correct setting in most situations. FORMAT=OLD is a migration aid and is only necessary if you have recently migrated from CICS V2 to CICS V4 and you want to use the FCT definitions you set up for CICS V2.
   If you are using CICS TS or CICS V2, FORMAT is ignored.

   NEW
      Specifies that the log is a forward recovery log.
      NEW is the default value.

   OLD
      Specifies that the log is an autojournal with update and add before. Use OLD when the autojournal records match the CICS V2 log format.

STARTAT(REDO | DSNAME)
   Specifies where on the log the recovery will begin.

   REDO
      Specifies all after-images found after the time specified in the STARTTIME keyword will be used in this recovery, provided a TUR has been found on the log for this VSAM sphere.

   DSNAME
      Specifies the first TUR found on the log after the time specified in the STARTTIME keyword will be used. All after-images after this TUR and before the time specified in the STOPTIME keyword will be used in the
recovery. So any after-images found after the time specified in the STARTTIME keyword, and before the TUR for this VSAM sphere will be ignored.

The STARTAT keyword is optional; if it is not specified, the default of REDO will be used.

If you use the STARTAT keyword without the STARTTIME keyword, STARTAT is ignored.

The STARTAT keyword has no synonyms.

Note: When a file is opened, the association between the file name and the data set name is recorded on the log by a TUR.

**HSMDATE**(date)
Identifies the date of the DFSMSHsm full volume dump which is to be used to restore the VSAM sphere. For more information about the DFSMSHsm restore process, refer to [z/OS DFSMSHsm Storage Administration Guide](https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsshm-storage-administration-guide).

date
Specifies the DFSMSHsm full volume dump date. The date must be in the format **yyddd**.

**VERSION**(DFSMShsm version number)
Specifies the DFSMSHsm version number. For more information about the DFSMSHsm restore process, refer to [z/OS DFSMSHsm Storage Administration Guide](https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsshm-storage-administration-guide).

DFSMShsm version number
Identifies the version number of the DFSMSHsm backup that is to be restored. The version number can be 1–999.

**VOLUME**(volume)
Specifies the volume serial number where the DFSMSHsm backup of the VSAM sphere will be restored to. This keyword can only be used if you have specified VERSION. If you specify VOLUME you must also specify the UNIT keyword. For more information about the DFSMSHsm restore process, refer to [z/OS DFSMSHsm Storage Administration Guide](https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsshm-storage-administration-guide).

volume
Identifies the volume which is to be used for the DFSMSHsm restore of the VSAM sphere. The volume is 1–6 characters.

**UNIT**(unit)
Specifies the unit name where the DFSMSHsm backup of the VSAM sphere will be restored to. This keyword can only be used if you have specified VERSION. If you specify UNIT you must also specify the VOLUME keyword. For more information about the DFSMSHsm restore process, refer to [z/OS DFSMSHsm Storage Administration Guide](https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsshm-storage-administration-guide).

unit
Identifies the unit name which is to be used for the DFSMSHsm restore of the VSAM sphere

**Usage Notes**

Consider these points when you use the RECOVER command:
- You can use multiple RECOVER commands in a single CICSVR run.
• It is assumed that the VSAM sphere to be recovered has been restored and is in a usable state. The CICSVR dialog interface recalls DFSMShsm or DFSMSdss backups for you and generates the appropriate RECOVER command and keywords.

• Only use the STARTTOD and STOPTOD keywords if the STARTTIME and STOPTIME values are not precise enough for your recovery run. You can print the MVS log stream to get the TOD values that you need for the recovery.

• RECOVER processing sequentially applies all changes that are found on the specified logs. It uses information in the dsname records to decide which after-images to apply. These records contain the file names of every data set that was opened for the base cluster or path. RECOVER applies the after-images to the VSAM sphere as follows:
  – Records that were updated are overwritten by their after-images.
  – Records that were added are added using their after-images.
  – Records that were deleted are deleted.

• You should specify the FCTCOMP command in your recovery run if you are using CICS V2 logs and recovering a file that is defined in the CICS FCT as being fixed record format.

• If the forward recovery run fails after starting to update the VSAM sphere, you must restore a new copy of the VSAM sphere before rerunning CICSVR.

• The base cluster specified in NEWSPHERE must have the same data set characteristics as the base cluster specified in SPHERE, regarding:
  – Control interval size
  – Maximum record size
  – Key length and offset
  – Data set format (that is, RRDS, VRRDS, KSDS, or ESDS)

Examples

```
RECOVER -
ONLY -
VERSION(01) -
APPLYCA -
SPHERE(PAYROLL.MONTH5.BASE)
```

Figure 94. RECOVER with DFSMShsm backup

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

Because no STARTTIME or STARTTOD keyword is specified, recovery starts after the first TUR at the first after-image encountered on the logs for that VSAM sphere.

Because no STOPTIME or STOPTOD keywords are specified, recovery stops when all logs have been exhausted.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

The VERSION keyword tells CICSVR to restore the VSAM sphere from DFSMShsm backup Version 1.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.
Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery. The STARTTTIME value must be prior to the initial tie-up record in the forward recovery log.

The STARTTIME keyword tells CICSVR to start the recovery on 01.001 at 11:59:59.

The STOPTIME keyword tells CICSVR to stop the recovery on 01.002 at 11:59:59.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

The STARTAT keyword tells CICSVR to start the recovery at the first TUR after the time specified that is in the STARTTIME keyword.

Figure 95. RECOVER with no DFSMShsm or DFSMSdss backup

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery. The STARTTTIME value must be prior to the initial tie-up record in the forward recovery log.

The STARTTIME keyword tells CICSVR to start the recovery on 01.001 at 11:59:59.

The STOPTIME keyword tells CICSVR to stop the recovery on 01.002 at 00:09:10.

Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery.

Figure 96. RECOVER with no DFSMShsm or DFSMSdss backup and CICS V2 log

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

Because no STARTTTIME or STARTTOD keywords are specified, recovery will start at the first after-image met on the logs for that VSAM sphere, after the first TUR.

The STOPTIME keyword tells CICSVR to stop the recovery on date 01.002 at 00:09:10.

Because no VERSION or HSMDATE keywords are specified, CICSVR assumes that the user has restored a backup copy of the VSAM sphere using a product other than DFSMShsm (for example, using DFSMSdss) prior to the forward recovery.
The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

The FORMAT keyword tells CICSVR that the CICS log is in the CICS V2 format.

```
RECOVER -
  ONLY -
  STARTTOD(85E727FC28544000) -
  VERSION(1) -
  VOLUME(TSO001) -
  UNIT(3390) -
  APPLYCA -
  SPHERE(CA_BASE02)
```

Figure 97. RECOVER with DFSMShsm backup and STARTTOD

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The ONLY keyword tells CICSVR that only one RECOVER step is needed.

The STARTTOD keyword tells CICSVR to begin the recovery at the value specified by the hex TOD STARTTOD keyword. This value is in the default, local format.

Because no STOPTIME or STOPTOD keywords are specified, recovery will stop when all logs have been exhausted.

The VERSION keyword tells CICSVR to restore the VSAM sphere from DFSMShsm backup Version 1.

The VOLUME and UNIT keywords tell CICSVR that the VSAM sphere will be restored to volume TSO001 on a 3390 disk.

The APPLYCA keyword tells CICSVR to apply the change accumulation data if a change accumulation data set exists.

```
RECOVER -
  SHADOW -
  NEWSPHERE(PAYROLL.MONTH5.SHADOW) -
  SPHERE(PAYROLL.MONTH5.BASE)
```

Figure 98. RECOVER with SHADOW copy

Forward recovery processing is requested for the VSAM sphere named in the SPHERE keyword.

The SHADOW keyword tells CICSVR that the data set specified with NEWSPHERE will be a shadow copy of the data set specified on the SPHERE keyword. You can use a shadow copy as a replacement to the user’s VSAM sphere. For information on how to recover using a shadow copy, see Setting up the journal archive job skeleton with shadow forward recovery on page 113.

The NEWSPHERE keyword tells CICSVR that the VSAM sphere to be recovered has a different data set name than the one recorded on the log.

Note: This RECOVER command should be run at regularly scheduled intervals so that the shadow copy is up-to-date with the user’s VSAM sphere.
Table 30 provides the RECOVER commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

**Table 30. RECOVER synonyms**

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVER</td>
<td>RO, RECO, RECONLY, RECOV</td>
</tr>
<tr>
<td>NEWSPHERE</td>
<td>NEWSPH, NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
<tr>
<td>SPHERE</td>
<td>SPH, DATASETNAME, DSNNAME, DSN</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>STARTDATETIME, STARTDATE, START</td>
</tr>
<tr>
<td>STOPTIME</td>
<td>STOPDATETIME, STOPDATE, STOP</td>
</tr>
<tr>
<td>VERSION</td>
<td>VER</td>
</tr>
</tbody>
</table>
RELATE—Relate a path to the base cluster

Use the RELATE command if updates have been made through the path on CICS V2 logs, and the path data set name of the restored backup copy of the path is different from the data set name of the originally logged path.

Format

```
RELATE PATH(path data set name) NEWPATH(new path data set name)
```

Keywords

**PATH(path data set name)**

Specifies the data set name of the originally logged path. Only one PATH keyword is allowed for each RELATE command. The PATH keyword is required.

*path data set name*

Specifies the MVS data set name of the path. This name is 1–44 characters.

**NEWPATH(new path data set name)**

Specifies the data set name of the restored backup copy of the path, if different from the data set name of the originally logged path. Only one NEWPATH keyword is allowed for each RELATE command. The NEWPATH keyword is required.

*new path data set name*

Specifies the MVS data set name of the path. This name is 1–44 characters.

Usage Notes

Consider these points when using the RELATE command:

- RELATE only applies to CICS V2 logs.
- You can use multiple RELATE commands in a single CICSVR step.
- If you specify the NEWSPHERE keyword of the RECOVER command, you may also need to specify the RELATE command. RELATE tells CICSVR that the restored backup copy of the path is different from the original path.

Examples

```
RELATE PATH(PAYROLL.BASE.PATH1) -
        NEWPATH(PAYROLL.BASE.PATH1.BACKUP)
```

Here is an explanation of this RELATE command:

1. The RELATE command tells CICSVR that the data set name of the restored backup copy of the path is different from the data set name of the originally logged path.
   
   The PATH keyword specifies the name of the original operational path.

2. The NEWPATH keyword specifies the name of the restored backup copy of the path.
Synonyms

Table 31 provides the RELATE commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

Table 31. RELATE synonyms

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATE</td>
<td>REL</td>
</tr>
<tr>
<td>PATH</td>
<td>DATASETNAME, DSNNAME, DSN</td>
</tr>
<tr>
<td>NEWPATH</td>
<td>NEWDATASETNAME, NEWDSNAME, NEWDSN</td>
</tr>
</tbody>
</table>
SEQCHKL—Define sequence check criteria for log records

Use the SEQCHKL command to define the action that CICSVR should take if it encounters a journal-label-record sequence error when moving from one log to the next.

Note: Only use this command if you want to force CICSVR to use logs that are not in time sequence order or logs that contain old data.

Format

```
SEQCHKL

GAPINSEQUENCE( STOP | WARNING | IGNORE )
OUTOFSEQUENCE( STOP | WARNING | IGNORE )
RESETSEQUENCE( STOP | WARNING | IGNORE )
```

Keywords

**GAPINSEQUENCE(STOP | WARNING | IGNORE)**

Specifies the action that CICSVR should take when it detects a step up or down in the journal-label-record sequence number, when moving from one log to another (and the journal-label records are in the correct time sequence).

The GAPINSEQUENCE keyword is optional.

If you specify:
- **STOP**, the recovery run will end with a severe message
- **WARNING**, processing will continue and a warning message will be issued
- **IGNORE**, gaps in journal-label-record sequence numbers will be ignored and no messages will be issued

**OUTOFSEQUENCE(STOP | WARNING | IGNORE)**

For CICS V4 logs, it specifies the action that CICSVR should take when it detects a step up or down in the journal-label-record sequence number, when moving from one log to another (and the journal-label records are out of time sequence).

The OUTOFSEQUENCE keyword is optional.

If you specify:
- **STOP**, the recovery run will end with a severe message
- **WARNING**, processing will continue and a warning message will be issued
- **IGNORE**, gaps in journal-label-record sequence numbers will be ignored and no messages will be issued

**RESETSEQUENCE(STOP | WARNING | IGNORE)**

Specifies the action that CICSVR should take when it detects that the journal-label-record sequence number has been reset to zero before previously reaching $4 \times 2^{32} = 2^{32}-1$, when moving from one log to another (and the journal-label records are in the correct time sequence).
The RESETSEQUENCE keyword is optional.

**Note:** If the journal-label records are not in the correct time sequence, this condition will be treated the same as the OUTOFSEQUENCE condition.

If you specify:
- STOP, the recovery run will end with a severe message
- WARNING, processing will continue and a warning message will be issued
- IGNORE, gaps in journal-label-record sequence numbers will be ignored and no messages will be issued

### Usage Notes

Consider the following when using the SEQCHKL command:

- Only use this command if you want to force CICSVR to use logs that are not in time sequence order, or use logs that contain old data.
- Only one SEQCHKL command can be specified per CICSVR step. If the SEQCHKL command is specified, one of the keywords and its value must also be specified.
- The default value of this command is used to ensure that the logs specified in the ALLOCATE command are in the correct sequence. ALLOCATE will always perform sequence checking; the SEQCHKL command is used to force CICSVR to take alternative action.
- If this command is omitted, CICSVR will always stop at a sequence error and issue the relevant messages.

### Examples

These sequence checking commands have this effect:

1. The SEQCHKL command tells CICSVR the action to take if a journal-label-record sequence error is detected when it moves from one journal to another.
   - The GAPINSEQUENCE keyword with the IGNORE option specifies that if CICSVR finds a gap in the journal-label-record sequence when it moves from one journal to another, then the gaps are ignored and no error message is issued. CICSVR continues processing.
   - The OUTOFSEQUENCE keyword with the WARNING option specifies that if CICSVR finds that the journal-label-records are not in the correct sequence when is moves from one journal to another, then CICSVR issues a warning message and continues processing.
   - The RESETSEQUENCE keyword with the IGNORE option specifies that if CICSVR finds that the journal-label-records sequence numbers are prematurely reset when it moves from one journal to another, then the gaps are ignored and no error message is issued. CICSVR continues processing.

2. The SEQCHKL command tells CICSVR the action to take if a journal-label-record sequence error is detected when it moves from one journal to another.
The OUTOFSEQUENCE keyword with the STOP option specifies that if CICSVR finds that the journal-label-records are not in the correct sequence when moving from one journal to another, then CICSVR issues an error message and terminates.

**Synonyms**

Table 32 provides the SEQCHKL commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

**Table 32. SEQCHKL synonyms**

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQCHKL</td>
<td>SEQL, LDS</td>
</tr>
<tr>
<td>GAPINSEQUENCE</td>
<td>GAPINSEQ, GAP</td>
</tr>
<tr>
<td>OUTOFSEQUENCE</td>
<td>OUTOFSEQ, OUT</td>
</tr>
<tr>
<td>RESETSEQUENCE</td>
<td>RESETSEQ, RESET</td>
</tr>
</tbody>
</table>
SEQCHKR—Define sequence check criteria for log records

Use the SEQCHKR command to define the action that CICSVR should take if it encounters a journal-label-record sequence error while processing a log.

**Note:** Only use this command if you want to force CICSVR to use logs that are not in the correct time sequence, or logs that contain old data.

**Format**

```
SEQCHKR
  GAPINSEQUENCE(STOP | WARNING | IGNORE)
  OUTOFSEQUENCE(STOP | WARNING | IGNORE)
  RESETSEQUENCE(STOP | WARNING | IGNORE)
```

**Keywords**

**GAPINSEQUENCE(STOP | WARNING | IGNORE)**

Specifies the action that CICSVR should take when it detects a step up or down in the journal-label-record sequence number, when moving between blocks within the same log (and the journal-label records are in the correct time sequence).

The GAPINSEQUENCE keyword is optional.

If you specify:
- **STOP**, the recovery run will end with a severe message
- **WARNING**, processing will continue and a warning message will be issued
- **IGNORE**, gaps in journal-label-record sequence numbers will be ignored, and no messages will be issued

**OUTOFSEQUENCE(STOP | WARNING | IGNORE)**

For CICS V4 logs, it specifies the action that CICSVR should take when it detects a step up or down in the journal-label-record sequence number, when moving between blocks within the same log (and the journal-label records are out of time sequence).

The OUTOFSEQUENCE keyword is optional.

If you specify:
- **STOP**, the recovery run will end with a severe message
- **WARNING**, processing will continue and a warning message will be issued
- **IGNORE**, journal-label-record sequence numbers that are not in the correct sequence will be ignored, and no messages will be issued

**RESETSEQUENCE(STOP | WARNING | IGNORE)**

Specifies the action that CICSVR should take when it detects that the journal-label-record sequence number has been reset to zero before previously reaching \(2^{32} - 1\), when moving between blocks from within the same log (and the journal-label records are in the correct time sequence).
Note: If the journal-label records are not in the correct time sequence, this condition will be treated the same as the OUTOFSEQUENCE condition.

The RESETSEQUENCE keyword is optional.

If you specify:
- STOP, the recovery run will end with a severe message
- WARNING, processing will continue and a warning message will be issued
- IGNORE, reset journal-label-record sequence numbers will be ignored, and no messages will be issued

Usage Notes

Consider these points when you use the SEQCHKR command:
- Only use this command if you want to force CICSVR to use logs whose label sequence numbers are not in the correct sequence, or use logs that contain old data.
- Only specify one SEQCHKR command per CICSVR step. If the SEQCHKR command is specified, one of the optional keywords and its value must also be specified.
- The default value of this command is used to ensure that logs are in the correct sequence. ALLOCATE will always perform sequence checking; the SEQCHKR command is used to force CICSVR to take alternative action.
- If this command is omitted, CICSVR will always stop at a sequence error and issue the relevant messages.

Examples

SEQCHKR GAPINSEQUENCE(WARNING) - 1
  OUTOFSEQUENCE(WARNING) -
  RESETSEQUENCE(IGNORE)
SEQCHKR OUTOFSEQUENCE(IGNORE) 2

These sequence checking commands have this effect:

1 The SEQCHKR command tells CICSVR the action to take if a journal-label-record sequence error is detected when processing a journal.
   The GAPINSEQUENCE keyword with the WARNING option specifies that if CICSVR finds a gap in the journal-label-record sequence when moving between blocks within the same journal, then CICSVR issues a warning message and continues processing.
   The OUTOFSEQUENCE keyword with the WARNING option specifies that if CICSVR finds that the journal-label-records are not in the correct sequence when moving between blocks within the same journal, then CICSVR issues a warning message and continues processing.
   The RESETSEQUENCE keyword with the IGNORE option specifies that if CICSVR finds that the journal-label-records sequence numbers are prematurely reset when moving between blocks within the same journal, then the gaps are ignored and no error message is issued. CICSVR continues processing.

2 The SEQCHKR command tells CICSVR the action to take if a journal-label-record sequence error is detected when moving between blocks within the same journal.
The OUTOFSEQUENCE keyword with the STOP option specifies that if CICSVR finds that the journal-label-records are not in the correct sequence when moving between blocks within the same journal, then CICSVR issues an error message and terminates.

### Synonyms

Table 33 provides the SEQCHKR commands or keywords along with acceptable synonyms that can be used in place of the commands or keywords.

**Table 33. SEQCHKR synonyms**

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQCHKR</td>
<td>SEQR, LREC</td>
</tr>
<tr>
<td>GAPINSEQUENCE</td>
<td>GAPINSEQ, GAP</td>
</tr>
<tr>
<td>OUTOFSEQUENCE</td>
<td>OUTOFSEQ, OUT</td>
</tr>
<tr>
<td>RESETSEQUENCE</td>
<td>RESETSEQ, RESET</td>
</tr>
</tbody>
</table>
SPHERE—Define spheres

Use the SPHERE command to define which spheres are in the CA group.

Format

```
SPHERE NAME(sphere name)
          BACKUPTIME(date and time,LOCAL|GMT)
```

Keywords

NAME(sphere name)
Identifies the name of the sphere in the change accumulation group. Recovery information is collected in the change accumulation data set for each sphere that is listed. At least one SPHERE keyword is required for each CA control statement.

sphere name
Defines the name of the sphere. This name is 1–44 characters.

BACKUPTIME(date and time,LOCAL|GMT)
Identifies the date and time of the backup. This parameter is optional and is only specified when DFSMSshsm or DFSMSdss is not being used for your data set backups. If this parameter is specified, you must update this date and time each time you create a new data set backup.

date and time
Must be in the format yyyy/ddd/hh/mm/ss, where:
  yyyy  Is the four digit year number (2001)
  ddd   Is the day of the year (001–366)
  hh    Is the hour of the day (00–23)
  mm    Is the number of minutes (00–59)
  ss    Is the number of seconds (00–59)

You can separate these values with a slash (/), period (.), or colon (:). You can also omit the separator character. For example:

```
BACKUPTIME(2001.159/:23:00)
```

You cannot substitute commas, blanks, and so on, for the time values, but you can omit values from the right. For example, if you specify STARTTIME(2001.159), CICSVR will assume that the time segment is 00:00:00.

LOCAL
Specifies that the date and time are in local format. You can specify LOCAL if you are using MVS log streams or CICS V4 logs. LOCAL is the default value.

GMT
 Specifies that the date and time are in Greenwich mean time (GMT) format. You can specify GMT only if you are using MVS log streams.

Usage Notes

Consider the following when you use the SPHERE command:

- This command is required if the CA command is specified.
• You can specify up to 50 SPHERE commands for each CA command.
• Do not specify duplicate sphere names for a CA group.
VSAMSTART—Start passing AMS commands to VSAM

Use the VSAMSTART command to tell CICSVR to start passing AMS commands to VSAM after a VSAM sphere has been recovered.

The VSAMSTART command has no synonyms.

Format

```
VSAMSTART DSN(data set name)
```

Keywords

**DSN(data set name)**

Specifies the data set for which recovery must be successfully completed before the AMS commands are passed to VSAM. The DSN keyword is required. You can only specify this keyword once per VSAMSTART command.

**data set name**

Specifies the data set name, which is 1–44 characters.

Usage Notes

Consider these points when you use the VSAMSTART command:

- The DSN keyword must be on the same line as the VSAMSTART command.
- You can use the VSAMSTART/VSAMEND combination to rebuild AIXs after recovering a VSAM sphere that was backed up using BWO.

Examples

```
VSAMSTART DSN(TEST.PAYROLL.BASE)
BLDINDEX -
  INDATASET(TEST.PAYROLL.BASE) -
  OUTDATASET(TEST.PAYROLL.BASE)
VSAMEND DSN(TEST.PAYROLL.BASE)
```

This pair of VSAMSTART and VSAMEND commands tells CICSVR to pass the AMS BLDINDEX command to VSAM when the recovery of TEST.PAYROLL.BASE has successfully completed. The AIX for TEST.PAYROLL.BASE will be built by VSAM when CICSVR has recovered TEST.PAYROLL.BASE.

For details of the VSAMEND command, see "VSAMEND—Stop passing AMS commands to VSAM" on page 203.
VSAMEND—Stop passing AMS commands to VSAM

Use the VSAMEND command to tell CICSVR to stop passing AMS commands to VSAM.

The VSAMEND command has no synonyms.

Format

```
VSAMEND DSN(data set name)
```

Keywords

**DSN(data set name)**

Specifies the data set in the corresponding VSAMSTART command. The DSN keyword is required. You can specify this keyword only once per VSAMEND command.

**data set name**

Specifies the data set name, which is 1–44 characters.

Usage Notes

Consider these points when you use the VSAMEND command:

- The DSN keyword must be on the same line as the VSAMEND command.
- You can use the VSAMSTART/VSAMEND combination to rebuild AIXs after recovering a VSAM sphere that was backed up using BWO.

Examples

```
VSAMSTART DSN(TEST.PAYROLL.BASE)
    BLDINDEX -
    INDATASET(TEST.PAYROLL.BASE) -
    OUTDATASET(TEST.PAYROLL.PATH)

VSAMEND DSN(TEST.PAYROLL.BASE)
```

This pair of VSAMSTART and VSAMEND commands tells CICSVR to pass the AMS BLDINDEX command to VSAM when the recovery of TEST.PAYROLL.BASE has successfully completed. The AIX for TEST.PAYROLL.BASE will be built by VSAM when CICSVR has recovered TEST.PAYROLL.BASE.

For details of the VSAMSTART command, see "VSAMSTART—Start passing AMS commands to VSAM" on page 202.
### Part 5. CICSVR Exits

**Chapter 14. Using the CICSVR archive exits**

- What you can use the exits for ........................................ 257
- User processing before copying the log record ...................... 257
- Terminating operations performed by the precopy exit ............. 257
- How the archive exits work .............................................. 257
- Calling the exits .................................................................. 257
- Exit programs ...................................................................... 257
- Register conventions ......................................................... 257
- Parameter lists .................................................................... 257
- Parameters ........................................................................... 257
- Precopy exit—process a copy of the log record ...................... 258
  - When the exit is called .................................................... 258
  - Parameter list .................................................................. 258
  - Parameters passed to the exit program .............................. 258
  - Parameter returned from the exit program ....................... 258
- Termination exit—terminate operations from the precopy exit .... 259
  - When the exit is called .................................................... 259
  - Parameter list .................................................................. 259
  - Parameter passed to the exit program .............................. 260
  - Parameter returned from the exit program ....................... 260

**Chapter 15. Implementing exits for forward recovery and backout**

- What you can use the exits for ........................................ 261
- User processing after an I/O error ...................................... 261
- Recovering an ESDS ......................................................... 261
- Logical recovery .............................................................. 262
- Changing the CICSVR return code ..................................... 262
- How the exits work ......................................................... 263
- Calling the exits .................................................................. 263
- Exit programs ...................................................................... 263
- Register conventions ......................................................... 263
- Parameter lists .................................................................... 264
- Parameters ........................................................................... 264
- Sample assemble and link-edit JCL ...................................... 264
- Error exit—pass I/O error information .................................. 265
  - When the exit is called .................................................... 265
  - Parameter list .................................................................. 265
  - Parameters passed to the exit program .............................. 267
  - Parameters returned from the exit program ....................... 269
- ESDS delete exit—pass ESDS deletion records ...................... 269
  - When the exit is called .................................................... 269
  - Parameter list .................................................................. 269
  - Parameters Passed to the Exit Program ............................ 270
  - Parameters Returned from the Exit Program ..................... 270
- Preapply Exit—Pass Log Records and VSAM Records ............ 271
  - When the Exit Is Called .................................................... 271
  - Parameter List .................................................................. 271
  - Parameters passed to the exit program .............................. 272
  - Parameters returned from the exit program ....................... 273
- Termination exit—pass CICSVR return code .......................... 273
  - When the exit is called .................................................... 273
  - Parameter list .................................................................. 273
  - Parameters passed to the exit program .............................. 274
Parameters returned from the exit program
Chapter 14. Using the CICSVR archive exits

Programming Interface information

This chapter discusses the CICSVR exit programs that you can use with the archive utility and the log stream copy utility and also discusses the exit interfaces. For information about the exits for forward recovery and backout, see [Chapter 15] Implementing exits for forward recovery and backout" on page 261.

What you can use the exits for

The CICSVR archive and log stream copy exits let you perform special processing at two different exit points. These optional exits and their uses are:

- **Precopy**
  - Copy each log record
- **Termination**
  - Terminate operations performed by the precopy exit

User processing before copying the log record

Use the **precopy exit** to copy each log record to another data set before it is copied to the output data set.

Terminating operations performed by the precopy exit

Use the **termination exit** to terminate operations that were performed by the precopy exit.

How the archive exits work

Before you start using an exit, you must know how it works with the CICSVR archive utility and log stream copy utility. The interface conventions for passing control and information between the archive utility or the log stream copy utility and the exit programs are described here.

For examples of exit coding, see the installation material provided with CICSVR.

Calling the exits

The CICSVR archive utility or log stream copy utility calls exit programs only if you supply a **DEFEXIT** command.

Exit programs

You can write exit programs in any programming language, if you follow the standard assembler register conventions described in this section.

The exits get control in 31-bit addressing mode. Link-edit each exit program as a separate load module and then place the load modules in a load library that is accessible to the CICSVR job. This can be, for example, a library that is concatenated to the one defined by the STEPLIB DD statement. If you specify DWWLOAD in your CICSVR JCL, the exit load modules must be in the DWWLOAD data set.

Register conventions

These register values apply at entry to the exit program:

- Register 1 contains the address of the exit parameter list.
Register 13 contains the address of the register save area. The exit program should save and restore the registers that it modifies, using the save area addressed by register 13.

Register 14 contains the return address that the exit program should branch to when it completes its work.

Register 15 contains the entry address of the exit program.

Parameter lists

All communication between the CICSVR and the exit programs is through parameters. When the archive utility or the log stream copy utility calls an exit, register 1 points to a parameter list containing the addresses of the parameters. The parameter list is an area whose length depends on the number of parameters to be passed. For each parameter, it consists of one fullword containing the address of the parameter.

Each exit has a different parameter list. You can find these lists and descriptions of the parameters, in the exit descriptions starting on page 258.

Communication from the exit programs to CICSVR occurs through the parameters.

Parameters

One parameter is used by both exits:

**Work area**

A 128-byte work area. The exit programs can use this area as a common work area. The address of this work area is passed to both the exit programs, enabling them to pass information among themselves. This area contains binary zeros at the first exit invocation.

Precopy exit—process a copy of the log record

Use the precopy exit to copy each log record to another data set, before the archive utility or log stream copy utility copies the record to the output data set.

When the exit is called

This exit is called just before each log record is copied to the output data sets.

Parameter list

This parameter list is passed to the precopy exit:

```
Log record, work area
```

Figure 99 on page 259 shows how you find the parameters from the contents of register 1.
Parameters passed to the exit program

These parameters are passed to the exit program:

**Log record**
The complete log record. This record is described by DSECT DFHJCRDS, which is provided with CICS.

**Work area**
A 128-byte work area that can be used to pass information between exit programs. See the description in "Parameters" on page 258.

Parameter returned from the exit program

This parameter is returned from the exit program:

**Work area**
The 128-byte work area that can be used to pass information between exit programs.

Termination exit—terminate operations from the precopy exit

Use the termination exit to terminate the actions that were performed by the precopy exit, such as closing the data set that the precopy exit copied the log records to.

When the exit is called
This exit is called when the CICSVR archive utility or log stream copy utility is ready to terminate.

Parameter list
This parameter is passed to the termination exit:
Figure 100 shows how you find the parameters from the contents of register 1.

Parameter passed to the exit program
This parameter is passed to the exit program:

Work area
A 128-byte work area that can be used to pass information between exit programs. See the description in Parameters on page 258.

Parameter returned from the exit program
This parameter is returned from the exit program:

Work area
The 128-byte work area that can be used to pass information between exit programs.

End of Programming Interface information
Chapter 15. Implementing exits for forward recovery and backout

Programming Interface information

This chapter discusses the CICSVR exit programs that you can use for forward recovery and backout and also discusses the exit interfaces. For information about exits for the CICSVR archive utility, see "Chapter 14. Using the CICSVR archive exits" on page 257.

What you can use the exits for

The CICSVR exits let you perform special processing at four different exit points. These optional exits and their uses are:
- **Error**  Handle I/O errors
- **ESDS delete**  Logically delete ESDS records
- **Preapply**  Perform logical recovery
- **Termination**  Change the CICSVR return code

These sections describe these exits and their uses.

Attention

Improper use of exits can cause loss of data integrity. All exits except the termination exit let you skip or modify records. Use exits only in well-controlled, familiar situations.

User processing after an I/O error

You can use the **error exit** if an uncorrectable data validity error occurs when CICSVR is reading a log, or if an error occurs when CICSVR is accessing a VSAM data set.

If you provide an error exit program, you can tell CICSVR to continue processing after an I/O error. CICSVR passes information about the error to the exit program. The exit program can tell CICSVR to ignore the record that caused the error and to continue processing with the next record.

Recovering an ESDS

You will need the **ESDS delete exit** if you use the CICSVR backout function to logically delete records from an ESDS.

You cannot physically delete a record from an ESDS, so you must update the existing record to indicate that it is **marked for deletion**. You insert a marked-for-deletion code in the record. This code is known to the applications that are accessing the ESDS. But, CICSVR does not know which field to amend. When CICSVR must back out a record that was added, it takes the ESDS delete exit. The exit program then returns the ESDS record (containing the marked-for-deletion code) that was required by the application.

If you must back out an ESDS that contains logged adds, you must provide an ESDS delete exit program. If you do not provide an exit program, CICSVR will terminate. The ESDS delete exit is taken for every add to an ESDS that is to be backed out.
CICSVR passes information about the ESDS record, including the length of that record, to the exit program. CICSVR obtains this information from the log. The exit program can modify the VSAM data part of the log record, to add the marked-for-deletion code that is recognized by your applications. CICSVR will then update the record in the ESDS. The exit can alternatively tell CICSVR to ignore this record, or even to terminate processing.

**Note:** CICSVR acquires large areas of storage when recovering an ESDS.

**Logical recovery**

The **preapply exit** is taken for each before-image or after-image that will be applied to a VSAM data set, before CICSVR applies the change, for data sets that are being recovered in this CICSVR run. For CICSVR backout, the exit is taken for every update, or for every add of a record that falls within the default or specified start and stop times. For forward recovery, it is taken for every update (if the original VSAM record has been successfully read) or for every add that falls within the default or specified start and stop times.

CICSVR passes the record that is read from the log and, where applicable, the VSAM record that will be updated, to the exit program. The exit program can modify the log record. It can also tell CICSVR to ignore the log record and continue with the next record.

This exit gives you the opportunity to perform selective processing of records, depending on your special requirements. This type of processing is useful to help in **logical recovery**; that is, recovering from logical errors in the VSAM data set that are caused by an application, or by transactions that did not work correctly. The preapply exit lets you use CICSVR for this purpose.

**Note:** If you try to perform logical recovery, there are situations that cannot be corrected by this exit. Although logical recovery might have recovered the data set to the state you require, other applications (online or offline) might have accessed the data set during the time it contained incorrect data.

You can use the preapply exit to:
- Tell CICSVR not to process certain before-images or after-images.
  You can do this to avoid updates that were made by an application or a transaction, or that were made during a specific time period.
- Modify certain before-images or after-images before CICSVR applies them to the VSAM data set.
  You can check the content of the log record and the content of the VSAM record to be updated, and then modify the VSAM data and key fields in the log record in the way you require.

**Changing the CICSVR return code**

You can use the **termination exit** to change the CICSVR completion code just before CICSVR terminates. For example, there might be times when you want to accept CICSVR results for further processing, even when CICSVR found errors during processing. CICSVR passes the completion code to the exit program, and the exit program can change the code to a desired code.
How the exits work

Before you start using an exit, you must know how it works with CICSVR. The interface conventions for passing control and information between CICSVR and the exit programs are described here.

For examples of exit coding, see the installation material provided with your CICSVR order.

Calling the exits

You tell CICSVR to call an exit by specifying the member name for the exit in the CICSVR exits secondary window (Figure 101):

CICSVR exits
Specify member names for the CICSVR exits. Press Enter to use the displayed member names in the recovery.

Preapply .
Error .
ESDS delete .
Termination .

Command ===>
F1=Help F5=Getdef F6=SaveDef F12=Cancel

Figure 101. Exits secondary window

For a description of the CICSVR ISPF dialog interface functions, refer to CICSVR V3R1 User’s Guide and Reference.

If you are not using the ISPF dialog interface, or if you are defining exits to be used with the CICSVR archive utility, CICSVR calls exit programs only if you supply a DEFEEXIT command.

Exit programs

You can write exit programs in any programming language, if you follow the standard assembler register conventions described in this section.

The exits get control in 31-bit addressing mode. Link-edit each exit program as a separate load module (see Figure 102 on page 263) and then place the load modules in a load library that is accessible to the CICSVR job. This can be, for example, a library that is concatenated to the one defined by the STEPLIB DD statement. If you specify DWWLOAD in your CICSVR JCL, the exit load modules must be in the DWWLOAD data set.

Register conventions

These register values apply at entry to the exit program:

- Register 1 contains the address of the exit parameter list.
- Register 13 contains the address of the register save area. The exit program should save and restore the registers that it modifies, using the save area addressed by register 13.
• Register 14 contains the return address that the exit program should branch to when it completes its work.
• Register 15 contains the entry address of the exit program.

Parameter lists
All communication between CICSVR and the exit programs is through parameters. When CICSVR calls an exit, register 1 points to a parameter list containing the addresses of the parameters. The parameter list is an area whose length depends on the number of parameters to be passed. For each parameter, it consists of one fullword containing the address of the parameter. Each exit has a different parameter list. You can find these lists and descriptions of the parameters, in the exit descriptions starting on page 265. When CICSVR calls an exit, register 1 points to a parameter list containing the addresses of the parameters. The parameter list is an area whose length depends on the number of parameters to be passed. For each parameter, it consists of one fullword containing the address of the parameter. Each exit has a different parameter list.

Communication from the exit programs to CICSVR occurs through the parameters. An action parameter is always needed to tell CICSVR what action to take. For some exits, the exit program might update one of the input parameters and return it to CICSVR.

Parameters
Two parameters are used by all exits:

Work area
A 128-byte work area. The exit programs can use this area as a common work area. The address of this work area is passed to all the exit programs, enabling them to pass information among themselves. This area contains binary zeros at the first exit invocation.

Action
A 1-byte character field containing an action code that all exit programs must return to CICSVR. The action code can be one of these:
C Continue as if no exit program was provided.
U Use the record that the exit program just updated.
I Ignore the log record and read the next log record.

Some exits accept only two of these action codes.

Sample assemble and link-edit JCL
This sample JCL is provided to let you assemble and link-edit your CICSVR exits. In Figure 102 on page 265, the exit is called DWWPREEX:
Error exit—pass I/O error information

You can use the error exit to make CICSVR continue processing after an I/O error.

When the exit is called

The error exit is called when:
- A data validity error occurs when CICSVR is reading a log
- A physical error occurs when CICSVR is reading a log
- A logical error occurs when CICSVR is processing a VSAM data set
- A physical error occurs when CICSVR is processing a VSAM data set

If used, the error exit is called by CICSVR when a VSAM error occurs during these CICS operations:
• Adding a record to the base cluster
• Deleting a record from the base cluster
• Updating an existing record on the base cluster

But, if the error can be corrected by CICSVR, the error exit is not called. This is true even if an I/O error message is produced.

For backup-while-open and for CICS/MVS logging, certain normal VSAM errors can occur. They are identified by these error codes:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Possible error codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding a record</td>
<td>008</td>
</tr>
<tr>
<td>Deleting a record</td>
<td>004, 016, 032, 116</td>
</tr>
<tr>
<td>Updating a record</td>
<td>004, 016, 032, 096, 116</td>
</tr>
</tbody>
</table>

These errors are considered to be serious:
• VSAM error codes (except those in the preceding table) for CICS/MVS logging
• All VSAM error codes for CICS V4 logging (except return code 016 when deleting a record)
• Log READ errors

If you do not provide an exit routine and the error is serious, CICSVR will terminate recovery.

**Parameter list**

This parameter list is passed to the error exit:

Data set name, error type, error information, length of error data, error data, work area, action code

Figure 103 on page 267 shows how you find the parameters from the contents of register 1.
Parameters passed to the exit program

These parameters are passed to the exit program:

**Data set name**

The data set name is dependent on these error types:
- **01–02** Shows that the data set name is for a log
- **11–14, 16, 17** Shows that the data set name is for the original VSAM base cluster
- **15** Shows that the data set name is for a VSAM path

The field is left justified and is padded with blanks.

**Error type**

A 2-byte character field that tells the exit which type of I/O processing was occurring when the error occurred. It can contain:
- **01** Log physical read
- **02** Log logical read
- **11** VSAM GET
- **12** VSAM PUT (new)
- **13** VSAM PUT (update)
- **14** VSAM ERASE
- **15** VSAM GET on a path (CICS/MVS logging only)
- **16** VSAM PUT (new) to an ESDS base (CICS V4 logging only)
- **17** VSAM ERASE when the relative byte address (RBA) already exists
**Error information**

A character field that provides information about the error.

For error type 01 (a log physical-read error), it is a 78-byte character field, containing bytes 50–127 of the SYNADF message buffer. This buffer contains information such as the ddname and error description.

For error type 02 (a log logical-read error), it is a 4-byte binary field that contains the current block-number count while reading forward or backward.

For error types 11–15 (a VSAM error), it is an 8-byte character field, containing:

**Bytes 0–1**

The second byte from the feedback field in the VSAM request parameter list (RPL). This field contains the return code (register 15). The contents are related to bytes 2–3, described below.

**Bytes 2–3**

The last 2 bytes from the feedback field in the VSAM RPL. This field contains the VSAM function code and feedback code.

**Byte 4**

A 1-character code indicating log type:

- 2: Indicates a CICS/MVS log
- 3: Indicates a CICS V4 log

**Byte 5**

A 1-character code showing the status of the sphere being recovered:

- N: The sphere was restored from a normal backup copy.
- B: The sphere was restored from a copy made using the backup-while-open facility.

**Bytes 6–7**

Not used.

For error type 16 (a VSAM insert error to an ESDS base), it is a 4-byte binary field containing the actual RBA used on the VSAM file. This error type only occurs when you are recovering from a CICS V4 log where the logged RBA differs from the real RBA.

**Note:** The record will have been appended to the file.

**Length of error data**

A fullword containing the length of the error data parameter.

**Error data**

Contains more data about the error.

For error types 01 and 02 (a log read error), this field contains the last log record that was read successfully.

For error types 11 and 15 (a VSAM GET error) and error type 14 (a VSAM delete error), it contains the key of the record to retrieve.
For error types 12, 13, and 16 (a VSAM insert or update error), it contains the record to process.

**Work area**

A 128-byte work area that can be used to pass information between exit programs. See the description in [Parameters](#) on page 264.

### Parameters returned from the exit program

These parameters are returned from the exit program:

- **Work area**
  The 128-byte work area that can be used to pass information between exit programs.

- **Action code**
  A 1-byte character field that tells CICSVR what action to take. The action code can be:
  - **C**: Continue as if the exit had not been taken, issue message DWW0211, and end processing with a return code of 12.
  - **I**: Ignore this log record or this VSAM error, issue message DWW0210, and continue processing.

**Note:** For error type 02, CICSVR will ignore the returned action code and issue message DWW0211 before terminating.

### ESDS delete exit—pass ESDS deletion records

You can use the ESDS delete exit with the CICSVR backout function to logically delete a record from an ESDS.

**When the exit is called**

The ESDS delete exit is called when a record is to be logically deleted from an ESDS.

**Parameter list**

This parameter list is passed to the ESDS delete exit:

- Base cluster name, log record, ESDS record length, ESDS record, work area, action code

Figure 104 on page 270 shows how you find the parameters from the contents of register 1.
Parameters Passed to the Exit Program

These parameters are passed to the exit program:

**Base cluster name**
- The name of the base cluster in question. The field is left justified and is padded with blanks.

**Log record**
- The complete log record. This record is described by DSECT DFHJCRDS, which is provided with CICS.

**Length of ESDS record**
- A fullword containing the length of the ESDS record.

**VSAM record**
- The ESDS record on the log that will be deleted.

**Work area**
- A 128-byte work area that can be used to pass information between exit programs. See the description in [Parameters on page 263](#).

Parameters Returned from the Exit Program

These parameters are returned from the exit program:

**Log record**
- The complete log record, which can be optionally modified by the exit program. The VSAM record key and data fields in this record will be used by CICSVR in further processing. Other fields in this record will not be used; instead, CICSVR will use these fields as they appeared in the record when it was passed to the exit program.
Work area
The 128-byte work area that can be used to pass information between exit programs.

Action code
A 1-byte character field that tells CICSVR what action to take. The action code can be:

- C: Continue processing as if no exit program were provided. CICSVR will issue error message DWW0505S and terminate processing.
- U: Use the log record returned by the exit program to update the ESDS.
- I: Ignore this log record; do not process it. The record will remain on the VSAM ESDS in its original form.

Preapply Exit—Pass Log Records and VSAM Records
You can use the preapply exit to modify a before-image or after-image that will be applied to a VSAM data set.

When the Exit Is Called
The preapply exit is called before CICSVR updates a VSAM data set with either the before-image (for backout) or the after-image (for forward recovery), for data sets that are being recovered in this CICSVR run. For CICSVR backout, the exit is taken for every update or add of a record that falls within the default or specified start and stop times. For forward recovery, it is taken for every update (if the original VSAM record has been successfully read) or for every add that falls within the default or specified start and stop times.

Parameter List
This parameter list is passed to the preapply exit:

- Base cluster name, log record, VSAM record length, VSAM record, work area, action code, format

Figure 105 on page 272 shows how you find the parameters from the contents of register 1.
Parameters passed to the exit program

These parameters are passed to the exit program:

**Base cluster name**

The name of the base cluster in question. The field is left justified and is padded with blanks.

**Log record**

The complete log record, containing the before-image or after-image to be written to the VSAM data set. For CICS Transaction Server records, the first four bytes are the length of the log record.

For CICS/MVS and CICS V4 V3 and V4, the first two bytes are the length of the log record, and the next two bytes are X'00'.

**Length of VSAM record**

A fullword containing the length of the VSAM record. In the case where CICSVR will add a record to the VSAM data set, the length is 0.

**VSAM record**

The VSAM record that is going to be updated or deleted. For forward recovery, this is the before-image; for backout, it is the after-image. In the case where CICSVR will add a record to the VSAM data set, this parameter contains zeros.

**Work area**

A 128-byte work area that can be used to pass information between exit programs. See the description in [Parameters](#) on page 264.

**Format**

Indicates the format of the CICS log record.

- **5.1** For CICS Transaction Server log records
- **ESA** For CICS V4 log records
Parameters returned from the exit program

These parameters are returned from the exit program:

Log record
The complete log record, which can be optionally modified by the exit program. The VSAM record key and data fields in this record will be used by CICSVR in further processing. Other fields in this record will not be used; instead, CICSVR will use these fields as they appeared in the record when it was passed to the exit program.

Work area
The 128-byte work area that can be used to pass information between exit programs.

Action code
A 1-byte character field that tells CICSVR what action to take. The action code can be:
- C Continue processing using the log record that was passed to the exit program.
- U Use the log record returned by the exit program.
- I Ignore this log record; do not apply it to the VSAM data set.

Note: For details of the old-style preapply exit, refer to an earlier edition of this book.

Termination exit—pass CICSVR return code

You can use the termination exit to modify the CICSVR return code.

When the exit is called
The termination exit is called when CICSVR is ready to end.

Parameter list
This parameter list is passed to the termination exit:

Completion code, work area, action code

Figure 106 on page 274 shows how you find the parameters from the contents of register 1.
Parameters passed to the exit program

These parameters are passed to the exit program:

**Completion code**

A fullword field containing the completion code that CICSVR will terminate with if the termination exit does not return its own value.

The code can be one of these:

- **00** Successful completion.
- **04** One or more warning messages have been issued.
- **08** One or more error messages have been issued.
- **12** CICSVR has terminated because of a severe error.

**Work area**

A 128-byte work area that can be used to pass information between exit programs. See the description in [Parameters](page 264).

Parameters returned from the exit program

These parameters are returned from the exit program:

**Completion code**

The fullword completion code as optionally changed by the termination exit program. The range is 0–4095; values outside this range will cause CICSVR to reset the modified completion code to 12.
Work area
The 128-byte work area that can be used to pass information between exit programs.

Action code
A 1-byte character field that tells CICSVR what action to take. The action code can be:
- C  Continue processing using the original completion code.
- U  Use the completion code returned by the exit program.

End of Programming Interface information
Appendix A. CICSVR V3R1 ddnames

Table 34 shows the ddnames for CICSVR Version 3 Release 1.

### Table 34. CICSVR V3R1 ddnames

<table>
<thead>
<tr>
<th>CICSVR ddname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWWARC1</td>
<td>Input log for the archive utility.</td>
</tr>
<tr>
<td>DWWCON1-3</td>
<td>CICSVR recovery control data set (RCDS). If the CICSVR server address space is active, you must preallocate this data set prior to bringing up the CICSVR server address space. You must use the following format for the data set name: prefix.DWWCONn. You must use the prefix value that is defined in the IGDSMSxx PARMLIB member and use 1, 2, and 3 for the n value. If the CICSVR server address space is not active, see &quot;Creating and maintaining the RCDS&quot; on page 42 if you use CICS TS or &quot;Creating and maintaining your RCDS&quot; on page 108 if you use CICS V4 for information on creating the RCDS.</td>
</tr>
<tr>
<td>DWWCOPYn</td>
<td>Output copies of the logs for the archive utility, log stream copy utility, and RCDS utility.</td>
</tr>
<tr>
<td>DWWIN</td>
<td>Data set that contains the CICSVR commands. You can either specify a sequential data set with 80-byte, fixed-length records, or include the CICSVR commands in-stream.</td>
</tr>
<tr>
<td>DWWLOAD</td>
<td>Load library for your CICSVR exits.</td>
</tr>
<tr>
<td>DWWLOG</td>
<td>Ddname used to allocate log data sets when not using the CICSVR ALLOCATE command.</td>
</tr>
<tr>
<td>DWWMSG</td>
<td>Output data set containing CICSVR messages. This ddname is used by the CICSVR server address space, the CICSVR dialog, and by CICSVR batch jobs. The DCB parameters for this data set are RECFM=FBA and LRECL=133. The block size can be provided on the DWWMSG DD statement and must be a multiple of 133. CICSVR server address space: If you plan to activate the CICSVR server address space you must preallocate the DWWMSG data set prior to bringing up the CICSVR server address space. You must use the following format for the data set name: prefix.DWWMSG.systemname. You must use the prefix value that is defined in the IGDSMSxx PARMLIB member and use the systemname that is defined in the IEASYSxx PARMLIB member. CICSVR dialog: Use a DWWMSG data set name for the CICSVR dialog that is different than the one that was dynamically created for use by the CICSVR server address space. Any valid data set name can be used. CICSVR batch jobs: In the CICSVR JCL skeleton, specify SYSOUT=’’ for the DWWMSG data set. Or, if you want the output to be written to a data set, use a data set name that is different than the one that was dynamically created for use by the CICSVR server address space, or the one used by the CICSVR dialog.</td>
</tr>
<tr>
<td>CICSVR ddname</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| DWWSORT       | Output data set that contains sort messages and control statements from the change accumulation job. Usually specified as SYSOUT=*.
| DWWPRINT      | Output data set that contains the reports produced by CICSVR. This ddname is used by the CICSVR dialog and by CICSVR batch jobs. The DCB parameters for this data set are RECFM=FBA and LRECL=133. The default block size is 133 bytes.

CICSVR dialog: Use a DWWPRINT data set name for the CICSVR dialog that is different than the one that was dynamically created for use by the CICSVR server address space. Any valid data set name can be used.

CICSVR batch jobs: In the CICSVR JCL skeleton, specify SYSOUT=* for the DWWPRINT data set. Or, if you want the output to be written to a data set, use a data set name that is different than the one that was dynamically created for use by the CICSVR server address space, or the one used by the CICSVR dialog.

| DWWDMSG       | Output data set that contains tracing and diagnostic information produced by individual CICSVR subroutines, as requested by CICSVR. This ddname is used by the CICSVR server address space and the CICSVR dialog. The DCB parameters for this data set are RECFM=VBA and LRECL=84. The default block size is 3120 bytes.

CICSVR server address space: If the CICSVR server address space is active, you must specify the data set using the following format: prefix.DWWDUMP.systemname. Use the prefix from the IGDSMSxx PARMLIB member and the systemname from the IEASYSxx PARMLIB member.

CICSVR dialog: Use a DWWDUMP data set name for the CICSVR dialog that is different than the one that was dynamically created for use by the CICSVR server address space. Any valid data set name can be used.

| DWWDUMP       | Output data set that contains tracing and diagnostic information, as requested by CICSVR. This ddname is used by the CICSVR server address space and the CICSVR dialog. The DCB parameters for this data set are RECFM=VBA and LRECL=84. The default block size is 3120 bytes.

CICSVR server address space: If the CICSVR server address space is active, you must specify the data set using the following format: prefix.DWWDUMP.systemname. Use the prefix from the IGDSMSxx PARMLIB member and the systemname from the IEASYSxx PARMLIB member.

CICSVR dialog: Use a DWWDUMP data set name for the CICSVR dialog that is different than the one that was dynamically created for use by the CICSVR server address space. Any valid data set name can be used.
Appendix B. Diagnosing change accumulation and DFSORT problems

Change accumulation calls DFSORT (or an equivalent sort product) to sort the log records before it updates the current change accumulation data set with the next log range. If DFSORT cannot process the records, CICSVR issues the following message:

DWW0905S
Unexpected result from the external sort product. The sort return code is nn.

This section explains how to get DFSORT messages and dumps as well as how to identify and eliminate other common errors.

How to get DFSORT messages

To diagnose problems using DFSORT, you must be able to see the DFSORT messages and control statements in the DWWSORT message data set. DFSORT messages start with ‘ICE’. If you receive a condition code of 20 or do not see the DFSORT messages and control statements, add the following to your change accumulation JCL:

//DWWSORT DD SYSOUT=A

and resubmit your job.

If you still do not see the DFSORT messages and control statements in the DWWSORT message data set, add the following to your change accumulation JCL:

//SORTDIAG DD DUMMY

to force out the DFSORT messages and control statements, and resubmit your job.

How to eliminate other common errors

DFSORT needs intermediate storage to sort the log records. IBM recommends that you specify DFSORT work data sets in your change accumulation JCL as follows:

//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100,25))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100,25))

SPACE=(CYL,(100,25)) is a good starting point for the space allocation. It provides 100 primary cylinders and 25*15 secondary allocation cylinders for each of the three DFSORT work data sets. If DFSORT is unable to sort the log records with this amount of intermediate storage, it will terminate with error message:

ICE046A SORT CAPACITY EXCEEDED - RECORD COUNT: n

In this case, you should increase the SPACE values or add additional work data sets, as appropriate, and resubmit your job.

Note: Tape sort work data sets are not supported.

If your site does not have enough work space available, you can reduce the amount of space and run the change accumulation job more frequently.
How to get a DFSORT dump

To get a DFSORT dump when an abend occurs, add the JCL and program control statements shown in Table 35 to your change accumulation JCL.

Table 35. Getting a dump

<table>
<thead>
<tr>
<th>If you received</th>
<th>Then use</th>
</tr>
</thead>
<tbody>
<tr>
<td>A user or system abend code</td>
<td>//SYSUDUMP DD SYSOUT=A</td>
</tr>
<tr>
<td>A condition code 16</td>
<td>//SYSUDUMP DD SYSOUT=A</td>
</tr>
<tr>
<td></td>
<td>//DFSPARM DD *</td>
</tr>
<tr>
<td></td>
<td>DEBUG ABEND,ESTAE</td>
</tr>
</tbody>
</table>

Common system abends related to storage

Table 36 shows common system abend codes that might appear when abends result from errors in storage parameters. The items are listed by system abend code. For details on DFSORT’s use of storage and storage parameters, refer to DFSORT Installation and Customization R14 and DFSORT Application Programming Guide R14.

Table 36. Common system abends related to storage

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>There was not sufficient storage to load a module. Ensure that the</td>
</tr>
<tr>
<td></td>
<td>values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.</td>
</tr>
<tr>
<td>804, 80A</td>
<td>The amount of storage requested exceeded that available. Ensure that</td>
</tr>
<tr>
<td></td>
<td>the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are</td>
</tr>
<tr>
<td></td>
<td>sufficient.</td>
</tr>
<tr>
<td>878</td>
<td>User exits required more storage than was available. Ensure that the</td>
</tr>
<tr>
<td></td>
<td>values for REGION, MAINSIZE, SIZE, RESALL or RESINV are</td>
</tr>
<tr>
<td></td>
<td>sufficient.</td>
</tr>
<tr>
<td>D37</td>
<td>A SORTWK or SORTOUT data set was specified with insufficient</td>
</tr>
<tr>
<td></td>
<td>primary space allocation and no secondary allocation. Increase the</td>
</tr>
<tr>
<td></td>
<td>primary allocation, or include a value for secondary allocation.</td>
</tr>
<tr>
<td>E37</td>
<td>A SORTOUT data set was allocated on a primary volume that had either</td>
</tr>
<tr>
<td></td>
<td>insufficient storage or excessive fragmentation, and no</td>
</tr>
<tr>
<td></td>
<td>secondary volume was specified. Specify more than one volume in the</td>
</tr>
<tr>
<td></td>
<td>VOLSER parameter for the SORTOUT data set or compress the</td>
</tr>
<tr>
<td></td>
<td>volume to allow it to contain the full 16 extents.</td>
</tr>
</tbody>
</table>

For more information on debugging and correcting problems using DFSORT, refer to DFSORT Messages, Codes and Diagnosis Guide R14. For general information on DFSORT, refer to DFSORT Application Programming Guide R14.
Appendix C. Diagnosing logging problems

If you encounter a wait which you believe to be logging related, look at the MVS console for messages prefixed with "IXG". These are the MVS system logger messages and may provide further information about the cause of the wait. The MVS system console may also reveal evidence of resource contention within MVS.

Logging problem categories

The following categories of problem (in order of ascending impact on the user) may be encountered by CICSVR:

1. Those problems within the MVS system logger that the MVS system logger resolves for itself. CICSVR has no involvement in this category and may only experience the problem as an increase in response times.

2. The MVS system logger is unable to satisfy the CICSVR request immediately. This problem state can be encountered on a 'STRUCTURE FULL' condition where the coupling facility has reached its capacity before offloading data to DASD. CICSVR is able to recognize this situation, issues DWW838I, and retries the request every three seconds until the request is satisfied. Typically, this can take up to a minute.

3. If the MVS system logger fails, CICSVR will attempt to trap the error and issue a 3999 condition code. The 3999 condition code indicates that no batch logging occurred for the sphere. Operator messages DWW251I - DWW260I or DWW262I indicate the reason for the termination. Since no batch logging was performed, you must take a backup of the sphere to ensure recoverability. If you use DFSMShsm, IBM recommends you set up your batch logging jobs to take a backup of your VSAM sphere if you receive a 3999 condition code from CICSVR VSAM batch logging.

Figure 107 shows a sample job step that takes a DFSMShsm backup of PAYROLL.FILE72 when a 3999 condition code is detected for STEP03.

```
//MYJOB JOB ....
....
//STEP03 EXEC PGM=....
//PAYROLL IS THE VSAM SPHERE DEFINED AS FRLOG(REDO)
//PAYROLL DD DISP=SHR,DSN=PAYROLL.FILE72
....
//BLBAD RUNS IF LOGGING TERMINATED AT STEP03 IS DETECTED
//BLBAD EXEC PGM=ARCINBAK,COND=(3999,NE,STEP03)
//ARCPRT DD SYSOUT**
//ARCSNAP DD SYSOUT**
//BACK01 DD DSN=PAYROLL.FILE72,DISP=SHR
```

Figure 107. Example of a DFSMShsm backup if 3999 condition code occurs

Figure 108 on page 283 shows a sample job step that takes a DFSMSdss backup (logical dump) of PAYROLL.FILE72 when a 3999 condition code is detected for STEP03.
If the MVS system logger fails and CICSVR cannot trap the error, the CICSVR server address space may terminate.

### MVS system logger availability

CICSVR checks for the availability of the MVS system logger. If it detects that the MVS system logger is not available, the following occurs:

- CICSVR issues DWW252I ATTEMPT TO CONNECT TO CICSVR ADDRESS SPACE FAILED.
- CICSVR returns a non-zero return code to VSAM OPEN.
- VSAM OPEN issues IEC161I with reason code 002 and SFI code indicating the reason CICSVR VSAM batch logging was not provided.
- CICSVR sets the condition code to 3999 for this job step.

You should set up your batch jobs so they can detect a 3999 condition code and trigger a DFSMShsm or DFSMSdss backup of your VSAM sphere to ensure recoverability. See Figure 107 on page 281 for a DFSMShsm example or Figure 108 for a DFSMSdss example.

If there is a problem with the logstream, you may need to delete it and redefine it.

### Diagnosis of MVS System Logger Problems

Extended waits by CICSVR for logging can be caused by problems within the MVS system logger or other areas of MVS. You can investigate these by looking at the MVS console messages. Look at the following information:

- Console messages and dumps
  - Look for the following information:
    - Outstanding WTOR messages
    - IXGxxx messages
    - Allocation, catalog and HSM error messages
    - I/O errors for log stream data sets and/or LOGR couple data sets

```plaintext
//MYJOB JOB ....
....
//STEP03 EXEC PGM=....
//* PAYROLL IS THE VSAM SPHERE DEFINED AS FRLOG(REDO)
//PAYROLL DD DISP=SHR,DSN=PAYROLL.FILE72
....
// BLBAD RUNS IF BATCH LOGGING TERMINATED IS DETECTED AT STEP03
//BLBAD EXEC PGM=ADDRSU,COND=(3999,NE,STEP03)
//SYSPRINT DD SYSOUT=*''
//DDUMPOUT DD DISP=(,CATLG),DSN=PAYROLL.FILE72.BACKUP,
//UNIT=3390,VOL=SER=USRPAK,
//SPACE=(CYL,(5,1),RLSE)
//SYSIN DD *
DUMP -
  DS(INCL(PAYROLL.FILE72)) -
  OPT(4) -
  COMPRESS -
  CICSVRBACKUP -
  SPHERE -
  OUTDDNAME (DDUMPOUT)
/*
```

Figure 108. Example of a DFSMSdss backup if 3999 condition code occurs
– IXCxxx messages, indicating problems with the coupling facility structure or couple data sets
– 1C5 abends, and other abends from the IXGLOGR address space

**Global resource serialization (GRS) resource contention**

To check GRS resource contention by displaying GRS enqueues and latch usage on all machines in the sysplex, issue either of the following MVS commands:

```
D GRS,C
D GRS,RES=(SYSZLOGR,*)
```

A normal response looks like:

```
D GRS,C

ISG020I 12.06.49 GRS STATUS 647
NO ENQ CONTENTION EXISTS
NO LATCH CONTENTION EXISTS

D GRS,RES=(SYSZLOGR,*)

ISG020I 14.04.28 GRS STATUS 952
NO REQUESTORS FOR RESOURCE SYSZLOGR *
```

A response showing GRS contention looks like this:

```
D GRS,C

ISG020I 12.06.31 GRS STATUS 619
LATCH SET NAME: SYS.IXGLOGER_LCBVT
CREATOR JOBNAM: IXGLOGR CREATOR ASID: 0202
LATCH NUMBER: 7
REQUESTOR ASID EXC/SHR OWN/WAIT
IXGLOGR 0202 EXCLUSIVE OWN
IXGLOGR 0202 SHARED WAIT

D GRS,RES=(SYSZLOGR,*)

ISG020I 19.58.33 GRS STATUS 374
S=STEP SYSZLOGR 91
SYSNAME JOBNAM ASID TCBADDR EXC/SHR OWN/WAIT
MV26 MSLDELC1 002F 008F6370 EXCLUSIVE OWN
S=STEP SYSZLOGR 93
SYSNAME JOBNAM ASID TCBADDR EXC/SHR OWN/WAIT
MV26 MSLWRTC1 002E 008DED90 EXCLUSIVE OWN
MV26 MSLWRTC1 002E 008DB990 EXCLUSIVE WAIT
MV26 MSLWRTC1 002E 008DB700 EXCLUSIVE WAIT
MV26 MSLWRTC1 002E 008F60C8 EXCLUSIVE WAIT
S=SYSTEMS SYSZLOGR LPAYROL.TESTLOG.TLOG1
SYSNAME JOBNAM ASID TCBADDR EXC/SHR OWN/WAIT
MV27 IXGLOGR 0011 008F7398 EXCLUSIVE OWN
MV26 IXGLOGR 0011 008F7398 EXCLUSIVE WAIT
```

This shows which tasks have exclusive enqueues on the log streams, and which tasks are waiting for them. It is quite normal for enqueues and latches to be obtained, occasionally with contention. They are indications of a problem only if they last for more than a minute or so.

Long term enqueuing on the SYSZLOGR resource can be a sign of problems even if there is no contention.

---

2. Log stream data sets are of the form IXGLOGR.stream_name.Annnnnnn. The high level qualifier (IXGLOGR) may be different if the HLQ parameter was specified when the log stream was defined.

3. You may also see latch set name SYS.IXGLOGER_MISC.
You can choose to display only those log streams exclusively enqueued on, or being waited on, by CICSVR jobs in the sysplex. Issue the following MVS command:

```
D GRS,RES=(DFHSTRM,*)
```

A typical response to this command looks like this:

```
ISGO20I 14.51.28 GRS STATUS 541
S=YES/STMS DFHSTRM PAYROL.CICSVRVR.DFHLOG
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 SHARE OWN
MV29 PAYROL93 0044 007D9138 SHARE OWN
S=YES/STMS DFHSTRM PAYROL.FWDRECOV.UTL3
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 SHARE OWN
MV29 PAYROL93 0044 007D9138 SHARE OWN
S=YES/STMS DFHSTRM PAYROL.IYK8ZET1.DFHJ02
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 SHARE OWN
MV29 PAYROL93 0044 007D9138 SHARE OWN
S=YES/STMS DFHSTRM PAYROL.IYK8ZET1.DFSHUNT
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 EXCLUSIVE OWN
MV29 PAYROL93 0044 007D9138 EXCLUSIVE OWN
S=YES/STMS DFHSTRM PAYROL.IYK8ZET3.DFHJ02
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 SHARE OWN
MV29 PAYROL93 0044 007D9138 SHARE OWN
S=YES/STMS DFHSTRM PAYROL.IYK8ZET3.DFSHUNT
SYSNAME JOBNAME ASID TCBADDR EXC/SHR OWN/WAIT
MV29 PAYROL91 0042 007D9108 EXCLUSIVE OWN
MV29 PAYROL93 0044 007D9138 EXCLUSIVE OWN
```

- **Coupling facility status**

To display the MVS system logger couple data set status, issue the following MVS command:

```
D XCF,CPL,TYPE=LOGR
```

A normal response looks like this:

```
D XCF,CPL,TYPE=LOGR
IXC358I 14.47.51 DISPLAY XCF 391
LOGR COUPLE DATA SETS
PRIMARY DSN: SYS1.SYSPLEX2.SEQ26.PLOGR
  VOLSER: P2S055 DEVN: 230D
  FORMAT TOD MAXSYSTEM
  12/20/95 09:25:48 8
ALTERNATE DSN: SYS1.SYSPLEX2.SEQ26.ALOGR
  VOLSER: P2S056 DEVN: 2C10
  FORMAT TOD MAXSYSTEM
  12/20/95 09:27:45 8
LOGR IN USE BY ALL SYSTEMS
```

If the response shows that LOGR is not in use by all systems, there may be a problem to investigate. Look for IXCxxx messages which might indicate the cause of the problem and issue the following command to attempt reconnection to the couple data set:

```
SETXCF CPL,TYPE=(LOGR),PCOUPLE=(couple_dataset_name)
```

To display all structures with Failed_persistent connections, issue the following MVS command:

```
D XCF,STR,STRNM=*,STATUS=FPCONN
```
The MVS system logger should resolve any failed connections.

- SMF and RMF statistics

SMF 88 log stream statistics records and RMF coupling facility usage reports are useful for analyzing problems that are affecting performance. Increasing the amount of coupling facility storage allocated to a structure may improve both MVS system logger performance and CICSvr performance.

### Collecting diagnosis information

If you suspect that there is a problem within the MVS system logger that is not a result of some other resolvable problem, you may need to collect additional diagnostic information. The dumps generated by CICSvr generally will not contain sufficient information about the MVS system logger.

A dump of XCF and MVS system logger address spaces from all systems are useful in the diagnosis of such problems. Issue the following series of MVS commands:

```plaintext
dump comm=(meaningful dump title)
r ww,jobname=(ixglogr,xcfas,cicsvr_jobname),dspname=('xcfas'.*),cont
r xx,strlist=(strname=structure,(listnum=all),acc=nolim),cont
r yy,remote=(syslist=*('xcfas','ixglogr'),dspname,sdata),cont
r zz,sdata=(couple,allnuc,lpa,lsqa,psa,rgn,sqa,trt,csa,grsq,xesdata),end
```

Use the `r xx,strlist=(strname=structure,(listnum=all),acc=nolim),cont` instruction only where you suspect a problem with the coupling facility structure.

Error records written to the MVS LOGREC data set may also be useful.

### Restarting the MVS system logger address space

If the MVS system logger address space has failed, it can be restarted using the `s ixglogrs` command. Note the ‘S’ at the end—IXGLOGRS restarts IXGLOGR as a system address space.

**CAUTION:**
If you forcibly cancel the MVS system logger address space (by issuing a `force ixglogr,arm` command) and coupling facility structures used by the MVS system logger (by issuing a `setxcf force,con,strname=structname,conname=all` command), CICSvr will set the condition code to 3999 for this job step and no CICSvr VSAM batch logging will be performed. You should set up your batch jobs so they can detect the 3999 condition code and trigger a DFSMShsm or DFSMSdss backup of your VSAM sphere to ensure recoverability. See Figure 107 on page 281 for a DFSMShsm example or Figure 108 on page 282 for a DFSMSdss example.
Appendix D. Sample CLIST (DWWCLIST)

Figure 109 on page 288 shows a sample command list (CLIST) that you can use to start the CICSVR ISPF dialog interface. This sample CLIST is member DWWCLIST of the SDWWCNTL data set.
PROC 0
CONTROL FLUSH NOPROMPT MSG
/* --------------------------------------------------------------- */
/* PROPRIETARY V3 STATEMENT */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* 5655-H91 */
/* (C) COPYRIGHT 1991,2001 IBM CORP. */
/* END PROPRIETARY V3 STATEMENT */
/* --------------------------------------------------------------- */
/* FUNCTION: SAMPLE CLIST TO INVOKE THE CICSVR ISPFDIALOG. */
/* (ENGLISH) */
/* --------------------------------------------------------------- */
ISPEXEC CONTROL ERRORS RETURN /* RETURN IPF ERRORS TO CLIST */
ISPEXEC VGET ZUSER SHARED /* GET USERID */
ISPEXEC VGET ZPFSHOW PROFILE /* CHANGE PFSHOW SO THAT */
SET PFSAVE = &ZPFSHOW /* PF-KEY ARE DISPLAYED DURING*/
SET PFCMD = &STR(PFSHOW ON) /* CICSVR DIALOG INTERFACE */
ISPEXEC DISPLAY COMMAND(PFCMD)
ISPEXEC LIBDEF ISPFFILE
ISPEXEC LIBDEF ISPPLIB
ISPEXEC LIBDEF ISPMLIB
ISPEXEC LIBDEF ISPTLIB
FREE FI(DWWCON1,DWWCON2,DWWCON3,DWWMSG,DWWPRINT,MYFILE)
FREE FI(DWWSLIB,DWWLLIB)
FREE FI(DWWLOAD)
/* ---------------------------------------------------- DWWCON1 */
/* ------------------------------------------------------------- */
/* REPLACE THE FOLLOWING DWW.DWWCON1 WITH WHAT YOU */
/* HAVE SPECIFIED IN DWWRUNAR IVP JOB FOR DEFINE STEP. */
/* ------------------------------------------------------------- */
ALLOC FI(DWWCON1) DA('DWW.DWWCON1') SHR
SET RCSAVE = &LASTCC
IF &RCSAVE == 0 THEN DO
SET DDNAME = DWWCON1
GOTO SETMSG3
END
/* ---------------------------------------------------- DWWCON2 */
/* ------------------------------------------------------------- */
/* REPLACE THE FOLLOWING DWW.DWWCON2 WITH WHAT YOU */
/* HAVE SPECIFIED IN DWWRUNAR IVP JOB FOR DEFINE STEP. */
/* ------------------------------------------------------------- */
ALLOC FI(DWWCON2) DA('DWW.DWWCON2') SHR
SET RCSAVE = &LASTCC
IF &RCSAVE == 0 THEN DO
SET DDNAME = DWWCON2
GOTO SETMSG3
END

Figure 109. Sample command list (Part 1 of 4)
/* REPLACE THE FOLLOWING DWW.DWWCON3 WITH WHAT YOU HAVE SPECIFIED IN DWWRUNAR IVP JOB FOR DEFINE STEP. */
ALLOC FI(DWWCON3) DA('DWW.DWWCON3') SHR
SET RCSAVE = &LASTCC IF &RCSAVE ^= 0 THEN DO
  SET DDNAME = DWWCON3
  GOTO SETMSG3
END

/* ALLOC FI(DWWMSG) SYSOUT(X) DEST(XXXX) */
ALLOC FI(DWWMSG) DA('DWW.&ZUSER..DWWMSG') SHR
SET RCSAVE = &LASTCC IF &RCSAVE ^= 0 THEN DO
  SET DDNAME = DWWMSG
  GOTO SETMSG3
END

/* ALLOC FI(DWWPRINT) SYSOUT(X) DEST(XXXX) */
ALLOC FI(DWWPRINT) DA('DWW.&ZUSER..DWWPRINT') SHR
SET RCSAVE = &LASTCC IF &RCSAVE ^= 0 THEN DO
  SET DDNAME = DWWPRINT
  GOTO SETMSG3
END

/* ISPLLIB */
/* THE FOLLOWING LINE (LIBDEF FOR ISPLLIB) IS ADDED ONLY IF THE CICSVR LOADLIB'S ARE NOT ALLOCATED TO ISPLLIB IN THE LOGON PROC. */
/* REPLACE THE FOLLOWING DWW.SDWWLOAD WITH WHAT YOU HAVE SPECIFIED IN DWWALLOC JOB */
ISPEXEC LIBDEF ISPLLIB DATASET ID('DWW.SDWWLOAD')

/* DWWLOAD */
/* THE FOLLOWING ALLOCATION IS REQUIRED ONLY IF THE CICSVR LOADLIB'S ARE NOT ALLOCATED TO ISPLLIB IN THE TSO LOGON PROC. */
/* REPLACE THE FOLLOWING DWW.SDWWLOAD WITH WHAT YOU HAVE SPECIFIED IN DWWALLOC JOB */
ALLOC FI(DWWLOAD) DA('DWW.SDWWLOAD') SHR
SET RCSAVE = &LASTCC IF &RCSAVE ^= 0 THEN DO
  SET DDNAME = DWWLOAD
  GOTO SETMSG3
END

Figure 109. Sample command list (Part 2 of 4)
/* ----------------------------------------------- */
/* REPLACE THE FOLLOWING DWW.SDWWLOAD WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB */
/* REPLACE THE FOLLOWING DWW.DWWEXLD WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB */
/* ----------------------------------------------- */

SET &LIBS = &STR('DWW.SDWWLOAD')
SET &DSN = &STR('DWW.DWWEXLD')
SET &LIBS = &STR(&LIBS , &DSN)
ALLOC FI(DWWLLIB) DA(&LIBS) SHR
SET RCSAVE = &LASTCC IF &RCSAVE ˇ= 0 THEN DO
   SET DDNAME = DWWLLIB
   GOTO SETMSG3
END

/* ----------------------------------------------- ISPPLIB */
/* REPLACE THE FOLLOWING DWW.SDWPENU WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB. */
/* ----------------------------------------------- */

ISPEXEC LIBDEF ISPPLIB DATASET ID('DWW.SDWPENU')
SET RCSAVE = &LASTCC IF &RCSAVE ˇ= 0 THEN DO
   SET DDNAME = ISPPLIB
   GOTO SETMSG3
END

/* ----------------------------------------------- ISPMLIB */
/* REPLACE THE FOLLOWING DWW.SDWMMENU WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB. */
/* ----------------------------------------------- */

ISPEXEC LIBDEF ISPMLIB DATASET ID('DWW.SDWMMENU')
SET RCSAVE = &LASTCC IF &RCSAVE ˇ= 0 THEN DO
   SET DDNAME = ISPMLIB
   GOTO SETMSG3
END

/* ----------------------------------------------- ISPTLIB */
/* REPLACE THE FOLLOWING DWW.SDWTENU WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB. */
/* ----------------------------------------------- */

ISPEXEC LIBDEF ISPTLIB DATASET ID('DWW.SDWTENU')
SET RCSAVE = &LASTCC IF &RCSAVE ˇ= 0 THEN DO
   SET DDNAME = ISPTLIB
   GOTO SETMSG3
END

/* ----------------------------------------------- DWWSLIB */
/* REPLACE THE FOLLOWING DWW.SDWSENU WITH WHAT YOU HAVE */
/* SPECIFIED IN DWWALLOC JOB. */
/* ----------------------------------------------- */

SET &LIBS = &STR('DWW.&ZUSER..ISPFILE')
SET &DSN = &STR('DWW.SDWSENU')
SET &LIBS = &STR(&LIBS , &DSN)
ALLOC FI(DWWSLIB) DA(&LIBS) SHR
SET RCSAVE = &LASTCC IF &RCSAVE ˇ= 0 THEN DO
   SET DDNAME = DWWSLIB
   GOTO SETMSG3
END

Figure 109. Sample command list (Part 3 of 4)
/* ---------------------------------------------------- ISPFILE */
ALLOC FI(MYFILE) DA('DWW.&ZUSER..ISPFIL') SHR
IF &LASTCC=0 THEN DO
  ALLOC FI(MYFILE) DA('DWW.&ZUSER..ISPFIL') NEW DATACLASS(LIBRARY)
END
SET RCSAVE = &LASTCC IF &RCSAVE = 0 THEN DO
  SET DDNAME = MYFILE
  GOTO SETMSG3
END

ALLOC FI(ISPFILE) DA('DWW.&ZUSER..ISPFIL') SHR
SET RCSAVE = &LASTCC IF &RCSAVE = 0 THEN DO
  SET DDNAME = ISPFILE
  GOTO SETMSG3
END

/**************************************************************/
/* INVOKE CICSVR DIALOG PROGRAM. */
/**************************************************************/
ISPEXEC SELECT PGM(DWWPM) NEWAPPL(DWW) PASSLIB
SET RCSAVE = &LASTCC IF &RCSAVE = 0 THEN +
  SET ZERRLM = &STR(CICSVR DIALOGS COMPLETED, RC = &RCSAVE)
GOTO FINISH

/**************************************************************/
/* SET ERROR MESSAGE IF ANY, TO BE DISPLAYED ON ISPF PANEL. */
/**************************************************************/
SETMSG3: +
  SET ZERRLM = &STR(ALLOCATE OF DDNAME &DDNAME FAILED WITH RC= &RCSAVE)
FINISH: +
  ISPEXEC LIBDEF ISPFILE
  ISPEXEC LIBDEF ISPPLIB
  ISPEXEC LIBDEF ISPMPLIB
  ISPEXEC LIBDEF ISPMLIB
  ISPEXEC LIBDEF ISPTLIB
  ISPEXEC LIBDEF ISPLLIB /** ONLY IF NOT IN LOGON PROC **/
  FREE FI(DWWCON1,DWWCON2,DWWCON3,DWWMSG,DWWPRINT,MYFILE)
  FREE FI(DWWSLIB,DWLLIB)
  FREE FI(DWWLOAD)
  IF &RCSAVE = 0 THEN +
    SET ZERRHM = &STR(ISR00003) /* SET TO TOP ISPF TUTOR PANEL */
  IF &RCSAVE = 0 THEN +
    GOTO EXIT1 /* NORMAL EXIT */
  SETERROR: +
    SET ZERRMSG = &STR(ISRZ002) /* USE THIS ISPF MESSAGE NUMBER. */
    SET ZERRALRM = &STR(YES) /* BEEP AT USER */
    SET ZERRNM = &STR(ISR00003) /* SET TO TOP ISPF TUTOR PANEL */
EXIT1: +
    SET PFCMD = &STR(PFSHOW &PFSAVE) /* RESTORE PFKEY DISPLAY */
    ISPEXEC DISPLAY COMMAND(PFCMD) /* PREVIOUS STATUS */
    EXIT CODE(&RCSAVE)

Figure 109. Sample command list (Part 4 of 4)
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
Programming Interface Information

This book primarily documents information that is NOT intended to be used as Programming Interfaces of CICSVR.

This book also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of CICSVR. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface information

End of Programming Interface information

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

CICS®, CICS/ESA®, CICS/MVS®, DFSMSdhp, DFSMSdss, DFSMSHsm, DFSMS/MVS®

IBM®, MVS, MVS/ESA, NetView®, OS/390®, Parallel Sysplex®, RACF®, RETAIN®, z/OS

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

The terms in this glossary are defined as they pertain to the CICS VSAM Recovery documentation. If you do not find the term you are looking for, view the IBM Dictionary of Computing located at:

http://www.ibm.com/networking/nsg/nsgmain.htm

This glossary includes terms and definitions from:

- The American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

A

access method services (AMS). A utility program for the definition and management of VSAM data sets.

after-image. Records that CICS writes to a forward-recovery log to show what the VSAM record will look like after it has been updated by the application. (Throughout the CICSVR library, the forward-recovery log is referred to as the log.)

AIX. Alternate index.

alternate index (AIX). A collection of index entries related to a given base cluster and organized by an alternate key; that is, a key other than the prime key of the associated base cluster data records. The AIX gives an alternative directory for finding records in the data component of a base cluster.

AMS. Access method services.

APAR. Authorized program analysis report.

application identifier (APPLID). The name that identifies a CICS system to VTAM. It can be a maximum of 8 characters.

APPLID. Application identifier.

archive utility. The CICSVR utility that registers details of a log on the RCDS and optionally copies it to a backup.

authorized program analysis report (APAR). A report of a problem that is suspected to be caused by a defect in a current, unaltered release of a program.

automatic journal archiving. A function provided by CICS V4. When a disk log, defined to use this function is ready for archiving, CICS automatically creates and submits an archive job. The log data set is not reused until archiving is complete, and CICS ensures that the archive jobs are submitted promptly.

B

back up. The process of copying a data set to a backup volume.

backout. The CICSVR function that you can use if CICS fails in the attempt to back out uncommitted changes on a VSAM sphere. Using information from the RCDS, CICSVR constructs a job to back out uncommitted changes on a VSAM KSDS, ESDS, or RRDS, as indicated on the log.

backout failing log record (BOFLGREC). The record that CICS stores on the system log (throughout the CICSVR library, the system log is referred to as the log). This allows CICSVR to start and stop its scan of the log in the correct places and to locate the relevant before-images. CICS issues a BOFLGREC the first time a backout failure is detected. CICS issues following BOFLGREC's if the same task suffers a backout failure through a different file, or if a different task suffers a backout failure. So, there is a BOFLGREC for each combination of file and task that fails backout.

backup. The copy of the VSAM sphere, either on disk or tape, that you make at regular intervals as a minimum precaution to protect a VSAM sphere.
backup-while-open facility (BWO). The facility supported by DFSMS/MVS, CICS V4, and CICSVR, that lets CICS VSAM data sets be backed up while CICS is concurrently updating them. The data sets can then be recovered if data is lost. For the software levels required to use this facility, refer to CICSVR MVS/ESA V2R3 Implementation Guide.

base cluster. A key-sequenced or entry-sequenced data set that one or more alternate indexes can be built over, or a relative-record data set.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also ICF catalog.

batch backout. See backout.

BCS. Basic catalog structure.

before-image. The copy of a VSAM record that CICS saves in the system log before CICS updates the record (throughout the CICSVR library, the system log is referred to as the log). Before-images are used to back out incomplete or incorrect changes if a failure occurs.

BOFLGREC. Backout failing log record

buffer. An area of processing storage that is used to hold a block of data while it is waiting to be processed or written to an I/O device.

BWO. Backup-while-open facility.

C

CA. See change accumulation.

CA. Control area.

CBIPO. Custom-Built Installation Process Offering.

CBPDO. Custom-Built Product Delivery Offering.

CEDA. The main CICS-supplied transaction used to define resources online. When you use CEDA, you can update the CICS system definition (CSD) data set, and the running CICS system. Refer to CICS V4 CICS-Supplied Transactions and CICS V4 Resource Definition (Online) or CICS/MVS CICS-Supplied Transactions and CICS/MVS Resource Definition (Online)

CEMT. A CICS-supplied transaction used to invoke all the master terminal functions. These functions include inquiring and changing the value of parameters used by CICS, altering the status of system resources, terminating tasks, and shutting down CICS. Refer to CICS Supplied Transactions or CICS/MVS CICS-Supplied Transactions

CF. Coupling Facility.

change accumulation. A CICSVR utility that reduces the time it takes to perform a forward recovery. CICSVR change accumulation consolidates forward recovery log records into a CA data set. CICSVR uses the CA data set in conjunction with the forward recovery log to reduce the number of log records that CICSVR needs to apply to get the sphere back to the exact state before the data was lost.

CI. Control interval.

CICS. Customer Information Control System.

CICS session. The time period during which a user has access to a CICS system.

CICS system definition (CSD) data set. A VSAM KSDS cluster with alternate paths. The CSD data set contains a resource definition record for every record defined to CICS using resource definition online (RDO).

CICSp lex. (1) A CICS complex. A CICSp lex consists of two or more regions that are linked using CICS intercommunications facilities. The links can be either intersystem communication (ISC) or interregion communication (IRC) links, but within a CICSp lex are more usually IRC. Typically, a CICSp lex has at least one terminal-owning region (TOR), more than one application-owning region (AOR), and might have one or more regions that own the resources that are accessed by the AORs. (2) In CICSp lex SM, a management domain. The largest set of CICS regions or systems to be manipulated as a single CICSp lex SM entity. CICS regions in a CICSp lex SM CICSp lex do not need to be connected to each other.

CICSp lex SM. IBM CICSp lex System Manager for MVS/ESA. An IBM CICS system management product that provides a single system image and a single point of control for one or more CICSp lexes, including CICSp lexes on heterogeneous operating systems.

CICSVR. CICS VSAM Recovery.

cluster. In VSAM, a named structure consisting of a group of related components. For example, when the data is key sequenced, the cluster contains the data and index components; for data that is entry sequenced, the cluster contains only a data component. See also base cluster and alternate index.

cold start. The standard CICS initialization sequence performed without regard for prior system activity.

Common User Access (CUA). Guidelines for the interface between a user and a workstation or terminal.

complete recovery. The CICSVR function that consists of forward recovery followed by backout, if needed. In CICSVR complete recovery, CICSVR restores a DFSMShsm backup for you.
concurrent copy. The facility supported by DFSMS/MVS, CICS V4, and CICSVR that increases the availability of data by letting you make a consistent backup or copy of data, concurrent with normal application program processing.

control area (CA). A group of VSAM control intervals used as a unit for formatting a data set before adding records to it.

control area split. The movement of the contents of some VSAM control intervals in a control area to a newly created control area, to aid the insertion, or lengthening of a record when no free control intervals remain in the original control area.

control interval (CI). A fixed-length area of auxiliary-storage space where VSAM stores records and distributes free space. It is the unit of information that is transmitted to or from auxiliary storage, by VSAM.

ccontrol interval split. The movement of some stored records in a VSAM control interval to a free control interval, to aid the insertion, or lengthening of a record that will not fit in the original control interval.

Coupling Facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a sysplex.

CSD. CICS system definition data set.

CUA. Common User Access.

D

Data Facility Storage Management Subsystem (DFSMS/MVS). An IBM licensed program that together with MVS/ESA SP compose the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

data integrity. The quality of data that exists as long as accidental destruction, change, or loss

ddname. Data definition name.

deregister. The CICSVR function that removes a VSAM sphere name from the RCDS, or removes all references to a log from the RCDS.

DFDSS. Referred to in this book by its new product name. See DFSMSdss.

DFHCSDUP. CICS system definition (CSD) data set utility program. It provides offline services for the CSD. You can invoke DFHCSDUP as a batch program, or from a user-written program running in batch mode, or under TSO.

DFHJCRDS. The CICS journal-control record-mapping macro.

DFHSM. Referred to in this book by its new product name. See DFSMShsm.

DFP. Referred to in this book by its new product name. See DFSMSdfp.

DFSMSdfp. Data Facility Storage Management Subsystem data facility product.

DFSMSdss. Data Facility Storage Management Subsystem data set services.

DFSMShsm. Data Facility Storage Management Subsystem hierarchical storage manager.

DFSMSrmm. Data Facility Storage Management Subsystem removable media manager.

DFSMS/MVS. Data Facility Storage Management Subsystem/MVS.

dsname record. A record on a log that equates an FCT file name to a data set.

DTB. Dynamic transaction backout.

dynamic transaction backout (DTB). The process of canceling changes that a transaction makes to a VSAM data set after the transaction fails, for whatever reason.

E

emergency restart. Initialization of the CICS system following an abnormal end, where the information recorded on the system log is used to recover the data files of all interrupted transactions, to the condition they
were in when the transactions started. (Throughout the CICSVR library, the system log is referred to as the log.)

**entry-sequenced data set (ESDS).** A VSAM data set whose records are physically in the same order in which they were added to the data set. An ESDS is processed by addressed direct access, or addressed sequential access and has no index. Records are added at the end of the data set.

**ESA.** Enterprise Systems Architecture.

**ESDS.** Entry-sequenced data set.

**Extended Recovery Facility (XRF).** A related set of programs that lets an installation reach a higher level of CICS availability to end users. Availability is improved by having a pair of CICS systems: an active system and a partially initialized alternate system. The alternate system stands by to continue processing if failures occur on the active system.

**file.** A CICS entity that relates to a data set. File names are 1–8 characters.

**file control table (FCT).** CICS table containing the characteristics of the files accessed by CICS file control.

**FMID.** Function modification identifier.

**forward recovery.** The CICSVR function that reapplies all changes to the VSAM sphere since the last backup. The sphere can be a KSDS, ESDS, RRDS, or VRRDS. CICSVR gets the information it needs to construct the recovery job from the RCDS. The contents of the logs are applied to the VSAM sphere to return it to its exact state before the data was lost. With CICSVR forward recovery, CICSVR restores a DFSMShsm backup for you.

**forward-recovery log.** A log that is being used for implementing forward recovery. (Throughout the CICSVR library, the forward-recovery log is referred to as the log.)

**function modification identifier.** A seven-character ID used to identify the release of a product.

**G**

**GDG.** Generation data group.

**generation data group (GDG).** A collection of data sets kept in chronological order; each data set is a generation data set.

**global user exit.** A point in a CICS module at which CICS can pass control to a program that you have written (an exit program) and then resume control when your program has finished. When an exit program is enabled for a particular exit point, the program is called every time the exit point is reached.

**ICF catalog.** Integrated catalog facility catalog.

**in-flight transaction.** A transaction that has uncommitted updates at the time of an abnormal CICS end.

**instance.** An instance of CICSVR starts when transaction VSAM is initialized as part of SMSVSAM address space initialization or enabled by operator command. It ends when transactional VSAM enters a quiesced or disabled state, or when the SMSVSAM address space is terminated.

**Integrated Catalog Facility (ICF) catalog.** A catalog that consists of a basic catalog structure (BCS) and its related volume table of contents (VTOCs), and VSAM volume data sets (VVDSs). The ICF catalog is the only catalog that is supported by DFSMS/MVS. See also **basic catalog structure (BCS), volume table of contents (VTOC), and VSAM volume data set (VVDS)**.

**Interactive System Productivity Facility (ISPF).** The MVS interactive facility that serves as a full-screen editor and dialog manager. ISPF can be used for writing application programs. It is used by CICSVR to provide an interactive dialog between the CICSVR user and the CICSVR functions.

**I/O.** Input/output.

**ISPF.** Interactive System Product Facility.

**J**

**JACD.** Journal archive control data set.

**JCT.** Journal control table.

**journal.** See log.

**journal archive control data set (JACD).** CICS V4 system data set for use by the CICS automatic journal archive facility to store information about the logs.

**journal control table (JCT).** The way by which the characteristics of the logs are described to CICS for access through journal control. The JCT contains journal information and operating system control blocks describing each log.

**journaling.** The recording of information onto a journal (including the system log) for processing by CICSVR. Also known as **logging**.
**journal-label record.** A special record type that is the first record written out by CICS in a block of log records.

**journal partitioned data set (JPDS).** A CICS V4 system data set used with the automatic journal archive facility. Each member of this data set contains skeleton JCL for use by the CICS automatic archive job submission program.

**JPDS.** Journal partitioned data set.

**K**

**keypoint.** The periodic recording of system information and control blocks on the system log (throughout the CICSVR library, the system log is referred to as the log).

**key-sequenced &dat (KSDS).** A VSAM data set whose records are loaded in key sequence and controlled by an index.

**KSDS.** Key-sequenced data set.

**L**

**linear data set.** A VSAM data set that contains data but no control information. A linear data set can be accessed as a byte-addressable string in virtual storage. See recovery control data set.

**link pack area (LPA).** In MVS, an area of virtual storage that contains re-enterable routines that are loaded at IPL time and that can be used concurrently by all tasks in the system.

**local shared resources (LSR).** Files that share a common pool of buffers and a common pool of strings; that is, control blocks supporting I/O operations.

**log.** A set of one or more sequential data sets to which records are written during a CICS session in these circumstances:
- By CICS, to implement user-defined resource protection (logging to the system log)
- By CICS, to implement user-defined automatic journaling (to a journal, including the system log)
- Explicitly, by the JOURNAL command (or macro), from an application program (to a journal, including the system log)

(Throughout the CICSVR library, all journals are referred to as logs.)

**log manager.** A CICS V4 domain introduced in CICS Transaction Server, which replaces the CICS journal control management function of earlier CICS versions. The CICS V4 log manager uses MVS system logger services to write CICS systems logs, forward-recovery logs, and user journals to log streams managed by the MVS system logger. (Throughout the CICSVR library, system logs, forward-recovery logs, and MVS log streams are referred to as logs.)

**log of logs.** A log created by CICS Transaction Server that contains records that are written each time a file is opened or closed. CICSVR scans the log of logs and saves information needed for recovery in the RCDS.

**log tail.** In CICSVR, the oldest log record of interest. Log tail deletion is the process of deleting unneeded records that are older than the oldest record of interest to CICSVR.

**local shared resources (LSR).** Files that share a common pool of buffers and a common pool of strings; that is, control blocks supporting I/O operations.

**logical unit of work (LUW).** A sequence of processing actions (for example, changes to a base cluster) that must be completed before the individual actions can be regarded as committed. Every CICS task that affects a recoverable resource consists of one or more LUWs. When changes are committed (by successful completion of the LUW and recording of the sync point on the system log), they need not be backed out after a later failure of the transaction or system. The end of an LUW is marked in a transaction by a sync point, issued either by the user program or by CICS when the transaction ends. In the absence of user sync points, the entire task is an LUW.

**LPA.** Link pack area.

**LSR.** Local shared resource.

**LUW.** Logical unit of work.

**M**

**master terminal operator (MTO).** A CICS operator who is authorized to use the master-terminal-functions transaction.

**menu bar.** The area at the top of a window that contains choices that let the CICSVR user access the actions available in that window.

**migration utility.** The utility provided by CICSVR that helps you upgrade your RCDS.

**MTO.** Master terminal operator.

**MVS/ESA.** An MVS operating system environment that supports ESA/370.

**MVS/ESA SP.** An IBM licensed system product. MVS/ESA SP together with DFSMS/MVS compose the base MVS/ESA operating environment.
object action. A process sequence in which the user selects an object and then selects an action to apply to that object.

online. Pertaining to a user’s access to a computer through a terminal. The term online is also used in this book to describe a resource (for example, a data set) being used by a user through a terminal.

P

path. A data set name for the relationship between an alternate index and its base cluster, or an alias for a VSAM data set.

PDF. Program Development Facility.

PMR. Problem management record.

problem management record (PMR). A record on the RETAIN database where all activity about your CICSVR problem is recorded.

program temporary fix (PTF). A temporary solution, or by-pass of a problem, diagnosed by IBM as resulting from a defect in a current, unaltered release of a program.

program update tape (PUT). A tape or cartridge on which IBM places PTFs so that you can install them on your system.

PTF. Program temporary fix.

pull-down. A list of choices associated with a choice on the menu bar. The CICSVR user selects a choice from the menu bar, and a pull-down appears in the secondary window, under the choice.

PUT. Program update tape.

R

RBA. Relative byte address.

RCDS. Recovery control data set.

RDO. Resource definition online.

record level sharing. See VSAM record level sharing.

recovery. (1) The general process of recovering VSAM data by CICS or CICSVR. (2) In DFSMSHsm, the process of copying a backup version of a data set from a backup volume to a specified volume, possibly to the volume from which the backup version was made.

recovery control. In CICSVR, the collective name for the functions that keep track of all the information needed to forward recover and back out protected VSAM spheres.

recovery control data set (RCDS). One of three identical linear VSAM data sets that contain information about the contents of archived logs and the ISPF dialog interface default values. CICSVR uses this stored information to construct recovery jobs. CICSVR uses three identical RCDSs to reduce the possibility of data loss.

recovery point time. The point in time that forward recovery starts from for VSAM data sets that were restored from a backup made using the backup-while-open facility. With the backup-while-open facility, recovery point time is a maximum of 30 minutes before the actual backup time.

register. See archive function.

relative byte address (RBA). The displacement of a stored record or control interval from the beginning of the storage space allocated to the VSAM data set to which it belongs.

relative-record data set (RRDS). A VSAM data set whose records are loaded into fixed-length slots. The records are accessed by a relative record number (RRN).

Remote Technical Assistance Information Network. See RETAIN.

request parameter list (RPL). In ACF/VTAM, a control block that contains the parameters needed for processing a request for data transfer.

resource definition macro. A method of defining resources to CICS using macros. You code and assemble special macro instructions, and then provide CICS with these assembled tables at initialization time.

resource definition online (RDO). The recommended method of defining resources to CICS by creating resource definitions interactively, or using the utility DFHCSDUP, and then storing them in the CICS system definition (CSD) data set. These definitions are then installed as CICS system tables, by specifying a list of definitions at CICS initialization time. Using the CEDA transaction, resource definitions can be installed while CICS is active, so they can be used immediately.

restore. (1) The process of copying a backup version of a VSAM data set from backup media, to the same media from which the backup version was created, or to another media. This restored copy can then be used in CICSVR forward recovery or backout. (2) In DFSMSHsm, the process of invoking DFSMSdss to perform the recover function before running CICSVR complete recovery.
RETAIN. A software system used by IBM Support Centers and other IBM offices to solve problems with IBM products. RETAIN is used to document each problem and the correction developed for it.

RPL. Request parameter list.

RLS. VSAM record level sharing.

RRDS. Relative-record data set.

S

SAA. Systems Application Architecture.

secondary window. The window you get when you select an option from a pull-down. A secondary window does not have a menu bar.

SIT. System initialization table.

SNA. System Network Architecture.

SMF. System Management Facility.

SMS. Storage Management Subsystem.

sphere. See VSAM sphere.

storage management subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, and ACS routine definitions.

sync point. See synchronization point.

synchronization point (sync point). A point in the processing of a task at which changes to recoverable resources are regarded as committed.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multi-system hardware components and software services to process customer workloads.

system initialization table (SIT). A CICS control table required for the system to be operational. The SIT controls the capability of the system through a set of system initialization parameters.

system log. A CICS log (ID=01) that is used by CICS to log changes to resources for backout. (Throughout the CICSVR library, the system log is referred to as the log.)

system logger. A central logging facility provided by MVS/ESA 5.2. The MVS system logger provides an integrated MVS logging facility that can be used by system and subsystem components. For example, it is used by the CICS Transaction Server log manager.

System Management Facility (SMF). An MVS component that collects and records system and job-related information.

Systems Application Architecture (SAA). A formal set of rules that enables applications to be run without modification, in different computer environments.

T

task. In CICS, a single instance of the execution of a transaction. Contrast with transaction.

tie-up record (TUR). The association between the file and data set, as recorded on the log.

transaction. Can be regarded as a unit of processing (consisting of one or more application programs) initiated by a single request, often from a terminal. A transaction might require the initiation of one or more tasks for its execution. Contrast with task.

transaction backout. The cancelation, because of a transaction failure, of all updates performed by a task.

TUR. Tie-up record.

U

uncommitted updates. The updates from an incomplete LUW that are left on the &sphere when a task or CICS abends.

upgrade set. All the alternate indexes that VSAM has been instructed to update whenever there is a change to the data part of the base cluster.

V

variable relative-record data set (VRRDS). A VSAM data set whose records are loaded into variable-length slots. The records are accessed by a relative record number (RRN).

volume table of contents (VTOC). A table on a direct access volume that describes each data set on the volume.

VRRDS. Variable relative-record data set

VSAM. Virtual Storage Access Method.

VSAM record level sharing (VSAM RLS). An extension to VSAM which provides direct record level sharing of VSAM data sets from multiple address spaces across multiple systems. Record level sharing utilizes the System/390 Coupling Facility to provide cross system locking, local buffer invalidation, and cross...
system data caching. With VSAM RLS, CICS regions that share VSAM data sets can reside in one or more MVS images within a parallel sysplex.

**VSAM sphere.** A base cluster, together with any alternate indexes defined with it.

**VSAM volume data set (VVDS).** A data set that describes the characteristics of VSAM data sets and system-managed data sets residing on a given disk; part of an ICF catalog.

**VTOC.** Volume table of contents.

**VVDS.** VSAM volume data set.

**X**

**XA.** Extended Architecture.

**XRF.** Extended Recovery Facility.
Index

A
ABARS 70, 136
activity keypoint processing  84, 147, 148
administrative procedures
naming conventions  89, 150
after-image  191, 234
  with preapply exit  262
AIX  146
AKPFREQ (SIT option)  84, 148
ALLOC TSO command  44, 110
allocating
  RCDS  43, 109
allocating logs  168, 206
alternate index (AIX)
  automatic rebuilding  7
  automatic removing  7
  considerations for BWO recovery  81
  recovery restriction  81
  reusable  7
AMP JCL DD parameter  31
AMS commands
  ALTER  46, 47, 81, 112, 113
  BLDINDEX  81
  BWO  86
  DEFINE CLUSTER  43, 45, 46, 47, 109, 111, 112, 113
  DELETE  46, 81, 112
  SHAREOPTION  47, 113
APPLID (application identifier)  81, 209
APPLYCA keyword  194, 237
archive utility
  exits  257, 260
  using  153
  using the exits  257, 260
  with the DEFEXIT command  175, 222
archive utility report  116
archived logs
  retention periods  113
archiving
  naming conventions  90, 150
AUTODEREG keyword  179, 210
autojournaling  65
automatic AIX processing  7
automatic archiving  114
automatic deregister  210
automatic journal archiving  135
automatic log deregister  114
back-level data set  81
backing up your VSAM spheres  69
backout
  changing the CICSVR return code  262, 273
  for an ESDS  261
  of application errors  262
  backout-failure control  10
backu-while-open  80, 145
  eligible data sets  80
backup copy
  naming conventions  89, 150
  unique names  89, 150
backup-while-open facility
  acceptable VSAM errors  266
  AKPFREQ SIT option  84, 148
  alternate index considerations  81
  back-level data set  81
  BACKUPTYPE (RDO file option)  67, 87
  CICS activity keypoints  84, 147, 148
  control-area splits  81
  control-interval splits  81
  DFSMSdss
    logical dump  81
    physical dump  80, 146
  file definitions  68, 88
  flags in ICF catalog  81
  operation  81
  recovery point time  82
  related CICS error messages  81, 82
  TUR (tie-up record)  84, 147
  with forward recovery  80
BACKUPTIME keyword  200, 251
BACKUPTYPE (RDO file option)  67, 87
batch logging
  enabling  28
  planning log streams  32
  RACF  33
  setting up  27
  setting up logging environment  33
  setting up the logging environment  32
  understanding  32
  using  27
before-image  214
  with preapply exit  262
BookManager  xix
BWO for CDS  147

C
CA (change accumulation)
  setting up CICSVR (for CICS TS)  52
  setting up CICSVR (for CICS V4)  121
  utility report (for CICS TS)  55
  utility report (for CICS V4)  124
CA command  173, 220
  calculating the RCDS size  42, 108
  capabilities of CICSVR  6
  CATALOG keyword  186
CEDA commands
  DEFINE FILE  67, 87
  CEDA DEFINE FILE command  133, 146
CEMT command  152
CEMT commands
  INQUIRE DSN  82
  SET DSN  11
change accumulation batch jobs 91, 151
change accumulation (CA)
  setting up CICSVR 121
  utility report (for CICS TS) 55
  utility report (for CICS V4) 124
change accumulation, setting up 52
change accumulation batch jobs 121
change accumulation groups 69, 136
change accumulation utility 5, 6
CICS
  abend 8
  activity keypoints 84, 147, 148
  after-image 262
  backout-failure control 10
  before-image 262
  DFHIJCRDS dsect 259, 270
  DTB (dynamic transaction backout) 8, 10
  emergency restart backout 8
  LUW (logical unit of work) 9
  MTO 81, 82
  online backout failure 9
  recoverable resources 8, 9
  restart data set 8
  sync point 9
  system abend 8
  transaction backout 8
  TUR (tie-up record) 84, 147
CICS logging and journaling 65
CICS/MVS
  acceptable VSAM errors 266
CICS Transaction Server 4, 5
  CICSVR functions 3
  log of logs scan utility 4
CICS Transaction Server for OS/390 Release 1
  VSAM RLS with 85
CICS V4 6
  archive utility 5
  backout 6
  backout-failure control 10
  CICSVR functions 5
  FCT definitions 87
  forward recovery 6
  making the CSD data set recoverable 89, 149
  preventing logical inconsistencies 11
CICSID keyword 209
CICSVR functions 13
CICSVR server address space
  activating 17
CID keyword 213
CLIST 39, 105
command reference 165
commands 188, 220, 231
  CA 173
  coding rules 161
  DEFEEXIT
    examples 175, 222
    format 175, 222
    keywords 175, 222
    usage 175, 222
    used with ARCHIVE 175, 222
commands (continued)
  LOGOFLOGS
    examples 180, 188
    keywords 179
    usage 179
  LOGSTREAMCOPY
    examples 184
    format 181
    keywords 181
    synonyms 185
    usage 183
  SPHERE 200, 251
  syntax explanation 161
  used with CICS TS 167
  used with CICS V4 205
  completion code 262, 273
  concurrent copy 85, 148
  control-area copy 81
  control-interval splits 81
  COPY keyword 181, 185, 210
  COPY command 71, 137
  COPY keyword 176, 186, 223
  copying logs 210
  copying MVS log streams 181
  coupling facility
    coupling facility logging 33, 49
    parallel sysplex 85
  CPS keyword 185
  creating a new RCDS 42, 108
  CSD (CICS system definition) data set
    making recoverable
      CICS SIT options 89
      CICS V4 SIT options 149
    CSDBKUP (SIT option) 89, 149
    CSDFRLOG (SIT option) 89, 149
    CSDRECOV (SIT option) 89, 149
D
DASD-only logging 33, 50
data class define panel 28
data loss
  logical 3, 265
  physical 3, 265
data set
  naming conventions 89, 150
  nonrecoverable 79
  recoverable 79
DATASETNAME keyword 198, 217, 242, 244
DD keyword 180
DDN keyword 180
DDNAME keyword 180
ddnames
  DWWARC1 154
  DWWCON1 37, 44, 103, 109
  DWWCON2 37
  DWWCON3 37
  DWWCOPY 154
  DWWDMSG 38, 104
  DWWDUMP 38, 104
  DWWIN 154
ddnames (continued)
DWWLLIB 38
DWWLOAD 154
DWWMSG 37, 103, 154
DWWPRINT 37, 154
DWWSLIB 38
ISPFILE 38, 103
ISPLL 38, 103
ISPLMLIB 38, 103
ISPPLIB 38, 103
ISPPROF 38, 104
ISPSLIB 38, 104
ISPTLIB 38, 104

ddnames for CICSVR V3R1 277
define a data class 29
defining
exits 175, 222
RCDS 43, 109
defining exits 177, 224
defining files 133
defining journals 134
defining log of logs 67
defining recovery attributes 67, 85
DEREG keyword 180, 213
DEREGISTER keyword 180
DFHJCRDS 259, 270
DFSMSdss (Data Facility Storage Management Subsystem data set services)
  logical dump 81
  physical dump 80, 146
DFSMSdss backup utility 71, 137
DFSMSdss COPY 71, 137
DFSMSdss DUMP 72, 138
DFSMSHsm (Data Facility Storage Management Subsystem hierarchical storage manager)
  CICS V4, without 81
  SETSYS command 81
DFSMSHsm backup utility 70, 136
diagrams
  error exit parameters 267
  ESDS delete exit parameters 270
  online backout failure in CICS V4 10
  preapply exit parameters 272
  precopy exit parameters 259
  termination exit parameters 260, 274
  transaction backout concepts 9
disaster recovery 91, 151
  making backups of VSAM spheres 91, 151
  preparing your remote site 98
DSN keyword 202, 203, 244, 253, 254
DSNAME (RDO file option) 67, 87
DSNAME keyword 198, 217, 242, 244
DTB (dynamic transaction backout) 8, 10
DWWAR 211
DWWARC1 211
DWWARC1 ddname 154
DWWCON1 ddname 37, 44, 103, 109
DWWCON2 ddname 37
DWWCON3 ddname 37
DWWCOPY 182, 210
DWWCOPY ddname 154
DWWDMSG ddname 38, 104
DWWDUMP ddname 38, 104
DWWIN ddname 154
DWWLC 184
DWWLLIB ddname 38
DWWLOAD ddname 154, 178, 225
DWWMSG ddname 37, 103, 154
DWWPRINT ddname 37, 154
DWWSLIB ddname 38

E
EDEL keyword 179, 226
editing the JCL skeleton 41, 107
calling conventions 257, 263
common parameters 258, 264
communication with CICS V4 258, 264
DEFEXIT command 263
defining 263
derror
  description 261
  finding the parameters 267
  parameter list 266
  parameters passed to program 267, 269
  parameters returned from program 269
  when called 265
editor (ISPF) 40, 106
editor (ISPF) panels
  main menu 39, 105
report descriptions
  archive statistics 117
  backout failing log records found 119
  redo file ID without tie-ups 118
  tie-up statistics 118
  VSAM data set statistics 118
reports
  log of logs scan 50
  log stream copy 95
secondary windows
  exits 263
  JCL skeleton 40, 107
exits
  calling conventions 257, 263
  common parameters 258, 264
  communication with CICS V4 258, 264
  DEFE exit command 263
defining 263
derror
  description 261
  finding the parameters 267
  parameter list 266
  parameters passed to program 267, 269
  parameters returned from program 269
  when called 265

Index 305
exits (continued)
ESDS delete
description 261
finding the parameters 270
parameter list 269
parameters passed to program 270
parameters returned from program 270
when called 269
installing 175, 222
link-editing 257, 263, 265
parameter lists 258, 264
preapply
description 262
finding the parameters 272
parameter list 271
parameters passed to program 272
parameters returned from program 273
when called 271
precopy (for archive)
description 257
finding the parameters 259
parameter list 258
parameters passed to program 259
parameters returned from program 259
when called 258
programming language 257, 263
register conventions 257, 263
sample link-edit JCL 265
secondary window 263
termination
changing the CICSVR return code 262, 273
description 262
finding the parameters 274
parameter list 273
parameters passed to program 274
parameters returned from program 274
when called 273
termination (for archive)
description 257
finding the parameters 260
parameter list 259
parameters passed to program 260
parameters returned from program 260
when called 259
EXPORT keyword 188, 231

forward-recovery log (continued)
log stream 66
TUR (tie-up record) 84, 147
forward recovery logging 65
functions
CICS TS 13
CICS V4 13
FWDRECOVLOG (RDO file option) 67, 87

G
GAPINSEQUENCE keyword 245, 248
general log streams 66
GMT keyword 192, 198, 200, 235, 242, 251
GROUP keyword 173, 220

H
hardware requirements xviii
HSMDATE keyword 195, 198, 238, 242

I
I/O
error handling 261, 265
ICF catalog
with the backup-while-open facility 82
IDCAMS ALTER
FRLOG(REDO) 30
IDCAMS DEFINE CLUSTER
FRLOG(R) 29
IMPORT keyword 188, 231
in-flight transaction 8
installing
exits 175, 222
the ISPF environment
allocating the data sets to TSO 37, 103
editing the JCL skeleton 41, 107
getting the JCL skeleton 40, 106
modifying the ISPF menu 39, 104
running the interface 38, 104
selecting from TSO 39, 105
using the ISPF select service 40, 106
installing exits 177, 224
introducing CICSVR
capabilities and limitations
alternate indexes 7
backout 7
DFSMShsm 7
logging 7
VSAM data set attributes 7
CICS transaction backout
description 8
DTB failure in CICS V4 10
failure 9
preventing logical inconsistencies 11
ISMF dialog interface 28
ISPF dialog interface
teditor 40, 106
installing 37, 103
ISPFILE ddname 38, 103

F
FCT (file control table)
resource definitions for CICS V4 87
FCTNAME keyword 227
force continuation of archive 230
FORMAT keyword 198, 237, 242
forward recovery 4
changing the CICSVR return code 262, 273
for an ESDS 261
from application errors 262
log failure 89
with the backup-while-open facility 80
forward-recovery log
failure 89
ISPLLIB ddname 38, 103
ISPMLIB ddname 38, 103
ISPLIB ddname 38, 103
ISPPROF ddname 38, 104
ISPSLIB ddname 38, 104
ISPSTART TSO command 39, 105
ISPTLIB ddname 38, 104

J
JCL (job control language)
automatic archiving 150
DWWARC1 154
DWWCON1 37, 44, 103, 109
DWWCON2 37
DWWCON3 37
DWWCOPY 154
DWWDMG 38, 104
DWWDUMP 38, 104
DWWIN 154
DWWLLIB 38
DWWLOAD 154
DWWMSG 37, 103, 154
DWWPRINT 37, 154
DWWSLIB 38
ISPFILE 38, 103
ISPLLIB 38, 103
ISPMLIB 38, 103
ISPLIB 38, 103
ISPPLIB 38, 103
ISPPROF 38, 104
ISPSLIB 38, 104
ISPTLIB 38, 104
running archive 154
skeleton 40, 107
substitution variable 150
to allocate the RCDS 44, 109
to link-edit an exit 265

JID keyword 213
JOBLIB ddname 178, 225
journal archive job skeleton 114
JOURNAL keyword 170, 208
JOURNALID keyword 209
JPDS (journal partitioned data set) 209

K
keywords
COPY 176, 223
PRECOPY 175, 176, 222, 223
SCAN 179
synonyms 164
TERM 176, 223
TERMINATION 175, 222

L
limitations of CICSVR 6
link-edit JCL 265
LOCAL keyword 192, 198, 200, 235, 242, 251
locks 88
LOG keyword 168, 206
log of logs
defining 67
improving performance 49
sample scan job 50
scan report 50
setting up 49
log of logs data sets
deregistering 47
registering 47
log of logs scan utility 48
improving performance 49
setting up 49
setting up as a batch job 50
update the RCDS 92
log stream blocks
retention periods 48
log stream copies
retention periods 48
log stream copy utility 4, 93
report 95
log tail 48
logging
collecting information 285
problem categories 281
logical dump 81
logical recovery 262
logon procedure (TSO) 39, 105
logs
archive naming conventions 90, 150
naming conventions 90, 150
record modification 262, 271
setting up retention periods 113
TUR (tie-up record) 84, 147
unique names 89, 150
LSR (local shared resources) 171, 218
LUW (logical unit of work) 9, 214

M
main menu 39, 106
master application menu (ISPF) 39, 104
migrating from CICSVR V2R3 to CISCVR V3R1 13
migrating RCDS 14
migrating to CICS TS 13
MOD 183, 211
MSGID 39, 105
MTO (master terminal operator) 81, 82
MVS
log stream
forward recovery 66
general 66
system 66
system clock 82
MVS log streams 66
MVS System Logger
availability 282
coupling facility status 284
dumps 282
GRS contention 283
MVS System Logger (continued)
  restarting 285
  RMF statistics 285
  SMF statistics 285
  MVSID keyword 209

N
NAME keyword 181, 186, 200, 251
naming conventions 89, 150
NaviQuest batch job 29
NEWDATASETNAME keyword 198, 217, 242, 244
NEWDSN keyword 198, 217, 242, 244
NEWDSNAME keyword 198, 217, 242, 244
NEWPATH keyword 243
NEWSPHERE keyword 193, 216, 236

O
online CICSVR books xviii
operational activities
  naming conventions 89, 150
optional software xvii
OUTOFSEQUENCE keyword 245, 248
overview
  capabilities 6
  CICS backout 8
  limitations 6
  VSAM RLS 85

P
panels
  main menu example 39, 105
parallel sysplex 85
passing record-format information 227
PATH keyword 243
performance considerations 171, 218
physical dump 80, 146
planning for CICSVR
  backup strategy 69, 135
  naming conventions 89, 150
  protecting VSAM spheres 112
  recovery strategy 69, 135
PREAPP keyword 179, 226
preapply exit 271
PREAPPLY keyword 177, 224
precopy exit 258
PRECOPY keyword 175, 222, 223
PREFIX keyword 173, 220
PROFILE TSO command 39, 105

R
RBA (relative byte address) 267, 268
RCDS (recovery control data set) (continued)
  maintaining 44
  migrating 14
  repairing 44, 110
  sizing
    CICS TS 42
    CICS V4 108
  understanding 41, 107
  what to do when they are full 44, 110
RCDS command 188, 231
RCDS EXPORT command 92
RCDS export/import utility 5, 6
RCDS utility 92, 152
RDO file definition 67, 87
RECFM keyword 229
RECFORMAT keyword 227
record-format information 227
RECORDFORMAT keyword 227
recovery
  automatic AIX processing 7
  changing the CICSVR return code 262, 273
  concepts 3
  for an ESDS 261
  from application errors 262
  strategy 69, 135
RECOVERY (RDO file option) 67, 87
recovery point time 82
RECOVERYREPORT keyword 179, 180, 210
RECREP keyword 180
REG keyword 180
REGISTER keyword 180
registering logs in the RCDS 209
Remote Recovery Data Facility 194, 237
remote site recovery 194, 237
report descriptions
  archive statistics 117
  backout failing log records found 119
  redo file ID without tie-ups 118
  tie-up statistics 118
  VSAM data set statistics 118
reports
  log of logs scan 50
  log stream copy 95
RESETSEQUENCE keyword 245, 248
resource definitions
  CICS CSD data set 89
  CICS V4 CSD data set 149
  FCT for CICS V4 87
retention periods, setting up 48
return code 262, 273
reusable AIX 7
RLSACCESS RDO definition 68, 87
RPL (request parameter list) 268
RRDF 194, 237
running CICSVR
  ISPF dialog interface 38, 104

S
sample JCL
  running archive 154
sample JCL  (continued)
  to link-edit an exit  265
SCAN keyword  179
scan utility  48
  improving performance  49
  setting up  49
  setting up as a batch job  50
  update the RCDS  92
scan utility report  50
secondary windows
  examples
    exits  263
    JCL skeleton  40, 107
SEL keyword  185
SELECT keyword  181, 185
sequence checking of logs  245, 248
SETSYS DFSMshsm command  81
setting up change accumulation  52
setting up CICSVR
  for CICS TS  37, 65
  for CICS V4  103, 133
setting up CICSVR VSAM batch logging  27
setting up scan utility  49
SHADOW  194, 237
shadow forward recovery  119
SHADOW keyword  198, 242
SIT options
  AKPFREQ  84, 148
  CSDBKUP  89, 149
  CSDFRLG  89, 149
  CSDRECOV  89, 149
sizing a new RCDS (using CICS V4)  108
sizing the RCDS (using CICS TS)  42
software requirements  xvi
SPHERE command  200, 251
SPHERE keyword  191, 214, 234
START keyword  180, 198, 217, 229, 242
STARTAT keyword  194, 237
STARTBLK keyword  185
STARTBLKID keyword  182, 185
STARTDATE keyword  198, 217, 229, 242
STARTDATETIME keyword  198, 217, 229, 242
STARTPOINT keyword  180
STARTTIME keyword  191, 214, 227, 234
STARTTOD keyword  192, 198, 235, 242
STEPLIB ddname  178, 225
STOP keyword  198, 217, 242
STOPBLK keyword  185
STOPBLKID keyword  183, 185
STOPDATE keyword  198, 217, 242
STOPDATETIME keyword  198, 217, 242
STOPTIME keyword  182, 192, 215, 235
STOPTOD keyword  193, 198, 236, 242
strategy planning  69, 135
sync point  9
system clock  82
system log streams  66
system logger
  with CICS  79
system logging  65

T
TERM keyword  176, 179, 223, 226
termination exit  259, 273
TERMINATION keyword  175, 177, 222, 225
TSO
  CLIST  39, 105
  commands
    ALLOC  44, 110
    ISPSTART  39, 105
    PROFILE  39, 105
  logon procedure  39, 105
  selecting CICSVR from  39, 105
  session  37, 103
TSO COPY  154
TUR (tie-up record)  84, 147, 194, 238

U
UNIT keyword  168, 173, 195, 198, 206, 220, 238, 242
unit of work (UOW)  66
UOW (unit of work)  66
upgrade set
  removing alternate indexes  81
user journaling  65
using CICSVR
  command reference  161, 165
  without DFSMShsm  81
using ISMF dialog interface  28
utility report (for CICS V4)
  change accumulation  55, 124

V
VERSION keyword  195, 198, 238, 242
VOLUME keyword  168, 173, 195, 198, 206, 220, 238, 242
VSAM
  acceptable errors  266
  control-area splits  81
  control-interval splits  81
  logical data loss  3
  lost or damaged data  3
  physical data loss  3
  RLS (record level sharing)
    and log failure  89
    description  85
    locks  88
    nonrecoverable data sets  79
    parallel sysplex  85
    understanding  85
  serious errors  266
  sphere
    I/O error  9, 10, 11
VSAM data set
  nonrecoverable  79
  recoverable  79
VSAM spheres
  scheduling backups  70
VSAM spheres
  additional backups  69, 135
VSAM sphres (continued)

daily backups 69, 135
how to group your backups 69, 136
nondaily backups 69, 135
scheduling backups 136

W

what CICSVR cannot do 6
write-to-programmer messages 39, 105
WTPMSG 39, 105
Readers’ Comments — We’d Like to Hear from You

CICS VSAM Recovery
Implementation Guide
Version 3 Release 1

Publication No. SH26-4126-00

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall satisfaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How satisfied are you that the information in this book is:

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  □ Yes  □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.