

Version 2 Release 1

*IBM Cloud Tape Connector for z/OS
User Guide*



Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

June, 2020 edition

This edition applies to Version 2 Release 1 of IBM Cloud Tape Connector for z/OS (product number 5698-ABM) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2016, 2020

© **Copyright International Business Machines Corporation .**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information.....	xi
Chapter 1. Summary of changes.....	1
Chapter 2. Introduction to Cloud Tape Connector.....	5
Cloud Connector Processing.....	5
Chapter 3. Configuring Cloud Tape Connector.....	7
Requirements.....	7
Choosing Cloud Connector configuration options.....	7
Method 1: Write directly to cloud.....	7
Method 2: Staging data to DASD.....	7
Method 3: History Processing.....	8
Determining which cloud copy method is best for your installation.....	9
Configuration Summary.....	12
Step 1: Create runtime libraries.....	12
Step 2: Create user parameter library.....	13
Step 3: Security Considerations.....	13
Parameter Library Security	13
Restore security.....	14
Step 4: Define access to the update dynamic LPA facility.....	14
Step 5: APF authorize SCUZLOAD on all systems.....	14
Step 6: Authorize TSO functions.....	14
Step 7: Create staging and restore aliases and define SMS information (Optional).....	15
Step 8: Copy ISPF members to a system CLIST library.....	15
Step 9: Customize ISPF CLIST.....	15
Step 10: Customize parmlib options via ISPF.....	16
Step 11: Customize history include list.....	16
Step 12: Define a z/OS repository file.....	16
Step 13: Copy and customize started task JCL.....	16
Step 14: Define the started task to your security system.....	17
Step 15: (Optional) Customize SSL/TLS security.....	17
Step 16: Check time format.....	20
Step 17: Start the started task.....	20
Chapter 4. Using the Cloud Tape Connector ISPF interface.....	21
Cloud Connector Main Menu.....	21
Specify Parmlib and Member.....	22
Cloud Connector Settings (Parmlib Options).....	23
General Options.....	24
Staging Options.....	26
Cloud Server Options.....	27
Create a cloud server definition.....	30
Specifying a S3-compatible cloud.....	31
Specifying an FTP cloud.....	34
Specifying a Hitachi Content Platform cloud.....	37
Specifying a SoftLayer cloud.....	39
Specifying an S3 cloud.....	42
Specifying a Proxy Server cloud.....	45
Using ICSF with S3 and CLS clouds.....	46

Using Proxy Server to connect to S3 and CLS clouds.....	46
Edit a cloud server definition.....	46
Delete a cloud server definition.....	47
Backup Filter Criteria.....	48
Create cloud filter criteria.....	49
Edit cloud filter criteria.....	49
Delete cloud filter criteria.....	50
Data Restore Options.....	50
Save parameter member.....	52
Viewing cloud server status.....	53
Chapter 5. Primary commands.....	55
Chapter 6. Column display functions.....	59
Accessing the CSETUP Primary Option Menu.....	59
Fixing a column.....	60
Repositioning columns.....	62
Resizing columns.....	63
Sort functionality.....	65
Defining sort columns.....	65
Fast-path SORT command.....	66
Resetting CSET customizations.....	68
Removing CSET customizations.....	68
Column scroll.....	69
Inner column scroll.....	69
Column numbers.....	69
Ruler display.....	69
Expanding columns.....	70
Restrictions.....	70
Chapter 7. Operational Considerations.....	71
Managing the started task.....	71
Displaying Diagnostic information.....	71
Updating parameter settings.....	72
Restoring a data set.....	72
Performing a stand-alone restore of the repository data set.....	73
Considerations for restoring GDG's.....	74
Starting the Cloud Tape Connector started task.....	79
Stopping the Cloud Tape Connector started task.....	79
Managing zIIP processing.....	80
Using the History console command.....	80
Working with cloud data sets.....	80
Specifying selection criteria.....	80
Entering commands.....	83
Restoring data from cloud storage.....	83
Deleting data from cloud storage.....	86
Repository Operations.....	87
Creating Repository Backups.....	87
Restoring a Repository.....	89
Catalog Support.....	90
Chapter 8. Backing up existing data sets.....	97
Specify Parmlib and Member to Include or Exclude.....	97
Working with history data sets.....	97
Using console commands with history data sets.....	100
Chapter 9. Active Tasks.....	101

Chapter 10. Utilities.....	105
Tape Compare Utility.....	105
Dataset Compare Report - Same Results.....	105
Dataset Compare Report - Different.....	106
Dataset Analysis Report.....	106
Sample Job to Include/Exclude Data Sets.....	108
Sample Job for Batch Restore	109
Sample Job for List Datasets	113
Diagnostic log files.....	113
Chapter 11. Cloud Tape Connector Repository.....	115
Repository Records.....	115
Sharing within a Sysplex.....	116
Chapter 12. Introduction to Virtual Tape Emulation.....	117
VTE concepts and benefits.....	117
Chapter 13. How VTE works.....	119
VTE components.....	119
Front-end components.....	119
Back-end components.....	121
Front-end to Back-end connection components.....	121
Tape Processing with VTE.....	122
Chapter 14. Implementation.....	125
Simulated virtual device types.....	125
Virtual Tape candidates.....	126
Allocating virtual devices.....	127
Using Virtual Tapes.....	129
Tapepools and Diskpools.....	131
Chapter 15. Virtual Tape Sharing and VTE complexes.....	135
Running multiple VTE servers in multiple MVS Systems.....	135
Running multiple VTE servers in a single MVS System.....	136
Virtual Tape Sharing.....	137
Virtual Tape sharing examples.....	141
Chapter 16. Dynamic Reallocation Logic.....	145
Dynamic Reallocation Rules.....	145
Dynamic Reallocation Decision Logic.....	145
Dynamic Reallocation Examples.....	147
Chapter 17. Managing the Virtual Tape Library.....	155
Overview.....	155
Virtual Tape Database - VDB.....	155
Virtual Tape Dataset.....	156
Virtual Tape Library.....	157
Virtual Tapes integrity and verification.....	158
Recovering a Virtual Tape.....	159
Controlling the scratch Virtual Tapes inventory.....	161
VDB integrity and verification.....	162
VDB backup and recovery.....	163
Journal File management.....	164
Virtual Tape Library Activity Tracking – VTE User SMF Record.....	164

Chapter 18. Managing the diskpools.....	165
Disk Space Management.....	165
Virtual tape data set de-stage (migration) and recall.....	171
Controlling Virtual Tape data sets de-staging.....	172
Virtual tape data set compression.....	180
Reconfiguring the Diskpool.....	183
Chapter 19. Integration with Tape Management Systems.....	185
Overview.....	185
Identifying the site's Tape Management System.....	185
Tape Management Systems common definitions.....	185
Chapter 20. Running VTE in JES3 environments.....	187
VTE in JES3 environments.....	187
Chapter 21. Handling exceptional events.....	189
VTE equipment check simulation.....	189
Chapter 22. e-Vault - VTE electronic vaulting.....	191
Traditional vault concept.....	191
VTE e-Vault concept.....	191
e-Vault types.....	191
Additional e-Vault features.....	195
e-Vault process flow.....	195
Type3 e-Vault processing.....	196
Type4 e-Vault processing.....	197
Type5 e-Vault processing.....	199
Recovery considerations for type3 e-Vault.....	200
Recovery considerations for type4 e-Vault.....	201
Recovery considerations for type5 e-Vault.....	202
Implementation samples.....	204
Chapter 23. Virtual Tape Replication.....	207
Chapter 24. zIIP Processing.....	209
Chapter 25. Set percent of zIIP-eligible work to be offloaded.....	211
Chapter 26. Electronic Migration (e-Mig).....	213
Highlights of electronic migration (e-Mig).....	213
How electronic migration works.....	214
Chapter 27. VAM Access Method.....	215
VAM Registers.....	216
VAM Syntax and Parameters.....	217
Chapter 28. The VTE Dialog Customization.....	223
The VTE Dialog basic installation.....	223
The VTE Dialog customization steps.....	223
TSO setup.....	223
ISPF setup.....	224
VTE server resources modification authorization.....	225
Enable the VTE Dialog.....	225
HVTDLGxx – VTE Dialog initialization statements.....	226

Chapter 29. VTE Customization.....	229
Introduction.....	229
Customization steps.....	229
Step 1: Hardware configuration.....	229
Step 2: Setup Steps.....	235
Step 3: Security Requirements and Definitions.....	237
Step 4: System Settings.....	238
Step 5: JES3 Additional Customization Steps.....	242
Step 6: e-Vault Installation.....	246
Step 7: Third-Party Product Prerequisites and Coexistence.....	246
Step 8: Start VTE.....	249
Step 9: Run the IVP jobs.....	249
Step 10: Add Virtual Tapes to the VDB.....	250
Chapter 30. VTE Initialization Statements.....	251
The Initialization Statements Residence.....	251
Initialization Statements Syntax.....	252
HVTMCSxx – Extended MCS Console Support.....	253
MCS.....	253
HVTOPTxx – Basic Options.....	256
IDENT.....	256
JRN.....	256
OPTIONS.....	257
PAT.....	259
PERFORM.....	260
SMF.....	262
SUBSYS.....	262
TCP.....	262
VDB.....	263
VDEVICE.....	264
VDSN.....	267
HVTPOLxx – Tape pools and Diskpools.....	268
DPOOL.....	269
TPOOL.....	271
HVTPRLxx – PAT Rules.....	277
PFILT.....	279
PPTRN.....	279
PRULE.....	280
HVTRULxx – Dynamic Reallocation Rules.....	282
RFILT.....	283
RPTRN.....	283
RULE.....	284
HVTSYSxx – Startup Selection.....	286
SYS.....	287
HVTTRLxx – Tape pool Rules.....	288
HVTVLTxx – e-Vault complex.....	289
VLTOWN.....	290
VLTRMT.....	292
Chapter 31. VTE Operation.....	297
Starting and Stopping VTE.....	297
The VTE Procedure.....	297
Starting VTE.....	299
Stopping VTE.....	300
Operator Commands.....	302
Enable/Disable Selected Features.....	303

Refresh Selected Server Parameters.....	309
Trace VTE Operation.....	311
Display Activity and Resources.....	313
Server Tuning.....	314
Server Management.....	315
Operator Commands Restrictions.....	316
Unsupported System Operator Commands.....	316
Chapter 32. VTE Offline Utilities.....	317
HVTUCTC – CTC and a Virtual Tape data set.....	317
HVTUDST – Set De-Stage Priority.....	317
Input.....	317
Return Codes.....	321
HVTUFTP – FTP a Virtual Tape Data set.....	321
Input.....	321
Return Codes.....	322
HVTUJFR – Journal File Format.....	322
Planning the Journal File Size.....	322
Input.....	322
Return Codes.....	324
HVTUJRC – VDB Recovery.....	324
Input.....	324
Return Codes.....	325
HVTURCV – Virtual Tapes Recovery.....	325
Input.....	325
Return Codes.....	325
HVTUREP – Database Reports.....	326
Input.....	326
Report Output.....	328
Return Codes.....	331
HVTUSCR – Virtual Tapes Status Synchronization.....	332
Input.....	332
Special HVTUSCR Utility Processes.....	340
Return Codes.....	341
HVTUTAP – Adding/Deleting Virtual Tapes.....	341
Input.....	341
Return Codes.....	343
HVTUV2V – Volume to Volume Copy.....	343
Special Considerations.....	344
Input.....	344
Return Codes.....	346
HVTUVBK – VDB Backup.....	346
Input.....	347
Return Codes.....	347
HVTUVSI – VDB Initialization.....	347
Input.....	347
Return Codes.....	348
Chapter 33. Implementing VTE and Tape Management Systems.....	349
Implementing in a CA CA-1 Environment.....	349
Add virtual tapes.....	349
Define Tape pools.....	350
Define Tape pool Rules to CA-1.....	350
Implementing in a CA-Dynam/TLMS Environment.....	351
Add Virtual Tapes.....	351
Define Tape pools.....	352
Define Tape pool Rules to CA-Dynam/TLMS.....	352

Implementing in an IBM DFSMSrmm Environment.....	353
Add Virtual Tapes.....	353
Define tapepools to VTE.....	354
Define tapepool rules to DFSMSrmm.....	355
Implementing in a BMC CONTROL-T Environment.....	355
Add Virtual Tapes.....	356
Define Tapepools.....	356
Define tapepool rules to CONTROL-T.....	357
Implementing in an ASG-Zara Environment.....	357
Add Virtual Tapes.....	358
Define Tapepools.....	358
Define tapepool rules to ASG-Zara.....	359
Chapter 34. VTE Exits.....	361
EXIT001.....	361
EXIT002.....	362
EXIT004.....	365
EXIT006.....	366
Chapter 35. VTE Storage Usage.....	371
VTE Server Main Storage Usage.....	371
Examples.....	372
Additional Main Storage Used by a Job Served by VTE.....	372
Disk Storage Usage.....	373
Installation and Product Libraries.....	373
Operation Files.....	373
Appendix A. Appendix A: List of Console Commands.....	375
Appendix B. Appendix B: Example of AT-TLS parameter setup.....	377
Appendix C. Appendix C: Using the Cloud Connector batch API.....	379
Using the APIB Obtain function.....	381
Using the APIB Free function.....	381
Using the APIB Get Cloud Status function.....	382
Using the APIB Build Cloud List function.....	383
Using the APIB Copy to Cloud function.....	384
Using the APIB Restore from Cloud function.....	386
Using the APIB Delete from Cloud function.....	390
Sample CUZ#APIT Member.....	391
Appendix D. Appendix D: Cloud Tape Connector Messages	401
ISPF Messages.....	401
Started Task Messages for Cloud Tape Connector.....	428
FEC Scroll Messages.....	446
Product Messages.....	451
Appendix E. Appendix E: List of VTE Messages.....	477
Notices.....	573
Trademarks.....	574
Terms and conditions for product documentation.....	574
Privacy policy considerations.....	575
Index.....	577

About this information

This guide provides instructions for configuring IBM Cloud Tape Connector for z/OS, and describes how to use it.

The typical user of this product is a systems programmer or administrator who understands the issues relating to tape, DASD, and cloud storage. This person must be familiar with z/OS and its conventions. The information in this document is designed to help data processing professionals perform these tasks:

- Establish parameters to define locations of cloud servers and tape drives.
- Specify criteria for determining which data sets to save to the cloud, and how long to keep the data before deleting it.
- Establish procedures for daily updates and maintenance.
- Use commands to restore data from cloud storage to tape or DASD.

Where to find support and product information

The IBM Knowledge Center provides current support information and product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

<http://www-01.ibm.com/support/knowledgecenter>.

For product support, refer to the following Web page:

<http://www.ibm.com/support>.

Accessibility features

The accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS® interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*, SC34-4822
 - *z/OS TSO/E Primer*, SA22-7787
 - *z/OS TSO/E User's Guide*, SA22-7794

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 1. Summary of changes

This section summarizes the significant improvements or enhancements for IBM Cloud Tape Connector for z/OS V2.1 and refers you to relevant sections of this book for more information. Minor changes to the text are not listed.

SC27-9403-03, June 2020

The following changes were made to the documentation in this release:

- *Change in terminology from "Cleversafe" to "S3-compatible" cloud*. The term "S3-compatible" is now used, where appropriate, to indicate that any cloud that is considered to fit in the S3-compatible category is acceptable for use with Cloud Tape Connector.
- *Product requirements*. A new section was added to list any software or hardware that is required for use with Cloud Tape Connector. See ["Requirements" on page 7](#) for details.
- *More on CUZJLIST job*. A new section was added to describe the CUZJLIST job in the SAMPLIB. The CUZJLIST job generates a list of the datasets on a specified cloud and indicates whether there is a match or a mismatch between the datasets found in the cloud and on the repository. See ["Sample Job for List Datasets" on page 113](#) for more information.
- *Change to General Options screen*. The "Reserve Repository on ENQ" field has been replaced with the "Share Repository across LPARs" field. Use this field to indicate whether to share a repository across multiple LPAR's within a single sysplex. See ["General Options" on page 24](#) for more information.
- *New messages*. Any new messages are listed in the message appendices and will have a change bar next to the text.

SC27-9403-02

The following changes were made to the documentation in this release:

- *More on defining an OMVS segment*. Information on defining an OMVS segment is introduced initially in ["Cloud Connector Processing" on page 5](#) and is mentioned in several other places in this guide, starting with ["Type5 e-Vault processing" on page 199](#).
- *Changes to the AT-TLS feature*. Information has been added or changed in the AT-TLS configuration information. See ["Step 15: \(Optional\) Customize SSL/TLS security" on page 17](#) for the details.
- *Change in terminology from "VTFM" to "VTE"*. Due to a change in the feature's name, Virtual Tape Facility Manager (VTFM) has now been changed to Virtual Tape Emulator (VTE). This text has been updated throughout this guide with a few exceptions based on context.
- *New "Time-out" parameter*. A new parameter has been added to all of the cloud server descriptions. This new parameter sets the maximum amount of time (in seconds) the sockets are allowed to be in control during read (BPX1RED) calls. See ["Cloud Server Options" on page 27](#) for more information.
- *New "Proxy Server" feature*. The new "Proxy Server" functionality enables you to use a Proxy Server to connect Cloud Tape Connector with offsite Cloud Servers, without breaking your organization's network rules. See ["Specifying a Proxy Server cloud" on page 45](#) for more information.
- *New messages*. Any new messages are listed in one of the message appendices and will have a change bar next to the text.

SC27-9403-01

The following changes were made to the documentation in this release:

- *More on the VTE "e-vault" feature*. Information has been added about Type5 electronic vaults (e-vaults). See [Chapter 22, "e-Vault - VTE electronic vaulting," on page 191](#) for more information.
- *New fields for optional parameters*. When creating cloud definitions, you have several optional parameters to choose from for each type of cloud. Previously you had to type those optional

parameters. Now each parameter is shown as a field with possible values listed for easier selection. In addition, a few new parameters have been added:

- Server-side Encryption (AES256)
- Number of Threads
- Virtual Host URL Style
- Connection Retry Time Sec

For more information, see [“Cloud Server Options”](#) on page 27 and locate the section describing the type of cloud used at your site.

- *New field for starting backup of history files.* When you make changes to the Include/Exclude list for history files, you can now run a backup of those changes immediately without having to wait until the next scheduled backup by simply specifying "Y" in the "Start History Processing in Started Task" field. See [“Working with history data sets”](#) on page 97 for more information.
- *New field for refreshing the parameters.* When you make changes to the parameters, you can now process those changes immediately without having to wait until the Cloud Tape Connector started task is restarted or the parameter member is refreshed. You can choose to have the parameters refreshed in the active started task by simply specifying "Y" in the "Refresh Parms in Started Task" field. See [“Save parameter member”](#) on page 52 for more information.
- *New console commands.* A few new console commands have been added to enable you to start repository backups, process history files, and restart orphaned staged files at any time, without needing to wait until the next defined interval (amount of time). Additional console commands have existed, but not been documented. All of the console commands are available in [Appendix A, “Appendix A: List of Console Commands,”](#) on page 375.
- *New VTE Electronic Migration (e-Mig) feature.* During electronic migration, VTE migrates selected virtual tape datasets off the managed diskpools, for example, off the MVS platform to any FTP server with assigned disk storage. See [Chapter 26, “Electronic Migration \(e-Mig\),”](#) on page 213 for more information.
- *New VTE Replication feature.* With the replication feature, VTE copies virtual tape datasets to remote locations, maintaining two copies of the same virtual tape dataset. See [Chapter 23, “Virtual Tape Replication,”](#) on page 207 for more information.
- *New VAM Access Method.* With this new access method, users are able to access virtual tapes without the VTE server services. By invoking the necessary VAM services, the invoker can read, write and create new virtual tapes which can be accessed later by applications as regular tapes using the VTE server. Virtual tapes created by applications using the VTE server can be accessed using the VAM services. See [Chapter 27, “VAM Access Method,”](#) on page 215 for more information.
- *More on zIIP processing.* Additional information is provided for zIIP processing, which allows most of the VTE CPU intensive work to be zIIP-eligible, that is, eligible to be dispatched on (or offloaded to) zIIP processor(s). This intensive CPU work includes general data moves, all types of compression, and other internal tasks (like translation). See [Chapter 24, “zIIP Processing,”](#) on page 209 for more information.
- *More on VDB-In-Storage.* When virtual tapes are added, deleted, scratched or a tapepool threshold interval has expired, the VDB-In-Storage is updated. To accelerate the scratch tape selection, the VDB is loaded in memory, and the search process is performed on the VDB-In-Storage. See [“Virtual Tape Database - VDB”](#) on page 155 for more information.
- *New messages.* A few new messages have been added. See the [Appendix D, “Appendix D: Cloud Tape Connector Messages ,”](#) on page 401 appendix for details on any message.

SC27-9403-00

The following changes were made to the documentation in this release:

- *New VTE feature.* The IBM Virtual Tape Emulation (VTE) product is now included with IBM Cloud Tape Connector for z/OS to enable virtual tapes to be saved to and retrieved from the Cloud. The use of VTE is not required, however, it is a new cloud feature that you may find very helpful. The documentation for installing, customizing, and using VTE is included in this guide. To learn about VTE, refer to [Chapter 12,](#)

[“Introduction to Virtual Tape Emulation,”](#) on page 117. For information on customizing the VTE Dialog, refer to [Chapter 28, “The VTE Dialog Customization,”](#) on page 223, and for the steps for customizing the VTE feature, refer to [Chapter 29, “VTE Customization,”](#) on page 229.

- *Life Cycle Encryption Control.* A new optional parameter, ENCRYPT, allows you to specify that you want the data encrypted on z/OS before moving the data to a cloud. The ENCRYPT parameter is valid for all cloud types. For more information, see [“Cloud Server Options”](#) on page 27.
- *More on staging alias.* Additional information and messages have been provided to help you understand the importance of a unique staging data set alias. Refer to [“Staging Options”](#) on page 26, and to [“Method 2: Staging data to DASD”](#) on page 7, and finally to [“Step 7: Create staging and restore aliases and define SMS information \(Optional\)”](#) on page 15.

Chapter 2. Introduction to Cloud Tape Connector

With the virtually unlimited storage available on private and public clouds, copying data to a cloud is an easy solution for companies who need a lot of space to store valuable data. By having copies in different storage locations, tape/disk and cloud, you can easily retrieve data that is lost, damaged, or unavailable. Conversely, if you need to restore data from the cloud, but the cloud is inaccessible due to connectivity issues, you can easily retrieve the data from backup tapes or DASD. With different storage options, you no longer have to worry about data getting lost or destroyed, and you have multiple methods for backing up and retrieving your data.

IBM Cloud Tape Connector for z/OS is designed to quickly and efficiently allow you to write sequential files concurrently to tape/disk and to a cloud repository, without having to run additional backup utilities. This product can be used in any supported z/OS environment. When you have a copy of your data saved to the cloud, you can recover data at a disaster recovery site without having the tape backups physically present. You can also restore data from the cloud if a tape has expired, has errors, or has been damaged in some way.

IBM Cloud Tape Connector for z/OS addresses several problems that are common to disaster recovery processes: delays in retrieving tapes, retrieving data from an expired tape, and retrieving data from a problem tape. Each of these problems is solved by saving the data on a cloud:

- You can recover your cloud data at a disaster recovery site without having to wait for tapes to be delivered or when tapes are otherwise not available. Cloud Tape Connector will connect to the cloud and retrieve all backups.
- You can recover your cloud data from a backup if a tape has expired and has been released back to the scratch pool.
- You can recover your cloud data if you receive an I/O error on the physical tape.

By specifying cloud storage, the data is copied from tape or disk to a cloud repository, providing two backup resources for your data and giving you peace of mind that your data is secure and accessible whenever you need it. You can specify several cloud locations and criteria to filter which data is selected for backup processing, so you only save the data you really need.

Cloud Connector Processing

Cloud Tape Connector (CUZ) backs up any sequential data from tape or disk to the cloud and also restores it to DASD or tape. You define the cloud servers, staging options, backup filter criteria, data restore settings, and other general options in a parameter member. Cloud Tape Connector uses the options in the parameter member to connect to the cloud servers that you specified. You can set up staging options so that the data from tapes is copied ("staged") to DASD before going to the cloud. This ensures that a copy of the data exists in case there is a connection issue with the cloud.

The data sets selected by your backup filter criteria will be saved to the cloud. You can set a retention period to determine how long the data sets will be stored in the cloud. You can display a list of the data sets that have been written to the cloud and can filter by data set name or cloud server.

When you restore a data set from the cloud, it will be written to either DASD or tape, based on the restore options you specify. Data sets with block sizes over 32K must be restored to tape.

There are two methods for backing up data to the cloud:

- *Capture data while it is being written to tape or DASD and copy to the cloud.* This is done through filter criteria defined in the parameter member, CUZ#PARM. You can filter by data set name, esoteric unit, or SMS storage class.
- *Create a list of existing data sets (history data sets) to copy directly to the cloud.* This method allows you to locate already existing sequential files and copy them directly to the cloud. The pre-existing data sets are identified by name in parameter member, CUZ#INEX.

Cloud Tape Connector provides two methods, described below, for capturing data:

- *Staging data to DASD.* This option will quickly save the tape data to a DASD data set called a *staging file*. After the batch job has completed, the contents of the staging file are written to the cloud by the Cloud Tape Connector Started Task. This method is recommended for two reasons:
 - Writing data directly to the cloud requires an uninterrupted transmission via a stable network connection to the cloud. If there is any interruption in that transmission, the data will not be written to the cloud.
 - If the data is staged to DASD, and the network connection is interrupted, the process to copy that data to the cloud will continue to restart until the data has been completely written to the cloud.
- *Writing data directly to the cloud.* This option will capture a copy of the data you are writing and send it to the cloud concurrently while the batch job executes. This option works best when transmission is uninterrupted and a stable network connection to the cloud is available. Writing data directly to the cloud could, in some cases, increase the elapsed times of your batch jobs if there is network delay.

If this method will be used, define a valid OMVS segment for the userid assigned to the batch jobs.

We recommend using the staging method due to the ability to restart the copy process. If the staging method is used, the data will be written to the cloud even if the copy process must be restarted to accomplish this task. If the "Direct Cloud Write" method is used, any interruptions in transmission will result in failure to write the data to the cloud.

Cloud Tape Connector has a started task with the default name of CUZCLOUD, which coordinates the movement of files to the cloud. Parameter member CUZ#PARM enables you to define the files you want to copy to the cloud based on data set name, esoteric unit, or SMS storage class. Another parameter member, CUZ#INEX, enables you to copy pre-existing files ("history files") to the cloud based on data set name.

The CUZCLOUD started task accesses one or more parameter data sets that contain the CUZ#PARM member (for the filter criteria and cloud definitions), and the CUZ#INEX member (for the data set names or masks for pre-existing files). The CUZCLOUD started task must be active in order for data sets to be written to or restored from the cloud.

Chapter 3. Configuring Cloud Tape Connector

After you install Cloud Tape Connector using the installation instructions provided in the *IBM Cloud Tape Connector for z/OS Program Directory* that is included with the product, you must configure Cloud Tape Connector for your environment.

Requirements

This section lists any hardware or software required for using Cloud Tape Connector.

Hardware requirements

Cloud Tape Connector runs on any hardware that supports the required version of z/OS.

Software requirements

Cloud Tape Connector can be used on any supported release of z/OS.

Choosing Cloud Connector configuration options

In order to determine the best configuration options for Cloud Tape Connector in your environment, three possible methods for moving data to a cloud server are described below. Review this information first, then configure the product using one of the methods described.

For a list of the steps to configure Cloud Tape Connector, refer to [“Configuration Summary” on page 12](#). An optional process to perform comparison testing is also provided by following the directions outlined in [“Comparing the options” on page 9](#).

Method 1: Write directly to cloud

This method will intercept the data while it is being written to tape or DASD and write it to a cloud server within the same job step.

To write data directly to a cloud server, filter criteria must be specified to inform Cloud Tape Connector which cloud to use to back up this data. From the Main Menu, a filter can be defined through Option 1 (Cloud Connector Settings – Parmlib Options), then by choosing Option 4 (Backup Filter Criteria). Filter criteria can be defined based on a data set name, esoteric unit or SMS storage class. You must define a valid OMVS segment for the user ID assigned to the batch jobs.

This method may cause your production batch process to be elongated while waiting for cloud writes to occur, depending on your cloud transmission speeds. Clouds defined within your data center (private clouds) will generally be very fast, and external clouds (public clouds requiring transmissions outside your data center) may be slow.

The one drawback to this method is that if the cloud server transmission is interrupted in any way, Cloud Tape Connector has no way of restarting the "cloud write" function. If Cloud Tape Connector detects a transmission error, one of these actions will occur:

- If the "Abend on Errors" option is set to YES, the job will abend.
- If the "Abend on Errors" option is set to NO, the job will continue with no data written to the cloud.

Method 2: Staging data to DASD

The staging method will intercept the data from tapes while it is being copied and save it to a DASD file very quickly. Once the data has been written to DASD, a copy process will be scheduled under the Cloud Tape Connector started task to copy the staging data set to the cloud.

This method is recommended because if the transmission to the cloud server is interrupted, the tape data has already been saved to DASD, and the "cloud copy" function will retry until the data set has been successfully written to the cloud. To use this method, the following items are required:

- Filter criteria must be specified to inform Cloud Tape Connector which cloud to use to back up this data. From the Main Menu, a filter can be defined through Option 1 (Cloud Connector Settings – Parmlib Options), then by choosing Option 4 (Backup Filter Criteria).
- A unique high-level alias (also known as the "staging" alias) must be created using the DEFINE ALIAS command in IDCAMS, and it will be used as the staging alias in Cloud Tape Connector. The staging alias MUST be a unique qualifier to be used only by Cloud Tape Connector. DO NOT use an alias being used by other users or products.

Note: The stage alias is required even if staging has been turned off. This alias is also used for creating backups of the repository and copying them to the cloud.

There are two methods for defining an alias: IDCAMS and the Main Menu.

- An example of how to define an alias using IDCAMS is shown below:

```
DEFINE ALIAS
(NAME(CUZSTAGE) -
RELATE(MY.USER.CATALOG)) -
CATALOG(MY.MASTER_CATALOG)
```

- From the Main Menu, a staging alias can be specified through Option 1 (Cloud Connector Settings – Parmlib Options), then by choosing Option 2 (Staging Options).
- Sufficient DASD must be available to hold the staged copy of the tape data set until the data set has been successfully written to the cloud. This DASD must be able to contain all of the data that is being asynchronously written to the cloud. A staging file is created when the tape data set is opened, and will be deleted after the data set has been written to the cloud.

Note: If the data is on DASD only and not on a tape, the data will be copied directly to the cloud and no staging file will be used.

Cloud Tape Connector "retry" logic occurs in two ways. The first is internal to the cloud copy function running under the Cloud Connector started task. On the Staging Options screen, which you can access from the Main Menu by choosing Option 1 (Cloud Connector Settings (Parmlib Options), then by choosing Option 2 (Staging Options), you may set values for the Error Retry Count parameter and the Error Retry Interval Seconds parameter.

- The Error Retry Count parameter determines the number of times Cloud Tape Connector will attempt to copy the data set from the staging file to the cloud.
- The Error Retry Interval Seconds parameter specifies the amount of time (in seconds) that Cloud Tape Connector will wait between each attempt to copy data.

If the retry count has been exhausted, the staging file will remain on DASD until the next Repository backup interval is triggered. This interval is based on the amount of time specified for the "Auto Bkup Repository Min" parameter, which you can access from the Main Menu by choosing Option 1 (Cloud Connector Settings - Parmlib Options), then by choosing Option 1 (General Options). In addition to backing up the Cloud Tape Connector Repository, the Repository backup function restarts any cloud copy functions that have previously failed.

Method 3: History Processing

This method allows you to copy existing sequential data sets (referred to as "history data sets") to the cloud. History processing supports sequential data sets on both DASD and tape.

In order to copy existing sequential data sets to the cloud, you must create a History Include/Exclude list. This can be accomplished from the Main Menu by choosing Option 4 (Backup History Datasets).

History processing is the only method available to copy existing data sets to the cloud. Methods 1 and 2 above capture the data as it is being written. If the data set has already been created and you want the data copied to the cloud, it must be specified in the History Include/Exclude list.

The History Include/Exclude list is processed when the auto-backup repository interval expires. This interval is based on the amount of time specified for the "Auto Bkup Repository Min" parameter, which you can access from the Main Menu by choosing Option 1 (Cloud Connector Settings - Parmlib Options),

then choosing Option 1 (General Options). The History Include/Exclude list is also processed 15 seconds after the Cloud Tape Connector started task has started .

The History Include/Exclude list is processed to schedule data sets to be copied to the cloud. A data set will be scheduled only if the data set does not already exist in the cloud. Since the History Include/Exclude list supports the use of "wildcard" characters when specifying data sets, it can be used to pick up newly created data sets, without recopying previously copied data sets. If the History Include/Exclude list is replaced, the updates will take effect during the next interval.

If you want to copy *new* backups to the cloud, without staging the data or writing directly to the cloud, you can use this history processing method. The only drawback is that the backup tape will be mounted twice: once to create the tape backup and the second time to read the tape to copy the data set to the cloud. This method is useful if you have a very short production batch window or you do not have the DASD storage available that is required to stage the data.

Another consideration to keep in mind is that history processing supports only one generation of the data set to be copied to the cloud. Filtering (Methods 1 and 2) allows up to 10 generations.

Determining which cloud copy method is best for your installation

Several parameters must be defined when using any of the cloud copy methods. Testing each method is also helpful in determining which cloud copy method is best for your installation.

Before testing the three methods to write to the cloud, the Cloud Tape Connector installation must be complete and the parameter library must be in place before starting the Cloud Tape Connector started task. It is recommended to use the Cloud Connector ISPF (Main Menu), specifically Option 1 and sub-options 1 through 5, to build and configure the parameter library.

Note: Manually changing the parameter members via ISPF Edit is not recommended because the Cloud Connector ISPF processing edits and validates every field.

The parameter library contains the general options (abend on errors, debugging info, auto-backup minutes, and cell pool size), staging options, cloud server definitions, filter criteria, and restore options. The Cloud Tape Connector started task requires that the parameters are error free before it will start. It also requires that at least one cloud server is properly defined and available for use.

Both the direct cloud write (Method 1) and the staging (Method 2) methods use 64-bit memory cell pool services to transfer data from the channel processor to the cloud writer or staging module. Certain conditions may occur that can cause auxiliary storage in the system to be exhausted. Parmlib general options (Cloud Connector ISPF Option 1.1) allow you to set the cell pool size and the number of memory cells that can be obtained for each data set being captured. If the memory cells are full, processing is suspended to prevent auxiliary storage shortages. Adjust the number of memory cells as needed.

Running DFSMSdss full volume backups with a block size greater than 32,768 may cause auxiliary storage shortages. We recommend re-blocking these data sets in the batch job to a value less than 32K when the staging method is used.

Comparing the options

In the following examples, the following assumptions are made:

- The Cloud Tape Connector started task name is CUZCLOUD.
- The parameter member name is CUZ#PARM.
- The History Include/Exclude member name is CUZ#INEX.

Run the same type of jobs that will be used in the production environment. Run as many jobs as you want to simulate a real production environment. Also run the same jobs for both direct cloud write and staging cloud copy so that you can compare the different methods.

Ensure that the parameter options have valid cloud server definitions and filter criteria set up so that Cloud Tape Connector knows to process the data sets used in these test jobs.

In order to have benchmark timings for comparison with the results of the various cloud options, it is recommended that you first run your test jobs without an active Cloud Tape Connector started task.

Method 1: Testing Direct Cloud Write

In order to enable direct cloud write, staging must be turned off. From the Main Menu, choose Option 1 (Parmlib Options) and then Option 2 (Staging Options) and set the "Stage Data on DASD" parameter to No. Save the parameters (Option 1.6) or press PF3 to exit the parameter options screens. You will be prompted to save the parameters if any changes were made.

1. Start the Cloud Tape Connector started task. If the Cloud Tape Connector started task is not running, start it using the following operator command:

```
S CUZCLOUD
```

If the Cloud Tape Connector started task is already active, refresh the active parameters by issuing the following operator command:

```
F CUZCLOUD,REFRESH MBR=CUZ#PARM
```

2. Verify that there were no parameter errors, and that the cloud server has made a successful connection (check the CUZOUT DD output in the Cloud Tape Connector started task job log).
3. Run your test job(s). After these jobs end, the data has been written to the cloud. You must define a valid OMVS segment for the userid assigned to the batch jobs.
4. Verify that the data sets are on the cloud by using the Cloud Tape Connector ISPF (Main Menu) Option 3. Scroll right to see the cloud data set name, which uses a naming convention of:

```
STAGEALIAS.JOBNAME.JOBID.DDNAME.RANDOM#
```

The Cloud Tape Connector started task must be active in order to view the data sets that have been backed up on the cloud.

Method 2: Staging

This staging method is used for data on tapes. In order to enable staging, the staging parameter option must be active. On the Cloud Tape Connector ISPF (Main Menu), choose Option 1 (Parmlib Options) then Option 2 (Staging Options), and set the "Stage Data on DASD" parameter to Yes.

Set the values for the allocation units large enough to save your largest file. Some staging data sets may appear larger than a real backup data set would be on DASD. Some backup utilities, such as DFSMSdss, write 256K blocks to tape. In those situations, Cloud Tape Connector divides the 256K block into 32K blocks in the staging file.

Save the parameters (Option 1.6) or press PF3 to exit the parameter options screens. You will be prompted to save the parameters if any changes were made.

1. Start the Cloud Tape Connector started task. If the started task is not running, start it using the following operator command:

```
S CUZCLOUD
```

If the Cloud Tape Connector started task is already active, refresh the active parameters by issuing the following operator command:

```
F CUZCLOUD,REFRESH MBR=CUZ#PARM
```

2. Verify that there were no parameter errors, and that the cloud server has made a successful connection (check the CUZOUT DD output in the Cloud Tape Connector started task job log).
3. Run your test job(s).
4. After these jobs end, the data sets have been staged to DASD. The Cloud Tape Connector started task will immediately schedule them to be copied to the cloud. Data sets to be staged use a naming convention of:

```
STAGEALIAS.JOBNAME.JOBID.DDNAME.RANDOM#
```

5. When the staging data set has been deleted from DASD, the data has been successfully written to the cloud. Use the Cloud Tape Connector ISPF (Main Menu) Option 3 to view the names of the data sets which have been backed up. Scroll right one screen to see the corresponding data set name used on the cloud. The Cloud Tape Connector started task must be active to view the data sets that have been backed up on the cloud.

Method 3: History processing

To test History processing, you will back up the same data sets that were copied to the cloud in the other two tests. History processing will only backup data sets to the cloud that currently do not exist in the cloud.

1. If you are going to use the same data set names as in the earlier tests, you will need to delete the backups from the cloud that were created in previous tests.

- a. On the Cloud Connector Main Menu, choose Option 3 (Cloud Datasets).
- b. Delete the backups by placing a "D" line command next to each one.

You will need to do this for all of the data sets that were created in the previous two tests.

2. After the cloud data sets have been deleted, you will need to create a History Include/Exclude list.

- a. Exit Option 3 (Cloud Datasets).
- b. On the Cloud Connector Main Menu, choose Option 4 (Backup History Datasets).
- c. On this screen, enter the data set masks for the data sets to be included in the copy to the cloud, and any to be excluded from the cloud.

The data set masks entered on this screen support "*" and "%" wildcard masking for both included and excluded data sets. As an example, assume that two DFSMSdss full volume backup copies are created in one job, where one copy is kept locally and the other is sent offsite. If these data sets are named MY.WEEKLY.VOLSER.BACKUP.LOC(+1) and MY.WEEKLY.VOLSER.BACKUP.OFF(+1), and you wanted to send only the local copy to the cloud, you would enter the mask of MY.WEEKLY.* on the include, and MY.WEEKLY.*.OFF on the exclude.

3. Each line to be included must point to a cloud server definition defined in the parameter member (Option 1.3). A retention period is also required for all included lines that identifies how long the data set should reside on the cloud before expiration. This retention period has no relationship to the retention period of the actual data set or the expiration date in the tape management system.
4. Save the changes to the History Include/Exclude list by pressing PF3. This will prompt for the location to save the list. The data set where this list is saved must be the same data set as the one specified on the CUZ#INEX DD in the Cloud Tape Connector started task. The member name must be the same as the member name specified in the "Copy Past History Parm Mbr" parameter (Option 1.1).
5. After saving this member, stop and restart the Cloud Tape Connector started task.
 - a. To stop the Cloud Cloud Tape Connector started task, use the "P CUZCLOUD" operator command.
 - b. To start the Cloud Tape Connector started task, use the "S CUZCLOUD" operator command.

Note: Refreshing the Cloud Cloud Tape Connector parameters will not initiate the History Processor task. History processing is triggered only on time intervals or through the use of the CUZJINCL job. If changes are made to the existing History Include/Exclude list while the Cloud Tape Connector started task is active and it is not restarted, the new list will be picked up when the next Auto Backup Repository interval is triggered.

The first Auto Backup Repository cycle is triggered 15 seconds after the Cloud Tape Connector started task has been started. At that time, messages will be written to indicate that data sets are being copied to the cloud. The information from the History Include/Exclude list can be seen in the output associated with the CUZOUT DD in the Cloud Tape Connector started task job log. All data sets scheduled to be copied to the cloud will be listed.

The number of data sets being copied to the cloud asynchronously is controlled by the parameter general option "Max Backup History Tasks" (Option 1.1). This parameter is used to govern the maximum number of tape drives that Cloud Tape Connector can utilize at one time.

If you wish to copy an existing sequential data set to the cloud without making changes to the History Include/Exclude list, you can copy and tailor the sample JCL member, CUZJINCL, provided in the SCUZSAMP data set. This job will initiate a process to immediately copy the identified data sets to the cloud without the need to wait for the next Auto Backup Repository cycle. This job enables you to process data sets one time only, while the History Include/Exclude used by the Cloud Tape Connector started task is processed repeatedly on the specified interval to locate and copy data sets to the cloud which have been created since the last execution.

Compare your results

After running the benchmark test without the use of Cloud Tape Connector, and the tests using the three different methods for writing to the cloud, you can compare the elapsed times of all of the jobs to determine which will be optimal for your environment.

Configuration Summary

This table summarizes the steps that are required to configure Cloud Tape Connector. Some of these steps are optional and may not apply to your environment.

Table 1. Summary of Configuration Steps	
Step	Description
1	“Step 1: Create runtime libraries” on page 12
2	“Step 2: Create user parameter library” on page 13
3	“Step 3: Security Considerations” on page 13
4	“Step 4: Define access to the update dynamic LPA facility” on page 14
5	“Step 5: APF authorize SCUZLOAD on all systems” on page 14
6	“Step 6: Authorize TSO functions” on page 14
7	“Step 7: Create staging and restore aliases and define SMS information (Optional)” on page 15
8	“Step 8: Copy ISPF members to a system CLIST library” on page 15
9	“Step 9: Customize ISPF CLIST” on page 15
10	“Step 11: Customize history include list” on page 16
11	“Step 10: Customize parmlib options via ISPF” on page 16
12	“Step 12: Define a z/OS repository file” on page 16
13	“Step 13: Copy and customize started task JCL” on page 16
14	“Step 14: Define the started task to your security system” on page 17
15	“Step 15: (Optional) Customize SSL/TLS security” on page 17
16	“Step 17: Start the started task” on page 20

Step 1: Create runtime libraries

JCL member CUZJRUNL will create a set of runtime libraries using the SMP/E target libraries as the source. CUZJRUNL is located in hlq.SCUZSAMP. To edit the JCL and submit the job, perform the following steps:

1. Provide a valid job card statement.
2. Change the PPFX=PPFX on the //CUZJRUNL statement to PPFX=desired.runtime.hlq.
3. Change the TGTHLQ=TGTHLQ on the //CUZJRUNL statement to TGTHLQ=smp.target.hlq.
4. Submit the job. Ensure that it ends with a completion code of 0 before proceeding to the next task.

Step 2: Create user parameter library

Two parameter members need to be copied from SCUZSAMP to the stand-alone parameter library/libraries: CUZ#PARM and CUZ#INEX.

- Member CUZ#PARM contains the parameters for configuring the product. This member contains general options, staging options, restore options, filtering criteria, and all of the cloud server definitions, including cloud sign-on information. This sign-on information will include the server locations, user IDs and passwords, and must be secured. Establishing security will be performed in [“Step 3: Security Considerations”](#) on page 13.
- Member CUZ#INEX contains filtering criteria for copying pre-existing sequential data sets, also known as the History Include/Exclude list, to the cloud. Member CUZ#INEX may not need the same level of security as member CUZ#PARM.

The CUZ#PARM parameter member contains sensitive information on accessing cloud servers, and access to the library in which this member resides needs to be limited to only those administrators who will be controlling cloud access.

The CUZ#INEX parameter member contains the names of pre-existing data sets to be copied to the cloud, and may need to be updated by a broader set of administrators.

There are two options for copying these members. One option is to copy each member to its own data set and the other option is to copy both members into the same data set. Using two different parameter libraries is recommended.

- *Different data sets.* If you choose to copy each member to its own data set, tailor and submit the JCL in member CUZJINEX from the SCUZSAMP library. This member will perform the following steps:
 1. Create the SCUZPARM data set.
 2. Copy member CUZ#PARM into the SCUZPARM data set.
 3. Create the SCUZINEX data set.
 4. Copy member CUZ#INEX into the SCUZINEX data set.
- *Same data set.* If you choose to copy both members into a single data set, tailor and submit the JCL in member CUZJSAMP from the SCUZSAMP library. This JCL member will perform the following steps:
 1. Create the SCUZPARM data set.
 2. Copy members CUZ#PARM and CUZ#INEX into the SCUZPARM data set.

Choose only one of the jobs, CUZJINEX or CUZJSAMP, to execute. It is recommended that these members should not be edited via ISPF edit, but rather be configured via the Cloud Connector ISPF interface. The steps for configuring these members are described later in this chapter.

Step 3: Security Considerations

When configuring Cloud Tape Connector, security issues must be thought through before any commands are issued. Review the topics below for security considerations.

Parameter Library Security

Because server locations, user IDs and passwords are required in the CUZ#PARM member, it is strongly recommended that the data set containing CUZ#PARM created in [“Step 2: Create user parameter library”](#) on page 13 be protected through SAF. The necessary authorizations are described below:

- The Universal Access for the SCUZPARM data set should be NONE.
 - The appropriate cloud administrators who will be defining the cloud servers and filter criteria must have UPDATE access.
 - The Cloud Tape Connector started task CUZCLOUD must have READ access.
- If the CUZ#INEX member was copied to the SCUZINEX data set, the Universal Access should be READ.

- Users who are allowed to change the Include/Exclude History data set masking must have UPDATE access.
- The Cloud Tape Connector started task must have READ access.

Restore security

Restore requests execute as subtasks of the Cloud Connector address space CUZCLOUD. As such, this address space needs the authority necessary to allocate and update any data set name that can potentially describe the target of a Restore request. Define CUZCLOUD to your SAF and give it the authority to perform Restore processing.

Step 4: Define access to the update dynamic LPA facility

Cloud Tape Connector uses the dynamic LPA facility, and the CUZ#MAIN program must be given SAF authorization to the CSVDYLPA FACILITY class.

To establish a RACF profile, allowing you to add and delete a module, issue an RDEFINE command, such as the following:

```
RDEFINE FACILITY CSVDYLPA.ADD.modname UACC(NONE)
RDEFINE FACILITY CSVDYLPA.DELETE.modname UACC(NONE)
```

Another option is to create a generic profile:

```
RDEFINE FACILITY CSVDYLPA.ADD.** UACC(NONE)
RDEFINE FACILITY CSVDYLPA.DELETE.** UACC(NONE)
```

To permit a user (in this case the CUZCLOUD started task) to add and delete module CUZ#MAIN to the LPA, issue the following command:

```
PERMIT CSVDYLPA.ADD.CUZ#MAIN CLASS(FACILITY) ID(CUZCLOUD) ACCESS(UPDATE)
PERMIT CSVDYLPA.DELETE.CUZ#MAIN CLASS(FACILITY) ID(CUZCLOUD) ACCESS(UPDATE)
```

Optionally, a RACF-defined group profile that is associated with the CUZCLOUD started task may be given access instead of the started task user ID.

Step 5: APF authorize SCUZLOAD on all systems

The SCUZLOAD data set needs to be added to the APF Authorization (Auth) list that is executed at the time of IPL. You can also enter the following console command:

```
SETPROG APF,ADD,DSN=MY.SCUZLOAD,SMS
```

If you need assistance, consult your systems administrator to have the SCUZLOAD library for Cloud Tape Connector added into the APF list and ensure appropriate access controls have been established.

Step 6: Authorize TSO functions

Consult your systems administrator to have the following changes made to the noted sections for the list of authorized commands and programs in the system PARMLIB member, IKJTSONn.

```
AUTHCMD NAMES(          /* Authorized Commands    */ +
  CUZ$TSOC             /* Cloud Connector      */ +
  )                   /* End of AUTHCMD      */ +

AUTHPGM NAMES(          /* Authorized Programs   */ +
  CUZ$TSOC             /* Cloud Connector      */ +
  )                   /* End of AUTHPGM      */ +
```

After the member IKJTSONn has been modified, activate the changes by using the TSO PARMLIB command.

Step 7: Create staging and restore aliases and define SMS information (Optional)

The following steps are required unless marked as optional. The optional steps are strongly recommended:

1. You must create a new high-level alias (Staging Alias) in order to create repository backups. If you plan to use the recommended method of staging the data from tapes to DASD, this will also require the use of the Staging Alias so that the only data sets being considered as candidates intended for the cloud will be the ones using this high-level name. The default value for the staging alias is CUZSTAGE.

Note: If the data sets associated with this alias will be going to SMS-managed DASD, add this alias to an existing SMS storage class or create a new SMS storage class.

The staging alias **MUST** be a unique qualifier to be used only by Cloud Tape Connector. **DO NOT** use an alias being used by other users or products.

Note: The stage alias is required even if staging has been turned off, so that backups of the repository can be copied to the cloud.

2. (Optional) Cloud Tape Connector is shipped with a tape compare program, which allows you to compare the data contents of the original backup on tape with the data contents of the restored data set on tape. For testing purposes, you may wish to define a unique alias to be used when restoring data sets. You can also optionally override the entire data set name at the time you request it to be restored.
 - Define a high-level alias for restored data sets. The default alias for restoring data sets is CUZRESTR. The alias used for restoring data sets cannot be the same as the staging alias (default CUZSTAGE).
 - If the data sets associated with this alias will be going to SMS-managed DASD then add this alias to an existing SMS storage class or create a new SMS storage class. Adding this alias to the same storage class as the staging alias is allowed.
3. (Optional) Create an SMS storage group with enough volumes to handle the largest amount of data to be written to the cloud at one time. When the data has been successfully written to the cloud, the staging data set is deleted. If your transmission is interrupted or down at the time of the backups, these staging files will reside on these volumes until the cloud write function has successfully completed.

If you are planning to copy a large amount of data, such as full-volume dumps to the cloud, you will need to have an identical number of DASD volumes available for staging those files.

Step 8: Copy ISPF members to a system CLIST library

Copy members CUZVP11, CUZVP110 and CUZTSOC to a system CLIST library allocated to SYSPROC.

Step 9: Customize ISPF CLIST

In member CUZVP11, change the data set name for the CUZVP11 file to the data set name of the system CLIST library these members were copied to by tailoring and entering the following command:

```
CHANGE '#MY.CLIST.LIB#' 'your.clist.library'
```

Edit member CUZVP110 and update the appropriate PROC statement variables. The steps are listed below.

1. CUZLVL – Change #HIGHLVL# to the high-level target data sets created in [“Step 1: Create runtime libraries”](#) on page 12.
2. CUZLOAD – Change #CUZ.RUNTIME.LOADLIB# to the Cloud Tape Connector load library.
3. PARMLDSN – Change #PARM.LIB.DSN# to the parameter library data set created in [“Step 2: Create user parameter library”](#) on page 13.

4. PARMLMBR – Change #PARMLBR# to the parameter member copied in [“Step 2: Create user parameter library”](#) on page 13. The default parameter is CUZ#PARM.
5. INEXDSN – Change #PARM.LIB.DSN# to the History Include/Exclude Data Set created in [“Step 2: Create user parameter library”](#) on page 13.
6. INEXMBR – Change #INEXMBR# to the History Include/Exclude Member copied in [“Step 2: Create user parameter library”](#) on page 13. The default parameter is CUZ#INEX.

Step 10: Customize parmlib options via ISPF

Once you have customized the CLIST members and placed them into a system CLIST library, you can invoke the Cloud Tape Connector ISPF interface by executing the command %CUZVP11 or defining an option through the ISPF panels.

On the Cloud Tape Connector Main Menu, choose Option 1, Cloud Connector Settings (Parmlib Options), to select the SCUZPARM data set you created in [“Step 2: Create user parameter library”](#) on page 13. You can customize the CUZ#PARM parameter member with your cloud definition information. For complete information on how to customize the parameter member, refer to [“Cloud Connector Settings \(Parmlib Options\)”](#) on page 23.

Step 11: Customize history include list

This customization will allow you to copy pre-existing backups or other sequential data sets to the cloud. These pre-existing data sets can be on DASD or tape.

Currently, the product only supports capturing backups on demand at the time the backup is being written to tape. In order to support copying pre-existing DASD or tape data sets to the cloud, these data sets must be identified in the history include lists, which are in member CUZ#INEX. Edit this member using Option 4, Backup History Datasets, from the Cloud Tape Connector Main Menu. Refer to [Chapter 8, “Backing up existing data sets,”](#) on page 97 for instructions on how to customize these entries.

Step 12: Define a z/OS repository file

The JCL member, CUZJDEFR, in the SCUZSAMP library will allocate the Cloud Tape Connector z/OS repository which will contain all of the information on data in the cloud.

Step 13: Copy and customize started task JCL

Copy member CUZCLOUD from the SCUZSAMP runtime library into a system procedure library, such as SYS1.PROCLIB.

You must customize the CUZCLOUD member to meet your installation requirements. Change #HIGHVL# to the high-level qualifier you used for your runtime libraries, and verify that all data set names are correct.

Verify that the STEPLIB, CUZ#PARM, and CUZ#INEX DD statements contain the correct data set names created in prior steps.

Ensure that the member listed in the statement below is the correct name of the parameter member you plan to use.

```
PARM='MBR=CUZ#PARM'
```

Note: At least one cloud server must be defined for use by Cloud Tape Connector before the CUZCLOUD started task will start. For information on defining a cloud, refer to [“Create a cloud server definition”](#) on page 30.

Step 14: Define the started task to your security system

Consult your systems administrator to define the CUZCLOUD started task to your security system. CUZCLOUD also requires a valid OMVS segment for the userid assigned to the started task. No special privileges are required.

Step 15: (Optional) Customize SSL/TLS security

Cloud Tape Connector uses Application Transparent Transport Layer Security (AT-TLS) to secure transmissions between the mainframe and cloud providers. Configuring and running AT-TLS enables SSL encryption to be performed on the TCP traffic of outbound connections from Cloud Tape Connector. To set up AT-TLS, the steps below will describe the process.

Preparing to use AT-TLS

Note: Other SSL/TLS automatic configuration solutions exist. If you are not using RACF, consult the documentation for your security management software.

The steps for setting up AT-TLS for use with Cloud Tape Connector are listed below, along with the corresponding section to refer to for more detailed information.

1. Configure AT-TLS to ensure a valid certificate is installed for each cloud provider. See [“Step 1: Configuring AT-TLS” on page 17](#).
2. Verify that policy-based networking (PAGENT) is enabled. See [“Step 2: Enabling PAGENT” on page 18](#).
3. Establish rules for determining which traffic will be encrypted. See [“Step 3: Defining AT-TLS rules” on page 19](#).
4. Refresh the PAGENT started task to ensure all changes are included. See [“Step 4: Refreshing PAGENT” on page 20](#).

Step 1: Configuring AT-TLS

If you would like to enable encryption for your cloud transfers, AT-TLS needs to be properly configured. For each cloud provider that you plan to use with Cloud Tape Connector, a valid certificate needs to be installed. Certificates for each cloud provider can be found from their respective SSL certificate provider, acting as a Root Certificate Authority (i.e., Geotrust, Digicert, etc).

1. Use a web browser to retrieve valid certificates using either of these methods:
 - For each cloud provider that you intend to use to store data, retrieve a certificate from a Secure Socket Layer (SSL) certificate provider (Geotrust, Digicert, etc.), acting as a Root Certificate Authority.
 - Alternatively, in the web browser's location field, type the URL for the cloud provider and then use the browser's certificate export feature to export the certificate into a file that can then be uploaded. You must export the Root CA.
2. Upload each certificate to its own variable block flat file on the mainframe. If you are uploading more than one certificate, each certificate must be uploaded to a different file. You are likely to acquire these certificates on your PC as *.cer or *.pem files.
 - If you have a *.cer, you should upload the data to the mainframe in Binary mode.
 - If you have a *.pem, you should upload the data to the mainframe in Text mode.
3. For each certificate, issue a command to define the certificate to your security management software. To add a certificate to RACF, issue the following command:

```
RACDCERT ID(USERID) ADD('CERT.DATASET.NME') CERTAUTH TRUST WITHLABEL('LABELNME')
```

Where:

USERID

Indicates that the new certificate is a user certificate associated with the specified user ID. It is recommended to use the USERID of the Cloud Tape Connector started task.

CERT.DATASET.NAME

Indicates the name of the data set that contains the uploaded certificate.

LABELNAME

Indicates the label name you want to use to help identify the certificate in RACF.

4. Identify the keyring that will house all of the certificates in a single addressable entity. You can either add the cloud certificates to an existing keyring, or create a new keyring. If you choose to create a new keyring, issue the appropriate command for your security management software.

For RACF, use the following command:

```
RACDCERT ID(USERID) ADDRING(RINGNAME)
```

Where:

USERID

Specifies the user ID of the key ring owner. It is recommended to use the USERID of the Cloud Tape Connector started task.

RINGNAME

Indicates the name of the keyring you want to define to RACF. This name can be any name you choose.

5. Connect the certificates to the keyring. For each certificate, issue the appropriate command for your security management software.

For RACF, use the following command:

```
RACDCERT ID(USERID) CONNECT(CERTAUTH LABEL('LABELNAME') RING(RINGNAME)
USAGE(CERTAUTH))
```

Where:

USERID

Specifies the user ID of the key ring owner.

LABELNAME

Indicates the label name that was used to add the certificate in RACF.

RINGNAME

Indicates the name of the existing keyring or the name of the new keyring that you created and added to RACF in Step 4 (above).

6. Finally, issue the appropriate refresh command for your security management software. For RACF, use the following command:

```
SETROPTS RACLIST(DIGTRING) REFRESH
```

Step 2: Enabling PAGENT

AT-TLS requires policy-based networking (PAGENT) to be enabled. If you do not already have this enabled, please see the "Policy-based networking" chapter of the *IBM z/OS Communications Server: IP Configuration Guide* (SC27-3650).

Step 3: Defining AT-TLS rules

AT-TLS uses rules to determine which traffic to encrypt. Encryption of Cloud Tape Connector traffic requires a valid AT-TLS rule. Be aware that most cloud providers use port 80 for unencrypted HTTP traffic and port 443 for encrypted SSL/TLS traffic.

If you do not already have an AT-TLS rules data set in place, refer to the "AT-TLS policy configuration" section of the *IBM z/OS Communications Server: IP Configuration Guide* (SC27-3650).

Define a valid AT-TLS rule for Cloud Tape Connector using the information in Appendix B, "Appendix B: Example of AT-TLS parameter setup," on page 377 as an example and a guideline.

As you define your AT-TLS rule, keep the following general recommendations in mind:

- AT-TLS is invoked only if all conditions of a rule are met. Therefore, it is recommended that you define the AT-TLS rule to be as minimally restrictive as necessary.
- It is recommended that you use the Jobname as the primary method for determining which traffic AT-TLS encrypts. The Jobname in your AT-TLS rule should be the name of the Cloud Tape Connector started task (CUZCLOUD). It is possible to use a wildcard as the Jobname. Another option is to use the Userid instead of Jobname.
- If you are using Backup Filter Criteria and write datasets directly (not using staging) from tape(s) to the cloud server via an encrypted connection, verify that the batch Jobname or USERID is used in the AT-TLS rule. In this case, data will be sent to the cloud server in the same job step that was used to write data to tape. The Jobname can be a wildcard.

If a USERID is needed and it is not part of the current AT-TLS rule, you can create a new AT-TLS rule for the USERID that is submitting the job. In both options, this USERID should have UPDATE authority to IRR.DIGTCERT.LISTRING in the FACILITY class for the keyring to use already created and defined AT-TLS rules

- If you are using a Restore Batch job with the WAIT_FOR_COMPLETION parameter set to YES and an encrypted connection, verify that the batch Jobname or USERID is used in the AT-TLS rule. In this case, data will be restored from the cloud server by this batch job. The Jobname can be a wildcard.

If a USERID is needed and it is not part of the current AT-TLS rule, you can create a new AT-TLS rule for the USERID that is submitting the job. In both options, this USERID should have UPDATE authority to IRR.DIGTCERT.LISTRING in the FACILITY class for the keyring to use already created and defined AT-TLS rules.

- If you are using the VTE e-Vaulting function to transfer the virtual tapes to Cloud servers via an encrypted connection, verify that the VTE started task or USERID is used in the AT-TLS rule. In this case, the virtual tape will be sent to the cloud server by the VTE started task. The Jobname can be a wildcard.

If a USERID is needed and it is not part of the current AT-TLS rule, you can create a new AT-TLS rule for the USERID that is submitting the job. In both options, this USERID should have UPDATE authority to IRR.DIGTCERT.LISTRING in the FACILITY class for the keyring to use already created and defined AT-TLS rules

- It is enough to add only one keyring and specify it in an AT-TLS rule as USERID/CLOUDrng, where USERID is the owner of keyring. All other USERIDs that are going to use this rule should have UPDATE authority to IRR.DIGTCERT.LISTRING in the FACILITY class.
- If you are using a Proxy Server through a secured (https) connection to cloud servers, you must create new AT-TLS rules for all jobs or users you are going to use through the proxy. It is necessary to have the **ApplicationControlled** parameter set to ON in your TTLSConnectionAdvancedParms statement for the port you are using to connect to the Proxy Server.

Note: To connect to an S3 and S3-compatible cloud server through a proxy via secured (https) connection, you must enable the **Use Proxy Server** parameter and use port 443 for the corresponding cloud servers. Be sure that your Proxy Server configuration allows forward https traffic via port 443 for the CONNECT method.

Step 4: Refreshing PAGENT

After the AT-TLS rules are in place, you must refresh the PAGENT started task to pick up any changes. Issue the following modify command:

```
F PAGENT,REFRESH
```

Where:

PAGENT

Indicates the name of the PAGENT started task.

Step 16: Check time format

Be sure that there are no configuration changes for your z/OS UTC time. It should be in GMT format to ensure a connection to remote cloud servers.

Step 17: Start the started task

To run Cloud Tape Connector, you must start the CUZCLOUD started task. You should add the start-up of this address space into your automated IPL procedures.

Chapter 4. Using the Cloud Tape Connector ISPF interface

You can specify parameter values by choosing Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. The values you enter in the fields on the ISPF screens will be saved in the SCUZPARM member, CUZ#PARM. The sample member as shipped contains several parameters that affect how Cloud Tape Connector will work, such as the cloud locations and the criteria for selecting the data that you want to save on the cloud.

Cloud Connector Main Menu

Invoke the Cloud Tape Connector ISPF interface by executing the command %CUZVP11 in ISPF option 6, or by selecting the ISPF panel option you previously defined. After you launch the Cloud Tape Connector ISPF interface, the Main Menu displays. The options listed on the Main Menu are briefly described below.

```
CUZ$MAIN V1R1 ----- IBM Cloud Tape Connector for z/OS -----
Option ==> _____

                                     2018/03/21 12:10:08
                                     User: USERID - CUZ
-----

Cloud Connector Started Task Status: Active

-----

1. Cloud Connector Settings (Parmlib Options)
2. Cloud Servers Status
3. Cloud Datasets
4. Backup History Datasets
5. Active Tasks

X. Exit
```

Figure 1. Main Menu

To choose an option, type that number on the Option line at the top of the screen.

1. **Cloud Connector Settings (Parmlib Options)**, Option 1, allows you to make changes to the CUZ#PARM member by making entries in the fields. Use this option to do the following:
 - Set general options for handling errors and WTO messages, debugging options, and memory cell values.
 - Specify staging options for backing up the data sets to DASD before copying the data sets to the cloud.
 - Define the cloud servers that will be used for backups.
 - Establish filter criteria for cloud backups.
 - Specify values for restoring data from the cloud.
 - Save or update the CUZ#PARM member.
2. **Cloud Servers Status**, Option 2, allows you to view a list of the cloud servers you defined in Option 1 and their connection status.

3. **Cloud Datasets**, Option 3, enables you to view the list of data sets that have been saved to the cloud. You can delete a cloud backup or restore a cloud data set to DASD or tape, depending on the blocksize.
4. **Backup History Datasets**, Option 4, is used to create the list of pre-existing data sets to be copied directly to the cloud. You can include or exclude data sets as needed.
5. **Active Tasks**, Option 5, displays the tasks that are active. You can use selection criteria to narrow the list by job name, data set name, or cloud name. The tasks are highlighted in different colors based on the type of activity, such as Filter Capture, Cloud Copy, Restore, History Queue, and Restore Queue.

Specify Parmlib and Member

When you choose Option 1, **Cloud Connector Settings**, on the Main Menu, a window displays. You must specify the **Parmlib Dataset** and the name of the **Parmlib Member** that you configured earlier.

```
CUZ$MAIN V1R1 ----- IBM Cloud Tape Connector for z/OS -----
Option ==> 1

                                     2016/04/28 17:54:56
                                     User: USERID - CUZ
-----

Cloud Connector Started Task Status: Active

-----

+----- Enter Parmlib Member -----+
| Parmlib Dataset  PROD.CLOUD.SCUZPARM |
| Parmlib Member   CUZ#PARM            |
+-----+

```

Figure 2. Enter Parmlib Member window

To view the product settings, you must specify the **Parmlib Dataset** and the name of the **Parmlib Member**.

Parmlib Dataset

Type the name of the data set that holds the Parmlib Member. The initial contents of the CUZ#PARM member will be copied from the sample provided in SCUZPARM. However, you may have specified a different data set name when you configured this product in [“Step 2: Create user parameter library” on page 13](#).

Parmlib Member

Type the one- to eight-character name for the parameter member. The default member name is CUZ#PARM. This member will be initially populated with parameters from the sample in SCUZSAMP after the customization job from [“Step 2: Create user parameter library” on page 13](#) has been run.

Cloud Connector Settings (Parmlib Options)

After you choose Option 1, **Cloud Connector Settings (Parmlib Options)**, on the Main Menu and specify the **Parmlib Dataset** and **Parmlib Member** to use, the **Parmlib Options Main Menu** displays. Several types of options are available for you to customize Cloud Tape Connector in your environment.

```
CUZ$PRML V1R1 ----- Parmlib Options Main Menu -----
Option  ===>

                                         2020/03/21 18:12:30
                                         User: USERID - CUZ
-----
Parmlib Dataset: PROD.CLOUD.SCUZPARM
Parmlib Member  : CUZ#PARM
-----

1. General Options
2. Staging Options
3. Cloud Servers
4. Backup Filter Criteria
5. Data Restore Options
6. Proxy Server
7. Save / Save As
X. Exit
```

Figure 3. Parmlib Options Main Menu

Type the number of the menu option you want in the Option line at the top of the screen. The menu options are briefly described below.

1 - General Options

Set values for handling errors and WTO messages, debugging options, and memory cell values.

2 - Staging Options

Specify options for creating a copy of the data sets to DASD before creating a backup on the cloud.

3 - Cloud Servers

Define the cloud servers that will be used for backups.

4 - Backup Filter Criteria

Establish filtering criteria for cloud backups. Only the data sets that match the criteria will be saved to the cloud.

5 - Data Restore Options

Specify values for restoring data from the cloud back to z/OS. Data can be restored to tape or to DASD.

6 - Proxy Server

Set values to define a single cloud server to be used as the proxy server. This server can be used to connect Cloud Tape Connector with offsite cloud servers without violating your organization's network rules.

7 - Save/Save As

Save or update the Parmlib Member.

General Options

The values you set under **General Options** cover several different areas including how to handle abends, old backups, WTO messages, and backup generations, and how to set up memory cell pools. The default values are displayed on the screen, but you can change those values to suit your environment.

```
CUZ$PRMG V1R1 ----- Parmlib General Options -----
Option ==>

                                         2019/06/27 12:21:37
                                         User: USERID - CUZ

-----
Parmlib Dataset: PROD.CLOUD.SCUZPARM
Parmlib Member : CUZ#PARM
-----

Abend on Errors . . . . . N          (Yes / No)
User Abend Return Code. . . . 55      (01 to 99)
Copy Past History Parm Mbr. . CUZ#INEX (Parmlib Mbr with INCL/EXCL)
Debug Mode. . . . . N          (None / All / Job)
Debug Jobname . . . . . MYJOB*      (Jobname Like)
Write to Operator Msgs. . . . N      (Yes / No)
Max Cloud Backup Gens . . . . 10      (1 to 10)
Max Backup History Tasks. . . 10      (1 to 99)
Auto Bkup Repository Min. . . 60      (5 to 9999)
Exec History Interval Min . . 60      (5 to 9999)
Memory Cell Pool Size . . . . 55242880 (2560000 to 99999999)
Memory Primary Cells. . . . . 50      (10 to 9999)
Share Repository across LPARs N      (Yes / No)
Write SMF Record Type . . . . 0       (128-255,0 - No SMF Records)
```

Figure 4. Parmlib General Options screen

The General Options are described below.

Abend on Errors

Indicate what to do if an error occurs.

- Y – Force the utility program to end prematurely (abend) if any errors are encountered.
- N – Do not force an abend if an error occurs. Allow the utility program to continue. (Default)

User Abend Return Code

Type a number between 01 and 99 to indicate the code you want to have returned when an error occurs. The default is 08.

Copy Past History Parm Mbr

Specify an eight-character name for the parameter member that will be used by the **Backup History Datasets** option. The default is CUZ#INEX. This is the History Include/Exclude data set member created during product configuration, which contains a list of data sets that are to be copied to the cloud. For configuration information for this member, refer to [“Step 2: Create user parameter library” on page 13](#) and to [“Step 11: Customize history include list” on page 16](#).

Debug Mode

Specify a debugging mode. Valid values are:

- N – None. Do not debug any jobs.
- A– All jobs. Debug all of the jobs.
- J – Job name. Only debug the jobs matching a specific job name.

Debug Jobname

If you type a "J" in the **Debug Mode** field, you must type a job name to indicate which job(s) to debug. You can also specify a "J" in the **Debug Mode** field and a partial job name followed by an asterisk (*) to indicate "any character" in the **Debug Jobname** field. For example, TEST* indicates to debug all job names that begin with "TEST".

Note: While it is possible to specify a "J" in the **Debug Mode** and only an asterisk (*) as the **Debug Jobname**, it is better to specify an "A" in the **Debug Mode** field to display debugging information for All Jobs.

Write to Operator Msgs

Indicate whether you want messages sent to a computer operator. Valid values are:

- Y – Send the messages to the operator's console.
- N – Send the messages to a file, but not to the operator's console. (Default)

It is recommended to specify an "N" in this field due to the significantly large number of messages that could be sent to the console (flooding).

Max Cloud Backup Gens

Type a number between 1 - 10 to indicate the maximum number of generations you want to save to a cloud backup.

Max Backup History Tasks

Type a number from 1 - 99 to indicate the maximum number of tape drives that can be allocated at the same time to backup previously created data sets. This parameter will prevent Cloud Tape Connector from utilizing all of the tape units in your system.

Auto Backup Repository Minutes

Type a number between 05 and 9999 for the number of minutes to wait between automatic backups of the cloud repository. The default is 60.

Note: If you specify a value in the **Auto Bkup Repository Min** field, you must create a Repository Backup filter to specify where to save a backup of the entire repository. To create a Repository Backup filter, choose the **Cloud Connector Settings (Parmlib Options)** option on the Main Menu, then choose the **Backup Filter Criteria** option.

Exec History Interval Min

Type a number between 5 and 9999 for the number of minutes to wait between History Include/Exclude processing tasks. The first interval begins 20 seconds after the started task (CUZCLOUD STC) is started. The default interval is 60 minutes.

Note: Previously this process was part of repository backups. Now it is a separate parameter so you can have different intervals for automatic backups of the repository and of the history data sets.

Memory Cell Pool Size

Type a number between 2560000 to 99999999 for the size of each cell pool. The default is 55242880.

Memory Primary Cells

Type a number between 10 and 9999 for the maximum number of primary cells in a pool that you want to define for storage. The default is 250.

Share Repository across LPARs

This option determines if the same repository is going to be shared across multiple LPAR's within a single sysplex. Valid options are:

- Y – VSAM record-level sharing (RLS) is enabled for the integrity of the repository.
- N – VSAM RLS is disabled, but only one CUZCLOUD STC (started task) can be started with the same repository.

The default is N.

Note: Cloud Tape Connector no longer supports repository sharing across sysplexes. Sharing is only allowed within one sysplex.

Write SMF Record Type

User SMF record types 128 - 255 can be used for tracking tasks like copying data sets to the cloud, deleting a data set from the cloud, and restoring a data set from the cloud. Specifying a value between 128 and 255 will turn on SMF processing within Cloud Tape Connector. If you specify a zero (0), SMF records will not be written. The default is 0.

Staging Options

The staging process captures the data from tapes to be copied to the cloud and creates a backup on DASD; this backup will be written to the cloud later. By staging the data, if there is an issue with connectivity to the cloud server (the network goes down or the server is unavailable), the data can be copied to the cloud at a later time. While this may require additional time and space, it will ensure that data is not lost if a server I/O problem occurs while attempting to copy the data to the cloud.

You may also choose to stage the data if your transfer rate to the cloud is slow, because a slow transfer rate could increase the elapsed time of your batch jobs for writing. Staging the data to DASD can reduce the elapsed time of your batch jobs.

If you choose not to stage the tape data and the "direct" transfer to the cloud is interrupted, the backup will not be written to the cloud. The job will abend if the **Abend on Errors** field on the **Parmlib General Options** screen is set to "Yes".

If you choose the "direct" transfer to the cloud, you must define the valid OMVS segment for the userid assigned to the batch jobs.

Note: If the data is on DASD only and not on tape, the data will be copied directly to the cloud and no staging file will be used.

When you choose Option 2, **Staging Options**, on the Parmlib Options Main Menu, the following screen displays.

```
CUZ$PRMS V1R1 ----- Parmlib Staging Options -----
Option ==>

                                     2016/06/29 18:11:08
                                     User: USERID - CUZ

-----
Parmlib Dataset: PROD.CLOUD.SCUZPARM
Parmlib Member : CUZ#PARM
-----

Staging a backup to DASD is in place to quickly capture the data being
backed up to the Cloud, so the job writing the data doesn't have to
wait for server IO. If you turn this on, make sure the Primary and
secondary track allocations are large enough to handle any backups.
-----

Stage Data on DASD. . . . . Y (Yes / No)
Staging Dataset Alias . . . . CUZSTAGE (z/OS Cataloged Alias)
Staging Dataset Allocation. . C (Cyl / Trks)
Primary Space Allocation. . . 5000 (1 to 99999999)
Secondary Space Allocation. . 5000 (1 to 99999999)
Staging Dataset Vol Count . . 005 (1 to 256)
SMS Storage Class . . . . . SCCUZSTG (Optional)
Error Retry Count . . . . . 10 (1 to 9999)
Error Retry Interval Secs . . 5 (1 to 999)
Nbr of IO Buffers . . . . . 50 (1 to 50)

*CMD
```

Figure 5. Parmlib Staging Options

The Staging Options for the Parmlib member listed on the screen are described below. The default values are displayed, but you can change those values to suit your environment.

Staging Data on DASD

Indicate whether you want to stage the data from tapes (Y) or skip this step and write the data only to the cloud (N). The default is Y.

Staging Dataset Alias

Specify up to an eight-character name for the z/OS catalog alias. The default name is CUZSTAGE.

The staging alias **MUST** be a unique qualifier to be used only by Cloud Tape Connector. **DO NOT** use an alias being used by other users or products.

Note: The stage alias is required even if staging has been turned off. This alias is also used for creating backups of the repository and copy them to the cloud.

For more information on defining a high-level alias for staging, see [“Step 7: Create staging and restore aliases and define SMS information \(Optional\)”](#) on page 15 and [“Method 2: Staging data to DASD”](#) on page 7.

Staging Dataset Allocation

Specify the type of allocation you want to use for a staging data set. Valid values are "C" for cylinders and "T" for tracks. The default is C.

Primary Space Allocation

Type a number between 1 and 99999999 for the amount of primary space to allocate. The default is 5000. This allocation parameter must be large enough to handle any size data set being written to the cloud.

Secondary Space Allocation

Type a number between 1 and 99999999 for the amount of secondary space to allocate. The default is 5000. This allocation parameter must be large enough to handle any size data set being written to the cloud.

Staging Dataset Vol Count

Type a number between 1 and 256 for the amount of volumes you need to allocate for a multi-volume data set. The default is 5 volumes.

SMS Storage Class (Optional)

This is an optional field. Specify up to an eight-character name for the storage class. The default is SCCUZSTG.

Error Retry Count

Type a number between 1 - 9999 to indicate the maximum number of times you want to try to copy the data sets again after an error occurs. The default is 10.

Error Retry Interval Secs

Type a number between 1 - 999 to indicate the number of seconds to wait between retry attempts. The default is 5.

Nbr of IO Buffers

Type a number between 1 and 50 for the maximum number of I/O buffers that can be used. The default is 10.

Cloud Server Options

When you choose option 3, **Cloud Servers**, on the Parmlib Options Main Menu, the **Cloud Server Display** screen appears. It lists all of the cloud servers that are defined for use by Cloud Tape Connector. You can also create a new cloud definition, modify a definition, or delete one.

```
CUZ$PRMC V2R1      ----- Cloud Server Display ----- 2019/07/29 18:17:01
Option  ==>  _____ Scroll ==> PAGE

Line Commands: C - Create E - Edit  D - Delete

                                     ROW 1 OF 6                                     >
-----
Cmd Cld Name  Type Userid  Passwor IP Address  Reposit Directory Path
--- MYCLOUD   HCP  JoeSmo1 M0reSpc ab0.cde1.hcp70 fg0.ten Dir1/
--- HCP       HCP  JoeSmo1 M0reSpc ab0.cde1.hcp70 fg0.ten Dir2/
--- CLS       CLS  raND0mJ MLP3A15 2.10.16.134 mystuff /
--- S3        S3   JCAT    tMpls0V      mystuf2 /
--- SFT       SFT  YOURID2 12345f6 cb250.objst8 /
--- FTP       FTP  myusrid puPyd0g 12.10.17.25
***** Bottom of Data *****
```

Figure 6. Parmlib Cloud Servers option showing Cloud Server Display screen

The information on this screen is described below. When using Cloud Tape Connector, you must page right to see all of the columns that are described below.

Cld Name

The cloud name is a user-defined name for linking filter criteria to a specific cloud definition. Type one to eight characters for the name of the cloud to be defined.

When you define filter criteria, which is described in [“Backup Filter Criteria”](#) on page 48, you assign a cloud name to a filter to specify which data you want to save to this cloud. For example:

1. You define a cloud named "PROD".
2. You create a filter, based on the data set name, with a data set mask of MY.PROD.DATASETS.*.
3. You assign this filter to cloud PROD.

The result is only the data sets that match this filter will be saved to cloud PROD.

Type or Cld Type

The type of protocol used for this cloud. Valid values are:

- CLS - S3-compatible
- FTP - File transfer protocol
- HCP - Hitachi Content Platform
- S3 - Amazon Simple Storage Service
- SFT - IBM SoftLayer
- PRX - Proxy Server

User ID or Userid

The User ID is the server ID that is required for signing into this cloud. The User ID can be up to 64 characters in length.

Password

The password associated with the User ID that allows access to this cloud. The password can be up to 64 characters in length.

IP Address

The Internet Protocol Address used to connect to the cloud. The IP Address can be up to 128 characters in length.

Repository

The name or location of the repository on the cloud where the data will be saved. The repository information can be up to 128 characters in length.

Directory or Path

The name and path of the directory where the data will be saved. The directory information can be up to 255 characters in length.

Port

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80 for CLS, HCP, S3, and SFT clouds. For FTP, the default is 21.

RetryAtt

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is 3.

RetryTime

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is 30.

DatRetry

Indicates the number of times to retry a data transfer. Keep in mind that a data transfer can be significantly more expensive than other network communications, because the Part Size can be 100MB and above. The default is 0.

VirtHost

Indicates whether the connection to a CLS or S3 server should have the bucket name in front of the host name. For example, if this is enabled and S3 is the server, the product will connect to

bucketname.s3.amazonaws.com, instead of the standard connection to s3.amazonaws.com. This is useful in some installations where advanced network routing is in place. Valid values are Y (yes) or N (no). The default is N.

Compression or Comp

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If either YES or Y is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, cannot be accomplished via software mode.

ConnRetr(xxx,xxx)

Indicates the number of times to attempt to reconnect with the cloud. Consists of a two-part value:

- The number of times the product will attempt to reconnect if the socket connection is dropped. Valid values are 1 - 999. The default is 5.
- The number of seconds to wait between connection attempts. Valid values are 1 - 999. The default is 30.

PartSize(xxx)

Indicates the maximum amount of data (part size), in MB, that will be transferred to the cloud during a transaction. Specify a number between 5 – 100. The default is 5MB.

The bigger the number, the better the performance will be, but at a memory cost. Specifying 100MB will gain the most in terms of performance, but does allocate a large memory area to hold the data.

Threads

Specifies that multiple concurrent threads will be used when writing data to S3 or CLS clouds. Using multiple threads can help large backups complete more quickly. With each thread that is specified, there is an increase in the amount of memory that is needed. If you specify a value of 0, no multi-threading will be performed. Valid values are 0-20. The default is 0.

Encrypt

Specifies whether server-side AES256 encryption will be supported. Possible values are:

- If a "Y" (yes) is specified, Cloud Tape Connector will add the proper header to S3 or CLS traffic to allow it to be in compliance with a bucket policy that requires server-side encryption.
- If an "N" (no) is specified or if the field is left blank, no additional header will be used. The default is N.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Time-out

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Use Proxy

Specifies whether a CLS or S3 server should be connected through the Proxy Server. The proxy server should be defined as a new cloud server type, PROXY. Valid values are Y (yes) or N (no). The default is N.

Consider the following values:

In this example, Cloud Tape Connector will try to connect to the cloud server using port 8080 instead of the default port of 80. On failed connections, Cloud Tape Connector will attempt to connect again to the cloud server 5 times, waiting 30 seconds between attempts. Compression through zEDC will be used when transferring data to the cloud, and the maximum part size is 25MB of data.

Line Commands

- C - Create a new cloud server definition.
- D - Delete an existing cloud server definition.
- E - Edit an existing cloud server definition.

Create a cloud server definition

Type the number that matches the type of cloud you want to define to Cloud Tape Connector.

```

+----- Cloud Server Create -----+
Cloud Server Create

Create Cloud Type
  1. CLS - S3-Compatible
  2. FTP - File Transfer Protocol
  3. HCP - Hitachi Content Platform
  4. SFT - SoftLayer
  5. S3 - Amazon Simple Storage Service
  6. PRX - Define a Proxy Server

PF12: Cancel

```

Figure 7. Cloud Server Create window shows list of Cloud Types that can be defined

After you choose the type of cloud you want to define, a "Create" screen for that type will display. Refer to the appropriate section below to see what information you need to provide to create a cloud definition.

Cloud Type	Link to Section
CLS - S3-Compatible	“Specifying a S3-compatible cloud” on page 31
FTP - File Transfer Protocol	“Specifying an FTP cloud” on page 34
HCP - Hitachi Content Platform	“Specifying a Hitachi Content Platform cloud” on page 37

Table 2. Links to cloud type fields (continued)	
Cloud Type	Link to Section
SFT - IBM SoftLayer	“Specifying a SoftLayer cloud” on page 39
S3 - Amazon Simple Storage Service	“Specifying an S3 cloud” on page 42
PRX - Proxy Server	“Specifying a Proxy Server cloud” on page 45

Cloud Tape Connector also supports the use of Virtual Tape Emulation (VTE) to "vault" tapes, which saves virtual tape data to disk. Through module CUZ#APIB, VTE can save data to and retrieve data from any cloud defined in Cloud Tape Connector.

See [Appendix C, “Appendix C: Using the Cloud Connector batch API,” on page 379](#) for more information on module CUZ#APIB.

Specifying a S3-compatible cloud

You can define a connection between Cloud Tape Connector and an S3-compatible (CLS) cloud, and assign a name to the cloud where you want to save data. Examples of clouds that are considered "S3 compatible" include IBM Cloud Object Storage, FlashBlade from Pure Storage, and others.

When you enter the "C" line command in the CMD column on the **Cloud Server Display** screen, a window displays asking you to indicate which type of cloud connection you want to create. To create a connection to a S3-compatible (CLS) cloud, type "1" in the **Cloud Server Create** window. The following screen displays, including any default values:

```
CUZ$PCL1 V2R1      ----- Cloud Server Display ----- 2019/07/13 13:35:12
Option ===>                               Scroll ===> PAGE

Line Commands: C - Create E - Edit D - Delete

+----- S3-Compatible Server Definition -----+
| Cld Type   CLS           (CLS,FTP,HCP,SFT,S3)
| Cld Name   _____ (Cloud Name used to Match Filter Criteria)
| Key Id     _____
| Secret Key _____
| IP Addr    _____
|
| Bucket Nme _____
| Directory  _____
|
| PKDS label _____
|
| Port                               Leave blank to disable client-side encryption
|                               80      (1 to 65535)
| Time-out of socket Sec          60      (0 to 999)
| Compress                        N        (Yes / No)
| Connection Retry Attempts       3        (0 to 99)
| Connection Retry Time Sec       30      (0 to 999)
| Data Transfer Retry              0        (0 to 99)
| Virtual Host URL Style          N        (Yes / No)
| Server-side Encryption          N        (Yes / No)
| Number of Threads               0        (0 to 20)
| Partsize MB                     5        (5 to 100)
| Use Proxy Server                N        (Yes / No)
+-----+

```

Figure 8. S3-Compatible Server Definition

The fields on this window are described below. You may need to page down to see all of the fields on this screen.

Cld Type

The type of cloud connection to be created. In this case, the value is **CLS** for S3-compatible clouds.

Cld Name

The name that Cloud Tape Connector will use to identify the cloud where you want to save data. Type one to eight characters for the name of this cloud. You can assign a different cloud name for the same cloud storage device if you are saving data to a different directory on that cloud or are using different credentials (User ID and Password) to access the cloud.

Key ID

The Key ID is the server ID that is required for signing into this cloud. The Key ID must be 1 - 128 characters in length.

Secret Access Key

The password associated with the Key ID that allows access to this cloud. The Secret Access Key must be 1 - 128 characters in length.

IP Addr

The Internet Protocol Address used to connect to the cloud. The IP Address can be up to 128 characters in length.

Bucket Nme

The name or location of the repository on the cloud where the data will be saved. The Bucket Name must be 3 - 63 characters in length.

Directory or Path

(Optional) The name and path of the directory where the data written to the cloud will be saved. The directory information can be up to 256 characters in length. Leave this field blank if you want to store data in the root folder. The directory designates the top level folder that will hold all of the Cloud Tape Connector data. For example, if you specify "Testdata/" in the Directory field, the data will be sent to the "Testdata/" directory. This is useful to keep Cloud Tape Connector data separate from other data you have stored in that particular cloud.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Port

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80.

Time-out of socket Sec

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Compress

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If Y (yes) is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, can not be accomplished via software mode.

Connection Retry Attempts

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is 3.

Connection Retry Time Sec

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is one (1) second.

Data Transfer Retry

Indicates the number of times to retry a data transfer. Keep in mind that a data transfer can be significantly more expensive than other network communications, because the Part Size can be 100MB and above, which is a big cost. The default is 0.

Virtual Host URL Style

Indicates whether the connection to a S3-compatible server should have the bucket name in front of the host name. For example, if this is enabled and S3 is the server, the product will connect to `bucketname.s3.amazonaws.com`, instead of the standard connection to `s3.amazonaws.com`. This is useful in some installations where advanced network routing is in place. Valid values are Y (yes) or N (no). The default is N.

Server-side Encryption

Specifies whether server-side encryption will be supported.

- If a "Y" (yes) is specified, Cloud Tape Connector will add the proper header to S3-compatible traffic to allow it to be in compliance with a bucket policy that requires server-side encryption. This may also be referred to as AES256 encryption.
- If an "N" (no) is specified or if the field is left blank, no additional header will be used.

The default is N.

Number of Threads

Specifies that multiple concurrent threads will be used when writing data to a S3-compatible cloud. Using multiple threads can help large backups complete more quickly. With each thread that is specified, there is an increase in the amount of memory that is needed. If you specify a value of 0, no multi-threading will be performed. Valid values are 0-20. The default is 0.

Part Size MB

Indicates the maximum amount of data (part size), in MB, that will be transferred to the cloud during a transaction. Specify a number between 5 – 100. The default is 5MB. The bigger the number, the better the performance will be, but at a memory cost. Specifying 100MB will gain the most in terms of performance, but does allocate a large memory area to hold the data.

Use Proxy Server

Indicates whether a S3-compatible server should be connected through the Proxy Server. If a proxy server has not been defined, you cannot set the **Use Proxy Server** parameter to "Y". You must define a proxy server prior to turning this option on. A proxy server should be defined as a new cloud server type, PROXY. Valid values are Y (yes) or N (no). The default is N.

An example of a completed S3-Compatible Server Definition is shown below. Some of the values listed on this screen are not valid, but are provided as an example of the type of values that are expected for a CLS server. When using Cloud Tape Connector, you may need to page down to see all of the fields on this screen.

```

CUZ$PCL1 V2R1      ----- Cloud Server Display ----- 2020/03/18 13:35:12
Option ===>                               Scroll ===> PAGE

Line Commands: C - Create E - Edit D - Delete

+----- S3-Compatible Server Definition -----+
| Cld Type      CLS              (CLS,FTP,HCP,SFT,S3)
| Cld Name      CLDEXAMP         (Cloud Name used to Match Filter Criteria)
| Key Id        s1dfkig0g3fFGF3c2c56
| Secret Key    pm0v9vj4in4tng50g05894gm-8elo0rr3r0fqf99
| IP Addr       8.19.212.56
|
| Bucket Nme    mybucket
| Directory     Testdata/
|
|
| PKDS label
|               Leave blank to disable client-side encryption
| Port          80              (1 to 65535)
| Time-out of socket Sec 60      (0 to 999)
| Compress       N              (Yes / No)
| Connection Retry Attempts 3    (0 to 99)
| Connection Retry Time Sec 30   (0 to 999)
| Data Transfer Retry 1          (0 to 99)
| Virtual Host URL Style N       (Yes / No)
| Server-side Encryption Y       (Yes / No)
| Number of Threads 3           (0 to 20)
| Partsize MB    5              (5 to 100)
| Use Proxy Server N            (Yes / No)
+-----+

```

Figure 9. Example of a S3-Compatible Server Definition

In this example:

- Cloud Tape Connector will try to connect to the cloud server using the default port of 80.
- Compression through zEDC will not be used when transferring data to the cloud.
- On failed network connections, Cloud Tape Connector will attempt to connect again to the cloud server 3 times, waiting 30 seconds between each attempt.
- If a data transfer fails, one more connection attempt will be made to transfer the data.
- The bucket name will not be placed in front of the S3-compatible host name.
- The AES256 server-side encryption will include a header.
- Three threads will be used when writing data to the cloud.
- The maximum part size is 5 MB.
- The Proxy Server will not be used for a connection to the cloud.

Specifying an FTP cloud

One of the cloud types that is supported is the File Transfer Protocol (FTP) server. You can define a connection between Cloud Tape Connector and an FTP cloud, and assign a name to the cloud where you want to save data.

When specifying an FTP server, please keep the following requirements in mind:

- The FTP server must support passive FTP.
- The z/OS FTP servers are not supported at this time.

When you type the "C" line command in the CMD column on the **Cloud Server Display** screen, a window displays asking you to indicate which type of cloud connection you want to create. To create a connection to an FTP cloud, type "2" in the **Cloud Server Create** window. The following screen displays, including any default values:

```

CUZ$PCL2 V2R1      ----- Cloud Server Display ----- 2019/07/13 13:35:12
Option ===>
                                Scroll ==> PAGE

Line Commands: C - Create E - Edit D - Delete

                                ROW 1 OF 6                                >
+----- File Transfer Protocol Server Definition -----+
| Cld Type      FTP              (CLS,FTP,HCP,SFT,S3)      |
| Cld Name      _____      (Cloud Name used to Match Filter Criteria) |
| User Id       _____      |
| Password      _____      |
| IP Addr       _____      |
| Directory     _____      |
|               _____      |
|               _____      |
| PKDS label    _____      |
|               Leave blank to disable client-side encryption |
| Port          21              (1 to 65535)              |
| Time-out of socket Sec 60      (0 to 999)              |
| Compress      N               (Yes / No)                |
| Connection Retry Attempts 3      (0 to 99)              |
| Connection Retry Time Sec 1      (0 to 999)              |
+-----+

```

Figure 10. File Transfer Protocol (FTP) Server Definition

The fields on this window are described below.

Cld Type

The type of cloud connection to be created. In this case, the value is FTP for File Transfer Protocol.

Cld Name

The name that Cloud Tape Connector will use to identify the cloud where you want to save data. Type one to eight characters for the name of this cloud. You can assign a different cloud name for the same cloud storage device if you are saving data to a different directory on that cloud or are using different credentials (User ID and Password) to access the cloud.

User ID or Userid

The User ID is the server ID that is required for signing into this cloud. The User ID can be up to 64 characters in length.

Password

The password associated with the User ID that allows access to this cloud. The password can be up to 64 characters in length.

IP Addr

The Internet Protocol Address used to connect to the cloud. The IP Address can be up to 128 characters in length.

Directory or Path

(Optional) The name and path of the directory where the data written to the cloud will be saved. The directory information can be up to 256 characters in length. Leave this field blank if you want to store data in the root folder. The directory designates the top level folder that will hold all of the Cloud Tape Connector data. For example, if you specify "Testdata/" in the Directory field, the data will be sent to the "Testdata/" directory. This is useful to keep Cloud Tape Connector data separate from other data you have stored in that particular cloud.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default for FTP is 21.

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If Y (yes) is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, can not be accomplished via software mode.

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is three (3) times.

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is one (1) second.

An example of a completed FTP Server Definition is shown below. Some of the values listed on this screen are not valid, but are provided as an example of the type of values that are expected for an FTP server.

Figure 11. Example of FTP Server Definition

- Cloud Tape Connector will try to connect to the cloud server using the default port of 21.
- Compression through zEDC will not be used when transferring data to the cloud.
- On failed connections, Cloud Tape Connector will attempt to connect again to the cloud server 3 times, waiting 30 seconds between each attempt.

Specifying a Hitachi Content Platform cloud

One of the cloud types that is supported is a Hitachi Content Platform (HCP) server. You can define a connection between Cloud Tape Connector and a cloud on the HCP, and assign a name to the cloud where you want to save data.

When you type the "C" line command in the CMD column on the **Cloud Server Display** screen, a window displays asking you to indicate which type of cloud connection you want to create. To create a connection to an HCP cloud, type "3" in the **Cloud Server Create** window. The following screen displays, including any default values:

```
CUZ$PLC3 V2R1      ----- Cloud Server Display ----- 2019/07/13 13:35:12
Option ===>                               Scroll ===> PAGE

Line Commands: C - Create E - Edit D - Delete

+----- Hitachi Server Definition -----+
| Cld Type   HCP          (CLS,FTP,HCP,SFT,S3)
| Cld Name    _____ (Cloud Name used to Match Filter Criteria)
| User Id     _____
| Password    _____
| IP Addr     _____
|
| Repositor   _____
|
| Directory   _____
|
| PKDS label   _____
|
| Port        80          (1 to 65535)
| Time-out of socket Sec  60 (0 to 999)
| Compress     N          (Yes / No)
| Connection Retry Attempts 3 (0 to 99)
| Connection Retry Time Sec 1 (0 to 999)
+-----+

Leave blank to disable client-side encryption
```

Figure 12. Hitachi Server Definition

The fields on this window are described below.

Cld Type

The type of cloud connection to be created. In this case, the value is HCP for Hitachi Content Platform.

Cld Name

The name that Cloud Tape Connector will use to identify the cloud where you want to save data. Type one to eight characters for the name of this cloud. You can assign a different cloud name for the same cloud storage device if you are saving data to a different directory on that cloud or are using different credentials (User ID and Password) to access the cloud.

User ID or Userid

The User ID is the server ID that is required for signing into this cloud. The User ID can be up to 64 characters in length.

Password

The password associated with the User ID that allows access to this cloud. The password can be up to 64 characters in length.

IP Addr

The Internet Protocol Address used to connect to the cloud. The IP Address can be up to 128 characters in length. The format for the URL is:

```
namespace.tenant.host.
```

As an example, if your HCP "namespace" value is ns0, the "tenant" name is ten1, and the host name is hcp123.hcphostname.com, the URL value you would enter in the **IP Addr** field is:

```
ns0.ten1.hcp123.hcphostname.com.
```

The host name and domain name will almost always be the same, unless your HCP box is specifically configured to have separate names.

Repository

The name or location of the repository where the data will be saved. The repository information can be up to 128 characters in length.

Directory or Path

(Optional) The name and path of the directory where the data written to the cloud will be saved. The directory information can be up to 256 characters in length. Leave this field blank if you want to store data in the root folder. The directory designates the top level folder that will hold all of the Cloud Tape Connector data. For example, if you specify "Testdata/" in the Directory field, the data will be sent to the "Testdata/" directory. This is useful to keep Cloud Tape Connector data separate from other data you have stored in that particular cloud.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Port

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80.

Time-out of socket Sec

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Compress

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If Y (yes) is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, can not be accomplished via software mode.

Connection Retry Attempts

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is three (3) times.

Connection Retry Time Sec

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is one (1) second.

An example of a completed Hitachi Server Definition is shown below. Some of the values listed on this screen are not valid, but are provided as an example of the type of values that are expected for a Hitachi server.

```

CUZ$PLC3 V2R1      ----- Cloud Server Display ----- 2019/07/13 13:35:12
Option ===>                               Scroll ===> PAGE

Line Commands: C - Create E - Edit  D - Delete

                                ROW 1 OF 6                >
+----- Hitachi Server Definition -----+
| Cld Type   HCP                (CLS,FTP,HCP,SFT,S3)
| Cld Name   HCPEXAMP           (Cloud Name used to Match Filter Criteria)
| User Id    JoeSmo1
| Password   M0reSpc
| IP Addr    ns0.ten1.hcp123.hcphostname.com
|
| Repositor  ns0.ten1.hcp123.hcphostname.com
|
| Directory  Testdata/
|
| PKDS label
|
|           Leave blank to disable client-side encryption
| Port       80                (1 to 65535)
| Time-out of socket Sec  60    (0 to 999)
| Compress    N                (Yes / No)
| Connection Retry Attempts  3    (0 to 99)
| Connection Retry Time Sec  30    (0 to 999)
+-----+

```

Figure 13. Example of Hitachi Server Definition

In this example:

- Cloud Tape Connector will try to connect to the cloud server using the default port of 80.
- Compression through zEDC will not be used when transferring data to the cloud.
- On failed connections, Cloud Tape Connector will attempt to connect again to the cloud server 3 times, waiting 30 seconds between each attempt.

Specifying a SoftLayer cloud

One of the cloud types that is supported is an IBM SoftLayer server. You can define a connection between Cloud Tape Connector and a cloud on the IBM SoftLayer (SFT) server, and assign a name to the cloud where you want to save data.

When you type the "C" line command in the CMD column on the **Cloud Server Display** screen, a window displays asking you to indicate which type of cloud connection you want to create. To create a connection to an IBM SoftLayer (SFT) cloud, type "4" in the **Cloud Server Create** window. The following screen displays, including any default values:

```

CUZ$PCL4 V1R1      ----- Cloud Server Display ----- 2016/07/13 13:35:12
Option ===>                               Scroll ==> PAGE

Line Commands: C - Create E - Edit D - Delete

+----- SoftLayer Server Definition -----+
| Cld Type      SFT          (CLS,FTP,HCP,SFT,S3)
| Cld Name      _____ (Cloud Name used to Match Filter Criteria)
| User Id      _____
| API Key      _____
| IP Addr      _____
|
| Container     _____
|
| Directory     _____
|
| PKDS label    _____
|
| Port          _____ Leave blank to disable client-side encryption
| Time-out of socket Sec  80 (1 to 65535)
| Compress       N         (Yes / No)
| Connection Retry Attempts 3 (0 to 99)
| Connection Retry Time Sec 1 (0 to 999)
+-----+

```

Figure 14. SoftLayer Server Definition

The fields on this window are described below.

Cld Type

The type of cloud connection to be created. In this case, the value is SFT for IBM SoftLayer.

Cld Name

The name that Cloud Tape Connector will use to identify the cloud where you want to save data. Type one to eight characters for the name of this cloud. You can assign a different cloud name for the same cloud storage device if you are saving data to a different directory on that cloud or are using different credentials (User ID and Password) to access the cloud.

User ID or Userid

The User ID is the server ID that is required for signing into this cloud. The User ID can be up to 64 characters in length.

API Key

The password associated with the User ID that allows access to this cloud. The password can be up to 64 characters in length.

IP Addr

The Internet Protocol Address used to connect to the cloud. The IP Address can be up to 128 characters in length.

Container

The name of the container (similar to a repository) where the data will be saved. The container name can be up to 128 characters in length.

Directory or Path

(Optional) The name and path of the directory where the data written to the cloud will be saved. The directory information can be up to 256 characters in length. Leave this field blank if you want to store data in the root folder. The directory designates the top-level folder that will hold all of the Cloud Tape Connector data. For example, if you specify "Testdata/" in the Directory field, the data will be sent to the "Testdata/" directory. This is useful to keep Cloud Tape Connector data separate from other data you have stored in that particular cloud.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the

corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Port

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80.

Time-out of socket Sec

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Compress

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If Y (yes) is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, can not be accomplished via software mode.

Connection Retry Attempts

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is 3.

Connection Retry Time Sec

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is one (1) second.

An example of a completed SoftLayer Server Definition is shown below. Some of the text values listed on this screen are not valid, but are provided as an example of the type of values that are expected for a SoftLayer server.

```
CUZ$PCL4 V1R1      ----- Cloud Server Display ----- 2016/07/13 13:35:12
Option ===>                               Scroll ==> PAGE

Line Commands: C - Create E - Edit  D - Delete

+----- SoftLayer Server Definition -----+
| Cld Type   SFT              (CLS,FTP,HCP,SFT,S3)
| Cld Name   SFTEXAMP         (Cloud Name used to Match Filter Criteria)
| User Id    JoeSmo1
| API Key     M0reSpc
| IP Addr     IBM0S123456-7:conn_administrator
|
| Container   myContainer
|
| Directory   Testdata/
|
|
| PKDS label
|
|          Leave blank to disable client-side encryption
| Port        80              (1 to 65535)
| Time-out of socket Sec  60   (0 to 999)
| Compress     Y              (Yes / No)
| Connection Retry Attempts  6   (0 to 99)
| Connection Retry Time Sec  10  (0 to 999)
+-----+
```

Figure 15. SoftLayer Server Definition

In this example:

- Cloud Tape Connector will try to connect to the cloud server using the default port of 80.
- Compression through zEDC will be used when transferring data to the cloud.
- On failed connections, Cloud Tape Connector will attempt to connect again to the cloud server 6 times, waiting 10 seconds between each attempt.

Specifying an S3 cloud

One of the cloud types that is supported is an Amazon Simple Storage Service (S3) server. You can define a connection between Cloud Tape Connector and an S3 cloud, and assign a name to the cloud where you want to save data.

When you type the "C" line command in the CMD column on the **Cloud Server Display** screen, a window displays asking you to indicate which type of cloud connection you want to create. To create a connection to an Amazon Simple Storage Service (S3) cloud, type "5" in the **Cloud Server Create** window. The following screen displays:

```
CUZ$PCL5 V2R1      ----- Cloud Server Display ----- 2019/07/13 13:35:12
Option ==>                               Scroll ==> PAGE

Line Commands: C - Create E - Edit  D - Delete

                                     ROW 1 OF 6                               >
+----- Amazon Simple Storage Service Server Definition -----+
Cld Type      S3              (CLS,FTP,HCP,SFT,S3)
Cld Name      _____    (Cloud Name used to Match Filter Criteria)
Key Id        _____
Secret Key     _____
Bucket Nme    _____
Directory     _____
PKDS label    _____
                Leave blank to disable client-side encryption
Port          80              (1 to 65535)
Time-out of socket Sec  60    (0 to 999)
Compress       N              (Yes / No)
Connection Retry Attempts  3  (0 to 99)
Connection Retry Time Sec  1  (0 to 999)
Data Transfer Retry       0    (0 to 99)
Virtual Host URL Style    N    (Yes / No)
Server-side Encryption    N    (Yes / No)
Number of Threads        0     (0 to 20)
Partsize MB             5      (5 to 100)
Use Proxy Server        N      (Yes / No)
```

Figure 16. Amazon Simple Storage Service (S3) Server Definition

The fields on this window are described below. When using Cloud Tape Connector, you may need to page down to see all of the fields on the screen.

Cld Type

The type of cloud connection to be created. In this case, the value is S3 for Amazon Simple Storage Service.

Cld Name

The name that Cloud Tape Connector will use to identify the cloud where you want to save data. Type one to eight characters for the name of this cloud. You can assign a different cloud name for the same cloud storage device if you are saving data to a different directory on that cloud or are using different credentials (User ID and Password) to access the cloud.

Key ID

The Key ID is the server ID that is required for signing into this cloud. The Key ID must be 1 - 128 characters in length.

Secret Access Key

The password associated with the Key ID that allows access to this cloud. The Secret Access Key must be 1 - 128 characters in length.

Bucket Name

The name or location of the repository on the cloud where the data will be saved. The Bucket Name must be 3 - 63 characters in length.

Directory or Path

(Optional) The name and path of the directory where the data written to the cloud will be saved. The directory information can be up to 256 characters in length. Leave this field blank if you want to store data in the root folder. The directory designates the top level folder that will hold all of the Cloud Tape Connector data. For example, if you specify "Testdata/" in the Directory field, the data will be sent to the "Testdata/" directory. This is useful to keep Cloud Tape Connector data separate from other data you have stored in that particular cloud.

PKDS Label

To encrypt data before sending it to the cloud server, you must specify the Public Key Data Set (*pkds*) label, which contains the public key that will be used to encrypt the data. During retrieval, the corresponding private key will be used to decrypt the data from the cloud server. The PKDS Label can be up to 64 characters in length. The default is to leave this field blank.

If you do not know the PKDS, contact your systems programmer for assistance.

Port

Identifies the port by which the product attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80.

Time-out of socket Sec

This parameter specifies the maximum amount of time (in seconds) that sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60.

Compress

Indicates whether to use zEnterprise Data Compression (zEDC) to compress data. The default is N (no).

If Y (yes) is specified, Cloud Tape Connector will attempt to use zEDC to compress the data when transferring to the cloud.

Note: For more information on zEDC, refer to <http://www.redbooks.ibm.com/redbooks/pdfs/sg248259.pdf>

Compression can only be completed on systems with full z/EDC support. For systems that do not have the required hardware for z/EDC, decompressing the data can still be accomplished via z/EDC's software inflate mode. Compressing data, however, can not be accomplished via software mode.

Connection Retry Attempts

Indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0 - 99. The default is 3.

Connection Retry Time Sec

Indicates the number of seconds to wait between connection attempts. Valid values are 0 - 999. The default is one (1) second.

Data Transfer Retry

Indicates the number of times to retry a data transfer. Keep in mind that a data transfer can be significantly more expensive than other network communications, because the Part Size can be 100MB and above, which is a big cost. The default is 0.

Virtual Host URL Style

Indicates whether the connection to an S3 server should have the bucket name in front of the host name. For example, if this is enabled and S3 is the server, the product will connect to `bucketname.s3.amazonaws.com`, instead of the standard connection to `s3.amazonaws.com`. This

- Cloud Tape Connector will try to connect to the cloud server using the default port of 80.
- Compression through zEDC will not be used when transferring data to the cloud.
- On failed connections, Cloud Tape Connector will attempt to connect again to the cloud server 3 times, waiting 30 seconds between each attempt.
- If a data transfer fails, one more connection attempt will be made to transfer the data.
- The bucket name will not be placed in front of the S3 host name.
- The AES256 server-side encryption will include a header.
- Three threads will be used when writing data to the cloud.
- The maximum part size is 5 MB.
- The Proxy Server will not be used for a connection to the cloud.

Specifying a Proxy Server cloud

The "Proxy Server" cloud type is used to define values, such as the IP address and port number, for a single proxy server for . Only one proxy server can be defined. An S3 or S3-compatible (CLS) cloud server can be connected through this Proxy Server.

When you type the "C" line command in the CMD column on the Cloud Server Display screen, a window displays asking you which type of cloud connection you want to create. To create a definition for this proxy server, type "6" (for Proxy Server) in the Cloud Server Create window. The following screen displays:

```
CUZ$PCL6 V2R1 ----- Proxy Server Definition ----- 2020/04/29 18:17:01
Option ==>----- Scroll ==> PAGE
Cld Type:  PROXY          (CLS,FTP,HCP,SFT,S3,PROXY)
Cld Name   PROXY          (Cloud Name PROXY required for Proxy Server)
Login *    -----
Password *  -----
Server Adr  -----
Port       80----- (1 to 65535)
* - Fields are optional
Proxy Servers are supported for S3 and CLS type clouds. To use this Proxy
server, set option Use Proxy Server to "Y" on S3 or CLS cloud definitions.
```

Figure 18. Initial Proxy Server Definition window

The fields on this window are described below.

Cld Type

Indicates the type of cloud connection to be created. In this case, the value is PROXY for Proxy Server.

Cld Name

Indicates the name of the cloud connection to be created. In this case, the value is PROXY for Proxy Server. It cannot be changed.

Login

(Optional) Specifies the Login name used for Proxy Authorization. This parameter is optional and should be specified if the Proxy Server in your network requires Proxy Authorization. The Login can be up to 128 characters in length.

Password

(Optional) Type the password for this proxy server. The password associated with the Login that is used for Proxy Authorization. The parameter is optional and should be specified if the Proxy Server in your network requires Proxy Authorization. The password can be up to 128 characters in length.

Server Address

Specify the Internet Protocol Address used to connect to the Proxy Server. The IP Address can be up to 128 characters in length.

Port

Identifies the port by which the product attempts to connect to the Proxy server. Valid values are 1 - 65535. The default is 80.

An example of a completed Proxy Server Definition is shown below. Some of the values listed on this screen are not valid, but are provided as an example of the type of values that are expected for a Proxy server.

```
CUZ$PCL6 V2R1 ----- Proxy Server Definition ----- 2020/04/29 18:17:01
Option ==>----- Scroll ==> PAGE
Cld Type:  PROXY          (CLS,FTP,HCP,SFT,S3,PROXY)
Cld Name   PROXY          (Cloud Name PROXY required for Proxy Server)
Login *    login
Password * password
Server Adr 192.168.1.1
Port       80            (1 to 65535)
* - Fields are optional
Proxy Servers are supported for S3 and CLS type clouds. To use this Proxy
server, set option Use Proxy Server to "Y" on S3 or CLS cloud definitions.
```

Figure 19. Example of values for Proxy Server Definition window

In this example:

- Cloud Tape Connector will try to connect to the Proxy server using the default port of 80.
- Login and Password credentials will be used if the Proxy Server in your network requires Proxy Authorization.

Using ICSF with S3 and CLS clouds

Cloud Tape Connector uses callable services provided by the IBM Integrated Cryptographic Service Facility (ICSF). In order for the product to function properly, ICSF needs to be present on the system with the ICSF started task running.

Cloud Tape Connector utilizes some functions of ICSF in order to do certain hashing that is necessary for S3 and S3-compatible (CLS) API requests. If ICSF is not running, S3 and CLS requests will fail with an error stating that the hash algorithm could not be performed. In this case, start the ICSF started task and try the process again.

Using Proxy Server to connect to S3 and CLS clouds

Cloud Tape Connector may use a "Proxy Server" to connect to Amazon Simple Storage Service (S3) and S3-compatible (CLS) clouds. To ensure proper function, the proxy server should be configured in your network.

Cloud Tape Connector does not check the proxy server configuration and connection. If the proxy server is not configured correctly, S3 and CLS requests will fail with an error.

An *unsecured* (http) connection through the proxy server works via any port that is specified for an S3 or CLS cloud. However, a *secure* (https) connection through a proxy server works only with port 443. In this case, be sure that your proxy server configuration allows forward https traffic via port 443 for the CONNECT method. A specific AT-TLS rule is required for this case.

Edit a cloud server definition

When you choose option 3, **Cloud Servers**, on the Parmlib General Options screen, the **Cloud Server Display** screen appears. When you type the "E" line command next to an existing cloud definition, a window displays the current definition for that cloud.

You can edit the information for that specific cloud. The fields you can modify will vary based on the type of cloud you are editing. Refer to the appropriate section in the table below to see what information you need to provide to edit a cloud definition.

Table 3. Links to cloud type fields

Cloud Type	Link to Section
CLS - S3-Compatible	“Specifying a S3-compatible cloud” on page 31
FTP - File Transfer Protocol	“Specifying an FTP cloud” on page 34
HCP - Hitachi Content Platform	“Specifying a Hitachi Content Platform cloud” on page 37
SFT - IBM SoftLayer	“Specifying a SoftLayer cloud” on page 39
S3 - Amazon Simple Storage Service	“Specifying an S3 cloud” on page 42
PRX - Proxy Server cloud	“Specifying a Proxy Server cloud” on page 45

Delete a cloud server definition

When you choose option 3, **Cloud Servers**, on the Parmlib General Options screen, the **Cloud Server Display** screen appears. When you type the "D" line command next to a cloud definition on this screen, the **Cloud Server Delete** window appears.

The **Confirm Delete** field, located at the bottom of the window, is asking you to confirm that you want to delete this server definition. Type one of these values in this field:

- **Y** - Yes, you want to delete this server definition.
- **N** - No, you want to keep this server definition. This is the default.

You can also press PF12 to cancel and exit this window. The server definition will not be deleted if you cancel.

You cannot delete a Proxy Server cloud if the **Use Proxy Server** parameter is enabled in any S3 or CLS cloud server definition. You must turn these options off before a delete of a proxy is allowed.

An example of a **Cloud Server Delete** window is shown below. The server information that appears on this window will vary depending upon the type of server you have chosen to delete.

```
CUZ$PRMC V1R1      ----- Cloud Server Display -----   2016/05/29   22:14:54  
Option ==>                               Scroll ==> PAGE
```

Line Commands: C - Create E - Edit D - Delete

ROW 1 OF 3 >

```
+----- Cloud Server Delete -----+  
Cld Type    FTP          (CLS,FTP,HCP, SFT, S3)  
Cld Name     xxxx        (Cloud Name used to Match Filter Criteria)  
User Id      userID4thiscloud  
Password     password4thiscloud  
IP Addr      123.456.7.89  
  
Repositor    \company  
  
Directory    y:\test\samplib  
  
Confirm Delete N (Y/N)                                PF12: Cancel
```

Figure 20. Example of a Cloud Server Delete confirmation window

Backup Filter Criteria

When you choose option 4, **Backup Filter Criteria**, on the Parmlib Options Main Menu, the **Cloud Filter Display** screen appears. It lists all of the filters that are defined for use by Cloud Tape Connector. You can also create a new cloud filter, modify a filter, or delete one.

```
CUZ$PRMF V1R1      ----- Cloud Filter Display ----- 2016/05/29 18:30:02
Option  ==>  ----- Scroll ==> PAGE

Line Commands: C - Create E - Edit  D - Delete

                                     ROW 1 OF 9
-----
  Filter      Cloud      Catalog      Retention      Filter
Cmd  Type     Name       To Cloud    Period    Criteria
-   -
-   Storclass  MARS       No          7          IBMVTL*
-   Dataset    MARS       Yes         2          PROD.DUMP01.*
-   Dataset    MARS       No          2          TESTLOG.*
-   Dataset    MARS       No          2          TESTSYS.*
-   Dataset    MARS       Yes         2          PROD.*
-   Esot Unit   JUPITER    No          2          VTAPE
-   Esot Unit   MARS       No          2          CART
-   Dataset    MARS       Yes         0          SYSID.*
-   Repo Bkup   MARS       No          0
-   Dataset    MARS       No          0          PROD.TEST.*
***** Bottom of Data *****
```

Figure 21. Example of Cloud Filter Display window for Backup Filter Criteria option

The information on this screen is described below. If you choose to Create or Edit a filter, these same fields appear on the **Cloud Filter Criteria Create** screen and the **Cloud Filter Criteria Edit** screen.

Filter Type

The type of filter used for this cloud. Valid values are:

- S - Storage class
- E - Esoteric Unit
- D - Dataset
- R - Repository Backup.

The Repository Backup filter specifies where to save a backup of the entire repository. No criteria is allowed for this filter type and only one Repository filter can be defined. A backup of the repository is performed every "n" minutes, based on the value you specified in the **Auto Bkup Repository Min** field on the General Options screen (under Parmlib Options).

Note: If you specified a value in the **Auto Bkup Repository Min** field, but do not create a Repository filter to specify where the repository backup should be saved, an error message displays to inform you that a Repository filter must be created to automatically backup the repository.

Be extremely cautious when creating a Storage Class or Esoteric Unit filter criteria that is associated with DASD devices. Do not include DASD devices as part of an Esoteric Unit filter. Creating a filter with "SYSDA" or "SYSALLDA" will copy JES sysout and joblog data sets to the cloud. Filter criteria on DASD devices should only be specified via Dataset Filters.

Cloud Name

The name of the cloud server where you want to save data. Type one to eight characters for the name of this cloud. The cloud must have already been defined through the Cloud Servers option (under Parmlib Options).

Catalog to Cloud

Specify Y or N to indicate whether to catalog a data set to the cloud. The default is N. If you specify Y, the data set will be copied to the cloud and cataloged (or re-cataloged) with a volume serial of CLOUD. The original data set will be deleted from disk or un-cataloged from tape.

Note: Re-cataloging a Repository Filter is not allowed.

Retention Period

Type a number between 0 and 9999 for the number of days you want to retain this backup. A value of zero is helpful for testing so you can run an expiration job and retest without having to wait days for the group of backed up data sets to expire. A value of 9999 means the data set will never expire or be deleted from the cloud via the expiration job. The only way to delete a data set from cloud with a Retention Period of 9999 is to manually delete it using Option 3 on the (ISPF) Main Menu.

Filter Criteria

Enter the data set names, esoteric unit names or storage class names to be used as selection criteria for this filter. A data set name can be up to 44 characters in length. The name of an esoteric or a storage class can be up to 8 characters. Wildcard characters are allowed (* %).

Line Commands

Choose one of these line commands for use on this screen:

- **C** - Create a new cloud filter.
- **D** - Delete an existing cloud filter.
- **E** - Edit an existing cloud filter.

Each of these commands displays a window where you complete the process.

Create cloud filter criteria

When you choose option 4, **Backup Filter Criteria**, on the Parmlib Options Main Menu, the **Cloud Filter Display** screen appears. When you type the "C" line command on this screen, the Cloud Filter Criteria Create window appears.

Specify values in the fields to create new cloud filter criteria. For more information about these fields, refer to [“Backup Filter Criteria” on page 48](#).

```
CUZ$PRMF V1R1      ----- Cloud Filter Display ----- 2016/05/29 18:30:02
Option ===>                                           Scroll ===> PAGE

Line Commands: C - Create E - Edit D - Delete

+----- Cloud Filter Criteria Create -----+
|
|  Filter Type      -      (Storclass,Esoteric Unit,Dataset,Repository)
|  Cloud Name      -      (Previously Defined Cloud Name)
|  Catalog to Cloud N      (Yes/No)
|  Retention Period ---- (# Days to Retain on Cloud)
|  Filter Criteria  -----
|
|                                                     PF12: Cancel
+-----+
```

Figure 22. Cloud Filter Criteria Create window

To save a backup of the entire Repository, you must specify "R" as the Filter Type and provide a Cloud Name and Retention Period. However, no value for the "Filter Criteria" field is allowed for this filter type because the whole repository will be saved to this location. Only one Repository (R) Backup filter can be defined.

Edit cloud filter criteria

When you choose option 4, **Backup Filter Criteria**, on the Parmlib General Options screen, the **Cloud Filter Display** screen appears. It lists all of the filters that are defined for use by Cloud Tape Connector. When you type the "E" line command on this screen, the **Cloud Filter Criteria Edit** window appears.

Specify values in the fields to edit an existing cloud filter. For more information about these fields, refer to [“Backup Filter Criteria” on page 48](#).

Figure 23. Cloud Filter Criteria Edit window

Delete cloud filter criteria

Figure 24. Confirm deletion of Cloud Filter Criteria

Data Restore Options

50 IBM Cloud Tape Connector for z/OS User Guide

```

CUZ$PRMR V1R1 ----- Parmlib Restore Options -----
Option ==>

2016/06/29 21:43:03
User: USERID - CUZ

-----
Parmlib Dataset: PROD.CLOUD.SCUZPARM
Parmlib Member : CUZ#PARM
-----

Datasets can be restored from the Cloud to a DASD unit as long as the
blksize is 32k or less. If the blksize is greater than 32K, the
data set will be restored to tape. You can also rename the DSN that
is being restored by entering a name in the Restore Alias field.
-----

Restore to DASD . . . . . Y (Yes / No)
Max Restore Tasks . . . . . 05 (01 to 99)
Restore DASD Unit . . . . . SYSALLDA (DASD Unit Device)
Restore TAPE Unit . . . . . TAPE (Tape Unit Device)
Restore Alias . . . . . CUZRESTR (Optional)
SMS Storage Class . . . . . SCCUZSTG (Optional)
Restore DSN Vol Count . . . . . 005 (1 to 256)
Retention Period. . . . . 0005 (0 to 9999)

```

Figure 25. Example of Parmlib/Data Restore Options screen with defaults displayed

The information on this screen is described below. The values displayed on this screen are the default values. Any and all of these values can be overridden at the time of restore.

Restore to DASD

Indicate whether you want to restore the data to DASD by typing Y (yes) or N (no). The default is Y. To restore the data to TAPE, specify N.

- If you type a "Y" in this field, you must also specify a value in the **Restore DASD Unit** field and in the **Restore TAPE Unit** field. If the block size is greater than 32K, the restored data must be sent to tape rather than DASD.
- If you type an "N" in this field, the data will not be restored to DASD, but will be restored to TAPE. Therefore, you must specify a value in the **Restore TAPE Unit** field.

Max Restore Tasks

Type a number between 01 and 99 for the maximum number of drives you want to use to perform this restore. Do not set the number higher than the actual number of drives you have at your site. The default is 5.

Restore DASD Unit

Specify a one- to eight-character name for the DASD device where the cloud data will be placed. The default name is SYSALLDA. To send data to DASD, you must have a value of "Y" in the **Restore to DASD** field. Also note the following:

- If the block size (blksize) is less than 32K, the cloud data will be restored to DASD.
- If the block size is greater than 32K, the restored data must be sent to tape rather than DASD.

Restore Tape Unit

Specify a one- to eight-character name for the TAPE device where the cloud data will be placed. The default name is TAPE.

If the block size (blksize) is greater than 32K, the restored data must be sent to tape rather than DASD. For this reason, you must specify a value in the **Restore TAPE Unit** field even if you specify a "Y" in the **Restore to DASD** field.

Restore Alias (Optional)

Specify a one- to eight-character high level alias to be used as a replacement high level when restoring the data set. The default alias name is CUZRESTR. The Restore Alias cannot be the same alias as the Staging Dataset Alias. For information on the Staging Dataset Alias, see ["Staging Options"](#) on page 26. This value is optional.

Restore DSN Vol Count

Type a number between 1 and 256 for the number of volumes you need to restore a multi-volume data set from the cloud. The default is 5 volumes.

SMS Storage Class (Optional)

Specify a one- to eight-character name for the storage class. This is an optional field.

Retention Period

Enter a number between 0 - 9999 to indicate the maximum number of days you want to keep the restored data when it is restored to tape. This Retention Period is a z/OS retention period which will be used by your Tape Management System.

Save parameter member

When you choose option 6, **Save/Save As**, on the Parmlib Options Main Menu, the **Save/Update Parmlib Member** window appears. On this window, you can edit and save the current parameter data set and member, or change the values and save them in a new data set or member.

The Save/Update Parmlib Member fields are described below. The default values are the data set and member name that you previously specified on the **Enter Parmlib Member** window when you chose Option 1, **Cloud Connector Settings (Parmlib Options)**, on the Main Menu. You can use these default values or specify a new parameter data set and member.

```
CUZ$PRML V1R1 ----- Parmlib Restore Options -----
Option ==> 6

                                     2018/04/29 21:51:58
                                     User: USERID - CUZ

-----
Parmlib Dataset: PROD.TEST.SCUZPARM
Parmlib Member : CUZ#PARM
-----

1. General Options
+----- Save / Update Parmlib Member -----+
| Parmlib Dataset  PROD.TEST.SCUZPARM          |
| Parmlib Member   CUZ#PARM                    |
|               |                               |
| Encrypt Cloud Definitions:  Y   (Yes/No)      |
| Refresh Params in Started Task: Y (Yes/No)    |
|               |                               |
|               PF12: CANCEL                     |
+-----+

5. Data Restore Options

6. Save / Save As

X. Exit
```

Figure 26. Example of Save/Update Parmlib Member window

The information on this screen is described below.

Parmlib Dataset

Specify the name of the data set that contains the parameter member. The SCUZPARM data set included with this product contains an example of the default parameter member, CUZ#PARM.

Parmlib Member

Type a one- to eight-character name for the parameter member. The default is CUZ#PARM.

Encrypt Cloud Definitions

Specify Y or N to indicate whether you want Cloud Tape Connector to encrypt the cloud server definitions in the Parmlib so that the ISPF screens display encrypted information after a cloud has been defined, rather than actual information. In addition, if you choose encryption (Y), the cloud definitions will not be displayed in write-to-operator (WTO) messages at startup, and will no longer be displayed when printing the cloud definitions in the parmliib. The default is Y (yes).

Refresh Parms in Started Task

Specify Y or N to indicate whether you want Cloud Tape Connector to immediately implement the changes you have made to the parameters. Normally, any changes made to the parameters do not take effect until the Cloud Tape Connector started task is restarted or the parameter member is refreshed. However, by specifying "Y" in this field, you can choose to have the parameters refreshed in the active started task.

Note: If the STC (started task) for Cloud Tape Connector, which is CUZCLOUD, is **not** active, the option to "Refresh Parms in Started Task" will not appear in this window. The STC must be running for this option to appear.

Additional method for refreshing the parameter member

If you prefer to refresh the parameters in the parmlib member through a console command, you can do so without having to stop and start the started task by entering the following z/OS console command:

```
/F CUZCLOUD,REFRESH MBR=CUZ#PARM (or the member name you saved it under)
```

In order for these new parameter options to take effect, the parameter member must be stored in the same data set specified in the CUZ#PARM DD statement in the Cloud Connector started task (CUZCLOUD).

Note: The "Refresh Parms in Started Task" option will do this for you automatically (run this console command) if you specify "Y" for that option.

Viewing cloud server status

When you choose Option 2, **Cloud Servers Status**, on the Main Menu, the **Cloud Server Status** screen appears. It lists all of the cloud servers that are defined for use by Cloud Tape Connector and their connection status.

Note: For information on how to define a cloud server, refer to [“Create a cloud server definition” on page 30](#).

```
CUZ$SERV V1R1      ----- Cloud Server Status ----- 2016/06/01 11:15:09
Option ==>                               Scroll ==> PAGE

-----
Cloud Name: MARS      Status: Connected      Type: HCP
User Id...: BestID1
IP Address: abc-de-fghi-JK.company.com
Repository: ab01.cd01.xyz-demo.abcdomain.com
Directory : mydata/
Total Byte: 15,062,925,049                      14 Gigabytes
-----
Cloud Name: xxxx      Status: Connected      Type: FTP
User Id...: MyuserID
IP Address: 123.456.7.89
Repository: \company10
Directory : y:\test\samplib
Total Byte: 0                      0 Bytes
-----
Cloud Name: CLS      Status: Connected      Type: CLS
Key Id...: slCXabCD1E1Fg23hIjKL
IP Address: 0.12.34.567
Repository: myrepos
Directory : /
Total Byte: 0                      0 Bytes
***** Bottom of Data *****
```

Figure 27. Example of Cloud Server Status screen

The information on this screen is described below

Cloud Name

The user-defined name given to a cloud server or location where the data will be saved.

Status

The connectivity state of the cloud server: Connected or Connection Error.

Type

The type of protocol used for this cloud:

- CLS - S3-Compatible
- HCP - Hitachi Content Platform
- FTP - File Transfer Protocol
- SFT- SoftLayer (IBM)
- S3 - Simple Storage Service (Amazon)
- PRX - Proxy Server

User ID or Key ID

The user or key identification required to login to the cloud server.

IP Address

The Internet Protocol Address used to connect to the cloud.

Repository or BucketName

The name or location on the cloud where the backup will be saved.

Directory or Path

The name and path of the directory where the backup will be saved.

Total Bytes

The amount of space that is being used for storage on this cloud.

Chapter 5. Primary commands

IBM Cloud Tape Connector for z/OS supports a number of primary commands that enable you to find information, navigate panels, modify the display of data, and print information.

FIND *abc*

Finds a unique string within a panel of data where *abc* is the string for which you are searching. If the specified string is found, the cursor moves to the first position of the found string. If the specified string is not found a message displays to indicate that is the case. You should be as specific as possible when using the **FIND** command to ensure the correct return.

The **FIND** command can be issued with the following keywords:

NEXT

Finds the next instance of the search string.

PREV

Finds the previous instance of the search string.

FIRST

Finds the first instance of the search string.

LAST

Finds the last instance of the search string.

ALL

Finds all instances of the search string.

The syntax is:

```
FIND <string> <keyword>
```

OR

```
FIND <keyword> <string>
```

where <string> is the text you want to find and <keyword> is a valid keyword for the **FIND** command (NEXT, PREV, FIRST, LAST, or ALL).

If none of the these keywords is explicitly specified, the default behavior is for the next instance of the search term to be found. If a keyword is the only parameter, it is treated as a search string. Two keyword parameters can coexist as long as one of them is identified by surrounding single quotes as the search string. Otherwise, the occurrence of multiple instances of keywords causes an error.

Examples:

To find the first instance of the word "apple", issue the command:

```
FIND apple FIRST
```

To find the next instance of the word "apple":

```
FIND apple
```

To find the last instance of the word "apple":

```
FIND apple LAST
```

To find all instances of the word "apple":

```
FIND apple ALL
```

To find all instances of the word "all", use single quotes to distinguish the search term from the keyword:

```
FIND 'all' ALL
```

Or:

```
FIND ALL 'all'
```

FORM

Reformats the display of a selected line item on a report panel such that each column becomes a row and values display in list format. To use the **FORM** command, type **FORM** in the option line, place your cursor on the line item of interest, and press Enter. The data for the selected line item will be displayed in list format showing column names and their associated values.

Notes:

1. To return to the original view from **FORM** view, press PF3.
2. CSETUP functions are not accessible when in **FORM** view. Exit **FORM** view to access CSETUP functionality.

NROW *n*

Displays the report for a subsequent row of interest where *n* is the number of rows (after the currently displayed row) that you want to scroll forward (when viewing reports in **FORM** view). The default value of *n* is 1.

Note: The **NROW** command is only valid when viewing a report in **FORM** view.

PROW *n*

Displays the report for a previous row of interest where *n* is the number of rows (prior to the currently displayed row) that you want to scroll back (when viewing reports in **FORM** view). The default value of *n* is 1.

Note: The **PROW** command is only valid when viewing a report in **FORM** view.

PRINTX

The **PRINTX** command takes a screen shot of a report and sends it to an output queue. The default output destination is the default output queue for your site. For example, if your site's default output class is configured to send output to the hold queue, the **PRINTX** command sends the currently displayed report to the hold queue. You can then view the output using SDSF.

You can change the output class designation for the **PRINTX** command by entering **PRINTX S** in the command line and pressing Enter. The following panel is displayed:

Figure 28. PRINTX Setup panel

```
SETUP ----- PRINTX Setup ----- 2010/02/25 14:27:15
Command ==> -----
Specify new output class and press ENTER
      or
press END to cancel.

If new output class is blank, default output class is used.

Current Output Class ==> DEFAULT OUTPUT MESSAGE CLASS
New Output Class      ==> _
```

Enter the desired output class in the **New Output Class** field and press Enter. The new output class is saved across sessions and remains in effect unless you change it. For appropriate output classes available at your site, check with your systems programmer. To change the class back to the default output message class, blank out the value in the **New Output Class** field.

For a snapshot of the current display (print screen), the ISPF Print command can be used. The ISPF Print command writes output to the ISPF LIST data set. See the *ISPF User's Guide* (SC34-4822, SC34-4823) for more information about ISPF Print.

SORT column_number direction

Sorts data (on panels of scrollable or tabular data) by column where *column_number* is the number of the column by which you want to sort and *direction* can be either **A** (to sort data in ascending order) or **D** (to sort data in descending order).

You can refer to columns only by the column number (not the column name). Column numbers are not displayed on the panel. The CMD column is column 1 and columns to the right are incremented sequentially.

Data can be sorted in ascending (A) or descending (D) order. To specify sort order, append the A or D to the end of the SORT command. The default is ascending (A). For example, to sort column 2 in descending order, type the following in the command line and press Enter:

```
SORT 2 D
```

The data will be sorted by column 2 in descending order.

Chapter 6. Column display functions

Column display functions (**CSETUP** functions) enable you to rearrange report columns, change the width of individual columns, and control the vertical ordering of columns.

CSETUP functionality enables you to:

- Rearrange report columns horizontally using the **CFIX** and **CORDER** options.
- Change the width of individual columns using the **CSIZE** option.
- Control the vertical ordering of columns using the **CSORT** option.

Additional column display functions enable you to:

- Scroll horizontally between columns, in both left and right directions.
- Scroll horizontally within a single report column while other report columns remain stationary on the screen.
- Insert column numbers above each display column.
- Generate a ruler at the top of the report columns beneath the headings.
- Display an entire row-column data element.

The customizations, or views, you configure using **CFIX**, **CORDER**, **CSIZE**, and **CSORT** can be saved across sessions.

The following syntax restrictions apply to the use of **CSETUP** functionality:

- Underlined text indicates the minimum acceptable abbreviation for each keyword.
- Variables are shown in italicized lowercase type.
- Keyword options are separated by vertical lines (|).

Accessing the **CSETUP** Primary Option Menu

The **CSETUP** primary option menu enables you to access the various **CSETUP** options and configure column display functions according to your display needs.

About this task

The **CSETUP** command uses the following syntax:

CSETUP

Launches the **CSETUP Primary Option Menu**.

To access and use the **CSETUP Primary Option Menu**:

Procedure

1. On any dynamic display (for example, the **Objects Profile Display** panel, the **Utilities Profile Display** panel, or the **Jobs Profile Display** panel), type **CSETUP** (or **CSET**) in the Option line and press Enter.
The **Setup Primary Option Menu** displays as shown in the following figure:

```

SETUP ----- Setup Primary Option Menu ----- YYYY/MM/DD HH:MM:SS
Command ==>
Temporary View

1 CFIX      Select columns to be fixed on the left side of the report
2 CORDER    Modify the horizontal placement of unfixed columns
3 CSIZE     Customize the size of columns
4 CSORT     Select columns to sort
5 CRESET    Reset column values
6 CREMOVE   Remove all customizations, including original defaults
7 PVIEW     Permanent View (toggle between temporary and permanent)

HELP      Setup Tutorial

```

Figure 29. Setup Primary Option Menu panel

2. Type the number corresponding to the option you want to access in the Command line and press Enter. The following options are available on the **Setup Primary Option Menu**:

CFIX

Option 1, **CFIX**, enables you to fix and unfix columns.

CORDER

Option 2, **CORDER**, enables you to reposition columns.

CSIZE

Option 3, **CSIZE**, enables you to change the displayed width of columns.

CSORT

Option 4, **CSORT**, enables you to select one or more columns for sorting and thus modify the order of the rows displayed.

CRESET

Option 5, **CRESET**, enables you to reset all customizations.

CREMOVE

Option 6, **CREMOVE**, enables you to remove all customizations.

PVIEW

Option 7, **PVIEW**, enables you to toggle between permanent view and temporary view.

Note: You can also directly invoke each **CSETUP** option by typing the corresponding command (for example, **CFIX**, **CORDER**, **CSIZE**, **CSORT**, **CRESET**, **CREMOVE**, or **PVIEW**) in the option line on any dynamic display and pressing Enter.

Fixing a column

The CFIX option enables you to fix and unfix columns. A fixed column is always located at the far left side of the display.

About this task

It does not shift horizontally (as unfixed columns do) when scrolling to the left or right. INNER COLUMN SCROLLING and CEXPAND may be used on a fixed column if the column is narrower than its maximum width. Certain columns may be permanently fixed in the report and cannot be unfixed by the user. Such a column has a fix status of P (permanently fixed).

A column cannot be fixed if it is larger than the available display area. There are also restrictions for fixing columns related to the size requirements of other columns.

To fix a column:

Procedure

1. Type **CFIX** in the option line on any display panel and press Enter.

The **Define Fixed Columns** panel displays as shown in the following figure:


```

CFIX ----- Define Fixed Columns ----- YYYY/MM/DD HH:MM:SS
Option  ==>                               Scroll ==> PAGE
-----+>
                                           ROW 1 OF 9

Column Function ==> 1 (1-Fix/Unfix, 2-Order, 3-Size, 4-Sort)
Permanent View  ==> Y (Y-Perm, N-Temp) Reset View ==> N (Y,N)

Device_Width   : 80
Old_Fixed_Width: 37      Old_Unfixed_Width: 43
New_Fixed_Width:         New_Unfixed_Width:

-----

Cmd New Old Len Column_Name
P  P  P   5  CMD
P  P  P  32  NAME
-      10  CREATOR
-      5  UPDT
-     32  DESCRIPTION
-     10  LAST_USER

Enter: Process selections; PF3: Exit and save; CAN: Exit without save
Line Cmds: F Fix U Unfix

```

Figure 30. Define Fixed Columns panel

The following fields appear on the **Define Fixed Columns** panel:

Column Function

Enables you to jump to any of the CSET functions by typing in the appropriate number. The number corresponding to the current option displays in this field.

Permanent View

Indicates whether the view you define is permanent or temporary. Valid values are:

- **Y**—View customizations are permanent.
- **N**—View customizations are temporary.

Reset View

Resets all customizations.

Device_Width

Shows the current display device size (screen width).

Old_Fixed_Width

Shows the sum of the FIXED column widths prior to any changes in the current CFIX panel.

Old_Unfixed_Width

Shows the UNFIXED area prior to any changes in the current CFIX panel. Old_Unfixed_Width = Device_Width - Old_Fixed_Width.

New_Fixed_Width

Shows the sum of the FIXED column widths that will result if the FIX/UNIFIX changes are saved.

New_Unfixed_Width

Shows the UNFIXED area that will result if the FIX/UNFIX changes are saved. New_Unfixed_Width = Device_Width - New_Fixed_Width.

Cmd

Field where you specify line commands. Valid line commands are F (fix) and U (unfix).

New

Displays the new CFIX view settings.

Old

Displays the previous CFIX view settings.

Len

Shows the length of the column.

Column_Name

Shows the name of the column.

2. Type **F** in the **Cmd** field next to column(s) you want to fix.
3. Type **U** in the **Cmd** field next to column(s) you want to unfix.
4. Press Enter.
The changed values display in the **New** column next to the corresponding column(s).
5. Press **PF3** to save changes and return to the display panel.

Repositioning columns

The **CORDER** option enables you to reposition report columns. If any columns are fixed, they are grouped together as the leftmost report columns. The unfixed columns are grouped together to the right of any fixed columns.

About this task

CORDER does not move a column out of its group. A fixed column cannot be relocated to the right of an unfixed column. Likewise, an unfixed column cannot be relocated to the left of a fixed column.

To reposition columns:

Procedure

1. Type **CORDER** in the option line on any display panel and press Enter.

The **Define Column Display Order** panel displays as shown in the following figure:

```
CTCORD ----- Define Column Display Order ----- YYYY/MM/DD HH:MM:SS
Option  ==>                                         Scroll ==> PAGE
----->
                                         ROW 1 OF 9

Column Function ==> 2 (1-Fix/Unfix, 2-Order, 3-Size, 4-Sort)
Permanent View  ==> N (Y-Perm, N-Temp)  Reset View ==> N (Y,N)

Cmd  Fix  New  Old  Column_Name
---  ---  ---  ---  ---
   P           1  CMD
   P           2  NAME
           3  CREATOR
           4  UPDT
           5  DESCRIPTION
           6  LAST_USER
           7  LAST_UPDATED
           8  CRTD_USER
           9  CREATED_USER
***** Bottom of Data *****

Enter: Process selections; PF3: Exit and save; CAN: Exit without save
Line Cmds: Specify number for column position
```

Figure 31. Define Column Display Order panel

The following fields appear on the **Define Column Display Order** panel:

Column Function

Enables you to jump to any of the CSET functions by typing in the appropriate number. The number corresponding to the current option displays in this field.

Permanent View

Indicates whether the view you define is permanent or temporary. Valid values are:

- **Y**—View customizations are permanent.
- **N**—View customizations are temporary.

Reset View

Resets all customizations.

Cmd

Field where you specify the number for column position.

Fix

Displays fixed columns. Valid values are:

- F—Indicates the column is fixed.
- P—Indicates the column is permanently fixed.

New

Displays the new CORDER view settings.

Old

Displays the previous CORDER view settings.

Column_Name

Shows the name of the column.

2. Type a number next to a column to specify its order.
3. Press Enter.

The new column order numbers display in the **New** column next to each column.

4. Press **PF3** to return to the display panel.

Resizing columns

The CSIZE option enables you to change the displayed width of columns.

About this task

This function is primarily intended for non-numeric data where there are large blank areas in all (or most) rows in a given column. Although the displayed width may change, the underlying data does not change.

If a column's size is less than the column maximum, it is possible that some data is not displayed. INNER COLUMN SCROLLING and CEXPAND can be used to see data outside the display range of the re-sized column.

Note: If the minimum and maximum column widths are equal, the column cannot be re-sized.

To re-size columns:

Procedure

1. Type **CSIZE** in the option line on any display panel and press Enter.

The **Define Column Size** panel displays as shown in the following figure:

```
CSIZE ----- Define Column Size ----- YYYY/MM/DD HH:MM:SS
Option  ==>                               Scroll ==> PAGE
-----+>
                                           ROW 1 OF 9

Column Function ==> 3 (1-Fix/Unfix, 2-Order, 3-Size, 4-Sort)
Permanent View  ==> N (Y-Perm, N-Temp)  Reset View ==> N (Y,N)

Device_Width   : 80
Old_Fixed_Width: 37      Old_Unfixed_Width: 43
New_Fixed_Width:         New_Unfixed_Width:

-----

Cmd New Old Min Max Fix Column_Name
  5   5   5   5   P   CMD
 32  32  32  32   P   NAME
 10  10  10  10     CREATOR
  5   5   5   5     UPDT
 32  32  32  32     DESCRIPTION
 10  10  10  10     LAST_USER

Enter: Process selections; PF3: Exit and save; CAN: Exit without save
Line Cmds: Column size, between MIN and MAX
```

Figure 32. Define Column Size panel

The following fields appear on the **Define Column Size** panel:

Column Function

Enables you to jump to any of the CSET functions by typing in the appropriate number. The number corresponding to the current option displays in this field.

Permanent View

Indicate whether the view you define is permanent or temporary. Valid values are:

- **Y**—View customizations are permanent.
- **N**—View customizations are temporary.

Reset View

Resets all customizations.

Device_Width

Shows the current display device size (screen width).

Old_Fixed_Width

Shows the sum of the FIXED column widths.

Old_Unfixed_Width

Shows the UNFIXED area.

New_Fixed_Width

Shows the sum of the FIXED column widths.

New_Unfixed_Width

Shows the UNFIXED area.

Cmd

Field where you specify the number for column position.

New

Displays the new CSIZE view settings.

Old

Displays the previous CSIZE view settings.

Min

Displays the minimum column length.

Note: If the minimum and maximum column widths are equal, the column cannot be re-sized.

Max

Displays the maximum column length.

Note: If the minimum and maximum column widths are equal, the column cannot be re-sized.

Fix

Displays fixed columns. Valid values are:

- **F**—Indicates the column is fixed.
- **P**—Indicates the column is permanently fixed.

Column_Name

Shows the name of the column.

2. Type the desired column size in the **Cmd** field next to the column you want to re-size.

Note: The column size you specify must be between the Min and Max values shown for that column.

3. Press Enter.

The new view criteria display in the **New** column.

4. Press **PF3** to return to the display panel.

Sort functionality

CSORT functionality enables you to select one or more columns for sorting and thus modify the order of the rows displayed on many product panels.

Columns are selected by sort priority and direction. Direction is either ascending (default) or descending. When more than one column is selected for sorting, the second column only differentiates when rows have matching data in the first column. Similarly, a third column only impacts the sort when data in both the first two columns are identical.

Defining sort columns

You can sort display data by columns. You can select up to nine columns for sorting.

About this task

A maximum of nine columns can be selected for sorting at one time. Internal requirements may create a smaller maximum. A message is issued if the maximum number of columns selected for sorting is exceeded.

Note: **CSORT** and **SORT** are synonymous.

Procedure

1. Type **CSORT** (or **SORT**) in the option line on any display panel and press Enter.

The **Define Sort Columns** panel displays as shown in the following figure:

```
SORT ----- Define Sort Columns ----- YYYY/MM/DD HH:MM:SS
Option ==>                               Scroll ==> PAGE
----->
                                      ROW 1 OF 9

Column Function ==> 4 (1-Fix/Unfix, 2-Order, 3-Size, 4-Sort)
Permanent View ==> N (Y-Perm, N-Temp)  Reset View ==> N (Y,N)
Stop Sorting    ==> N (Y,N)

Cmd Dir New Old Column_Name
- - - - -
- - - - - NAME
- - - - - CREATOR
- - - - - UPDT
- - - - - DESCRIPTION
- - - - - LAST_USER
- - - - - LAST_UPDATED
- - - - - CRTD_USER
- - - - - CREATED_USER

Enter: Process selections; PF3: Exit and save; CAN: Exit without save
Ord: 1-9 Dir: A Asc D Desc
```

Figure 33. Define Sort Columns panel

The following fields appear on the **Define Sort Columns** panel:

Column Function

Enables you to jump to any of the CSET functions by typing in the appropriate number. The number corresponding to the current option displays in this field.

Permanent View

Indicate whether the view you define is permanent or temporary. Valid values are:

- **Y**–View customizations are permanent.
- **N**–View customizations are temporary.

Stop Sorting

Indicates whether to stop sorting as specified. Valid values are:

- **Y**—Stop sorting.
- **N**—Continue sorting.

Cmd

Field where you specify the sort order.

Dir

Specifies the lexicographic order for the column. Valid values are:

- **A**—(Default) Values are listed in ascending order, smallest to largest.
- **D**—Values are listed in descending order, largest to smallest.

New

Displays the new CSORT view settings.

Old

Displays the previous CSORT view settings.

Column_Name

Shows the name of the column.

2. Type **A** or **D** in the **Cmd** field next to the columns on which you want to base your sort.
3. Press Enter. The new sort preferences are displayed in the **New** column.
4. Press **PF3** to return to the display panel.

Fast-path SORT command

The SORT command can be used as a primary (fast-path) command by typing the appropriate SORT syntax in the Option line of any report panel and pressing Enter.

The functionality supports both single and multi-column sorting and enables users to specify sort order (ascending or descending) for each column in the sort.

Syntax for single-column sorting

The syntax for single-column sorting is as follows:

SORT *column_identifier* *dir*

Where *column_identifier* is either the **column name** or the **relative column number** and *dir* is the direction in which to sort the column data. Valid values for *dir* are:

asc

(Default) Sorts data in ascending order.

desc

Sorts data in descending order.

Notes:

1. There must be a space between the *column_identifier* and its *dir* (if used).
2. The **relative column number** for a column is determined based on the column's placement when visible on the screen. Thus, relative column numbers are only available for columns currently visible on the screen. Relative column numbers are determined by counting the displayed columns from left to right, with the leftmost visible column being assigned the number '1' and each successive column (reading left to right) being assigned a relative column number that is incremented by 1. **Hint:** To quickly determine the column number, use the **CNUM** command to toggle on the column numbers for each display column.
3. You can sort on a column that is not displayed if you use the **column name** (instead of the **relative column number**) as the *column_identifier* in the SORT syntax.

Multi-column sorting

The syntax for multi-column sorting is as follows:

`SORT column_identifier dir column_identifier dir`

Where *column_identifier* is either the column name or the relative column number and *dir* is an optional indication of the direction in which to sort the column data. Valid values for *dir* are:

asc

(Default) Sorts data in ascending order.

desc

Sorts data in descending order.

The *column_identifier* and *dir* values must all be separated by spaces. The maximum number of columns that can be sorted at once is 9.

Usage examples

For a report display that has three columns, all of which display on the screen:

Column 1: Name

Column 2 Creator

Column 3: Status

The following examples show how you can sort these columns:

SORT NAME

Sorts display data in ascending order based on the value in the **Name** column (when no *dir* value is specified, the default sort order is ascending, thus **SORT NAME** and **SORT NAME A** are synonymous).

SORT NAME D

Sorts display data in descending order based on the value in the **Name** column.

SORT NAME DESC

Sorts display data in descending order based on the value in the **Name** column.

SORT NAME A CREATOR D

Sorts display data first in ascending order based on the value in the **Name** column and then sorts data in descending order based on the value in the **Creator** column.

SORT NAME ASC CREATOR DESC

Sorts display data first in ascending order based on the value in the **Name** column and then sorts data in descending order based on the value in the **Creator** column.

SORT 1 A

Sorts display data in ascending order based on the value in the **Name** column.

SORT 1 A CREATOR D

Sorts display data first in ascending order based on the value in the **Name** column and then sorts data in descending order based on the value in the **Creator** column.

SORT 3 2 1

Sorts the display data first in ascending order based on the value in the **Status** column, then in ascending order based on the value in the **Creator** column, and finally in ascending order based on the value in the **Name** column.

Note: When you specify a column name using any of the above formats, you may enclose it in single quotes, double quotes, or be without any quotes. For example, the following are equivalent:

SORT NAME D

SORT 'NAME' D

SORT "NAME" D

Resetting CSET customizations

The **CRESET** option enables you to reset all customizations.

About this task

After **CRESET** is issued, all fixed columns are unfixed (except for any permanently fixed columns), all selected sort columns are deselected and sorting is disabled, all column sizes are set to the initial values or maximum values if no suggested value previously existed, and original column locations are restored.

Procedure

1. To issue the **CRESET** option, access the **Setup Primary Option Menu** by typing **CSET** in the option line of any report display and pressing Enter.

The **Setup Primary Option Menu** displays.

2. Type **5** in the command line and press **Enter**.

CRESET is issued and all fixed columns are unfixed (except for any permanently fixed columns), all selected sort columns are deselected and sorting is disabled, all column sizes are set to the initial values or maximum values if no suggested value previously existed, and original column locations are restored.

3. Alternatively, you can issue the **CRESET** command as a primary command using the following syntax:

CRESET

Resets all customizations (unfixes fixed columns, deselects selected sort columns, sorting disabled, column sizes set to initial values, original column locations restored).

Note: **CRESET** differs from **CREMOVE** in that **CREMOVE** sets all column sizes to their maximum values ignoring any initial, suggested sizes.

Removing CSET customizations

The **CREMOVE** option enables you to remove all customizations.

About this task

After you issue the **CREMOVE** command, all fixed columns are unfixed (except for those that are permanently fixed), all selected sort columns are deselected and sorting is disabled, all column sizes are set to their maximum values, and original column locations are restored.

Procedure

1. To issue the **CREMOVE** option, access the **Setup Primary Option Menu** by typing **CSET** in the option line of any report display and pressing Enter.

The **Setup Primary Option Menu** displays.

2. Type **6** in the Command line and press Enter.

The **CREMOVE** command is issued.

3. Alternatively, you can issue the **CREMOVE** command as a primary command using the following syntax:

CREMOVE

Removes all customizations (unfixes fixed columns, deselects selected sort columns, sorting disabled, column sizes set to maximum values, original column locations restored).

Note: **CREMOVE** differs from **CRESET** in that **CREMOVE** sets all column sizes to their maximum values ignoring any initial, suggested sizes.

Column scroll

Column scrolling enables you to scroll horizontally between columns, in both left and right directions.

Use the following commands when viewing any dynamic display panel to scroll horizontally between columns:

CRIGHT *n*

Enables you to scroll the left side of the display window *n* report columns to the right.

CLEFT *n*

Enables you to scroll the left side of the display window *n* report columns to the left.

Inner column scroll

Inner column scroll enables you to scroll horizontally within a single report column while other report columns remain stationary on the screen.

Inner column scrolling may be useful for columns that have been shortened using the **CSIZE** functionality. Use the following commands when viewing any dynamic display panel to scroll horizontally within a single report column:

ICRIGHT

Enables you to scroll to the right within one report column while the other report columns remain stationary.

ICLEFT

Enables you to scroll to the left within one report column while the other report columns remain stationary.

Column numbers

Column numbers can be inserted above each display column.

The inserted column numbers are relative to the leftmost display column. Use the following command to invoke column numbering:

CNUM

Enables you to toggle on/off the column numbers above each display column.

Notes:

1. The leftmost displayed column is always numbered one (1) regardless of how far to the right you scroll.
2. You can use column numbers when issuing the **SORT** fast-path command.
3. Column numbers are not removed by **CRESET** nor **CREMOVE**. To remove column numbers, reissue the **CNUM** command.

Ruler display

The **COLS** command enables you to generate a ruler at the top of the report columns beneath the headings.

This ruler tracks the current position within the column. The < > symbols indicate whether there is additional column data to the left or right of the displayed data. For example:

```
<-5----2----5->
```

In this example, positions 13 through 28 are displayed. There is data both to the left and right of the currently displayed area.

The **COLS** command can be issued by itself, as a toggle switch, or with one parameter (ON|OFF). The syntax is as follows:

COLS (ON|OFF)

Enables you to generate a ruler at the top of the report columns to track the current position within the column.

Expanding columns

The **CEXPAND** command enables you to display an entire row-column data element.

About this task

This command can be useful in instances when the **CSIZE** command has reduced a column to a width that is too narrow to display all data. Expanding columns using the **CEXPAND** command provides you with an alternative to inner column scrolling.

Procedure

To invoke CEXPAND, place the cursor on a row-column element and issue the **CEXPAND** command. The cursor position determines the row-column that expands. The **CEXPAND** command can be issued by itself or with two parameters (row and column). The syntax is as follows:

CEXPAND (row column)

Enables you to display an entire row-column data element where *row* is the number of the row and *column* is the number of the column (non-heading lines only) that you want to expand.

Restrictions

The following restrictions apply to CSET options.

- Total fixed column sizes cannot exceed screen width.
- Total fixed column sizes must leave enough unfixed space for the minimum allowed size for all unfixed columns. If a column is not eligible for resizing, the column's minimum size requirement is the same as its maximum size. Minimum and maximum sizes for all columns are shown in the CSIZE display.
- If a column has been resized, then its current width is treated as its smallest allowable size. When a column is resized its current size must fit on the screen completely. For example, on an 80-byte screen with no fixed columns, a 128-byte column can only be resized to 80 bytes or less (assuming no conflicting minimum size associated with the column). If there were two 10-byte fixed columns, for a total fixed area size of 20-bytes, the 128-byte column would be limited to 60 bytes or its minimum allowed size, whichever was smaller.

Chapter 7. Operational Considerations

When using Cloud Tape Connector, several features help you operate the product, view the connection status of the cloud servers, and manage the data sets that are saved on the cloud or restored to tape or DASD. These product features and operational commands are described below.

Managing the started task

This section describes the available console commands. All commands are intended to be operands of the z/OS MODIFY (F) command.

The general command format is as follows:

```
F CUZCLOUD,keywords
```

Table 4. Command format	
Item	Description
F	Specifies the commonly used abbreviation used for the z/OS MODIFY command.
CUZCLOUD	Refers to the job name of the Cloud Tape Connector started task, which in this example is the default, CUZCLOUD.
keywords	Indicates the Cloud Tape Connector command to be executed. All of the operations are described in this chapter.

For example, if an operator wanted to display diagnostic information, the operator would type the following command:

```
F CUZCLOUD,DISPLAY DIAG
```

Please note that changes made to the Cloud Tape Connector configuration with console operator commands only apply to this execution of the Cloud Tape Connector started task and are not saved after the started task is stopped.

Cloud Tape Connector issues message CUZ0112E if the specified command is invalid and message CUZ0115E if a supported command contains invalid syntax.

For a complete list of available console commands, refer to [Appendix A, “Appendix A: List of Console Commands,”](#) on page 375.

Displaying Diagnostic information

Use the DISPLAY DIAG command to display diagnostic information about the Cloud Tape Connector started task. IBM Technical Support may ask for the information provided by this command as part of problem diagnosis. The Cloud Tape Connector started task issues messages CUZ0113I and CUZ0114I in response to this command.

The syntax for the DISPLAY DIAG command is:

```
F CUZCLOUD,DISPLAY DIAG
```

Keywords

DISPLAY - Cloud Tape Connector is to initiate Display processing.

Parameters

DIAG - Request a display of diagnostic information.

Diagnostic log files

To write errors and other informational messages that pertain to the transfer of data to and from a cloud server, you need to allocate the LOGMSG DD file. When LOGMSG is allocated, additional DDs can also be used to collect diagnostic information for IBM Technical Support. These additional DDs, as well as LOGTRACE and LOGDEBUG, should only be used when directed to do so by IBM Technical Support.

To obtain diagnostic data from any of these DD statements, a DD statement must be added to the CUZCLOUD started task procedure. The format of these DD statements is as follows:

```
//LOGMSG DD SYSOUT=*  
//LOGDEBUG DD DUMMY  
//LOGTRACE DD DUMMY
```

Updating parameter settings

You can make changes to the parameter values used by the Cloud Tape Connector started task (CUZCLOUD) by entering the new values in the parameter fields displayed via the Cloud Tape Connector ISPF interface. Be sure to save the parameter changes to the Parmlib member, which is CUZ#PARM by default. You can then use the REFRESH command to implement those changes immediately so that the new parameter values are used by the CUZCLOUD started task without having to restart that task.

Use the REFRESH command to implement the parameter changes listed in the specified parameter member without having to restart Cloud Tape Connector. The syntax is:

```
F CUZCLOUD,REFRESH MEMBER=member
```

Keywords

REFRESH - Update the active parameters in the CUZCLOUD started task.

Parameters

MEMBER - Replace *member* with the name of the parameter member to process. The default name for this member is CUZ#PARM. Ensure that changes to the parameter values have been saved in the Parmlib member, CUZ#PARM, and that this member has been saved in the parameter library before issuing the REFRESH command. The command will fail if the specified member name is not in the parameter library in use by the CUZCLOUD started task.

Example of command

This example shows how to use the REFRESH command to reload the parameter values from the Parmlib member, CUZ#PARM. These values will be used by the CUZCLOUD started task without having to restart the task.

```
F CUZCLOUD,REFRESH MEMBER=CUZ#PARM
```

Restoring a data set

Use the RESTORE command to initiate a restore request for a data set or initiate a stand-alone restore of the repository data set. The RESTORE operator command provides limited options for restoring data sets from cloud storage locations. For more flexibility in restoring data sets, use of the ISPF dialog is recommended.

Use the RESTORE DSN command to restore a data set. The command syntax is:

```
F CUZCLOUD,RESTORE DSN=dsname [GEN=nnn] [NEWNAME=new dsname]
```

Keywords

RESTORE - Requests Cloud Tape Connector to restore a specified data set.

Parameters

DSN - Replace *dsname* with the name of the data set that you wish to restore. Cloud Tape Connector issues messages CUZ088E and CUZ091E if a backup for the specified data set does not exist.

GEN - Replace *nnn* with a number from 1-10 to indicate the previous generation to restore. Use this value to select a backup version other than the most recent version. For example, GEN=1 requests the previous (-1) backup version. Cloud Tape Connector issues message CUZ091E if a matching prior backup version does not exist.

NEWNAME - Replace *new dsname* with the name you wish to assign to the restored data set. The RESTORE command will not delete or replace a currently cataloged data set with the same name as the one being restored. If this is the case, use this parameter to specify a new data set name that does not conflict with a name currently in the catalog.

Example 1

This example shows how to restore the current version of a data set that is no longer cataloged:

```
F CUZCLOUD,RESTORE DSN=DATA.SET
```

Example 2

This example shows how to restore the previous version of a data set that is currently cataloged:

```
F CUZCLOUD,RESTORE DSN=DATA.SET GEN=1 NEWNAME=DATA.SET.GEN1
```

Example 3

This example shows how to restore previous version 3 of a data set that is no longer cataloged:

```
F CUZCLOUD,RESTORE DSN=DATA.SET GEN=3
```

Performing a stand-alone restore of the repository data set

Use the RESTORE CLD command to perform a stand-alone restore of the repository data set from cloud storage when the repository is not available. This section only describes the command syntax, keywords, and parameters. Refer to the stand-alone restore procedures in [“Restoring a Repository” on page 89](#) for complete details on recovering and restoring the Cloud Tape Connector Repository.

The command syntax is:

```
F CUZCLOUD,RESTORE CLD=cllname CLDLIST=clclist
```

Keywords

RESTORE - Requests Cloud Tape Connector to restore a specified data set.

Parameters

CLD - Replace *cllname* with the name of the cloud storage location that contains the backup of the Repository. Cloud Tape Connector records this name in message CUZS056I when creating a copy of the Repository in Cloud storage.

CLDLIST - Replace *clclist* with the name of the file containing the Repository backup. This name is a fixed name constructed using the Staging alias as the high-level qualifier, the system name as the second-level qualifier, the Repository DDNAME from the started task as the third-level qualifier, and LIST as the last qualifier. Cloud Tape Connector records this name in message CUZS056I when creating a copy of the Repository in cloud storage.

Example of command

This example shows how to use the RESTORE CLD command to initiate a stand-alone restore of the Cloud Tape Connector Repository.

```
F CUZCLOUD,RESTORE CLD=MYCLOUD CLDLIST=CUZSTAGE.SYSA.CUZCLOUD.LIST
```

Considerations for restoring GDG's

When restoring a Generation Data Set (GDS), which is a member of a Generation Data Group (GDG), it is important to consider how the dataset will be restored. This section explains the terminology of GDG's and then describes how the dataset will be restored.

When defining a GDG, there are a few factors to consider:

- Will the GDG be SMS Managed or Non-SMS Managed?
- If SMS Managed, will the data set be located on a disk or on tape?

These factors are discussed in more detail below.

SMS-Managed GDG's

For a GDG to be managed by SMS (System-Managed Storage), a GDG base must first be defined to the z/OS Catalog, which requires a batch job to be run. The use of parameters in this batch job will determine how a GDG is restored

To define a GDG, run an IDCAMS DEFINE GDG batch job. The important parameters in this job are the LIMIT and SCRATCH/NOSCRATCH parameters. An example is shown below.

```
// JOBCARD
// *
//DEFGDG EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
DEFINE GDG (NAME('CUZ.MLGP00.DUMP.BKP') SCRATCH LIMIT(3))
DEFINE GDG (NAME('CUZ.MLGP01.DUMP.BKP') NOSCRATCH LIMIT(3))
/*
```

The LIMIT will determine the number of generations that can be addressed using Relative Generations Numbers. Relative generations are data sets that can be referenced with (0), (-1). An Absolute Generation refers to the data set name assigned to the catalog, such as MY.DATASET.G0004V00.

In the sample job above, two GDG's were created with a LIMIT of 3:

- The first data set was defined with the SCRATCH option.
- The second data set was defined with the NOSCRATCH option.

The SCRATCH option will delete and uncatalog data sets that are rolled off. The NOSCRATCH option will leave the old dataset cataloged, but it cannot be referenced by a relative generation number like (0) or (-1). It must be referenced by its absolute generation, like MY.DATASET.G0001V00.

Example of DSLIST results based on SCRATCH and NOSCRATCH parameters

Below is a DSLIST of data sets generated with the two GDG's that were created from the IDCAMS DEFINE GDG job described above. Five generations have been created for each GDG.

CUZ	*ALIAS
CUZ.MLGP00.DUMP.BKP	??????
CUZ.MLGP00.DUMP.BKP.G0003V00	STAG12
CUZ.MLGP00.DUMP.BKP.G0004V00	STAG01
CUZ.MLGP00.DUMP.BKP.G0005V00	STAG06
CUZ.MLGP01.DUMP.BKP	??????
CUZ.MLGP01.DUMP.BKP.G0001V00	STAG12
CUZ.MLGP01.DUMP.BKP.G0002V00	STAG08
CUZ.MLGP01.DUMP.BKP.G0003V00	STAG09
CUZ.MLGP01.DUMP.BKP.G0004V00	STAG08
CUZ.MLGP01.DUMP.BKP.G0005V00	STAG13

Notice that the first GDG, CUZ.MLGP00.DUMP.BKP, has only generations 3, 4, and 5 (listed as G0003V00, G0004V00, G0005V00 in the above example). This is because the SCRATCH option is used and generations 1 and 2 have been rolled off, deleted, and uncatalogued.

The second GDG, CUZ.MLGP01.DUMP.BKP, still shows all five generations. This is because the NOSCRATCH option was used. The data sets were not deleted or uncatalogued. Generations 1 and 2 have a status of ROLLED-OFF. These data sets (1 and 2) cannot be referenced by relative generation number

(-3), (-4). You will need to reference them by their absolute generation number, CUZ.MLGP01.DUMP.BKP.G0001V00.

Note: The NOSCRATCH option has no effect on GDG data sets that are on tape or when the Cloud Tape Connector Filter option, Catalog to Cloud, is turned on.

Example of use of LISTCAT command

To determine the status of a GDS, issue an IDCAMS LISTCAT command on the dataset or the GDG Base. The status values that are important in Cloud Tape Connector processing are the following:

Active

The dataset (GDS) is in the GDG index and can be referenced by a relative generation number.

Defer

The dataset (GDS) has been restored, but is outside of the GDG index and cannot be referenced by a relative generation number.

Rolled Off

The dataset (GDS) has been removed from the GDG index and can't be referenced by a relative generation number.

If you issue a LISTCAT command on the GDG base, it will only display data sets that are in the Active GDG index. It will not display Deferred or Rolled-Off generations. Below is a LISTCAT on GDG base CUZ.MLGP01.DUMP.BKP. Generations 1 and 2 are not displayed because they have been rolled-off.

```
GDG BASE ----- CUZ.MLGP01.DUMP.BKP
ATTRIBUTES
LIMIT-----3      NOSCRATCH  NOEMPTY    LIFO
ASSOCIATIONS
NONVSAM--CUZ.MLGP01.DUMP.BKP.G0003V00
NONVSAM--CUZ.MLGP01.DUMP.BKP.G0004V00
NONVSAM--CUZ.MLGP01.DUMP.BKP.G0005V00
```

Below is an example of a LISTCAT on generation 1 (listed as G0001V00), which has been rolled-off.

```
NONVSAM ----- CUZ.MLGP01.DUMP.BKP.G0001V00
HISTORY
DATASET-OWNER----- (NULL)      CREATION-----2018.068
RELEASE-----2      EXPIRATION-----0000.000
ACCOUNT-INFO----- (NULL)
STATUS-----ROLLED-OFF
```

Below is an example of a LISTCAT on generation 3 (listed as G0003V00), which is still active in the GDG index.

```
NONVSAM ----- CUZ.MLGP01.DUMP.BKP.G0003V00
HISTORY
DATASET-OWNER----- (NULL)      CREATION-----2018.068
RELEASE-----2      EXPIRATION-----0000.000
ACCOUNT-INFO----- (NULL)
STATUS-----ACTIVE
```

Non-SMS managed GDG's

Two of the STATUS values, Rolled-off and Defer, are *not* supported by non-SMS managed GDG's. All of the data sets in this type of GDG are considered to be active.

The NOSCRATCH parameter is ignored and data sets rolling-off will be deleted and uncatalogued.

Restoring SMS-Managed GDG's

Cloud Tape Connector takes caution to avoid modifying the GDG index. If a dataset has a status of Active, Cloud Tape Connector will restore it as Active. If the dataset has a status of Rolled-Off or Deferred, it will be restored as Deferred. However, there may be valid reasons to modify the GDG index. Therefore, a new parameter, **Roll into GDG Base**, has been created to allow you to modify the GDG index.

The "Roll into GDG Base" parameter has been added to the ISPF Restore screen, the batch restore job, and the Restore API macro. This new parameter has three options: Yes, No, or Conditional. Examples of how to specify this new parameter are provided below.

ISPF Restore pop-up panel

When you choose Option 3, **Cloud Datasets**, from the Main Menu, and then specify to restore (R) a data set, the "Restore Dataset Confirmation" pop-up panel displays. The **Roll into GDG Base** field is listed on this screen. An example is shown below.

```
CUZ$CDSN V1R1      ----- Cloud Dataset Display ----- 2016/02/04 15:22:19
Option ===>                               Scroll ===> PAGE

Line Commands: R - Restore V - View D - Delete

-----
Dataset Like *
+----- Restore Dataset Confirmation -----+
|
| Dataset Name      CUZ.MLGP01.DUMP.BKP.G0005V00
| Backup Timestamp  2018-03-09 07:17:51
| Restore Unit Type T (Tape / Disk)
| Restore DASD Unit SYSALLDA (Restore to DASD Esoteric Unit)
| Restore TAPE Unit CART (Restore to Tape Esoteric Unit)
| Restore Vol Count 005 (1 to 256)
| SMS Storage Class SCCUZSTG (Restore to SMS Storage Class)
| Retention Period  0004 (# days to retain restored DSN)
| Delete Old Backup N (Delete old backup DSN ?)
| Roll into GDG Base C (Yes/No/Conditional)
| Restore to Alias  CUZRESTR (Blank will Restore to Orig Name)
| Restore to Dataset -----
|
| Restore Dataset  N (Yes / No / All)
|
| PF12: Cancel
+-----
```

Valid values are Yes, No, and Conditional (default). See [“Values for Roll into GDG Base parameter”](#) on page 77 for more detailed information.

Batch Restore control cards

If you are using a batch job to run a restore process, you can specify the **ROLL_INTO_GDG_BASE** parameter. See the example below.

```
RESTORE_FROM_CLOUD (
  RESTORE_TO_ALIAS      CUZRESTR
  RESTORE_TO_UNIT       SYSALLDA
  RESTORE_SMS_STORAGE_CLASS SCCUZSTG
  RESTORE_VOLUME_COUNT  5
  RELATIVE_GENERATION    '0'
  RETENTION_PERIOD      0000
  DELETE_OLD_BACKUP     YES
  WAIT_FOR_COMPLETION   YES
  ROLL_INTO_GDG_BASE    CONDITIONAL
)
```

Valid values are: Yes, No, and Conditional (default). See [“Values for Roll into GDG Base parameter”](#) on page 77 for more detailed information.

API Restore Macro Parameter List

If you are using the API Restore macro to run a restore process, you can specify the **RESTORE_ROLL_INTO_GDG_BASE=ROLL_IN_GDG** parameter. See the example below.

```
CUZ#APIB FUNC=RESTORE_FROM_CLOUD,      +
        APIB_REG=(R8),                  +
        RESTORE_DSN=MY_DATASET,         +
        RESTORE_REL_GEN=_REL_GEN,       +
        RESTORE_TO_DASD_UNIT=DASD_UNIT, +
```



```

RESTORE_TO_TAPE_UNIT=TAPE_UNIT,      +
RESTORE_TO_ALIAS=MY_ALIAS,            +
RESTORE_TO_STORCLASS=STORCLASS,       +
RESTORE_TO_DASD_TAPE=D,                +
RESTORE_DELETE_OLD=Y,                  +
RETENTION_PERIOD=RETENTION_PERIOD,     +
RESTORE_WAIT_FOR_COMPLETION=Y,          +
RESTORE_ROLL_INTO_GDG_BASE=ROLL_IN_GDG

```

Values for Roll into GDG Base parameter

The values specified for the **Roll into GDG Base** parameter have different results depending on whether you are using SMS DASD data sets or SMS Tape data sets. This section describes the values and their results.

Values for Roll into GDG Base parameter for SMS DASD data sets

This section presents the values for the **Roll into GDG Base** parameter as they apply to SMS DASD data sets.

Yes

The generation data set (GDS) will be rolled into the active generation data group (GDG) index, regardless of its current state.

No

The GDS will be restored in Deferred state, regardless of its current state.

Conditional (default)

The restore process varies, depending on the present state of the GDS. Refer to the table below.

<i>Table 5. Conditional processing for SMS-managed DASD data sets</i>	
Current Status of GDS	Conditional processing results
Active	Restore as Active.
Defer	Restore as Deferred.
Rolled off	Restore as Deferred.
Not cataloged	Processing depends on whether or not the GDG Index is full: <ul style="list-style-type: none"> • If GDG Index is full, restore as Deferred. • If GDG Index is not full, restore as Active.

Roll into GDG Base for SMS Tape data sets

This section presents the values for the **Roll into GDG Base** parameter as they apply to SMS Tape data sets.

Yes

The generation data set (GDS) will be rolled into the active generation data group (GDG) index, regardless of its current state.

No

The GDS will be restored, but the data set will be renamed from G###V## to G###X##.

Conditional (default)

The restore process varies, depending on the present state of the GDS. Refer to the table below.

<i>Table 6. Conditional processing for SMS-managed tape data sets</i>	
Current Status of GDS	Conditional processing results
Active	Restore as Active.

Table 6. Conditional processing for SMS-managed tape data sets (continued)	
Current Status of GDS	Conditional processing results
Defer	Restore with the data set renamed from G####V## to G####X##.
Rolled off	Restore with the data set renamed from G####V## to G####X##.
Not cataloged	Restore with the data set renamed from G####V## to G####X##.

Restoring a GDS with an Active status

To restore a generation data set (GDS) in Active Status, the z/OS restore process allocates the dataset with DISP=(NEW,CATLG). By allocating the data set this way, the dataset will always be placed in the GDG index and will remove the oldest generation if the GDG index is full. In the examples provided in “SMS-Managed GDG's” on page 74, generations 3, 4 and 5 are active.

The following will occur when restoring generation 1 to Active Status:

- Generation 1 will be placed into the GDG index.
- Generation 3 is deleted.

After the restore, the following changes will have been made:

- Generations 2 and 3 have been deleted.
- Generations 1, 4 and 5 will be in the GDG index.
- Generation 5 will have a relative generation of (0).
- Generation 4 will have a relative generation of (-1).
- Generation 1 will have a relative generation of (-2).

Restoring a GDS in Deferred status

To restore a generation data set (GDS) in Deferred status, the restore process allocates the dataset with DISP=(NEW,KEEP). Since SMS requires *disk* data sets to be cataloged, Cloud Tape Connector will restore the dataset in Deferred Status and also catalog it.

When restoring a *tape* dataset with DISP=(NEW,KEEP), the dataset will *not* be cataloged. To ensure that a tape data set is restored and also cataloged with DISP=(NEW,CATLG), Cloud Tape Connector renames the generation level of the data set from G####V## to G####X##. Notice that the “V” has been renamed to an “X”. Renaming the data set will prevent the data set from being placed on Active status, allowing it to be in a Deferred status.

Restoring Non-SMS managed GDG's

Since Non-SMS managed generation data groups (GDG's) do not support a GDG status of Deferred or Rolled-Off, the generation data set (GDS) must be restored as Active, or placed in the GDG index.

Values for Roll into GDG Base parameter for Non-SMS data sets (Disk or Tape)

This section presents the values for the **Roll into GDG Base** parameter as they apply to both DASD (disk) or Tape data sets that are *not* SMS-managed.

Yes

The generation data set (GDS) will be rolled into the active generation data group (GDG) index, regardless of its current state.

No

The GDS will be restored, but the data set will be renamed from G####V## to G####X##.

Conditional (default)

The restore process varies, depending on the present state of the GDS. Refer to the table below.

Table 7. Conditional processing for non-SMS managed data sets	
Current Status of GDS	Conditional processing results
Cataloged	Restore as Active.
Not cataloged	Restore with data set renamed from G#V# to G#X#.

Starting the Cloud Tape Connector started task

Use the z/OS START command to initiate the Cloud Tape Connector started task. Use the following command:

```
[START|S] CUZCLOUD
```

Use either START or the "S" abbreviation.

Stopping the Cloud Tape Connector started task

Use either of the command formats below to stop the Cloud Tape Connector started task.

[P|STOP]CUZCLOUD

Enter this command to terminate the started task.

F CUZCLOUD,STOP

Enter this command to terminate the started task.

Example 1

This example shows how to stop the Cloud Tape Connector address space with the z/OS STOP command:

```
P CUZCLOUD
```

Example 2

This example shows how to stop the Cloud Tape Connector address space with the z/OS MODIFY (F) command:

```
F CUZCLOUD,STOP
```

Resolving problems with stopping the started task

If the Cloud Tape Connector started task fails to terminate normally, the started task may be cancelled with the CANCEL (or C) console command.

```
C CUZCLOUD
```

If this occurs and the started task is cancelled, the SVC intercepts may not be properly disabled. In this case, tailor and submit the JCL from the CUZJDSVC member in the SCUZSAMP data set to disable the SVC intercepts.

If a batch job was cancelled or forced out of the system while Cloud Tape Connector was processing that job, it may be necessary to cancel the CUZCLOUD address space due to residual storage and processing issues. If this condition should occur, tailor and submit the JCL from the CUZJDSVC member in the SCUZSAMP data set to disable the SVC intercepts. The Cloud Tape Connector started task can then be started again.

Managing zIIP processing

This command can be used to enable or disable zIIP processing. The IBM z Systems Integrated Information Processor (zIIP) provides an option to help free-up general computing capacity in your processor. If your processor is zIIP-enabled, Cloud Tape Connector can take advantage of this feature.

zIIP processing will be set based on the presence of the NOZIIP DD in the CUZCLOUD started task JCL procedure. If the DD is not present or is commented out, zIIP processing will be active by default. To enable or disable zIIP processing without having to restart the started task, issue one of the following commands. Message CUZ0121I is issued in response to this command to indicate the current state of the zIIP processing option.

The command syntax is:

```
F CUZCLOUD,ZIIP [ENABLE|DISABLE]
```

Keywords

ZIIP - Cloud Tape Connector is to modify zIIP status.

Parameters

ENABLE - Enable the zIIP processing option.

DISABLE - Disable the zIIP processing option.

Example of command

This example shows how to enable the zIIP processing option:

```
F CUZCLOUD,ZIIP ENABLE
```

Using the History console command

"History" data sets are either no longer actively used or are still in use, but have never been backed up to the cloud. If you need to start processing history files immediately, you can use the EXECUTE=HISTORY console command to start that process prior to a scheduled backup.

Normally history files are processed at a regular, defined interval, based on the value you specify in the **Auto Backup Repository Minutes** field listed on the **General Options** on the Parmlib Option Main Menu. This parameter not only initiates an automatic backup of the repository, but will process or re-process the history "include" list. However, if you want to begin the history processing immediately, without having to wait for the next scheduled interval for a backup, you can specify the following console command:

```
F CUZCLOUD,EXECUTE=HISTORY
```

For more information on history data sets, refer to [Chapter 8, "Backing up existing data sets,"](#) on page 97. For a list of available console commands, refer to [Appendix A, "Appendix A: List of Console Commands,"](#) on page 375.

Working with cloud data sets

After Cloud Tape Connector has created a backup on a cloud server, you can manage the data sets that are saved to the cloud. When you specify selection criteria, a list of data sets matching that criteria displays. You can then restore data sets from the cloud to tape or disk (depending upon the blocksize), and/or delete data sets from the cloud. This section will describe different ways in which to work with cloud data sets.

Specifying selection criteria

When you choose Option 3, **Cloud Datasets**, on the Main Menu, the **Enter Dataset Selection Criteria** window displays. You can use the default values or type different values to display a list of data sets currently saved on the cloud that match your criteria.

```

CUZ$MAIN V1R1 ----- IBM Cloud Tape Connector for z/OS -----
Option ==> 3

2016/07/01 11:23:17
User: MYUSERID - CUZ

-----

Cloud Connector Started Task Status: Active

-----

1. +----- Enter Dataset Selection Criteria -----+
   |
2. | Dataset Like *
   | Cloud Like *
3. |
4. |
   +-----+
X. Exit

```

Figure 34. Enter Dataset Selection Criteria window

The fields on this window are described below.

Dataset Like

Specify a name or part of a name for the data set you are trying to select. The default is an asterisk (*), which means all.

Cloud Like

Specify a name or part of a name for the cloud you are trying to select. The default is an asterisk (*), which means all.

After specifying selection criteria, the **Cloud Dataset Display** screen displays the list of data sets that match the selection criteria. Line commands enable you to delete a data set or restore a data set. You can also change the selection criteria to display a different group of cloud data sets on this screen.

```

CUZ$CDSN V1R1 ----- Cloud Dataset Display ----- 2016/05/22 12:20:41
Option ==> Scroll ==> PAGE
Primary Commands: REFresh display REStore all in display
Line Commands : R - Restore D - Delete X -Exclude
-----Row 1 of 12
Dataset Like *
Cloud Like *
Total Bytes 15,062,946,941 14 Gigabytes
-----
Cmd Dataset Name Backup Timestamp
- DB2PROD.ARCHLOG1.D16020.T1732224.B00000617 2016-05-20 22:32:22
- DB2PROD.ARCHLOG1.D16020.T1732224.A00000617 2016-05-20 22:32:27
- DB2PROD.ARCHLOG2.D16020.T1732224.B00000617 2016-05-20 22:32:28
- DB2PROD.ARCHLOG2.D16020.T1732224.A00000617 2016-05-20 22:32:33
- DB2PROD.ARCHLOG1.D16021.T1412347.B00000618 2016-05-21 19:12:35
- DB2PROD.ARCHLOG1.D16021.T1412347.B0000069 2016-05-21 19:12:35
- DB2PROD.ARCHLOG1.D16021.T1412347.A00000618 2016-05-21 19:12:40
- DB2PROD.ARCHLOG1.D16021.T1412347.A0000069 2016-05-21 19:12:40
- DB2PROD.ARCHLOG2.D16021.T1412347.B00000618 2016-05-21 19:12:40
- DB2PROD.ARCHLOG2.D16021.T1412347.B0000069 2016-05-21 19:12:40
- DB2PROD.ARCHLOG2.D16021.T1412347.A00000618 2016-05-21 19:12:45
- DB2PROD.ARCHLOG2.D16021.T1412347.A0000069 2016-05-21 19:12:46

```

Figure 35. Example of Cloud Dataset Display screen

The "Total Bytes" amounts display underneath the **Cloud Like** field. These numbers indicate the amount of space used by all of the data sets that match the selection criteria.

The information in each column of the Cloud Dataset Display screen is described below, but cannot be changed. You can scroll to the right or left to see additional data that is not displayed on the initial screen.

Dataset Name

The fully qualified z/OS data set name, up to 44 characters in length, that is being stored on the cloud.

Backup Timestamp

The date and time that the backup was created. The date is in the format of YYYY-MM-DD. The time format is HH:MM:SS.

Jobname

The one- to eight-character name of the backup job.

Stepname

The one- to eight-character name of the step name of the backup job.

Job Nbr

The one- to eight-character job number assigned to the job.

DD Name

The six- to eight-character DD name.

Esot Unit

The one- to eight-character name of the esoteric unit.

Recfm

The type of record format being used. Values include:

- F: Fixed
- FB: Fixed block
- FBA: Fixed-block architecture
- FBS: Fixed-block, spanned
- V: Variable-length, unblocked
- VB: Variable-length, blocked
- VS: Variable-length, unblocked, spanned
- VBS: Variable-length, blocked, spanned
- U: Undefined

Dsorg

The type of data set organization. Possible values include:

- PS: Physical sequential
- DA: Direct access

Lrecl

The length, in bytes, of each record in the data set.

Blksize

The maximum length, in bytes, of a data block.

Byte Count

The maximum number of bytes for each block.

Block Count

The maximum number of blocks allowed.

Cloud Name

The name of the cloud where the data set is stored.

Cloud Data Set Name

The fully qualified cloud data set name, up to 44 characters, that is assigned when the data set is stored in the cloud. The Cloud DSN may be different than the z/OS DSN due to multiple generations of the same DSN being stored on the cloud.

Entering commands

On the Cloud Dataset Display, you can enter line commands to the left side of a data set name in the Cmd column, or you can enter primary commands on the command line (Option line).

Line Commands

You can enter a line command next to one or more data sets. A window will appear asking you to confirm the command for each data set.

Note: Before processing any of the line commands, Cloud Tape Connector will verify that you have the authority to perform the function. If you do not have the correct SAF authority, the line command will not be processed and an error message will display.

Valid line commands are:

- **R** - Restore (copy) this data set from cloud to tape or DASD. Use this line command when you have only a few data sets to restore from the cloud or when you want to restore a data set that is not the most current generation of the data set (the 0 generation).
- **D** - Delete a data set from the cloud.
- **X** - Exclude this data set. This line command is used in combination with the RESTORE primary command, which restores the most current generation (the 0 generation) of every data set from the list that is displayed on the screen. By using data set name masking, you can selectively identify the list of data sets to be restored. However, you can further exclude specific data sets that also match the mask by using the Exclude line command (X) to select the data sets you want to exclude from RESTORE processing.

Excluded data sets are removed from the list displayed on the screen, but remain in the cloud. Type the REFRESH primary command, as described below, on the Option line to show the excluded data set in the list again.

Primary Commands

The primary commands are useful when you want to process a large group of data sets. Valid primary commands are:

- REFRESH or REF

Use this command to re-create the list of data sets. This is especially helpful if you previously used the "X" (Exclude) line command to remove a data set from the list displayed on the screen. The REFRESH primary command will cause the previously excluded data set to appear in the list again.

- RESTORE or RES

Use this command to restore the most current generation (0 generation) of all data sets in the current display list. For example, if a data set has 5 generations in the cloud, only the most current generation will be restored. If you want to restore the previous generation (the "-1" generation), this can be accomplished with the "R" (Restore) line command or by excluding the other generations on the screen and then issuing the RESTORE command.

Use the "X" (Exclude) line command, prior to entering the RESTORE command, to remove data sets from the list that you do not want restored. You do not have to manually Exclude the older generations.

Restoring data from cloud storage

Cloud Tape Connector restores data from cloud storage locations through explicit requests. The product provides three mechanisms to initiate these requests:

- **Cloud Connector ISPF Interface (Main Menu).** This is the recommended way to request data set restores. From the Main Menu, you can choose several Restore options that provide much more function than the Restore Operator command.
- **Restore Operator Command.** This command does not require the Cloud Connector ISPF interface, but it provides limited function.

- **Batch Restore job.** This is an alternate method of restoring data sets from the cloud without using the ISPF interface. This restore method supports the use of wildcard characters or restoring full data sets. The instructions for performing a batch restore process are found in the SCUZSAMP library in member CUZJREST. For more details on how to modify the JCL in this member, refer to [“Sample Job for Batch Restore”](#) on page 109.

From this point and for simplicity, this section assumes the recommended method, Cloud Connector ISPF Interface, is used to initiate Restore requests. Also, this section is an overview of Restore processing. Refer to [“Restore commands”](#) on page 84 for specific details.

Restore commands

When you choose Option 3, **Cloud Datasets**, on the Main Menu and specify selection criteria, a list of backup data sets appears on the **Cloud Dataset Display** screen. You can restore one data set, several data sets, or all of the data sets.

The Restore "R" *line* command and the RESTORE (or RES) *primary* command can be used to restore data sets. These commands are described in more detail below.

Using line commands

To restore individual data sets from the cloud, type "R" in the Cmd column next to the data set you want to restore. A confirmation window appears. The name and characteristics of the data set that you selected are listed in the confirmation window. An example is shown below.

```
CUZ$CDSN V1R1      ----- Cloud Dataset Display ----- 2016/02/04 15:22:19
Option ===>                               Scroll ==> PAGE

Line Commands: R - Restore V - View D - Delete

-----
Dataset Like *
+----- Restore Dataset Confirmation -----+
|
| Dataset Name      PRODTEST.ARCHLOG1.D16020.T1732224.B0000617
| Backup Timestamp  2016-07-20 22:32:22
| Restore Unit Type D (Tape / Disk)
| Restore DASD Unit SYSALLDA (Restore to DASD Esoteric Unit)
| Restore TAPE Unit TAPE (Restore to Tape Esoteric Unit)
| Restore Vol Count 005 (1 to 256)
| SMS Storage Class STORCLAS (Restore to SMS Storage Class)
| Retention Period  0005 (# days to retain restored DSN)
| Delete Old Backup N (Delete old backup DSN ?)
| Roll into GDG Base C (Yes/No/Conditional)
| Restore to Alias  CUZRESTR (Blank will Restore to Orig Name)
| Restore to Dataset -----
|
| Restore Dataset  N (Yes / No / All)
|
| PF12: Cancel
+-----
```

The values shown in the fields on this screen are the restore option values in the Parmlib. All of the values can be overridden at this time.

If you type an "R" on two or more Cmd lines on the Cloud Dataset Display screen, a confirmation window displays for each data set to verify that you want to restore it. If you typed an "R" in the Cmd line of a large number of data sets and are certain that you want to restore all of them, type "A" in the Restore Dataset field to restore them all.

Note: Before processing any of the "R" line commands, Cloud Tape Connector will verify that you have the authority to restore the data sets. If you do not have the correct SAF authority, an error message will display.

The fields on this screen are described below.

Restore Unit Type

Specify one of the following values in this field:

- T -- Tape
- D -- Disk/DASD

If you specify a "D" and the block size is too large to store on DASD, an error message will display to inform you that this data set must be restored to tape. It is required that you specify a value in the **Restore DASD Unit** field and the **Restore TAPE Unit** field in the event that a data set is too large to restore to DASD.

Restore DASD Unit

Specify a one- to eight-character name for the DASD device where the cloud data set will be placed. The default value is the Parmlib value. To send a data set to DASD, you must have a value of "D" in the **Restore Unit Type** field. Also note the following:

- If the block size (blksize) is less than 32K, the cloud data set will be sent to DASD.
- If the block size is greater than 32K, the restored data set must be sent to tape rather than DASD.

Restore Tape Unit

Specify a one- to eight-character name for the tape device where the cloud data set will be placed. The default value is the Parmlib value.

If the block size (blksize) is greater than 32K, the restored data set must be sent to tape rather than DASD. For this reason, you must specify a value in the **Restore TAPE Unit** field even if you specify a "D" in the **Restore Unit Type** field.

Restore Vol Count

Type a number between 1 and 256 for the number of volumes needed to restore a multi-volume data set from the cloud. The default value is the Parmlib value.

SMS Storage Class (Optional)

Specify a one- to eight-character name for the storage class. This is an optional field.

Retention Period

Enter a number between 0 - 9999 to indicate the maximum number of days you want to retain the restored data set on tape. This Retention Period is a z/OS retention period which will be used by your Tape Management System for releasing the tape back to the scratch pool.

Delete Old Backup?

Indicate whether to delete (Y) or keep (N) the local copy of the data set before restoring the data set from the cloud. The default is "N". If you are restoring a data set from the cloud to a data set that already exists, the old data set must be deleted before the restore can occur. A message will appear if the "restore to" data set already exists and **Delete Old Backup** is set to "N".

Roll into GDG Base?

Indicate whether to modify the generation data group (GDG) index. Valid values are:

- Yes
- No
- Conditional (This is the default.)

See [“Restoring SMS-Managed GDG's” on page 75](#) for more detailed information on these values and how they are applied to SMS DASD data sets and SMS Tape data sets.

Restore to Alias

If you want to restore the data set to a different name, enter the name of a high-level alias to be used when restoring the data set. If you prefer to use the original data set name, rather than an alias, leave this field blank. This field is mutually exclusive with **Restore to Dataset**.

Restore to Dataset

Type the name of the data set where you want to place the restored data. If you prefer to use the original data set name, leave this field blank. This field is mutually exclusive with **Restore to Alias**. If you want to restore to the original data set, both "Restore to Alias" and "Restore to Dataset" must be blank.

Restore Dataset

Indicate whether you want to restore this data set:

- **Y** - Yes
- **N** - No (Default)
- **A** - All

Using primary commands

When you type the RESTORE (or RES) *primary* command in the Option line at the top of the screen, all of the most current versions (0 generation) of the data sets currently listed in the Cloud Dataset Display screen are selected for restoration. You can remove data sets from the list on the screen prior to issuing the RESTORE command by using the "X" line command to mark a data set as Excluded. Using the RESTORE primary command eliminates the need to confirm the restore for each individual data set.

The RESTORE primary command only restores the most recent version of the data set. If multiple versions of the same data set are selected, older versions are ignored unless you specifically select them with the "R" line command or exclude the other generations.

Also the RESTORE primary command does not support the **Restore to Dataset** field. The options on this screen (below) are applied to all of the data sets displayed in the list. Thus, restoring all data sets to the same "Restore to Dataset" is not supported.

When you use this command, a window displays that is similar to the one for the "R" line command. For field descriptions, refer to ["Using line commands" on page 84](#).

```
CUZ$CDSN V1R1      ----- Cloud Dataset Display ----- 2016/06/22 10:44:20
Option ==> RES                                           Scroll ==> PAGE

Primary Commands: REFresh display  REStore all in display
Line Commands   : R - Restore  D - Delete  X -Exclude
+----- Restore All Datasets Confirmation -----+
|
| Restore All Datasets - This option will restore the latest
| generation of all datasets in the list. To remove datasets
| from this list, exit out and Exclude them prior to setting
| Restore All Datasets to Yes.
|
| Restore Unit Type      D      (Tape / Disk)
| Restore DASD Unit      SYSALLDA  (Restore to DASD Esoteric Unit)
| Restore TAPE Unit      TAPE      (Restore to Tape Esoteric Unit)
| Restore VOL count      10        (1 to 256)
| SMS Storage Class      STORCLAS  (Restore to SMS Storage Class)
| Retention Period       0005      (# days to retain restored DSN)
| Delete Old Backup      N         (Delete old backup DSN ?)
| Roll into GDG Base     N         (Yes/No/Conditional)
| Restore to Alias       CUZRESTR   (Blank will Restore to Orig Name)
|
| Restore All Datasets In List  N  (Yes / No)
|
| PF12: Cancel
+-----+

```

Figure 36. Restore All Datasets Confirmation window

You can specify the restore settings for the data sets you want to process. The restore options you choose will be applied to all data sets that are restored when using the RESTORE primary command. You are also given the opportunity to confirm that you want to restore all of the data sets in the list.

To cancel this RESTORE process and leave this window, press PF12.

Deleting data from cloud storage

When you choose Option 3, **Cloud Datasets**, on the Main Menu and specify selection criteria, a list of backup data sets appears on the **Cloud Dataset Display** screen. To delete one of these data sets from the cloud, type "D" in the Cmd column next to the data set you want to delete. A confirmation window appears for you to indicate whether you want to delete this data set or keep it.

The name of the data set that you selected for deletion is listed in the confirmation window. Indicate whether you want to delete this backup data set by typing "Y" (yes) or "N" (no) in the **Delete Cloud Backup** field. The default is N. An example is shown below.

```
CUZ$CDSN V1R1      ----- Cloud Dataset Display ----- 2076/02/04 15:27:12
Option ===>                               Scroll ===> PAGE
Primary Commands: REFresh display  REStore all in display
Line Commands: R - Restore V - View  D - Delete

-----
Dataset Like *
+----- Delete Cloud Backup Confirmation -----+
|
| Dataset Name      PRODTST.ARCHLOG1.D16250.T1234567.B0000617
| Backup Timestamp  2076-07-20 22:32:22
|
| Delete Cloud Backup N (Yes / No / All)
|
| PF12: Cancel
+-----+
|
| PROD001.ARCHLOG1.D16310.T1234567.B0000618    2076-07-21 19:12:35
| PRODTST.ARCHLOG1.D16310.T1234567.B000069      2076-07-21 19:12:35
| PROD001.ARCHLOG1.D16310.T1234567.A0000618    2076-07-21 19:12:40
| PRODTST.ARCHLOG1.D16310.T1234567.A000069      2076-07-21 19:12:40
| PROD001.ARCHLOG2.D16310.T1234567.B0000618    2076-07-21 19:12:40
| PRODTST.ARCHLOG2.D16310.T1234567.B000069      2076-07-21 19:12:40
| PROD001.ARCHLOG2.D16310.T1234567.A0000618    2076-07-21 19:12:45
| PRODTST.ARCHLOG2.D16310.T1234567.A000069      2076-07-21 19:12:46
```

Figure 37. Delete Cloud Backup Confirmation window

If you typed a "D" on two or more lines on the Cloud Dataset Display screen, a confirmation window displays for each data set to verify that you want to delete it. If you selected a large number of data sets and are certain that you want to delete all of them, type "A" in the **Delete Cloud Backup** field to delete all of the selected data sets.

Note that you can delete expired backups from the cloud through the use of a batch job. Sample JCL is available in the CUZJEXPR member of the SCUZSAMP library. This batch job will delete all backups from the cloud and z/OS repository where the retention period has expired.

Repository Operations

The Cloud Tape Connector Repository contains information about all of the data sets currently residing in cloud storage locations and is updated whenever a data set is copied to a cloud. The Repository can also be copied to a designated cloud storage location for recovery purposes.

Two parameters, **Cloud Filter Criteria** and **Auto Bkup Repository Min**, determine whether the Repository is copied to a cloud storage location, and if so, how often. The Repository copy stored in the cloud can be used in the event of a disaster recovery event or to recover the Repository in the event of loss.

Creating Repository Backups

To request Repository backup processing, first define **Cloud Filter Criteria** to establish a destination for the backup. Only one Repository Filter Type can be defined. To create a Repository definition, follow these steps:

1. On the Main Menu, choose option 1, **Cloud Connection Settings (Parmlib Options)**.
2. On the Parmlib Options menu, choose option 4, **Backup Filter Criteria**. The **Cloud Filter Display** screen appears.
3. Type the "C" line command to create a cloud filter. The **Cloud Filter Criteria Create** window appears.

```
CUZ$PRMF V1R1      ----- Cloud Filter Display ----- 2016/05/29   18:30:02  
Option ==>                               Scroll ==> PAGE
```

Line Commands: C - Create E - Edit D - Delete

ROW 1 OF 9

```
+-----+  
|----- Cloud Filter Criteria Create -----+  
|  
| Filter Type          R              (Storclass,Esoteric Unit,Dataset,Repository)|  
| Cloud Name           MYCLOUD        (Previously Defined Cloud Name)         |  
| Retention Period 3___             (# Days to Retain on Cloud)               |  
| Filter Criteria      -----+  
|                                                                    PF12: Cancel |  
+-----+
```

***** Bottom of Data *****

Figure 38. Example of Cloud Filter Criteria Create screen with Repository as Filter Type

- . For this cloud definition, use **Filter Type** "R" for Repository.
- . Specify the **Cloud Name** of the cloud storage location to contain the Repository copy. In the example above, the Cloud Name is MYCLOUD.
- . Specify a **Retention Period**. The Retention Period should be set to a value that will retain the cloud copy at least until another backup is taken, as specified by the **Auto Bkup Repository Min** parameter (described below).
- . Do not specify any values in the **Filter Criteria** field because the Filter Type of "R" designates this as a Repository backup filter, so the entire Repository will be saved to this location.

Next, specify a value in the **Auto Bkup Repository Min** parameter to define the Repository backup time period. This value sets the number of minutes to elapse between backups. To specify a value for this parameter, do the following:

- Choose Option 1, **Cloud Connector Settings (Parmlib Options)** on the Main Menu.
- Choose Option 1, **General Options**, on the Parmlib Options menu. The **Parmlib General Options** screen displays.

```
CUZ$PRMG V1R1 ----- Parmlib General Options -----
Option ===>

2016/07/03 12:21:37
User: USERID - CUZ

-----
Parmlib Dataset: PROD.CLOUD.SCUZPARM
Parmlib Member : CUZ#PARM
-----

Abend on Errors . . . . . N (Yes / No)
User Abend Return Code. . . . 01 (01 to 99)
Copy Past History Parm Mbr. . . CUZ#INEX (Parmlib Mbr with INCL/EXCL)
Debug Mode. . . . . J (None / All / Job)
Debug Jobname . . . . . TEST* (Jobname Like)
Write to Operator Msgs. . . . N (Yes / No)
Max Cloud Backup Gens . . . . 1 (1 to 10)
Max Backup History Tasks. . . . 5 (1 to 99)
Auto Bkup Repository Min. . . . 60 (5 to 9999)
Memory Cell Pool Size . . . . 55242880 (2560000 to 99999999)
Memory Primary Cells. . . . . 250 (10 to 9999)
Share Repository across LPARs N (Yes / No)
Write SMF Record Type . . . . 0 (128-255,0 - No SMF Records)
```

Figure 39. Parmlib General Options screen

The **Auto Bkup Repository** Mininterval dictates the frequency of the Repository backup. However, the interval should correlate to actual usage. For example, specifying a value of 60 (minutes) is unnecessary if data is only backed up to the cloud on a weekly basis. Conversely, if cloud backups occur throughout the day, 30 might be a more appropriate value. Refer to [“General Options” on page 24](#) for more details on how to specify these options.

Once the started task is activated with these parameters, the Repository is copied to the cloud storage location designated on the **Cloud Filter Criteria** screen on the interval designated by the **Auto Bkup Repository Min** parameter. The Repository backup is stored in the cloud as an IDCAMS REPRO copy of the VSAM data set. Once retrieved from the cloud, it can be used as input to reconstitute the damaged or missing VSAM Repository.

Using console commands for repository backups

If you want to start a repository backup immediately, without having to wait for the next interval, you can use a console command to do that. You can also restart the staging process of orphaned files that were not backed up previously. Before issuing a console command, verify that the started task, CUZCLOUD, is active.

The following console command will start a repository backup immediately:

```
/F CUZCLOUD,EXECUTE=BACKUP
```

The following command will restart the staging process of orphaned files:

```
/F CUZCLOUD,EXECUTE=RESTART
```

Restoring a Repository

There are three ways to restore the Repository. The first two methods require that the Repository currently exists and is accessible via the CUZCLOUD address space.

Restore Repository with ISPF Interface and the "R" line command

If the VSAM Repository is still available and accessible via the ISPF Interface (Option 3, **Cloud Datasets**, on the Main Menu), use the "R" line command to restore the Repository backup from the displayed list. You must specify either a new name or new alias for the restored data set, because the Repository backup data set name is the VSAM Repository.

Restore processing retrieves the IDCAMS REPRO copy of the VSAM Repository from its cloud storage location and recreates it on either Tape or DASD. Define a new VSAM Repository and use the restored data set as input to an IDCAMS REPRO process to repopulate the VSAM Repository. The ISPF Interface restores to either Tape or DASD, depending upon the storage media selected on the Restore panel. Refer to ["Restore commands"](#) on page 84 for more information.

Restore Repository with Operator command

If the VSAM Repository is still available, but the ISPF Interface is *not* accessible, use the operator command to restore the Repository backup, specifying a new name so as not to conflict with the existing VSAM Repository. For example, enter the following command:

```
F CUZCLOUD,RESTORE DSN=PROD.SCUZCLOUD NEWNAME=PROD.SCUZCLOUD.REPRO
```

Restore processing retrieves the IDCAMS REPRO copy of the VSAM Repository from its cloud storage location and recreates it on either Tape or DASD. Define a new VSAM Repository and use the restored data set as input to an IDCAMS REPRO process to repopulate the VSAM Repository. The operator command restores to the storage media recorded in the repository record, DASD.

Stand-Alone Repository Restore

If the VSAM Repository is unavailable or empty, follow the Stand-Alone Repository Restore steps below to rebuild the repository.

The Stand-Alone Repository Restore is typically used during a disaster recovery event. In this situation, the VSAM Repository is empty and the only copy of it resides in its cloud storage location. Follow these steps to restore and repopulate the VSAM Repository. These steps assume you have configured the Cloud Tape Connector started task.

1. Confirm that the CUZ#REST Samplib member is the same as the CUZ#PARM member, except for Filter Criteria. CUZ#REST must not have any Filter Criteria specified in the CLOUD_BACKUP_UTILITY keyword. This prevents CUZCLOUD from taking new backups while it is being restored.
2. Modify the CUZCLOUD started task JCL by changing CUZ#PARM to CUZ#REST.
3. Define a new VSAM Repository data set using the name specified in the CUZCLOUD started task. It should be empty as this process will populate it with data from the Repository cloud backup.
4. Start the CUZCLOUD started task with parameter member CUZ#REST. CUZCLOUD will be running in an idle state with no Filter Criteria dictating any backup processing.
5. Issue the RESTORE CLD operator command, specifying the Cloud List name of the VSAM Repository, in the following format:

```
stageinghlq.systemname.CUZCLOUD.LIST
```

An example is shown below:

```
F CUZCLOUD,RESTORE CLD=MYCLOUD CLDLIST=CUZSTAGE.SYSA.CUZCLOUD.LIST
```

This command requires two parameters, the cloud name specified in the Repository Filter Criteria and the Cloud List name. The Cloud List specifies a fixed format file name for the backup copy of the Repository data set stored in the cloud. It contains a list of the VSAM Repository backup entries in the cloud. It retrieves the most recent Repository backup entry from this list and uses that information to retrieve the Repository copy from its cloud storage location.

The Cloud List file name is constructed using the Staging alias as a high-level qualifier, the system name as a second-level qualifier, and CUZCLOUD.LIST for the suffix. The list is replaced each time the Repository is copied to the cloud.

Messages CUZS102I and CUZS056I record the Cloud name and Cloud List name at the end of each copy process.

```
CUZS102I-CUZ#REPM-Cloud Connector Repository List staged to
CUZSTAGE.SYSA.CUZCLOUD.LIST
```

```
CUZS056I-CUZ#REPM-Cloud Connector Repository being backed up to Cloud MYCLOUD
```

After invoking the command, the CUZCLOUD address space populates the VSAM Repository. After completing the process, access the ISPF Interface to view the repository contents. If you wish to begin cloud backup processing at the disaster recovery site, shutdown the address space, modify the CUZCLOUD started task JCL to point to the original CUZ#PARM, and restart it. Otherwise, continue processing using the restore-based parameter member, CUZ#REST.

Messages CUZS102I and CUZS056I are uniquely named so that they may be recorded via any message automation process. The information contained in these two messages is critical to the stand alone restore process.

Catalog Support

When using catalog support within Cloud Tape Connector, the original data set will be deleted from DASD or uncataloged from tape, and re-cataloged with a volume serial of CLOUD after the data set has been successfully written to the cloud.

Backup Filter Criteria

A field, **Catalog to Cloud**, is available on the Filter Criteria screens. The default is "N" (No). To turn on cataloging to the cloud, choose to edit (E) the cloud filter and type "Y" in the **Catalog to Cloud** field.

An example of the "Cloud Filter Criteria Edit" screen is shown below, which includes the **Catalog to Cloud** field. For more information on specifying filter criteria, see ["Backup Filter Criteria" on page 48](#).

Figure 40. Cloud Filter Criteria Edit window

Assuming you used the entries listed in the above sample screen, the list of data sets matching the filter criteria could be similar to the example screen below:

Figure 41. Example of Cloud Filter Display screen

An example of the Include/Exclude History screen with the "Catalog Cloud" (Catlg Cloud) column is shown below. For more information on backing up history data sets, see ["Working with history data sets"](#) on page 97.

```

CUZ$INEX V1R1      ----- Include / Exclude History ----- 2016/06/01 13:12:48
Option ==>
Scroll ==> PAGE

Line Commands: I - Insert  D - Delete  R - Repeat
-----
In/Ex Dataset: PROD.CUZ0110.SCUZSAMP
In/Ex Member : CUZ#INEX
-----
Row 1 of 4
-----
Cmd Inc Dataset Filter Cloud Retn Catlg
  Exc Name Perd Cloud
- I PROD12.BKPDATA.* CITYABC 0090 Y
- I CLDCON.DRDATA.* INHOUSE 0120 N
- E PROD12.BKPDATA.TEST
- E CLDCON.DRDATA.TEST
***** Bottom of Data *****

```

Figure 42. Example of Include/Exclude History screen with one data set selected for cataloging to the cloud

Processing changes with batch job

An alternative to copying data to the cloud using the history "include" list is to tailor the sample JCL member, CUZJINCL, in the SCUZSAMP library and run it as a batch job. This batch job requires you to manually enter the include and exclude history data set masks, cloud name, retention period and catalog to cloud status for files to be copied to the cloud. When you run this job, the include history list is processed immediately.

As an example, part of the sample JCL in CUZJINCL is shown below:

```

CLOUD_NAME      FTP      -\* Copy to this Cloud*\
RETENTION_PERIOD 0        -\* Retain Cloud Days *\
CATALOG_TO_CLOUD YES     -\* Catalog DSN Cloud *\
INCLUDE(         -\* Include the following DSNs *\
  PDUSER.JCLLIB-
)
EXCLUDE(         -\* Exclude the following DSNs *\
)

```

For more information on the CUZJINCL sample library member, see [“Sample Job to Include/Exclude Data Sets” on page 108](#).

Running the batch job is a one-time include process. If you need data sets to be copied to the cloud regularly, they must be specified in the history include list member, CUZ#INEX.

Running and verifying cataloging of data sets to the cloud

After updating the Parmlib Options with filter criteria cataloging options and history include options, do the following:

- Recycle the CUZCLOUD started tasks.
- Run your normal batch jobs.

Several messages will appear in your jobs and in the SYSLOG stating that the data sets have been re-cataloged to the cloud. An example of these messages is shown below:

```

06.50.49 RS04 STC00030 CUZS200I CUZ#RCAT-DSN CUZTEST.CUZ.EXPRES01.SHIST000.P00000.
R200020 has been successfully Re-Cataloged to Cloud

06.50.49 RS04 STC00030 CUZ0101I-CUZSTCPY-Stage to copy process completed
successfully

```

If you use the ISPF DSLIST option or option 3.4 to check the job, you will see that the data sets now have "CLOUD" as the Volume Serial (VOLSER). An example of this screen is shown below.


```

Command ===> _____ Scroll ===> CSR
Command - Enter "/" to select action      Message      Volume
-----
.
.
.
ABCSTAGE                                     ?ALIAS
ABCTEST.ABC.EXPRES01.SCUST000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SDIST000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SHIST000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SITEM000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SNORD000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SODLN000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SORDR000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SSTCK000.P00000.R200020 CLOUD
ABCTEST.ABC.EXPRES01.SWARE000.P00000.R200020 CLOUD

```

Figure 43. Sample ISPF DSLIST screen with CLOUD in the Volume column

At this point, Cloud Tape Connector is backing up data sets and re-cataloging them. When the data set is allocated, it will be restored back from the cloud and re-cataloged on a z/OS volume. The data sets in the example above are just a few of the data sets that were the result of a DB2 image copy of an entire database. When you run a DB2 restore job, these data sets will be brought back.

When each data set is allocated, you will see the following messages in your job and the SYSLOG:

```

06.58.05 RS04 JOB00040 CUZS187I CUZSVH26-Jobname DB2#RST .RECOVER-Cloud Dataset
CUZTEST.CUZ.EXPRES01.SSTCK000.P00000.R200020 Scheduled for Restore

06.58.05 RS04 JOB00040 CUZS169I CUZ#API-Func R-DSN
CUZTEST.CUZ.EXPRES01.SSTCK000.P00000.R200020 Scheduled for Restore from Cloud

06.58.05 RS04 JOB00040 CUZS084I CUZ#XMRR Scheduling restore,
DSN=CUZTEST.CUZ.EXPRES01.SSTCK000.P00000.R200020

06.58.05 RS04 STC00030 CUZS116I CUZ#RTDR-Attaching restore for
DSN(CUZTEST.CUZ.EXPRES01.SSTCK000.P00000.R200020)

```

When you refresh the DSLIST, you will see that the data sets have been restored back to z/OS. The Volume is no longer "CLOUD".

```

Command ===> _____ Scroll ===> CSR
Command - Enter "/" to select action      Message      Volume
-----
.
.
.
ABCSTAGE                                     ?ALIAS
ABCTEST.ABC.EXPRES01.SCUST000.P00000.R200020 STAG06
ABCTEST.ABC.EXPRES01.SDIST000.P00000.R200020 STAG08
ABCTEST.ABC.EXPRES01.SHIST000.P00000.R200020 STAG07
ABCTEST.ABC.EXPRES01.SITEM000.P00000.R200020 STAG08
ABCTEST.ABC.EXPRES01.SNORD000.P00000.R200020 STAG02
ABCTEST.ABC.EXPRES01.SODLN000.P00000.R200020 STAG10
ABCTEST.ABC.EXPRES01.SORDR000.P00000.R200020 STAG02
ABCTEST.ABC.EXPRES01.SSTCK000.P00000.R200020 STAG06
ABCTEST.ABC.EXPRES01.SWARE000.P00000.R200020 STAG09

```

Figure 44. Sample ISPF DSLIST screen without CLOUD in the Volume column

After a data set has been restored to z/OS, it will not be re-cataloged back to the cloud. The backup still resides on the cloud, but is not cataloged to the cloud.

If you ran another DB2 image copy job, you would see that the new data sets are re-cataloged to the cloud and the older restored data sets are not.

Command ===>	Scroll ===> CSR
Command - Enter "/" to select action	Message Volume
ABCTEST.ABC.EXPRES01.SCUST000.P00000.R200020	STAG06
ABCTEST.ABC.EXPRES01.SCUST000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SDIST000.P00000.R200020	STAG08
ABCTEST.ABC.EXPRES01.SDIST000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SHIST000.P00000.R200020	STAG07
ABCTEST.ABC.EXPRES01.SHIST000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SITEM000.P00000.R200020	STAG08
ABCTEST.ABC.EXPRES01.SITEM000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SNORD000.P00000.R200020	STAG02
ABCTEST.ABC.EXPRES01.SNORD000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SODLN000.P00000.R200020	STAG10
ABCTEST.ABC.EXPRES01.SODLN000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SORDR000.P00000.R200020	STAG02
ABCTEST.ABC.EXPRES01.SORDR000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SSTCK000.P00000.R200020	STAG06
ABCTEST.ABC.EXPRES01.SSTCK000.P00000.R200021	CLOUD
ABCTEST.ABC.EXPRES01.SWARE000.P00000.R200020	STAG09
ABCTEST.ABC.EXPRES01.SWARE000.P00000.R200021	CLOUD

Figure 45. Sample ISPF DSLIST screen with a mix of data sets restored from cloud and new data sets cataloged to the cloud

Expiring data sets

You can run the expire job, CUZ#EXPR, or Samplib (SCUZZSAMP) member CUZJEXPR to generate a report of expired data sets. A new column has been added to the expire report generated in the job. This column indicates whether Cloud Tape Connector un-cataloged the data set from z/OS.

Note: If a data set is expiring, meaning deleted from the cloud and the data set has been re-cataloged to the cloud, Cloud Tape Connector will un-catalog the data set from z/OS.

An example of the "Expired Cloud Backups" report is shown below. The data set names have been shortened and the number of lines reduced to ensure the report is easy to read.

12017-10-05 11:16:37	z/OS Cloud Connector for Tape			
Expired Cloud Backups				
Dataset Name Uncataloged	Cloud Dataset Name	Cloud Name		
-----	-----	-----		
ABCTEST.EXPRES01.SSTCK000.R200020	ABCSTAGE.DB2#IC.JOB00039.SYS00001.R3102658	FTP		NO
ABCTEST.EXPRES01.SCUST000.R200020	ABCSTAGE.DB2#IC.JOB00039.SYS00005.R9556793	FTP		NO
ABCTEST.EXPRES01.SHIST000.R200020	ABCSTAGE.DB2#IC.JOB00039.SYS00007.R1831880	FTP		NO
ABCTEST.EXPRES01.SORDR000.R200020	ABCSTAGE.DB2#IC.JOB00039.SYS00009.R4356680	FTP		NO
.				
.				
.				
ABCTEST.EXPRES01.SSTCK000.R200021	ABCSTAGE.DB2#IC.JOB00041.SYS00001.R5208251	FTP		Yes
ABCTEST.EXPRES01.SCUST000.R200021	ABCSTAGE.DB2#IC.JOB00041.SYS00005.R9511233	FTP		Yes
ABCTEST.EXPRES01.SHIST000.R200021	ABCSTAGE.DB2#IC.JOB00041.SYS00007.R5508555	FTP		Yes
ABCTEST.EXPRES01.SORDR000.R200021	ABCSTAGE.DB2#IC.JOB00041.SYS00009.R3186122	FTP		Yes
.				
.				
.				

Figure 46. Sample report for Expired Cloud Backups

In the example above, only data sets cataloged with *volser*=CLOUD were uncataloged from z/OS.

When you refresh the DSLIST, the data sets will no longer show "CLOUD" under the Volume column. An example screen is shown below.

Command	Message	Volume
====>	Scroll	====> CSR
Command - Enter "/" to select action		
.		
.		
.		
ABCSTAGE		?ALIAS
ABCTEST.ABC.EXPRES01.SCUST000.P00000.R200020		STAG06
ABCTEST.ABC.EXPRES01.SDIST000.P00000.R200020		STAG08
ABCTEST.ABC.EXPRES01.SHIST000.P00000.R200020		STAG07
ABCTEST.ABC.EXPRES01.SITEM000.P00000.R200020		STAG08
ABCTEST.ABC.EXPRES01.SNORD000.P00000.R200020		STAG02
ABCTEST.ABC.EXPRES01.SODLN000.P00000.R200020		STAG10
ABCTEST.ABC.EXPRES01.SORDR000.P00000.R200020		STAG02
ABCTEST.ABC.EXPRES01.SSTCK000.P00000.R200020		STAG06
ABCTEST.ABC.EXPRES01.SWARE000.P00000.R200020		STAG09

Figure 47. Sample ISPF DSLIST screen without CLOUD in the Volume column

Chapter 8. Backing up existing data sets

Cloud Tape Connector is most often used to backup data sets that are created frequently. However, it can also be used to back up data sets that are either no longer actively used or that are still in use, but have never been backed up to the cloud. These data sets are referred to as "history" data sets. The **Backup History Datasets** option on the Main Menu enables you to copy and manage these history data sets.

The data sets you add to the "include" history list must have the data set organization of PS (Physical Sequential). Partitioned data sets (PDS and PDSE) and VSAM files are not supported. To copy a VSAM file or partitioned data set to the cloud, you will first need to run a utility program, such as IDCAMS or IEBCOPY, to create a sequential data set, and then have that output data set sent to the cloud, either by filter criteria or through the use of the include history data set list.

Specify Parmlib and Member to Include or Exclude

When you choose Option 4, **Backup History Datasets**, on the Main Menu, a window displays. You must specify the **Parmlib Dataset** and the name of the **Parmlib Member** that you want to use when determining which data sets to include or exclude.

```
CUZ$MAIN V1R1 ----- IBM Cloud Tape Connector for z/OS -----
Option ==> 4

                                     2016/07/25 17:54:56
                                     User: USERID - CUZ
-----

Cloud Connector Started Task Status: Active

-----

+----- Enter History Include/Exclude Member -----+
| In/Ex Dataset  PROD.TEST.SCUZPARM                    |
| In/Ex Member   CUZ#INEX                               |
|                                                        |
|                                                        |
|                                                        |
|                                                        |
| PF12: Cancel                                         |
+-----+
```

Figure 48. Enter History Include/Exclude Member window

To view a list of backup history data sets, you must specify the **Include/Exclude Dataset** and the name of the **Include/Exclude Member**.

In/Ex Dataset

Type the name of the Include/Exclude history data set that holds the Include/Exclude Member. The default data set name is SCUZPARM.

In/Ex Member

Type a one- to eight-character name for the Include/Exclude history member that you want to use when identifying which pre-existing sequential data sets to include or exclude. You can also modify the sample member, CUZ#INEX, for your environment. This is the default member name. The Include/Exclude Member can be in a separate Include/Exclude history data set or reside in the same library as the CUZ#PARM member.

Working with history data sets

When you choose option 4, **Backup History Datasets**, on the Main Menu and specify an Include/Exclude Dataset and an Include/Exclude Member, the **Include/Exclude History** screen appears. On this screen you can create new data set filters, delete existing filters, and specify whether to include or exclude a data set filter when determining which history data sets to copy to the cloud.

One example of how you could use the Include (I) and Exclude (E) values is when you want to copy a large group of data sets to the cloud, but exclude some of the data sets via a mask so they are not copied. In

the example below, the PRODMP.BKPDATA group of data sets is included, but part of that group, PRODMP.BKPDATA.TEST, is excluded.

```
CUZ$INEX V1R1      ----- Include / Exclude History ----- 2016/06/01 13:12:48
Option ==>                               Scroll ==> PAGE

Line Commands: I - Insert  D - Delete  R - Repeat
-----
In/Ex Dataset: PROD.CUZ0110.SCZSAMP
In/Ex Member  : CUZ#INEX
-----
Row 1 of 4
-----
Cmd  Inc  Dataset Filter
  Exc
-   I   PRODMP.BKPDATA.*
-   I   CLDCON.DRDATA.*
-   E   PRODMP.BKPDATA.TEST
-   E   CLDCON.DRDATA.TEST
***** Bottom of Data *****
```

Figure 49. Example of Include/Exclude History screen with large group included and subset group excluded

The fields on the Include/Exclude History screen are described below.

Inc/Exc

Indicate whether to include (I) or exclude (E) the data sets that match this filter when making a backup on the cloud.

Dataset Filter

Type up to 44 characters for the filter criteria. Data set masking is supported for both the include and exclude data sets. Use an asterisk (*) for one or more characters and a percent sign (%) to represent a single character anywhere in the data set name. An example is shown below:

```
MY.TEST%.DATA*.SETS.
```

Cloud Name

Type up to eight characters for the name of the cloud where the filtering criteria applies.

Retn Perd

Type a number between 0 and 9999 for the number of days you want to retain this backup. A value of 9999 means that the backup is kept permanently and does not expire. A value of zero is helpful for testing purposes to ensure your data was moved to the cloud successfully.

Catalog to Cloud

Specify Y or N to indicate whether to catalog a history data set to the cloud. The default is N (no). Type a "Y" for each "include" line you would like to re-catalog to the cloud with a volume serial of CLOUD.

Note: This option is available on "include" (I) lines only.

Line commands

Type one of these line commands in the Cmd column:

- **I** - Insert a line to add a new data set filter.
- **D** - Delete a line (if blank) or delete an existing data set filter.
- **R** - Repeat a line to copy an existing data set filter to a new line.

Saving changes

After you have completed your changes to the history data set include list, press PF3. The **Save/Update Include History Member** window displays, as shown below.

```

CUZ$MAIN V1R1 ----- IBM Cloud Tape Connector for z/OS -----
Option ==> 4

2016/07/25 17:54:56
User: USERID - CUZ

-----

Cloud Connector Started Task Status: Active

-----

+----- Save/Update Include History Member -----+
| In/Ex Dataset  PROD.TEST.SCUZPARM                |
| In/Ex Member   CUZ#INEX                          |
|                                                        |
| Start History Processing in Started Task:  _  (Yes/No) |
|                                                        |
|                                                        | PF12: Cancel
+-----+

```

Figure 50. Save/Update Include History Member

In/Ex Dataset

Verify the data set name where you want the history data to be saved.

In/Ex Member

Verify the member name where you want the history data to be saved.

Start History Processing in Started Task

Specify a Y or N to indicate whether to begin the history processing immediately without having to wait for the next history interval to trigger a backup.

Specifying "Y" in this field is the same as entering this console command:

```
F CUZCLOUD,EXECUTE=HISTORY
```

Note: If the STC (started task) for Cloud Tape Connector, which is CUZCLOUD, is **not** active, the option to "Start History Processing in Started Task" will not appear in this window. The STC must be running for this option to appear.

For more information on console commands that can be used to backup repository files, start history processing, or restart repository processing, refer to ["Using console commands with history data sets"](#) on page 100. For a longer list of available console commands, refer to [Appendix A, "Appendix A: List of Console Commands,"](#) on page 375.

The repository maintenance interval will backup the repository, start history processing and restart orphaned staging files. When this process is running on multiple members on a guest, this repository maintenance will run on **ONLY** one LPAR. The "Start History Processing in Started Task" option will appear in the window if you updated the history list on the same LPAR that's running the repository maintenance.

For example, let's assume that Cloud Tape Connector is running on LPAR10 and LPAR25. However, Repository Maintenance is only running on LPAR10. If you update the history list on LPAR 10 and the STC (started task) for Cloud Tape Connector is up, the "Start History Processing in Started Task" option will appear in the window. If you update the history list on LPAR25, this option will not appear in the window.

If you wish to change the location or name of the member, you must update the **Copy Past History Parm Mbr** parameter in the **Parmlib General Options** of the ISPF interface. For more information, see ["General Options"](#) on page 24.

Important: If you specify "N" in the "Start History Processing in Started Task" field, the data sets specified in the **Copy Past History Parm Mbr** parameter will not be processed until the next interval cycle defined in the **Auto Bkup Repository Min** parameter. This parameter not only initiates an automatic backup of the repository, but will process or re-process the history "include" list. As an example, if the last time the repository was backed up was 15 minutes ago, and you have the auto backup repository minutes value set to 60, it will be 45 more minutes before the include history list is processed again. The default value for **Auto Bkup Repository Min** is 60.

Alternate method for processing history changes with batch job

The easiest method for processing history changes is to specify "Y" in the "Start History Processing in Started Task" field, as described above, which will begin processing the history files immediately, assuming the started task is active.

However, an alternative to copying data to the cloud using the history include/exclude list is to tailor the sample JCL member, CUZJINCL, in the SCUZSAMP library and run it as a batch job. This batch job requires you to manually enter the include and exclude history data set masks, cloud name, retention period and catalog to cloud status for files to be copied to the cloud. As an example, part of the sample JCL in CUZJINCL is shown below:

```
CLOUD_NAME          FTP      -\* Copy to this Cloud*\
RETENTION_PERIOD    0        -\* Retain Cloud Days *\
CATALOG_TO_CLOUD    YES      -\* Catalog DSN Cloud *\
INCLUDE(            -\* Include the following DSNs *\
    PDUSER.JCLLIB-
)
EXCLUDE(            -\* Exclude the following DSNs *\
)
```

For more information on the CUZJINCL sample library member, see [“Sample Job to Include/Exclude Data Sets” on page 108](#).

When you run this job, the include history list is processed immediately, just like the "Start History Processing in Started Task" field. You do not have to wait for the started task to hit the interval specified in the **Auto Bkup Repository Minutes** parameter.

Running the batch job is a one-time include process. If you need data sets to be copied to the cloud regularly, then they must be specified in the history include list member, CUZ#INEX.

Using console commands with history data sets

Some data sets are considered "history" data sets, meaning that they are no longer actively used or have never been backed up to the cloud. Option 4, **Backup History Datasets**, on the Main Menu enables you to copy and manage these history data sets.

The **Auto Backup Repository Minutes** field listed in the **General Options** on the Parmlib Option Main Menu specifies the amount of time (interval) between backups of history files. However, if you want to start a repository backup or perform history processing immediately, you can use console commands to do that.

New console commands have been added to allow you to kick off a variety of actions immediately, without having to wait for the next interval. For a complete list of console commands, refer to [Appendix A, “Appendix A: List of Console Commands,” on page 375](#)

Start Repository Backup

The following command is used to start the repository backup:

```
/F CUZCLOUD,EXECUTE=BACKUP
```

Start History Processing

The following command is used to start processing history data sets:

```
/F CUZCLOUD,EXECUTE=HISTORY
```

Restart Orphaned Staging Files

The following command is used to restart the staging process of orphaned files:

```
/F CUZCLOUD,EXECUTE=RESTART
```


Chapter 9. Active Tasks

The **Active Tasks** option allows you to view the current status of any active tasks. Different colors are used to indicate the status, such as green for "Cloud Copy" activity.

When you choose Option 5, **Active Tasks**, on the Main Menu, a window displays where you can specify selection criteria for the active tasks you want to review. An example is shown below:

```
+----- Active Tasks Selection Criteria -----+
|
|   Jobname Like  *
|   Dataset Like  *
|   Cloud   Like  *
|
+-----+
```

Figure 51. Active Tasks Selection Criteria

The fields on the "Active Tasks Selection Criteria" window are described below.

Jobname Like

Display only the Active Tasks that are like this job name. The default is an asterisk (*), which means all.

Dataset Like

Display only the Active Tasks that are like this data set name. The default is an asterisk (*), which means all.

Cloud Like

Display only the Active Tasks that are like this Cloud Name. The default is an asterisk (*), which means all.

After specifying the selection criteria, the Active Tasks Display screen appears. On this screen, the jobs matching your selection criteria are displayed in a list. If you decide to change the list, the same criteria fields are at the top of the screen so you can adjust the selection criteria as needed.

An example of the screen is shown below without any data.

```
RCC$DISP V1R1      ----- Active Tasks Display ----- 2018/05/01 11:22:58
Option  ===>      Scroll  ===> PAGE

Display Types:  Filter Capture  Cloud Copy  Restore  History Queue  Restore Queue

----- No rows to display -----
Jobname Like  *
Dataset Like  *
Cloud   Like  *
-----
Dataset Name                                     Total Byte Count   Bytes/Sec
```

Figure 52. Active Tasks Display screen with color-coded status at top

The next example shows the Active Tasks Display screen with active data.

RCC\$DISP VIRL ----- Active Tasks Display ----- 2018/05/01 14:56:59
Option ==> Scroll ==> CSR

Display Types: Filter Capture Cloud Copy Restore History Queue Restore Queue

Jobname Like w
Dataset Like w
Cloud Like w

Row 1 of 39 +>

Dataset Name	Total	Byte Count	Bytes/Sec
RCCTEST.RCC.EXPRES01.SSTCK000.P00006.R200003	22,765,716	4,553,143	
RCCTEST.RCC.EXPRES01.SSTCK000.P00017.R200003	22,765,716	4,553,143	
RCCTEST.RCC.EXPRES01.SODLN000.P00003.R200003	40,943,764	13,647,921	
RCCTEST.RCC.EXPRES01.SCUST000.P00002.R200003	148	148	
RCCTEST.RCC.EXPRES01.SODLN000.P00002.R200003	22,507,520	11,253,760	
RCCTEST.RCC.EXPRES01.SODLN000.P00005.R200002	22,536,192	11,268,096	
RCCTEST.RCC.EXPRES01.SCUST000.P00004.R200002	21,274,624	10,637,312	
RCCTEST.RCC.EXPRES01.SSTCK000.P00009.R200002	21,561,344	10,780,672	
RCCTEST.RCC.EXPRES01.SSTCK000.P00021.R200002	22,765,568	11,382,784	
RCCTEST.RCC.EXPRES01.SCUST000.P00001.R200003	974,848	0	
RCCTEST.RCC.EXPRES01.SODLN000.P00006.R200002	0	0	
RCCTEST.RCC.EXPRES01.SODLN000.P00007.R200002	0	0	
RCCTEST.RCC.EXPRES01.SODLN000.P00008.R200002	0	0	
RCCTEST.RCC.EXPRES01.SORDR000.P00001.R200002	0	0	

Figure 53. Active Tasks Display screen with data shown in various colors

The "Active Tasks Display" uses different colors to indicate the status of active tasks:

Filter Capture Tasks (Pink)

Filter Capture tasks are jobs where the data is currently being captured and directly written to the cloud or being staged. They are displayed in pink on the screen.

Cloud Copy Tasks (Green)

Cloud Copy tasks are jobs where a dataset is being copied to the cloud. The tasks can consist of active history tasks, staged data sets being copied, or data sets residing on DASD during filter capture. They are displayed as green on the screen.

Restore Tasks (Yellow)

Restore tasks are data sets currently being restored from the cloud. They are displayed as yellow on the screen.

History Queue Tasks (White)

History Queue tasks are history data sets that are currently queued to be copied to the cloud. These tasks are not currently active copy tasks. They are displayed as white on the screen.

Restore Queue Tasks (Blue)

Restore Queue tasks are restore data sets that are currently queued to be restored from the cloud. These tasks are not currently active restore tasks. They are displayed as blue on the screen.

There are several columns of information about the Active Tasks Display. Page right to see additional columns. All of the columns are described below.

Dataset Name

Lists the fully qualified z/OS data set name, up to 44 characters in length, that is being stored to the cloud or being restored from the cloud.

Total Byte Count

Indicates the number of bytes copied or restored at this time.

Bytes/Sec

Indicates the average number of bytes per second that were transferred to or from the cloud.

Block Count

Indicates the number of blocks copied or restored at this time.

Elapsed Time

Lists the amount of time, in hours, minutes, and seconds, that the active task has been running.

Jobname

Lists the one- to eight-character name of the backup job.

Stepname

Lists the one- to eight-character name of the step that was performed in this job.

Job Nbr

Lists the one- to eight-character job number assigned to the job.

Start Timestamp

Indicates the date and time that the task started. The date is in the format of YYYY-MM-DD. The time format is HH:MM:SS.

Cld Name

Lists the name of the cloud where the data set is stored to or restored from.

DD Name

Lists the six- to eight-character DD name.

Cloud Dataset Name

Specifies the fully-qualified cloud data set name (DSN), up to 44 characters, that is assigned when the data set is stored on the cloud. The Cloud DSN may be different than the z/OS DSN due to multiple generations of the same DSN being stored on the cloud.

Chapter 10. Utilities

The utilities described in this chapter provide additional functionality to maximize your use of Cloud Tape Connector. The purpose of each utility and instructions on how to use the utility are provided.

Tape Compare Utility

In some cases, a data set may be copied multiple times, such as to tape, then to the cloud, then back to tape at a later date. In this situation, you may want to verify that the data set is still the same and no data has been lost through all of the copying processes. The Tape Compare Utility allows you to compare two data sets to ensure that they are the same. Different reports are generated to show if the data sets are the same or where they differ if they are not the same.

The SCUZSAMP library contains sample JCL for a job to compare the contents of two tape data sets in member CUZJCOMP. Run this job to compare two data sets or to analyze a specific data set. Follow the instructions at the top of the CUZJCOMP job to make changes to the job before you submit it.

When you run the compare tape job, it will generate one or both of the following reports.

- Dataset Compare Report -- Results show whether the data sets are the same or different.
- Dataset Analysis Report -- Results provide an analysis of a specific tape showing more basic information, such as tape label, number of blocks and records, and block sizes. If this analysis is performed on one data set, the Dataset Analysis Report only provides this basic type of information. This report is also written as part of the Dataset Compare Report for each tape when two tapes are being compared.

Dataset Compare Report - Same Results

The sample "Dataset Compare Report" below shows how the report would look if the data sets were the same. Notice the last line in the report which states "The datasets compared are equal".

```

                                DATASET COMPARE REPORT

*****
*****
***** FILE COMPARE NUMBER:      1      BELOW 2 FILES WILL BE
COMPARED
*****
*****

DDNAME: DSN1 DSN: TSDEM.GENER.PUTA      FIRST VOLSER: C10057 FILE SEQ
00001
DDNAME: DSN2 DSN: TSDEM.GENER.PUTB      FIRST VOLSER: C10059 FILE SEQ
00001

HDR2 DCB INFORMATION WAS
EQUAL

LAST 17 BYTES OF DSN FROM HDR1 LABELS NOT EQUAL DSN1: TSDEM.GENER.PUTA DSN2:
TSDEM.GENER.PUTB

TWO JFCB DSNs NOT EQUAL DSN1: TSDEM.GENER.PUTA      DSN2:
TSDEM.GENER.PUTB

THE DATASETS COMPARED ARE
EQUAL
```

Figure 54. Dataset Compare Report -- Results show data sets are the same.

Dataset Compare Report - Different

The sample "Dataset Compare Report" below shows how the report would look if the data sets were found to be different. Notice the line toward the bottom of the report which states "The datasets compared are not equal".

```

                                DATASET COMPARE REPORT

*****
*****
***** FILE COMPARE NUMBER:      1      BELOW 2 FILES WILL BE
COMPARED
*****
*****

DDNAME: DSN1 DSN: TSDEM.GENER.PUT3      FIRST VOLSER: C10047 FILE SEQ
00001
DDNAME: DSN2 DSN: TSDEM.GENER.ABN3      FIRST VOLSER: C10045 FILE SEQ
00001

HDR2 DCB INFORMATION WAS
EQUAL

LAST 17 BYTES OF DSN FROM HDR1 LABELS NOT EQUAL DSN1: TSDEM.GENER.PUT3 DSN2:
TSDEM.GENER.ABN3

TWO JFCB DSNs NOT EQUAL DSN1: TSDEM.GENER.PUT3      DSN2:
TSDEM.GENER.ABN3

THE DATASETS COMPARED ARE NOT EQUAL !!!      <----- ALERT
-----
THE LAST BLOCK READ ON EACH DATASET WAS NOT
EQUAL
  DATA IN BLOCK NUMBER:      1  IS DIFFERENT AT BYTE DECIMAL:      168  HEX:
000A8
***** BOTTOM OF DATA
```

Figure 55. Dataset Compare Report -- Results show data sets are different.

Dataset Analysis Report

The sample "Dataset Analysis Report" below shows information about a single data set, such as tape label, number of blocks and records, and block sizes.

```

                                DATASET ANALYSIS
REPORT

DDNAME: DSN1      DEVICE TYPE: 3490      UNIT ADDRESS:
05AE

*****
*****
***** FILE NUMBER:
1
*****
*****

DSN: TSDEM.GENER.PUTA                      FIRST VOLSER: C10057 FILE SEQ
00001

HEADER LABELS FOLLOW. CURRENT VOLSER:
C10057
VOL1C10057

HDR1TSDEM.GENER.PUTA C1005700010001      01612501612700000004IBM OS/VS
370
      DSN LAST 17: TSDEM.GENER.PUTA      1ST VOLSER: C10057      VOLSQ:      1      FILESQ:      1      BLK
COUNT:      0

HDR2F279200008000TSDEMG1/S010      P      B
14F2C      RECFM=FB      BLKSZ=27920      LRECL=
80

***** TAPEMARK ENCOUNTERED
*****

** THE DATASET REACHED EOF ON THIS
VOLUME
BLOCKS DATASET HAD ON THIS VOLUME:
4

***** TAPEMARK ENCOUNTERED
*****

TRAILER LABELS FOLLOW. CURRENT VOLSER:
C10057
EOF1TSDEM.GENER.PUTA C1005700010001      0161250161270000004IBM OS/VS
370
      DSN LAST 17: TSDEM.GENER.PUTA      1ST VOLSER: C10057      VOLSQ:      1      FILESQ:      1      BLK
COUNT:      4

EOF2F279200008000TSDEMG1/S010      P      B
14F2C      RECFM=FB      BLKSZ=27920      LRECL=
80

***** TAPEMARK ENCOUNTERED
*****

      DATASET TOTALS FOR DATASET:
      TSDEM.GENER.PUTA

      NUMBER OF BLOCKS READ:
4
      SIZE OF LARGEST BLOCK:
27920
      SIZE OF SMALLEST BLOCK:
5200
      AVERAGE BLOCK SIZE:
22240
      NUMBER OF LOGICAL RECS READ:
1,112
      NUMBER OF BYTES READ:

```

Sample Job to Include/Exclude Data Sets

If existing data sets need to be copied to the cloud, you can use the CUZZINCL job in the SCUZSAMP library to specify which data sets to include and exclude for the copy process. This same information can be specified by using Option 4, **Backup History Datasets**, on the Main Menu of the ISPF interface.

Note: For information on using the ISPF interface to specify backup history data sets, see [“Working with history data sets”](#) on page 97.

Be sure to review the instructions at the top of the CUZZINCL job. You will need to tailor the JCL before submitting it. An example of the CUZZINCL member is shown below.

```
//JOB CARD JOB      , '?' , REGION=0M
//*****
//* COPYRIGHT ROCKET SOFTWARE, INC. 2017
//* ALL RIGHTS RESERVED.
//*****
//* * * * *
//* Member: CUZZINCL
//*
//* This job will copy existing backups (tape or DASD) to
//* the cloud servers. You may add as many data sets (included /
//* excluded) as you like. Data set wildcarding is also supported.
//*
//* Instructions: Change the steplib to the runtime load library
//*
//* 1. Change #HIGHLVL# to the high level qualifier for your
//* runtime load library
//* 2. Set the cloud server you want the data be copied to by
//* changing #MYCLOUD# to your predefined cloud name
//* ** The cloud name must be a 1 to 8 character cloud name
//* defined in the parmlib member CUZ#PARM
//* 3. Set the Retention Period in nbr of days you want the data
//* to reside on the cloud. Change #RETPD# to a numeric value
//* ** The retention period must be 1 to 4 characters with
//* values between 0 and 9999.
//* 4. Set the Catalog to Cloud option to YES or NO. Setting this
//* value to a YES will delete your original disk dataset or
//* uncatalog your original tape dataset and Re-Catalog it with
//* VOLSER=CLOUD. The dataset will be restored back to z/OS
//* when allocated. Change #CATALOG# to YES or NO.
//* 5. Add data sets to the include / exclude list
//*
//* Examples:
//*
//* 1. Data sets going to a single cloud
//*
//* CLOUD_NAME MYCLOUD1 -\* COPY TO THIS CLOUD
//* RETENTION_PERIOD 30 -\* RETAIN 30 DAYS
//* CATALOG_TO_CLOUD YES -\* Re-Catalog to Cloud
//* INCLUDE ( -\* INCLUDE DATASETS
//* MY.DATASETS.TEST1* -
//* MY.DATASETS.TEST2* -
//* MY.DATASETS.TEST3* -
//* ) -\*
//* EXCLUDE ( -\* EXCLUDE DATASETS
//* MY.DATASETS.TEST1.BKP -
//* ) -\*
//*
//* 2. Data sets going to multiple clouds with Re-Cataloging
//* datasets to MYCLOUD1 only.
//*
//* CLOUD_NAME MYCLOUD1 -\* COPY TO THIS CLOUD
//* RETENTION_PERIOD 30 -\* RETAIN 30 DAYS
//* CATALOG_TO_CLOUD YES -\* Re-Catalog to Cloud
//* INCLUDE ( -\* INCLUDE DATASETS
//* MY.DATASETS.TEST1* -
//* MY.DATASETS.TEST2* -
//* MY.DATASETS.TEST3* -
//* ) -\*
//* EXCLUDE ( -\* EXCLUDE DATASETS
//* MY.DATASETS.TEST1.BKP -
//* ) -\*
//* CLOUD_NAME MYCLOUD2 -\* COPY TO THIS CLOUD
//* RETENTION_PERIOD 30 -\* RETAIN 30 DAYS
//* CATALOG_TO_CLOUD NO -\* Don't Re-Catalog
//* INCLUDE ( -\* INCLUDE DATASETS
//* MY.DATASETS.TEMP1* -
```



```

/*          MY.DATASETS.TEMP2*          -          *
/*          MY.DATASETS.TEMP3*          -          *
/*          )                          -\*          *
/*          EXCLUDE (                  -\* EXCLUDE DATASETS *
/*          )                          -\*          *
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/*CUZ#INEX EXEC PGM=CUZ#INEX,REGION=0M
/*STEPLIB DD DISP=SHR,DSN=#HIGHVL#.SCUZLOAD
/*SYSPRINT DD SYSOUT=*
/*CUZOUT DD SYSOUT=*
/*CUZ#INEX DD *
/*      CLOUD_NAME          #MYCLOUD#      -\* COPY TO THIS CLOUD      *\
/*      RETENTION_PERIOD    #RETPD#        -\* RETAIN X NBR OF DAYS   *\
/*      CATALOG_TO_CLOUD    #CATALOG#      -\* Re-Catalog to Cloud      *\
/*      INCLUDE (           -\* INCLUDE DATASETS          *\
/*          INCLUDE.MY.DATASETS*           -          *
/*      )                               -\*          *
/*      EXCLUDE (           -\* EXCLUDE DATASETS          *\
/*          EXCLUDE.SOME.DATASETS*         -          *
/*      )                               -\*          *
/*
/*
/*

```

Required Changes

The items in bold are the variables you **must** change in the JCL to customize the job for your environment.

#HIGHVL#

Change #HIGHVL# to the high level qualifier for your runtime load library.

#MYCLOUD#

Set the cloud server you want the data be copied to by changing #MYCLOUD# to your predefined cloud name. The cloud name must be a 1 to 8 character cloud name defined in the parmlib member CUZ#PARM.

#CATALOG#

Change #CATALOG# to YES or NO. The default is NO. A value of YES will delete your original disk data set or uncatalog your original tape data set and re-catalog it with VOLSER=CLOUD after a successful copy has been placed on the cloud. When the data set is allocated, it will be restored back from the cloud and re-cataloged on a z/OS volume.

#RETPD#

Change #RETPD# to a numeric value to indicate the number of days you want the data to reside on the cloud. The retention period must be between 0 and 9999 days.

INCLUDE.MY.DATASETS*

Replace this with data set names that you want to include in the job.

EXCLUDE.MY.DATASETS*

Replace this with data set names that you want to exclude from this job.

Sample Job for Batch Restore

The Batch Restore job provides an alternate method for restoring data sets from the cloud without having to use the ISPF interface. The JCL provided in the sample job can be modified for your environment. This job supports the use of wildcard characters and full data set restoration.

The sample job for performing a batch restore process is located in the SCUZSAMP library in member CUZJREST. Be sure to follow the instructions listed in this member. You will need to tailor the JCL before submitting it. An example of member CUZJREST is shown below.

```

//CUZJREST JOB , 'CTC Batch Restore', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID
//*
//*-----*\
//*          *\
//* Cloud Tape Connector Batch Restore - This JCL is an alternate*\
//* method of restoring datasets from the cloud without      *\
//* having to use the ISPF interface. This method of         *\
//* restore supports wild carding or full dataset restores.*\

```

```

/**
/** Required Parameters:
/** 1. INCLUDE -
/** 2. RESTORE_TO_UNIT - 8 Character unit for Restore
/** to Dataset
/** - Can be tape or Disk if
/** Blocksize < 32K
/** - If Blocksize > 32K, unit
/** must be tape device
/**
/** Optional Parameters:
/** 1. RESTORE_TO_DATASET - 44 Character DSN of restored
/** Dataset
/** - Mutually exclusive with
/** RESTORE_TO_ALIAS
/** - Can be used only if 1 Dataset
/** is being restored
/** - To Restore to original DSN,
/** Don't enter parms
/** RESTORE_TO_DATASET or
/** RESTORE_TO_ALIAS
/** 2. RESTORE_TO_ALIAS - 8 Character Alias to be
/** substituted in Restore DSN
/** - Mutually exclusive with
/** RESTORE_TO_DATASET
/** 3. RETENTION_PERIOD - Nbr of days to retain the
/** newly created restore DSN
/** - 0000 through 9999 days
/** 4. RESTORE_SMS_STORAGE_CLASS - 8 Character SMS storage class
/** for allocating restore DSN
/** 5. RESTORE_VOLUME_COUNT - Nbr of volumes the restore
/** DSN may span. If omitted,
/** DASD units defaults to 1 and
/** TAPE units defaults to 5
/** 6. RELATIVE_GENERATION - Generation to restore.
/** - Generation can be positive or
/** negative value.
/** - Range from '0' to '-10'
/** - If negative value entered,
/** value must be in quotes. "'"
/** - Default is most recent
/** generation or 0
/** 7. DELETE_OLD_BACKUP - If Restore DSN is already
/** cataloged, delete it?
/** - Must be YES or NO
/** - Default is NO
/** 8. WAIT_FOR_COMPLETION - Wait for each restore to
/** complete before ending this
/** job?
/** - Must be YES or NO
/** - Wait = NO, Restore is
/** scheduled in CUZCLOUD STC
/** - Wait = YES, Job will not
/** end until all restores
/** complete
/**
/** Examples:
/**
/** 1. Restore all 0 generation datasets starting with "MYDSN"
/** with a new alias CUZRESTR on Disk using SMS Storage Class
/** SCCUZRST and retention period of 1 day (Disk DSN),
/** Delete the old backups and Wait for the restore to
/** complete.
/**
/** RESTORE_FROM_CLOUD (
/** RESTORE_TO_ALIAS CUZRESTR
/** RESTORE_TO_UNIT SYSDA
/** RESTORE_SMS_STORAGE_CLASS SCCUZRST
/** RESTORE_VOLUME_COUNT 5
/** RELATIVE_GENERATION '0'
/** RETENTION_PERIOD 0001
/** DELETE_OLD_BACKUP YES
/** WAIT_FOR_COMPLETION YES
/** )
/** INCLUDE ( -\*INCLUDE THE FOLLOWING DSNS
/** MYDSN.* -
/** )
/** EXCLUDE ( -\*EXCLUDE THE FOLLOWING DSNS
/** )
/**
/** 2. Restore dataset MY.CLOUD.DSN to MY.CLOUD.DSN.RESTORE
/** on Disk using SMS Storage Class SCCUZRST with no

```

```

/* retention period. Delete the old backup and wait for the restore to complete.
/*
/*
/* RESTORE_FROM_CLOUD (
/* RESTORE_TO_DATASET MY.CLOUD.DSN.RESTORE
/* RESTORE_TO_UNIT SYSDA
/* RESTORE_SMS_STORAGE_CLASS SCCUZRST
/* RESTORE_VOLUME_COUNT 5
/* RELATIVE_GENERATION '0'
/* RETENTION_PERIOD 0001
/* DELETE_OLD_BACKUP YES
/* WAIT_FOR_COMPLETION YES
/* )
/* INCLUDE ( -\*INCLUDE THE FOLLOWING DSNS
/* MY.CLOUD.DSN -
/* )
/* EXCLUDE ( -\*EXCLUDE THE FOLLOWING DSNS
/* )
/*
/*
/* 3. Restore all 0 generation datasets for disaster recovery coming back as the original dataset names and restore them to TAPE. Retain the tapes for 5 days, don't wait for completion and delete all restored cataloged datasets
/*
/* RESTORE_FROM_CLOUD (
/* RESTORE_TO_UNIT CART3590
/* RELATIVE_GENERATION '0'
/* RETENTION_PERIOD 0005
/* DELETE_OLD_BACKUP YES
/* WAIT_FOR_COMPLETION NO
/* )
/* INCLUDE ( -\*INCLUDE THE FOLLOWING DSNS
/* * -
/* )
/* EXCLUDE ( -\*EXCLUDE THE FOLLOWING DSNS
/* )
/*
/*
/* -----
/*
/* CUZJREST EXEC PGM=CUZJREST,REGION=0M
/* STEPLIB DD DISP=SHR,DSN=#HIGHLVL#.SCUZLOAD
/* SYSPRINT DD SYSOUT=*
/* SYSUDUMP DD SYSOUT=*
/* CUZOUT DD SYSOUT=*
/* CUZJREST DD *
/* -----
/*
/* 5697-I80
/* (C) COPYRIGHT ROCKET SOFTWARE, INC. 2015 ALL RIGHTS RESERVED.
/* -----
/*
/* RESTORE_FROM_CLOUD (
/* RESTORE_TO_ALIAS CUZRESTR
/* RESTORE_TO_UNIT SYSDA
/* RESTORE_SMS_STORAGE_CLASS SCCUZRST
/* RESTORE_VOLUME_COUNT 5
/* RELATIVE_GENERATION '0'
/* RETENTION_PERIOD 0001
/* DELETE_OLD_BACKUP YES
/* WAIT_FOR_COMPLETION YES
/* )
/* INCLUDE ( -\*INCLUDE THE FOLLOWING DSNS
/* MYDSN.* -
/* )
/* EXCLUDE ( -\*EXCLUDE THE FOLLOWING DSNS
/* )
/*
/*

```

Required Changes

The items in bold are the variables you **must** change in the JCL to customize the job for your environment.

#HIGHLVL#

Change #HIGHLVL# to the high level qualifier for your runtime load library.

MYDSN*

Replace this with data set names that you want to restore from the cloud.

RESTORE_TO_UNIT

Specify a one- to eight-character name for the DASD device or tape where the cloud data set will be placed when it is restored.

- If the block size is less than 32K, the restored data set can be sent to tape or disk.
- If the block size is greater than 32K, the restored data set must be sent to a tape device.

Optional Parameters

The parameters below are **optional** and can be included in the JCL as needed.

RESTORE_TO_DATASET

Type up to 44 characters for the data set name to assign to the restored data set. If you prefer to use the original data set name, leave this field blank. If you specify a value for this parameter, you cannot specify anything for the RESTORE_TO_ALIAS parameter as they are mutually exclusive. The RESTORE_TO_DATASET parameter can only be used if one data set is being restored, not multiple data sets.

RESTORE_TO_ALIAS

If you want to restore the data set to a different name, type 1 - 8 characters for the name of a high-level alias to be used when restoring the data set. If you prefer to use the original data set name, rather than an alias, leave this field blank. If you specify a value for this parameter, you cannot specify anything for the RESTORE_TO_DATASET parameter as they are mutually exclusive.

RETENTION_PERIOD

Enter a number between 0000 - 9999 to indicate the maximum number of days you want to keep the newly created restore DSN. This Retention Period is a z/OS retention period which will be used by your Tape Management System.

RESTORE_SMS_STORAGE_CLASS

Specify a one- to eight-character name SMS storage class for allocating the restore DSN.

RESTORE_VOLUME_COUNT

Type a number between 1 and 256 for the number of volumes needed to restore a multi-volume data set from the cloud. If omitted, the number of DASD units defaults to 1 and the number of TAPE units defaults to 5.

RELATIVE_GENERATION

Type a number in the range from '0' to '-10' to specify which generation of the data set to restore. When multiple copies of the same DSN have been saved to the cloud, you can specify which of those copies (generations) to restore. The default is the most recent generation, which is 0. You can specify a positive or negative value, however negative values must be in quotes, such as '-5' or '-10'. For example, the generation before the most recent backup can be specified as '-1' or simply 1.

DELETE_OLD_BACKUP

Indicate whether to delete (YES) or keep (NO) the local cataloged copy of the data set before restoring the data set from the cloud. Specify YES or NO. The default is NO. If you are restoring a data set from the cloud to a data set that already exists, the old data set must be deleted before the restore can occur. A message will appear if the "restore to" data set already exists and "Delete Old Backup" is set to "N".

WAIT_FOR_COMPLETION

Indicate whether to wait for each restore to complete before ending this job. Specify YES or NO. If YES, the job will not end until all of the restores are complete. If NO, the Restore is scheduled in the CUZCLOUD started task so that the job can end. Look in the sysout file or data set specified on the CUZOUT DD in the CUZCLOUD started task for results of the restore.

Sample Job for List Datasets

The List Dataset job, CUZJLIST, provides a list of datasets located on the cloud. The JCL provided in the sample job can be modified for your environment. This job supports the use of wildcard characters and full data set restoration.

The List Dataset job merges in the z/OS repository to show which datasets match between the Cloud Tape Connector (CTC) repository and the cloud. This job will also show datasets that do not match between the two. There is a column in the report with the one of the following statuses:

- M – Data Set found on cloud and in CTC Repo.
- R – Data Set found on cloud, but not in CTC Repo.
- Z – Data Set found in CTC Repo, but not on Cloud.

The sample job for performing a list process is located in the SCUZSAMP library in member CUZJLIST. Be sure to follow the instructions listed in this member. You will need to tailor the JCL before submitting it. An example of member CUZJLIST is shown below.

```
//JOB CARD JOB      , '?' , REGION=0M
//*****
//* COPYRIGHT ROCKET SOFTWARE, INC. 2016
//*
//* ALL RIGHTS RESERVED.
//*****
//* * * * *
//* Member:  CUZJLIST
//*
//* This job will produce a list of datasets out on the cloud.
//* It will merge in the z/OS repository to show which datasets
//* match between the repository and the cloud. It will also show
//* datasets that do not match between the two.
//*
//* Change #HIGHLVL# to the high level qualifier for your runtime
//* load library
//*
//* Change #CLOUD# to the Cloud Connector Cloud Name to acquire
//* a dataset list.
//*
//* * * * *
//CUZ#LIST EXEC PGM=CUZ#LIST,PARM='#CLOUD#',REGION=0M
//STEPLIB DD DISP=SHR,DSN=#HIGHLVL#.SCUZLOAD
//CUZ#LIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

Required Changes

The items in bold are the variables you **must** change in the JCL to customize the job for your environment.

#HIGHLVL#

Change #HIGHLVL# to the high level qualifier for your runtime load library.

#CLOUD#

Change #CLOUD# to the Cloud Name used by Cloud Tape Connector to acquire a dataset list.

Diagnostic log files

To write errors and other informational messages that pertain to the transfer of data to and from a cloud server, you need to allocate the LOGMSG DD file. When LOGMSG is allocated, additional DDs can also be used to collect diagnostic information for IBM Technical Support. These additional DDs, as well as LOGTRACE and LOGDEBUG, should only be used when directed to do so by IBM Technical Support.

To obtain diagnostic data from any of these DD statements, a DD statement must be added to the CUZCLOUD started task procedure. The format of these DD statements is as follows:

```
//LOGMSG DD SYSOUT=*
//LOGDEBUG DD DUMMY
//LOGTRACE DD DUMMY
```

Chapter 11. Cloud Tape Connector Repository

The Cloud Tape Connector Repository contains information about all of the data sets copied to clouds defined in Cloud Tape Connector. The repository data set can only be accessed through the CUZCLOUD started task.

The CUZCLOUD address space acts as a service provider for users that need access to the Repository. This is known as synchronous cross-memory communication. Through this process, jobs that create data sets which are copied to the cloud will perform those actions on the cloud, but do not update the Repository directly. That is also the case for any actions on data sets that are generated through the Cloud Connector ISPF dialog. Instead, the CUZCLOUD address space allocates any updates to the Repository.

Repository Records

The Repository uses different types of records to create information about the data sets stored in the clouds defined in Cloud Tape Connector.

The Cloud Connector Repository is a z/OS VSAM Keyed Sequential data set (KSDS) with a 52-byte key. The first 8-bytes of the key are either x'00's or the store clock (STCK) value when the Cloud Connector created the entry. The data set name copied to cloud storage makes up the remaining 44-bytes. There is a minimum of two records in the repository for each data set stored in cloud storage, a single Profile record and one or more Discrete records.

- A **Profile** record names a data set that has at least one copy of itself in cloud storage. It also holds information necessary to find related Discrete entries for specific data sets. A Profile record in the Repository must have at least one Discrete record. For a Profile record, the first 8-bytes of the key are x'00's and the remaining 44-bytes are the data set name stored in the cloud.
- A **Discrete** record uniquely names a data set that lives in cloud storage and has all the information necessary to retrieve the data set from the cloud. Explicitly removing all of the Discrete records for a named data set will also remove the associated Profile record. For a Discrete record, the first 8-bytes are the STCK value at the time of creation and the remaining 44-bytes are the data set name stored in the cloud.

Cloud Tape Connector uses this format to support multiple copies or Generations of a data set in cloud storage. For example, a data set named A.B.C will have a Profile record and one or more Discrete records having the name A.B.C in the data set name part of the key. The total number of discrete records equate to the largest number of Generations allowed.

See [“Step 12: Define a z/OS repository file”](#) on page 16 for instructions on defining the Cloud Tape Connector Repository. In the sample library (SCUZSAMP), member CUZIDEFR has JCL that defines and populates the Repository. Submit this job to execute a job that defines the Repository and populates it with a single null record. Cloud Tape Connector requires that this record exist to allow sharing the Repository.

Note: The Repository at early installations may have incorrect Share Options (SHROPTNS) specified. Correct this using the JCL in the sample library (SCUZSAMP) member CUZJALTR. Submit this JCL to execute a job that changes the Repository to SHROPTNS(4 3), which is the correct specification.

After starting the Cloud Tape Connector address space, the filter criteria determines which data sets to copy to the cloud, and records describing the copied data sets are placed in the Repository. This is the data necessary to restore data sets from the cloud.

Sharing within a Sysplex

For Sysplex environments with more than one LPAR, if jobs creating data destined for the cloud can execute on any of the LPARs, run the Cloud Tape Connector address space on all the LPARs. This requires that Cloud Tape Connector share the Repository between multiple LPARs.

To share the repository, specify DISP=SHR on the CUZCLOUD DD statement in the SCUZSAMP library, member CUZCLOUD. When Cloud Tape Connector detects sharing, the following message is issued during address space startup:

```
CUZS111I CUZ#WTDR-Repository Cross System Sharing active
```

To facility Repository sharing, Cloud Tape Connector employs ENQ/DEQ logic to keep Repository update integrity across the LPARs. The ENQ major name is **SYSZCUZV** and the ENQ minor name is the Repository data set name (DSN=) on the started task CUZCLOUD DD statement. The type of control is determined by the process being attempted:

- Update, Delete, and Insert attempts request Exclusive control.
- Read attempts request Shared control.

The ENQ requests specify a SYSTEMS scope to allow the Sysplex environment to ensure integrity across the defined LPARs.

Chapter 12. Introduction to Virtual Tape Emulation

This chapter gives a brief description of the concepts and benefits of Virtual Tape Emulation (VTE).

VTE concepts and benefits

Historically, data in the enterprise has been stored on disks and tapes. While disks were relatively expensive and required capacity management and I/O optimization, tapes were the best media to manage backups, migrations, disaster recovery, and other applications.

Over the years, as the volume of data has grown rapidly, the data management aspect of tapes has become more complex and a source of many problems, like capacity limitations and low media utilization. From the physical management perspective, there are problems like floor space limitations, disaster recovery requirements to have tapes in multiple locations. Issues with drives include drive contention, reliability, and more.

Virtual Tape Emulation can solve these problems and streamline business processes by removing the need to handle real tapes.

Virtual Tape Emulation (VTE) is an MVS software product that simulates IBM-compatible tape controllers, tape drives, and tape volumes. Real tape devices are replaced by logical, memory resident, virtual tape devices, and tape volumes are replaced with virtual tape data sets that are maintained on any IBM compatible 3380/3390 disk subsystem. VTE simulates 3480/3490/3590/3592 devices. During this process, VTE remains fully transparent to the host, the MVS applications, the tape management systems, and the users.

Today, VTE can be configured in a "VTE complex" that enables concurrent access to tape volumes. VTE is a Virtual Tape product that does not need real tapes.

VTE supports standard-label and no-label tapes. Multiple data set volumes and multiple volume data sets are fully supported.

Unlike tape mount managers that allocate tape data sets on disk and, thereby, can cause jobs to experience Sx37 abends, no Sx37 abends can ever occur to jobs using virtual tapes. VTE is not limited to cataloged, nor to SMS-managed data sets. Its virtual tapes continue to use the original retention defined through the Tape Management System (TMS).

In addition, special tape commands (like Read Backward) and large blocks (up to 256 K) continue to be supported.

The implementation of VTE does not require any changes to existing JCL or applications. VOL=REF groups are supported by a special mechanism that ensures that all data sets on a given tape will be allocated exclusively either on a real tape or on a virtual tape. UNIT=AFF chains are supported by splitting these chains to one that allocates real tapes and another that allocates virtual tapes.

A special feature of VTE is its ability to enable the equivalent of DISP=SHR on a virtual tape. MVS does not allow the specification of SHR on a tape unit like it does on a disk device, but with rules coded into the VTE server's parameter list, virtual tapes can now be shared by applications within the same MVS system.

VTE can provide up to 256 virtual tape devices per VTE server, thus eliminating contention for tape devices. VTE gives back the floor space originally dedicated to real tape devices and eliminates the mechanical delays, failures, and tape Data Checks (DCK) that frequently occur when dealing with real tapes. All of this happens in a hands-off, secured, disk environment.

VTE handles an unlimited number of virtual tapes, so, unlike manual or other virtual tape systems, the number of tapes is no longer an issue. In addition, the virtual tape library may be shared among systems. Gradual increase in virtual tapes media capacity is possible by just adding more disk space to the system.

VTE can be fully integrated with any MVS environment. Its virtual tapes are managed by the site's tape management system. VTE virtual tape data sets, being disk data sets, can be system managed by all

means of SMS and disk management products. Also, VTE offers full control of these virtual tape dataset locations.

VTE virtual tapes are standard disk data sets. They may be browsed using ISPF or other online browser.

In conjunction with a choice of IBM's XRC or IBM's PPRC or similar tools, combined with VTE, you have a solution that offers instantaneous recovery of and access to virtual tape libraries in a disaster recovery situation. With VTE, "day one processing" at the recovery center is not exposed to limited Silo recovery space due to the lack of tape transports meeting the backup and batch processing window. VTE electronically transfers all tape data while migrating or merging data centers. Also, in conjunction with any type of FTP, VTE electronically vaults virtual tapes to one or more remote off-site storage locations.

Using VTE (CTC), it is possible to electronically vault virtual tapes to remote off-site storage locations on a Cloud.

With VTE, overall system performance and storage utilization are increased to an optimal state while cost of ownership is reduced to a minimum.

Unlike hardware-based virtual tape systems, VTE does not introduce new hardware. It leverages investments in existing tape hardware and software. VTE introduces the least amount of change and complexity, while offering the greatest flexibility and scalability leaving the choice to the user of how much resources to allocate to virtual tape tasks.

Chapter 13. How VTE works

This chapter explains some basics about the VTE components and the lifecycle of the virtual tapes.

VTE components

The VTE components can be grouped into four distinct groups:

- Front-end components
- Back-end components
- Components that enable the relationship between the front-end and back-end components
- Offline utilities

Before proceeding with the component descriptions, it is worthwhile to understand the terms *front-end* and *back-end*.

The front-end refers to all tasks performed by MVS or other applications against real tapes or cartridges. These tasks include allocating tape drives, mounting of tapes, opening tape data sets, reading/writing tape data sets, unloading tapes, cataloging tape data sets, etc.

The back-end refers to all tasks performed by the VTE server against the virtual tape data sets. All these tasks are performed exclusively by VTE and the VTE server is the only entity that monitors and manages the relationship between the front-end and the back-end.

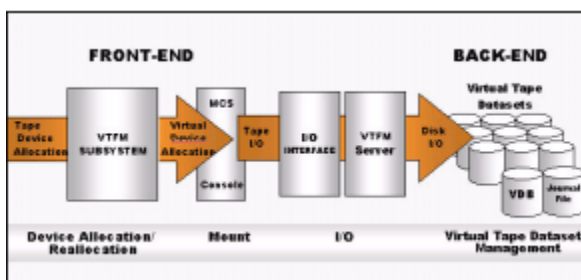


Figure 57. VTE Components

Front-end components

The front-end components, described below, work in conjunction with tasks performed by MVS.

VTE Subsystem

The Virtual Tape Emulation (VTE) subsystem is required to perform dynamic reallocation, to participate in type3, type4, and type5 e-Vault, and to enable Parallel Access Tape (PAT). For more information on PAT, refer to [“Virtual Tape Sharing” on page 137](#).

- The VTE subsystem changes allocations from 3480/3490/3590/3592 or equivalent real devices to VTE virtual devices. The subsystem scans every tape allocation (when a job enters the system or dynamically allocates tapes) and matches each job’s DD statements or dynamic allocation definitions with the predefined dynamic reallocation rules. If a match is found, the subsystem changes the real tape device allocations to a VTE virtual tape device allocation.

This way, jobs and applications may be assigned VTE virtual devices while VTE is active, without any changes to their JCL or tape allocation definitions.

- The VTE subsystem scans every tape allocation (when a job enters the system or dynamically allocates tapes) and subject to predefined VTE PAT rules, it substitutes the virtual tape volume serial numbers with PAT alias volume serial numbers.

This way, jobs and applications may share virtual tapes within the same MVS system.

- The VTE subsystem passes data between the server and the HVTUFTP utility in behalf of type3 and type4 e-Vault.
- The VTE subsystem passes data between the server and the HVTUCTC utility for a type5 e-Vault.

Extended MCS

The VTE extended MCS is responsible for capturing tape mount requests and triggering the VTE server to handle these mounts.

VTE Virtual Device

A VTE virtual device is a standard simulated 3480/3490/3590/3592 device defined to MVS using a UIM and a regular HCD process. No tape-related hardware is required. The virtual devices do not need to be connected to a Control Unit and they are attached through virtual CHPIDs (channels that do not necessarily exist in the CPU).

```
Command ==> _____ Scroll ==> CSR
I/O Device List Row 13 of 109 More:
Select one or more devices, then press Enter. To add, use F11.
-----Device-----#-----Control Unit Numbers + -----
/ Number Type + CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0570,32 VT3490D2 4 -----
- 05B0,32 VT3490D4 4 -----
- 05F0,16 VT3490D6 4 -----
- 0700,36 3390 1 4 0700 -----
- 0724,42 3390 1 4 0700 -----
```

Figure 58. VTE virtual devices on the I/O Device List panel of the HCD

VTE identifies its virtual devices according to the name defined in the ESOTERIC parameter of the VDEVICE initialization statement. Immediately after its startup and interface establishment, and if so specified in its initialization statements, the VTE server issues a V cuu1-cuu2, ONLINE operator command to the required amount of devices defined as part of this esoteric name. From this point on, these devices are fully operational under the VTE server control and any application can use them as real 3480/3490/3590/3592 devices.

An example of the code for varying a virtual device online is shown below:

```
V 570,ONLINE
IEE302I 0570 ONLINE
D U,,,570,1
IEE457I 13.30.16 UNIT STATUS 221
UNIT TYPE STATUS VOLSER VOLSTATE
0570 VT34 0-NRD /REMOV
```

Virtual tape

The tapes mounted on VTE virtual devices are called virtual tapes. Although the application considers them as real tapes and they are managed by the tape management system, no real tapes exist.

The capacity of virtual tapes is set by the user.

Tapepool

The virtual tapes may be grouped in different volser ranges called *tapepools*. These ranges are similar to the volser ranges defined in the local tape management system. Each tapepool is associated with a diskpool (explained later).

Tapepools Scratch Virtual Tapes Monitor

The VTE tapepools scratch virtual tapes monitor constantly checks that sufficient scratch virtual tapes exist on the tapepools. When the scratch virtual tapes number falls below a given threshold, appropriate warning messages are issued.

Back-end components

The back-end components, described below, are managed by the VTE server.

Virtual Tape data set

The virtual tape dataset is a physical, sequential, multi-volume, disk dataset that contains the data written to virtual tapes by applications.

VTE allocates and maintains one virtual tape dataset for each virtual tape, that is, one tape volume data is contained by one disk dataset.

The virtual tape data sets reside on a pool of disks defined as a diskpool in the VTE tapepools and diskpools definitions.

The virtual tapes structure is similar to the real tapes structure.

A virtual tape contains all the user data and the access method meta-data such as header labels, trailer labels, and tapemarks. In addition, the virtual tape dataset contains VTE meta-data needed for proper processing of the virtual tape.

Virtual Tape Database - VDB

VTE keeps track of the relationship between virtual tapes and virtual tape datasets in its virtual tapes database—the VDB. The VDB is a VSAM KSDS file that contains a record for each virtual tape with some details about the virtual tape status (for example: scratch/not scratch, virtual tape dataset name, etc.).

The first record in the VDB is a control record that relates the VDB to a VTE server and holds synchronization information.

Virtual Tape Library

The virtual tapes, along with the VDB, comprise the so-called virtual tape library. The virtual tape library can be partially or fully set to Read_Only mode.

Journal File

The journal file is a sequential wrap-around file that records all the updates done to the VDB.

If a recovery of a VDB is required, it can be forward-recovered to an up-to-date status according to the journal file.

e-Vault Manager

The VTE e-Vault manager is responsible for managing and moving virtual tape datasets between the local and remote off-site storage locations. As part of this management, VTE establishes and uses TCP/IP sessions with other VTE servers in other MVS systems.

Diskpool

A group of disks defined under an esoteric name or a Storage Group and associated with a specific tapepool is called a diskpool. More than one tapepool can be associated with the same diskpool.

Diskpools free monitor space

The VTE diskpools free space monitor constantly checks that sufficient free space exists on diskpools that contain virtual tape datasets. When the free space falls below a given threshold, appropriate warning messages are issued.

Front-end to Back-end connection components

The components described below enable the relationship between the front-end and backend components.

The VTE Server

The VTE server is the started task that is responsible for the VTE virtual devices simulation. It processes in its back-end the MVS and applications front-end requests intercepted by the I/O interface.

When the VTE server is started, it verifies all the initialization statements, initializes all VTE components, and varies online the VTE virtual devices.

I/O Interface

The I/O interface intercepts any I/O request directed to the VTE virtual devices. When intercepted, the request is directed to the VTE server for processing. This interface is implemented via a front-end to the MVS STARTIO routine.

Offline Utilities

A set of offline utilities are provided with VTE. These utilities manage the VDB and the virtual tape data sets. For more information on offline utilities, see [Chapter 32, “VTE Offline Utilities,” on page 317](#).

Tape Processing with VTE

This section describes how virtual tapes are processed from the submission of a job by the user in the front-end until the storing of the data on the disk in the back-end.

Front-end Processing

When the VTE subsystem identifies that a job has started, it scans its allocation table and matches the job's DD statements with the predefined dynamic reallocation rules. If a match is found, VTE virtual devices are set for the job.

When PAT is implemented, the job's DD statements are checked for PAT-eligibility. If the DD statement is PAT-Eligible, the respective volume serial numbers are substituted with PAT alias volume serial numbers.

When the mount is requested on the VTE virtual device, the virtual tape is mounted on the VTE virtual device and the I/O requests are sent by the application to the device.

An output virtual tape is used until an End-of-Volume is set by the VTE server (according to the tape capacity defined by the user), then it is unloaded and a new virtual tape is mounted.

An input virtual tape is used until the trailer label or tapemark is encountered by the VTE server, then it is unloaded and the next virtual tape (if any) is mounted.

Back-end Processing

When the virtual tape is mounted in the front-end, a virtual tape data set is allocated and opened in the back-end:

- For a scratch tape request, a new virtual tape data set is allocated on the diskpool as defined in the diskpools and tapepools definitions. The first extent is allocated according to the expected tape capacity and the number of eligible disks. More extents are allocated as needed during writing.
- For a non-scratch tape request, the relevant virtual tape data set is allocated and used.

From this point on, every I/O done by the job to the tape data set defined by the DD statement in the front-end is captured by the VTE I/O interface and directed to the VTE server which performs the appropriate operation against the virtual tape data set on disk:

- If the job reads a block in the front-end, the VTE server reads a whole disk track from the respective virtual tape data set in the back-end and passes the first block to the job. Subsequent read blocks are passed from the already read buffer, without accessing the disk again.

If the job writes a block in the front-end, the VTE server keeps it in an internal buffer in the back-end. Subsequent written blocks are kept this way until a full track-sized buffer is filled, in which case, it is written to the virtual tape data set on disk.

This way, very high performance and high throughput are achieved.

- After a predefined number of written blocks, and with each tape channel command that causes synchronization between the tape drive and the controlling computer, a checkpoint is written to the virtual tape data set. This checkpoint is then used for recovery purposes when a system failure occurs while processing a tape.

Note: The virtual tape data sets are written to disk using ECKD CCWs. That way, if a site defined a PAV alias device for disks that contain virtual tape data sets, VTE can access multiple virtual tape datasets simultaneously, without getting a busy condition from the disk controller.

When the job ends execution and unloads the virtual tape in the front-end, the VTE server closes and deallocates the virtual tape data set in the back-end. In addition, the appropriate entry in the VDB is updated accordingly.

Offline utilities processing

If a virtual tape data set is not properly closed or if its associated VDB entry is damaged, an offline utility can recover it.

If, according to the tape management system, the virtual tape becomes scratch, a VTE offline utility deletes the entire virtual tape data set from the disk and the appropriate VDB entry is updated accordingly.

You can produce reports about the contents of the virtual tape library and the status of the virtual tapes using an offline utility.

Using an offline utility, virtual tape data sets are prioritized for de-staging, according to their expected activity. The list of prioritized data sets is prepared for the disk management product according to the desired utilization level of the VTE diskpools.

Chapter 14. Implementation

This chapter explains how to initially implement VTE and how to use VTE virtual devices and virtual tapes. It also describes the VTE tapepools and diskpools concepts.

Simulated virtual device types

VTE simulates four different tape device types, which are the following:

- 3480
- 3490
- 3590
- 3592

Virtual devices general considerations

The devices are defined to MVS using a single UIM and a regular HCD process. The UIM (through VTE EXIT002) includes all potential device types definitions. This technique allows for several devices of the same type to be defined (perhaps for several VTE servers running in the same MVS system).

While various real devices differ by means of capacity, speed, recording technology, reliability and media compatibility, VTE uses an identical format to store the data on its back-end disks regardless of the simulated device type. The virtual tape capacity and the compression is controlled solely by VTE. Consequently, virtual tapes created on one virtual device type can be accessed on any other virtual device type.

According to this, no special preference should exist for using, for example, 3490 simulation over 3590 simulation. The only need for a specific simulation can be when an application that uses duplexing (like HSM) writes the data on a virtual tape and needs to write the duplex on a real tape. In this case, both devices should be identical and the virtual device should be selected according to the real device.

3592 devices considerations

Although the real 3592 devices can write up to 300GB (full capacity), VTE virtual tapes capacity (uncompressed) cannot be larger than 52GB. Therefore, if 3592 devices are used for duplexing, the real 3592 device should be scaled to 60GB (optimal performance).

Real 3592 devices emulate already supported real 3490 or 3590 devices. Consequently, virtual 3592 devices are simulated through the existing 3590 devices.

For real devices, allocation ambiguities between 3592 devices and the actual 3590 emulated devices should be handled by the user. As with virtual 3592 devices (simulated through virtual 3590 devices), this problem does not exist. The VTE UIM sets different device type values to each of its defined devices letting MVS allocate the virtual tapes according to these different device types.

The following table describes the various device types usage and implementation:

Table 8. Simulated device characteristics				
Simulated device	Simulated media	UIM device	TPOOL MTYPE parameter	VDEVICE EMULATE parameter
3480	0	3480	N/A	N/A
3490	1	3490	STANDARD	N/A
3490	2	3490	ENHANCED	N/A
3590	3	3590	N/A	N/A

<i>Table 8. Simulated device characteristics (continued)</i>				
Simulated device	Simulated media	UIM device	TPOOL MTYPE parameter	VDEVICE EMULATE parameter
3592	5	3590	N/A	3592

Virtual Tape candidates

Before implementing VTE, a decision must be made to define what are the best tape datasets candidates to be reallocated by VTE to virtual tapes. The decision varies from site to site according to its special needs. For example:

- Complete Tape Replacement - VTE can be used to completely replace all real tape processing so no real tape device will be attached to the mainframe anymore.
- Disaster Recovery - VTE can be used to reallocate disaster recovery data that used to reside on tape and now can be mirrored remotely using a remote replication product.
- Shorter Batch Window - VTE can be used to backup application data, thus shortening the backup time and the application down time.
- Short-term backups - VTE can be used to hold daily/weekly/monthly backup that eventually dies on disk without any real tape involved.
- Workload sharing of tape data sets. Can be used to parallel process data on tape.
- Tape Caching - VTE can be used to cache the tape work on a diskpool as a station before the data set expires on the cache or migrated to real tape.
- Electronic Vaulting - VTE can be used to electronically vault tapes off-site, instead of using a manual process (the so-called 'Truck Access method').
- Data center migration - VTE can be used to merge or migrate a data center. The virtual tapes data can be moved over a remote replication product.
- Media for Migration Level 2 -VTE can be used as a media to hold all ML2 data for ABR, CA-Disk or DFSMSshm.

The following data sets should never be written onto virtual tapes:

- The VDB
- The journal file
- The user catalog that contains the VDB and the journal file
- The user catalog that contains the virtual tape data set's entries
- The master catalog
- The virtual tape data sets
- Data sets cataloged in the catalog that contains the VDB. The best way to avoid this is to define the VDB in a separate user catalog.

The above applies to discrete data sets, as well as to full disk dumps that may contain the above data sets.

IBM presents the VTE Assessor tool which can analyze a customer's tape environment and make recommendations on VTE diskpools sizing, simulations of tape workload over VTE, and identification of the most suitable tape candidates for VTE in the event of a partial tape replacement.

The VTE Assessor is part of the IBM Professional Services which also assists the customer in implementing VTE strategies ranging from library replacements to full Business Continuity and data recovery schemes.

Allocating virtual devices

The allocation of a VTE virtual device is done either explicitly in the JCL or, without changing JCL, by coding VTE dynamic reallocation rules.

VTE offers four methods by which a job can allocate virtual devices:

- Manual allocation
- Automatic allocation using dynamic reallocation rules
- Automatic allocation using SMS Data Class ACS routines
- Implicit allocation

VTE offers one method by which a job can bypass allocation of VTE virtual devices: Hyper 'never-reallocate'

Manual allocation

Manual allocation is achieved by changing the JCL to explicitly point the VTE virtual device in the UNIT parameter of the DD statement that you wish to assign a VTE virtual device.

Example:

```
//M20TEST JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1 EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=M20.KUKU2
//SYSUT2 DD DISP=(,KEEP),
// DSN=M20.FIRST.TAPE.FILE,
// UNIT=VT90ES
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
```

In the above example, VT90ES is the esoteric name assigned to the VTE virtual devices. SYSUT2 DD statement will allocate a VTE virtual device.

Automatic allocation using dynamic reallocation rules

Automatic allocation means allocating a VTE virtual device without modifying an existing job's JCL or an existing application's allocation parameters. A series of dynamic reallocation rules can be defined to instruct VTE to change existing real tape device allocations to VTE virtual device allocations.

The dynamic reallocation rules define the criteria for reallocation and consist of parameters like jobname, data set name, step name, program name and more. A set of parameters can be grouped into a filter list.

The reallocation is done transparently when the job starts execution or when a tape device is dynamically allocated. As each job step ends, the created data sets are cataloged (if requested) as residing on a virtual tape.

The dynamic reallocation rules logic is described in [Chapter 16, “Dynamic Reallocation Logic,”](#) on page 145.

For example, assume that the following dynamic reallocation rule is defined:

```
RULE INC,JOBNAME=M20*,UNIT=3480
```

The following job is submitted:

```
//M20TEST JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1 EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=M20.KUKU2
//SYSUT2 DD DISP=(,KEEP),
// DSN=M20.FIRST.TAPE.FILE,
// UNIT=3480
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
```

In the above example, SYSUT2 DD statement will allocate a VTE virtual device due to the dynamic reallocation rule.

Automatic allocation using SMS Data Class ACS Routines

The implementation with SMS Data Class ACS routines is only available in JES2 environments.

Automatic allocation can be achieved by using the SMS Data Class ACS routine in conjunction with VTE dynamic reallocation rules. This way VTE determines which DD statement should be reallocated based on the information made available to the ACS routine by MVS. Using this method, the ACS routine automatically assigns a pre-defined Data Class to all DD statements that should use a VTE virtual device. Only one dynamic reallocation rule should be defined to tell VTE to reallocate all the DD statements with this specific Data Class.

The Data Class properties are meaningless since no data set is really allocated with this Data Class.

For example, assume that the following Data Class ACS routine is coded:

```
PROC DATACLAS
FILTLIST TAPES          INCLUDE ('3420','3480','TAPE','3490',
                              'CART8','CART9')
FILTLIST M20            INCLUDE (M20.TEST.***)
  IF &UNIT=&TAPES AND &DATACLAS='' THEN
    SELECT (&DSN)
      WHEN (&M20)      SET &DATACLAS='DCELCANA'
  END
END
```

The following dynamic reallocation rule is defined:

```
RULE      INC,
          DATACLAS=DCELCANA
```

The following job is submitted:

```
//M20TEST  JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1       EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=M20.KUKU2
//SYSUT2   DD DISP=(,KEEP),
//          DSN=M20.TEST.DATA1,
//          UNIT=3480
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
```

In the above example, SYSUT2 DD statement will allocate a VTE virtual device due to the Data Class ACS routine and the dynamic reallocation rule.

Implicit allocation

Implicit allocation is done automatically by VTE using dynamic reallocation (without needing dynamic reallocation rules or SMS Data Class ACS routines). Whenever a job DD or dynamic allocation specifies a virtual tape volser and a unit which is NOT a virtual unit, VTE will cause this allocation to allocate a virtual device.

The following job is submitted:

```
//M20TEST  JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1       EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=M20.KUKU2
//SYSUT2   DD DISP=OLD,
//          DSN=M20.FIRST.TAPE.FILE,
//          UNIT=3480,
//          VOL=SER=EC0026
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
```

In the above example, EC0026 is a virtual tape. SYSUT2 will allocate a VTE virtual device due to implicit allocation.

Hyper 'Never-Reallocate'

Hyper 'never-reallocate' means not reallocating a tape allocation even though the dynamic reallocation rules dictate that or an implicit allocation condition exists.

This option is useful when an external real tape is to be accessed and a similar volser exists already in the virtual tape library.

In order to use this option, the DCB of the involved DD statement should specify the **CPRI=E** parameter.

The following job is submitted:

```
//M20TEST    JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1         EXEC PGM=IEBGENER
//SYSUT1     DD DISP=SHR,DSN=M20.KUKU2
//SYSUT2     DD DISP=OLD,
//           DSN=M20.FIRST.TAPE.FILE,
//           DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB,CPRI=E),
//           UNIT=3480,
//           VOL=SER=EC0026
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
```

In the above example, EC0026 is a virtual tape. SYSUT2 will not allocate a VTE virtual device due to the DCB=CPRI=E specification (although implicit allocation condition exists).

Using Virtual Tapes

VTE maintains the VDB that contains a record for each virtual tape. When virtual tapes are to be mounted, appropriate virtual tapes are selected from the VDB and presented to MVS and to the tape management system as if they were actually mounted on the requested device.

If tapepools are implemented in the tape management system, the same tapepools should be defined for VTE (with the DPOOL and TPOOL initialization statements) so that VTE will present the expected virtual tape when the mount request is for a specific tapepool.

Sites where some applications use non-specific scratch tapes and some applications use specific pre-defined scratch tapes may take advantage of the 'Mount-on-specific-request-only' attribute.

Virtual tapes that bear this attribute will NOT be eligible for mount on a non-specific scratch request even though they are scratch.

That way, applications that mount non-specific scratch tapes will not interfere with tapes to be later mounted as specific mount requests by the other applications.

VTE distinguishes between the following virtual tape states:

Known virtual tape

A virtual tape that has been used or is being used by VTE and has a record in the VDB. There are three types of 'known' virtual tapes:

- *In-use* virtual tape:

A virtual tape that is currently in use for read or write by jobs served by VTE. The 'in-use' value indicates the sharing level (that is, one write or one or more read applications) of the virtual tape among systems that share a VTE complex.

- *Scratch* virtual tape:

A virtual tape that is marked 'scratch' in the VDB by the HVTUSCR utility or added by the HVTUTAP utility and never used. A scratch virtual tape can be marked with the 'Mount-on-specific-request-only' attribute.

- *Active* virtual tape:

A virtual tape that has been used in the past and is not currently mounted and is not marked 'scratch' by the HVTUSCR utility.

Unknown virtual tape

A virtual tape that has never been mounted on a VTE virtual device or has never been added to the VDB by any VTE utility.

Using Known Virtual Tapes

In the following cases, a "known" virtual tape is mounted:

- When a DD statement refers to a VTE virtual device (either explicitly or dynamically reallocated), and the request is for a non-specific (scratch) virtual tape (from a specific or the default tapepool), and the VDB contains scratch virtual tapes, VTE will assign and mount a scratch virtual tape from the appropriate tapepool. Virtual tapes that bear the 'Mount-on-specific-request-only' attribute will not be eligible for mount on a non-specific scratch request, even though they are scratch.
- When a DD statement refers to a cataloged data set residing on a virtual tape, VTE will assign and mount the virtual tape on which the referred data set resides.
- When a DD statement refers to a VTE virtual device (either explicitly or dynamically reallocated), and to a specific virtual tape, VTE will assign and mount the specific virtual tape.

Using Unknown Virtual Tapes

In the following cases, an "unknown" virtual tape is added to the VDB:

- When using the HVTUTAP utility.

This utility performs a mass add of new virtual tapes to the VDB. These virtual tapes will be later eligible for mount as scratch with no need to further label them.

Note: As long as a virtual tape is not mounted, no virtual tape data set is created on disk.

This is the recommended method to add virtual tapes to the VDB.

- When requesting a scratch virtual tape and no scratch virtual tapes exist in the VDB.

In this case, the VTE server issues the HVT317W message asking the operator to define a new virtual tape volume serial number. Once defined, the virtual tape is added to the VDB and becomes a 'known' virtual tape.

- When requesting a specific "unknown" virtual tape and opening it for output.

In this case, IEC704A message is issued by MVS. Once the operator replies U, the virtual tape is added to the VDB and becomes a "known" virtual tape.

Adding virtual tapes is subject to the employed tape management system restrictions.

Example 1 for known and unknown virtual tapes

Consider the following job:

```
//OUT01 DD DISP=(NEW,KEEP),
//          DSN=M20.FIRST.TAPE.FILE,
//          VOL=SER=EC0001,
//          UNIT=VT90EC
```

- If EC0001 is a "known" virtual tape, EC0001 will be mounted and used with no further operator intervention. EC0001 can be either scratch or active. When unloaded, EC0001 is marked "active" in the VDB.
- If EC0001 is an "unknown" virtual tape and the application opens EC0001 for input, an IEC512I message is issued by MVS and EC0001 is unloaded. This occurs so an application cannot read a new unlabeled tape and VTE cannot open an "unknown" virtual tape for input. EC0001 will not be added to the VDB nor allocated on disk.
- If the application opens EC0001 for output, an IEC704A message is issued by MVS asking the operator to confirm the tape label. Once the operator replies 'U', EC0001 is added to the VDB, gets a standard label and can be used. When unloaded, EC0001 is marked "active" in the VDB.

Note: This option is subject to the restrictions of the tape managements system.

Example 2 for known and unknown virtual tapes

Consider the following job:

```
//OUT01      DD DISP=(NEW,KEEP),  
//              DSN=M20.FIRST.TAPE.FILE,  
//              UNIT=VT90EC
```

- If the VDB contains scratch virtual tapes, one of them is mounted on the requested VTE virtual device and no further operator intervention is required. When unloaded, the virtual tape is marked "active" in the VDB.
- If the VDB does not contain scratch virtual tapes, the VTE server mounts an arbitrary "unknown" virtual tape and issues an HVT317W message asking to reply with a new label. Upon reply, the virtual tape is added to the VDB, gets a standard label and can be used. No further operator intervention is required. When unloaded, the virtual tape is marked "active" in the VDB.

Scratching Virtual Tapes

Once a virtual tape is used by VTE, it is marked as "active" in the VDB and it is not eligible for mount as scratch. These virtual tapes are scratched by the HVTUSCR utility (as directed by the tape management system). As the utility scratches the virtual tapes, it releases the disk space used by them while they were active. The actual scratch of the virtual tapes may be deferred according to the value specified for the DEFERBY parameter in the VTE TPOOL initialization statement.

Deleting Virtual Tapes from the VDB

A virtual tape in a "scratch" status can be totally deleted from the VDB by the HVTUTAP utility. When deleted, its entry is dropped from the VDB and the virtual tape becomes "unknown". When reused, it will have to be added as a new virtual tape.

Tapepools and Diskpools

Tapepools are used by tape management systems to indicate the grouping of scratch tapes into different groups. That way, a tape pool can contain one or more range(s) of tapes with common characteristics. Whenever a job requests a scratch tape, the tape management system may (according to its internal logic) request that a scratch from a specific group (tapepool) be mounted.

VTE groups the same ranges of virtual tapes in tapepools. These tapepools share a common set of virtual tape attributes, for example, the tape capacity, the compression method, etc.

Each VTE tapepool is associated with one VTE diskpool. A VTE diskpool is a logical grouping of disks under one esoteric name (for non-SMS-managed disks) or under one Storage Group (for SMS-managed disks).

These diskpools share a common set of attributes, for example, free space threshold warning values, SMS classes, etc.

Each VTE diskpool is associated by one or more VTE tapepools.

Tapepools and Diskpools relationship

When a scratch virtual tape from a specific tapepool is requested, VTE extracts all relevant values for that tapepool. In addition and according to the diskpool associated by the requested tapepool, it decides on which disks (along with their attributes) the virtual tape data set should be allocated.

The tapepools and diskpools attributes are defined by the TPOOL and DPOOL initialization statements in the HVTPOOL member in the SAMPLIB library, respectively.

The following example illustrates the relationship between the TPOOL and DPOOL initialization statements, the job requesting a scratch virtual tape, and as a result, the virtual tape presented by VTE:

```
DPOOL      NAME=DISKPOOL1,
           UNIT=(UNIT1),
           THRESH=(85,180,50)
TPOOL      NAME=TAPEPOOL1,
           DPOOL=DISKPOOL1,
           RANGE1=(EC0001,EC0010),
           RANGE2=(EC0101,EC0110),
           TAPECAP=(610,4),
           CKPINTVL=0,
           DEFERBY=48,
           COMPRESS=(1,2)
```

Suppose that a certain job requests a scratch virtual tape from tapepool TAPEPOOL1.

VTE assigns a virtual tape that is contained in the range defined under one of the RANGE_n parameters in the TPOOL initialization statement. The virtual tape data set is allocated on disks defined in the UNIT parameter of the associated DPOOL initialization statement (DISKPOOL1). The virtual tape data set is allocated on non-SMS-managed disks.

The virtual tape's capacity is 610 MB and its associated virtual tape data set spans up to four disks.

The default Tapepool

There are cases when the tape management system does not associate any tape pool to a scratch tape request. In such cases, the operator must mount a virtual tape whose volser is not defined in any of the tape management system tape pools.

However, VTE must associate a diskpool with this request. To achieve this, VTE defines a special tapepool—the default tapepool. The default tapepool contains all the virtual tapes that are not defined in any of the regular tapepools. The TPOOL initialization statement that defines RANGE1=(0,0) is designated to be the default tapepool. One and only one default tapepool should be defined. If a default tapepool is not defined, message HVT247S is issued and the VTE server fails initialization.

Tapepool rules

When a site does not employ a tape management system or the tape management system does not implement tape pools or no tape pool has been assigned, the TRULE initialization statement can be used to assign tapepools according to certain conditions.

This way, a job DD statement will use a virtual tape from a specific tapepool without the tape management system's intervention. If no tapepool rules are implemented, the default tapepool is assigned.

Note: The tapepools implemented via the VTE tapepool rules mechanism (and pointed to by the TRULE initialization statements) are transparent to the tape management system. That means that the virtual tapes volume serial numbers defined under these tapepools should not be defined under any of the tape management system tape pools, otherwise it will reject their mounts.

A tapepool assignment is done based on specific DD statement parameters and TRULE initialization statements.

There are two types of TRULE initialization statements:

- INC type

Each INC type initialization statement starts a new TRULE initialization statement group that contains none, one or more EXC type initialization statements.

- EXC type

Each EXC type initialization statement is associated with its preceding INC type initialization statement and defines EXclude conditions for the INclude definitions in that specific TRULE initialization statements group.

The TRULE coding logic is as follows:

- Each DD statement values are successively matched against each INC type TRULE initialization statement that starts a new TRULE initialization statements group:
 - For the first matched INC type TRULE initialization statement, the DD statement values are matched successively against all the associated EXC type TRULE initialization statements in that TRULE initialization statement group.
 - If no associated EXC type TRULE initialization statement matches the DD statement, the tapepool is changed from the default tapepool to the tapepool value coded on the INC type TRULE initialization statement and the tapepool assignment process ends.
- If one associated EXC type TRULE initialization statement matches the DD statement, the current TRULE initialization statements group is skipped and matching is resumed with the next TRULE initialization statements group (if any).
- If no TRULE initialization statements group matches the DD statement, no tapepool is assigned to the mount request and it defaults to the default tapepool.

The TRULE initialization statements are defined in the HVTTTLxx member in the SAMPLIB library.

The following example describes the relationship between the TPOOL and TRULE initialization statements and the assigned tapepool.

TPOOL initialization statement sample in the HVTPOLxx member:

```
TPOOL      NAME=T_VTSWIZ,
           DPOOL=D_VTSWIZ,
           RANGE1=(EC1000,EC1099),
           RANGE2=(EC1200,EC1299),
           TAPECAP=(610,10),
           CKPINTVL=0,
           DEFERBY=12,
           COMPRESS=(1,2)
```

TRULE initialization statement sample in the HVTTTLxx member:

```
TRULE      INC, JOBNAME=M20*,
           PGM=IEBGGENER,
           TPOOL=T_VTSWIZ
```

According to the above example, a DD statement is assigned the T_VTSWIZ tapepool, if all the following conditions are true:

- The tape management systems did not previously assign a tapepool by itself.
- The jobname starts with M20.
- The PGM is IEBGENER.
- The DD statement requests a scratch virtual tape.

VTE Tapepool selection logic

The following logic is employed by VTE when selecting the right scratch tape to be mounted or the right tapepool to be assigned to an explicit virtual tape mount request:

- For a scratch virtual tape request:
 - VTE tries to mount a scratch virtual tape found in the requested tapepool's range.
 - If no scratch virtual tapes are found in the tapepool's range, message HVT317W is issued requesting the operator to label a new virtual tape in that tapepool.
 - If the mount request does not require an explicit tapepool, VTE tries to find a matching tapepool according to the TRULE initialization statements (if any). If no TRULE initialization statement matches the mount request, the tapepool defaults to the default tapepool.
 - If the mount request's tapepool defaults to the default tapepool, VTE tries to mount a scratch virtual tape whose volser is not defined in any of the RANGES parameter of the TPOOL initialization

statements. If such a scratch virtual tape is not found, message HVT317W is issued requesting the operator to label a new virtual tape from the default tapepool.

- For explicit virtual tape requests:

VTE automatically assigns it a tapepool (by locating the RANGE this volser belongs to). If no RANGE contains this virtual tape, the tapepool defaults to the default tapepool.

Chapter 15. Virtual Tape Sharing and VTE complexes

This chapter describes how to share the virtual tape library among several MVS systems, how to create several virtual tape libraries on a single MVS system and how to share virtual tapes in the same or among several MVS systems.

Running multiple VTE servers in multiple MVS Systems

VTE can be installed and operated in multiple MVS systems. As each VTE server can support up to 256 devices, using the product in several systems multiplies the number of supported devices.

When the VDB, the journal file and the virtual tape datasets are shared between all the MVS systems, a VTE complex is achieved and all virtual tapes can be accessed from all participating systems.

In such a complex, the VDB and the journal file are RESERVE-d with a QNAME of V_TAPE. If a contention or this resource is detected, message HVT314W is periodically issued, listing the owner and those requesting the resource.

The virtual tape data sets are never RESERVE-d.

When VTE is installed on more than one logical partition (LPAR) of the same processor, the VTE virtual devices can be defined once and be shared by all MVS systems. Being virtual, all of the 256 devices can be online and operational on all the MVS systems, simultaneously.

Requirements

The following requirements must be satisfied in order to form a VTE complex:

- VTE should be installed in all systems.
- The following resources must be shared:
 - The VDB and the journal file.
 - The virtual tape data sets.
 - The user catalogs where the VDB, the journal file and the virtual tape data sets are cataloged. In addition, the master catalogs in all participating systems must contain the appropriate ALIAS to the above user catalogs.
- The following definitions must be identical in all participating systems:
 - QUAL1/QUAL2/QUAL3 values of the VDSN initialization statement.
 - The value specified for UCBTBYT1 in EXIT002 specifications that are used to define the UCB device type.
 - The tapepools and diskpools definitions. Make sure that the esoteric names and the Storage Groups define the same disks in all participating systems.

The following illustration depicts the configuration of two VTE servers in two MVS systems that comprise a complex:

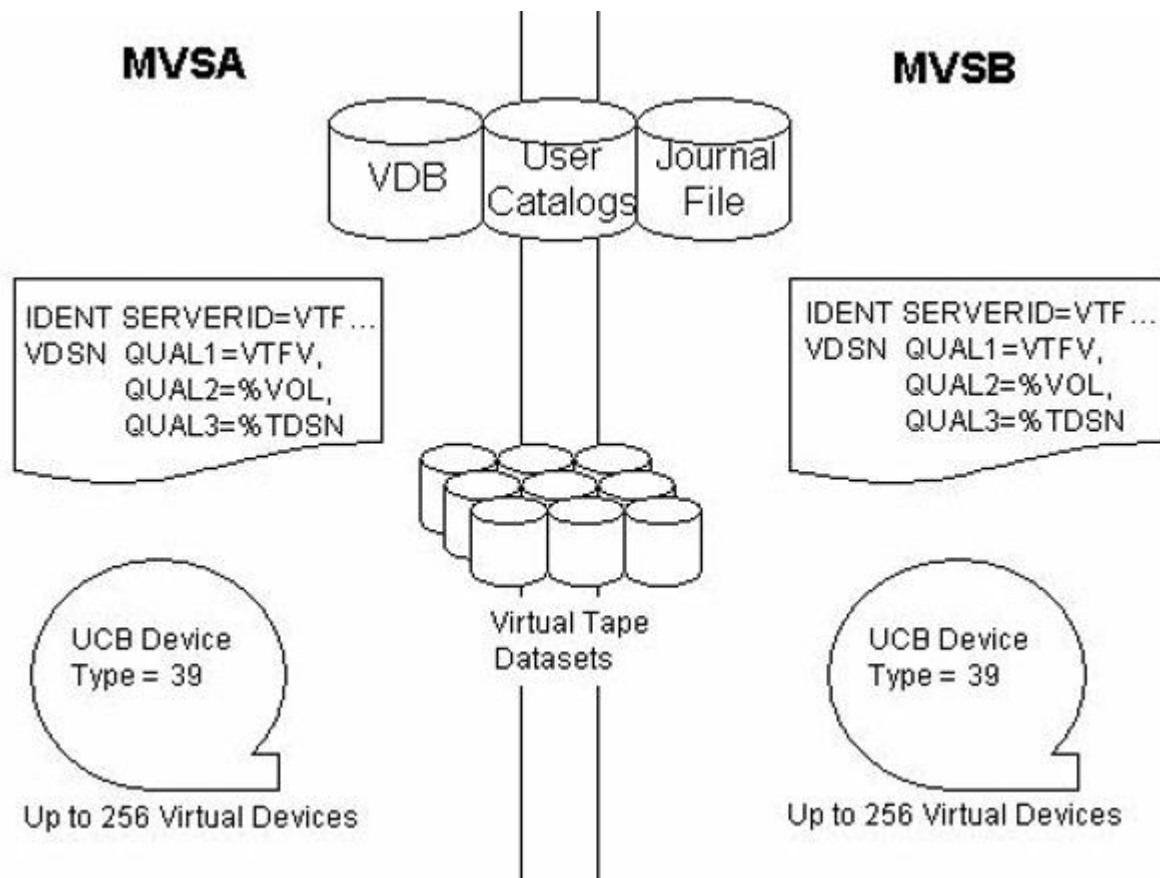


Figure 59. VTE complex configuration in multiple MVS systems

Running multiple VTE servers in a single MVS System

Multiple VTE servers may run concurrently in the same system. That may occur when managing separate virtual tape libraries or when testing a new version of VTE.

The following requirements must be satisfied in order to run multiple VTE servers in a single MVS system:

- Each VTE server must have its own initialized VDB file.
- Each VTE server must have its own journal file.
- Each of the device numbers grouped under the VTE virtual devices esoteric name should not overlap.
- Each VTE server should have its own device unit type, MVS generic name and UCB device type. This ensures that virtual tapes belonging to one VTE server will not be mounted on another VTE server's devices.
- The following parameters in the initialization statements of the different VTE servers should have different values:
 - SERVERID value (as specified in the IDENT initialization statement).
 - ALIASPFX value (as specified in the PAT initialization statement). This holds even when PAT is not ENABLE-d.
 - Subsystem name (as specified in the SUBSYS initialization statement).
 - PORT value (as specified in the TCP initialization statement). This holds only if the TCP option is ENABLE-d.
 - ESOTERIC value (as specified in the VDEVICE initialization statement).
 - VCHPID value(s) (as specified in the VDEVICE initialization statement).
 - QUAL1 value (as specified in the VDSN initialization statement). The different ALIASes may be defined in the same user catalog

The user (through VTE EXIT002) can define the device unit type, the MVS generic name and the UCB device type of all servers.

Note that EXIT002 (as part of the VTE UIM) is defined only once in the system (regardless of the number of VTE servers or installations).

Therefore, make sure that only one VTE installation contains the EXIT002 definitions for all the installed servers in the system.

The following illustration depicts the configuration of two VTE servers in a single MVS system:

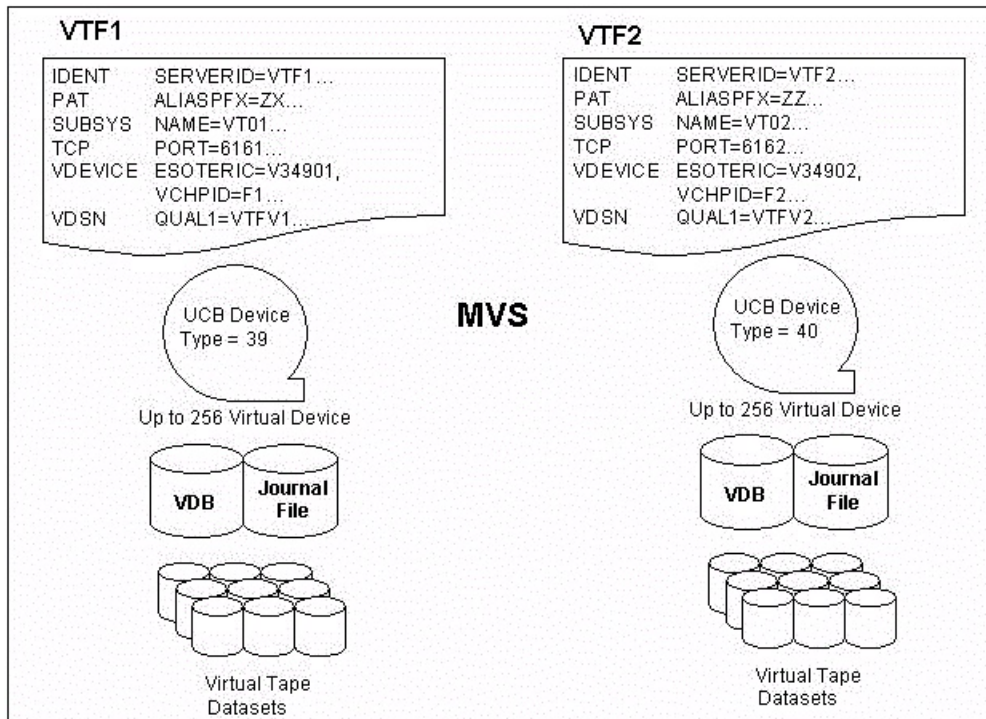


Figure 60. Multiple VTE (VTF) servers configuration in a single MVS system

Virtual Tape Sharing

VTE allows the sharing of virtual tapes for read access. When the virtual tape library is shared among different systems, jobs from up to four different systems can mount the same virtual tape concurrently for read (as if DISP=SHR is specified in the corresponding DD statement). All this is possible without any special user intervention.

Additionally, with the implementation of Parallel Access Tape (PAT), specific DD statements, that are found to be PAT-Eligible, can mount the same virtual tapes for read in the same system or in a shared VTE virtual tape library.

Sharing Virtual Tapes from multiple MVS Systems

If VTE is operated in a VTE complex, virtual tapes can be written in one system, then accessed from other systems. Read access to the virtual tapes can be performed concurrently from up to four different MVS systems in the VTE complex without any special definitions.

Sharing Virtual Tapes in single MVS system - Parallel Access Tape

Tapes, by nature, are exclusively used by one user at a time. When a tape is mounted on a tape drive, it is opened and available for read or write operations by the user, who requested the mount. Given that there is only one entity that accesses the tape at a time, the one who receives ownership of the tape can perform read/write operations without any sharing considerations. In order to protect the tape from sharing violations, and to serialize the access to tapes, some protection mechanisms are established. One

of the mechanisms is the SYSZVOLS enqueue, which serializes mount requests for tapes that are already mounted.

VTE virtual tapes are allocated on disk storage. This fact provides the ability to overcome the limitations of real tapes, since the access to data sets on disks can be shared.

However, the protection mechanisms that are integrated within the operating system, treat the virtual tapes just as real tapes, and enforce the same policy over virtual tapes.

Using Parallel Access Tape (PAT), the serialization and protection mechanisms that prevent the concurrent access to virtual tapes can be bypassed for virtual tapes that are used for read according to pre-defined criteria.

PAT is implemented by substituting the explicitly specified tape volumes or by adding tape volumes on DD statements that specify cataloged data sets by PAT alias volumes.

A mount request for PAT-Eligible DD statement will be satisfied by using the PAT alias volume for the requested virtual tape. This way, up to 10,000 DD statements of multiple jobs can mount the same virtual tape for read. Each mount will use a different PAT alias volume. The use of PAT alias volumes allows the bypass of all the protection mechanisms that serialize the mount of the same tape volume.

Setting PAT eligibility

You can select one of the following three methods to implement PAT and mark specific DD statements as PAT-Eligible:

Disposition

DD statements that specify DISP=SHR for tape allocations. This method considers that DD statements that define a disposition of SHR, and intend to use the virtual tapes as input, should be marked automatically as PAT-Eligible.

DCB

DD statements that specify DCB=CPRI=S for tape allocations. This method gives the user the option to change JCL such that only specific DD statements will be marked as PAT-Eligible.

PAT rules

DD statements can be marked as PAT-Eligible according to certain criteria defined by the PAT rules, without any changes to existing JCL.

The method upon which DD statements are marked as PAT-Eligible is defined by the **INCLUDE** parameter of the PAT initialization statement in the HVTOPTx member.

The following should be considered when marking a DD statement as PAT-Eligible:

- A PAT-Eligible DD statement is able to mount a tape for read operation (e.g. input) only.
- Due to the use of PAT alias volumes to enable PAT, the use of the virtual tapes for read is not recorded by the tape management system. The tape management system refers to the PAT alias volumes as external tapes.

When a job starts execution it is scanned by the VTE subsystem for PAT-Eligibility. If a DD statement is marked as PAT-Eligible according to one of the above methods, a PAT alias volume is assigned to the mount request instead of the tape volume that was explicitly or implicitly coded in the DD statement. The mount request is issued for the PAT alias volume with EXPDT=98000 such that the tape management system ignores it. When the PAT alias volume is mounted, the HVT022I message is issued, indicating the original and the PAT alias volume.

```
HVT022I MOUNTTAP ACCEPTED FOR VOLUME=DT0812(ZZ0001), CUU=0547 ($$DFLT
```

All the read operations are satisfied from the virtual tape dataset that contains the expected virtual tape (represented by the PAT alias volume). If a write operation is attempted by the job, the VTE server fails the job with the HVT332E message.

When using PAT alias volumes, the original tape volume is hidden and the tape management system does not update its database with the original tape volume usage.

Care should be taken when reports or statistics are derived from the database, as they do not reflect the real use of the tape. In addition, certain scratch controls should be revised when using PAT. For example, Days Since Last Used Control may cause tapes to get prematurely scratched even though they were used (through a PAT alias volume).

The PAT rules

To implement PAT without changing the JCL of production jobs, the user may specify certain criteria for VTE to dynamically mark specific DD statements as PAT-Eligible. The criteria are defined using PAT rules and may contain several filters or filter lists like jobname, stepname, datasetname, SMS classes, and more.

The PFILT initialization statement defines the filter list name, while the PPTRN initialization statement contains the values associated with each filter list.

The PRULE initialization statements in conjunction with the PFILT and PPTRN initialization statements define the PAT-Eligibility criteria. The PFILT initialization statement defines the filter list name, while the PPTRN initialization statement contains the values associated with each filter list. The PFILT, PPTRN, and PRULE initialization statements are located in member HVTPRLxx in the SAMPLIB library.

During initialization, the VTE server reads the PAT rules and loads them to the Extended Common System Area (ECSA). When a job starts execution or a tape device is dynamically allocated, the PAT rules are matched against the job's DD statements and when a match is found, this DD statement is dynamically marked as PAT-Eligible.

There are two types of PRULE initialization statements:

- INC type

Each INC type initialization statement starts a new PRULE initialization statement group that contains none, one or more EXC type initialization statements.

- EXC type

Each EXC type initialization statement is associated with its preceding INC type initialization statement and defines EXclude conditions for the INclude definitions in that specific PRULE initialization statements group.

For more information on how to code PAT rules, refer to [Chapter 30, “VTE Initialization Statements,” on page 251](#).

The PAT rule definitions can be dynamically modified by swapping HVTPRLxx members with different set of PAT rules.

PAT restrictions

The following environments or JCL specifications will cause the respective virtual tapes allocation not to be PAT-Eligible:

- Environments
 - Static (batch) allocations in JES3 environments.
- Special DD definitions
 - DDs under a JOBCAT/STEPDAT range.
 - DDs that contain VOL=REF=* . . or VOL=REF=DSN or are REF-ed by another DD statement in the job.
 - Cataloged dataset that contains a UNIT=AFF request to an unlike device type.

Under normal circumstances the system detects the AFF request to an unlike device, issues the IEF245I message and fails the job.

If the PAT facility adds PAT alias volumes and a unit value, the dataset looks like a not-cataloged dataset so the system is unable to detect the error and the virtual tape is mounted on a real device.

- DDs that define GDG-ALL requests (that is, specify the GDG name without a generation number in the DSNNAME parameter).
- DDs that contain the CHKPT=EOV parameter.

IBM Checkpoint retains the mounted volume serial number and requests its mount, should the job be restarted.

If the PAT facility replaces the volume, that volume will be requested by the restart, but that volume does not really exist.

- Subsystem files.
- Dummy DDs.

- Special jobs

- Since the PAT facility uses PAT alias volumes, no DD statement that catalogs, recatalogs or uncatalogs the dataset should be PAT-Eligible. The following is a detailed list of these cases:
- Catalog or re-catalog in the same job:
- Data sets cataloged or re-cataloged in the current DD.
- For data sets cataloged or re-cataloged in a previous step, VTE picks the volume serial numbers when the job enters the system (before the re-catalog takes place).
- For data sets cataloged or re-cataloged in a next step, there is a chance that JobRestart products (like Control-R or CA-11) uncatalog data sets due to GDG adjustment, prevent 'not catlg 2' situation, and so on.
- DISP=MOD:

Under certain circumstances the system may re-catalog the dataset.

- DISP=(xxx,CATLG) or DISP=(xxx,xxx,CATLG).
- DISP=(xxx,DELETE) or DISP=(xxx,xxx,DELETE):

For cataloged dataset (that is, no VOLSER or UNIT are coded), the system uncatalogs the dataset.

For this cataloged dataset, if the PAT facility adds PAT alias volumes and a unit value, the dataset looks like a not-cataloged dataset so the system does not uncatalog it.

Due to this inconsistency the DD statement is not PAT-Eligible.

Note that if DISP=(xxx,UNCATLG) or DISP=(xxx,xxx,UNCATLG) the dataset is always uncataloged.

- DISP=NEW:

Implies DISP=(NEW,DELETE).

- No DISP at all:

Implies DISP=(NEW,DELETE).

- Special applications

Certain common products that use dynamic allocation to allocate virtual tapes are not able to use PAT. The reason is that their internal logic does not allow for concurrent mount of the same tape. Examples of such products are IBM's DFSMSHsm and CA's CA-Disk.

- Manual modifications

- Jobs that use the IBM Checkpoint/Restart facility not through the external EOVS interface (that is, by specifying CHKPT=EOV on selected DD statements), should manually exclude these DD statements from PAT. This is so since the PAT facility cannot anticipate these Checkpoint requests.

Virtual Tape sharing examples

The following examples show how can a VTE complex be used to share virtual tapes, use different tape ranges or share virtual tapes in the same MVS system.

Example 1: Sharing a Virtual Tape in a VTE Complex (without PAT)

The following scenario depicts the process that occurs when different MVS systems that comprise a VTE complex try to concurrently access the same virtual tape.

Suppose that JOBA in SYSA writes a dataset on virtual tape EC0001.

Suppose also that JOBB in SYSB wishes to read the dataset being written by JOBA in MVSA on EC0001.

- If JOBA in SYSA completed and unloaded EC0001, the VTE server running in SYSB will mount and use it.
- If JOBA in SYSA has not yet completed, the VTE server running in MVS B will issue the following message:

```
HVT323W VOLUME=EC0001 (JOBA/JOB000001) REQ=READ STAT=WRITE
(SYSA) TO CANCEL WAIT REPLY 'N'
```

The operator may reply 'N' to this message in which case JOBB will not wait for JOBA to complete in SYSA and will be canceled.

Otherwise, JOBB will wait for JOBA in SYSA to complete, at which time JOBB will automatically get EC0001 mounted.

While JOBB in SYSB reads the virtual tape, other jobs from three additional MVS systems that belong to the VTE complex can mount EC0001 for read purposes.

If a job from a 5-th MVS system tries to mount the tape, message HVT323W is issued and the operator has the option to wait or cancel the job.

Example 2: Sharing a Virtual Tape in the Same MVS System (using PAT)

The following scenario depicts the process that occurs when two jobs that run in one MVS system try to concurrently access the same virtual tape for read.

The following conditions are assumed to be true:

- The PAT facility for batch jobs is enabled in the HVTOPTx member and PAT-Eligible tapes are selected according to PAT rules.
- Volume EC0001 is a virtual tape and M20.GDG2.G0017V00 data set is cataloged on it.
- The following PRULE initialization statements are coded:

```
PRULE      INC,
           JOBNAME=JOBA,DISP=SHR
PRULE      INC,
           JOBNAME=JOBB,DISP=SHR
```

The following two jobs are submitted:

```
//JOBA      JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1        EXEC PGM=IEBGENER,REGION=0M
//SYSUT1    DD DISP=(SHR),
//          DSN=M20.GDG2(0)
//SYSUT2    DD DUMMY
//SYSIN     DD DUMMY
//SYSPRINT  DD SYSOUT=*
```

```
//JOBB      JOB 0,M20,CLASS=A,MSGCLASS=X,REGION=4000K
//S1        EXEC PGM=IEBGENER,REGION=0M
//SYSUT1    DD DISP=(SHR),
//          DSN=M20.GDG2(0)
```

```
//SYSUT2 DD DUMMY
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
```

As a result, VTE will use PAT alias volumes to read the data set on SYSUT1:

```
$HASP373 JOBA      STARTED - INIT 2 - CLASS A - SYS MM90
IEF403I JOBA - STARTED - TIME=14.32.08
HVT308I /PROC      /STEP      /DDNAME /VOLSER/ALIAS /DSNAME
HVT308I / /S1      /SYSUT1    /EC0001 /ZX0000/M20.GDG2(0)
IEF233A M 0571,ZX0000,,JOBA,S1,M20.GDG2.G0017V00
IEF234E K 0571,ZX0000,PVT,JOBA,S1
        JOBA      #01 S1      IEBGENER - COMP CODE=R0000
IEF404I JOBA - ENDED - TIME=14.32.14
$HASP395 JOBA ENDED
```

```
$HASP373 JOBB STARTED - INIT 3 - CLASS A - SYS MM90
IEF403I JOBB - STARTED - TIME=14.32.09
HVT308I /PROC      /STEP      /DDNAME /VOLSER/ALIAS /DSNAME
HVT308I / /S1      /SYSUT1    /EC0001 /ZX0001/M20.GDG2(0)
IEF233A M 0570,ZX0001,,JOBB,S1,M20.GDG2.G0017V00
IEF234E K 0570,ZX0001,PVT,JOBB,S1
        JOBB      #01 S1      IEBGENER - COMP CODE=R0000
IEF404I JOBB - ENDED - TIME=14.32.14
$HASP395 JOBB ENDED
```

SYSUT1 DD statement should allocate volume EC0001. However, due to the PAT rules, two PAT alias volumes (ZX0000 and ZX0001) are substituted and mounted and message HVT308I indicates that substitution.

In addition, at unload time message HVT023I (not shown here) details the original and the associated PAT alias volumes.

If one of the jobs tried to write on the virtual tape, message HVT332E would have been issued and the job abended.

An example is shown below of a job trying to write through a PAT alias volume.

```
$HASP373 JOBA STARTED - INIT 2 - CLASS A - SYS MM90
IEF403I JOBA - STARTED - TIME=14.41.34
HVT308I /PROC      /STEP      /DDNAME /VOLSER/ALIAS /DSNAME
HVT308I / /S1      /SYSUT2    /EC0001/ZX0000/M20.GDG2(0)
IEF233A M 0571,ZX0000,,JOBA,S1,M20.GDG2.G0017V00
HVT341I *****
HVT332E * I/O ERROR ON VOLUME=EC0001, RC=40 (WRITE THRU PAT ALIAS VOLUME ) *
HVT342E * JOBA      /JOB04186 I/O FAILED BY 0_RDIREC SERVER - EQC GENERATED *
HVT341I *****
IOS000I 0571,FB,EQC,07,0E00,,**,ZX0000,JOBA 001
1040202C4000000200000(0000000000000000)0070(00000000)878F1E7114720000
IEC513D I/O ERR 0571,ZX0000,SL,JOBA,S1,M20.GDG2.G0017V00
58 IEC513D REPLY 'U'-ABEND OR 'M'-MOUNT SCRATCH
R 58,U
IEC502E R 0571,ZX0000,SL,JOBA,S1
IEC147I 613-10,IFG0196T,JOBA,S1,SYSUT2,0571,,M20.GDG2.G0017V00
```

Example 3: Sharing a virtual tape library in a VTE complex when each system uses different virtual tapes

The following example details the steps to be taken when two MVS systems that comprise a VTE complex have dedicated volumes range and each system should use only its volumes (that is, not use scratch virtual tapes from the other system's range).

The following conditions are assumed to be true:

- There are two systems, SYSA and SYSB, that share the same VTE complex. SYSA virtual tapes range is A00000-A09999 and SYSB virtual tapes range is B00000-B09999.
- The local tape manager does not use tape pools for its mounted tapes.

The requirement can be achieved if each system will define two tapepools—a specific tapepool and the default tapepool.

In the specific tapepool, each system will define the other system's range.

An example below shows separate tape ranges in a VTE complex.

```
TPPOOL definitions in SYSA
TPPOOL NAME=T_SYSA,
        DPOOL=dpool,
        RANGE1=(B000000,B099999) ,
        ..
        ..
TPPOOL NAME=$$DFLT,
        DPOOL=dpool,
        RANGE1=(0,0) ,
        ..
        ..
```

```
TPPOOL definitions in SYSB
TPPOOL NAME=T_SYSB,
        DPOOL=dpool,
        RANGE1=(A000000,A099999) ,
        ..
        ..
TPPOOL NAME=$$DFLT,
        DPOOL=dpool,
        RANGE1=(0,0) ,
        ..
        ..
```

When the tapepools are defined as shown above, mount requests for scratch virtual tapes will be satisfied from the default tapepool which contains all the virtual tapes, except the ones defined in the specific tapepool or, in other words, the virtual tapes that belong to that system.

This example can be easily generalized to more than two systems. In such cases, each system defines the other systems virtual tapes ranges in specific tapepools leaving its own virtual tapes range undefined (or defaulted to the default tapepool).

Chapter 16. Dynamic Reallocation Logic

This chapter describes how the VTE subsystem is used in order to allocate VTE virtual devices to jobs without changing their JCL.

Dynamic Reallocation Rules

In order to implement VTE without changing the JCL of production jobs, the user may specify certain criteria for VTE to dynamically reallocate tape allocations from a real tape device to the VTE virtual tape device. The criteria are defined using dynamic reallocation rules and may contain several filters or filter lists like jobname, stepname, datasetname, SMS classes, and more.

The RFILT initialization statement defines the filter list name, while the RPTRN initialization statement contains the values associated with each filter list.

The RULE initialization statements, in conjunction with the RFILT and RPTRN initialization statements, define the dynamic reallocation criteria. The RFILT, RPTRN and RULE initialization statements are located in member HVTRULxx in the SAMPLIB library.

During initialization, the VTE server reads the dynamic reallocation rules and loads them to Extended Common System Area (ECSA). When a job starts execution or a tape device is dynamically allocated, the dynamic reallocation rules are matched against the job's DD statements or the dynamic allocation parameters. When a match is found, the allocation defined by this DD statement or the dynamic allocation request is dynamically changed to allocate a VTE virtual device.

There are two types of RULE initialization statements:

INC type

Each INC type initialization statement starts a new RULE initialization statement group that contains none, one or more EXC type initialization statements.

EXC type

Each EXC type initialization statement is associated with its preceding INC type initialization statement and defines EXclude conditions for the INclude definitions in that specific RULE initialization statements group.

For more information on how to code dynamic reallocation rules, refer to [Chapter 30, "VTE Initialization Statements,"](#) on page 251.

The dynamic reallocation rule definitions can be dynamically modified by swapping HVTRULxx members with different set of dynamic reallocation rules.

Dynamic Reallocation Decision Logic

The dynamic reallocation rules are used to let VTE reallocate real device allocations to VTE virtual device allocations. This dynamic reallocation process is performed by the VTE subsystem. The factors and logic according to which the VTE subsystem reallocates DD statements allocations or dynamically allocated tape devices are described below.

Terminology

The following terminology applies to the dynamic reallocation process.

Must reallocate

An indication that dynamic reallocation must be done for a specific DD statement regardless of the dynamic reallocation rules specifications.

Never reallocate

An indication that dynamic reallocation will never be done for a specific DD statement regardless of the dynamic reallocation rules specifications.

VOL=REF group

All the DD statements that are related through VOL=REF parameters. The VOL=REF specification forces all the related datasets to reside on the same volume

UNIT=AFF chain

All the DD statements that are related through a UNIT=AFF chain. The UNIT=AFF specification forces all the related DD statements to allocate their volumes on the same unit.

Making the decision

Before checking the dynamic reallocation rules, several special DD statements are marked as **must reallocate** or **never reallocate** and others are not considered for dynamic reallocation.

- System jobs (jobs whose jobname is IEESYSAS) are never subject to dynamic reallocation.
- The following DD statements are not considered for dynamic reallocation:
 - DD statements that refer to cataloged data sets (that is, they do not specify VOL or UNIT parameters) are assumed to be cataloged correctly.
 - DD statements that specify a UNIT parameter with the esoteric name of the VTE virtual devices.
 - DD statements that do not specify a UNIT parameter (explicitly or implicitly using an SMS class).
 - DD statements that specify a device number of a VTE virtual device (by explicitly specifying UNIT=cuu or UNIT=/cuu).
- The following DD statements cannot be dynamically reallocated.
 - Tape allocations of programs that access the catalog directly (like IEHMOVE, IEHPROGM or IDCAMS).
 - The following special DD statements are marked as never reallocate:
 - STEPLIB, STEPCAT, JOBLIB, and JOBCAT DD statements.
 - Subsystem files (instream data, sysout files, etc.).
 - DUMMY DD statements.
 - DD statements that refer to members of partitioned data sets.
- If a DD statement explicitly specifies a 'known' virtual tape (using VOL=SER parameter), it is marked as must reallocate. If this DD statement leads a VOL=REF group, all related DD statements are marked as must reallocate. A hyper '**never-reallocate**' parameter can be added to a DD statement in order to prevent the dynamic reallocation of specific 'known' tapes.
- If a DD statement explicitly specifies an 'unknown' virtual tape (using VOL=SER parameter), it is marked as never reallocate. If this DD statement leads a VOL=REF group, all related DD statements are marked as never reallocate.

After the never reallocate or must reallocate indications are set, each DD statement is checked for dynamic reallocation eligibility according to the dynamic reallocation rules.

- The DD statement values are successively matched against each INC type dynamic reallocation rule that starts a new dynamic reallocation rules group. Each rules group starts with an INC type dynamic reallocation rule, and may contain several EXC type dynamic reallocation rules. If no INC type dynamic reallocation rule is found to match, the DD statement is skipped (not marked for dynamic reallocation).

The INC type dynamic reallocation rule is considered to match a given DD statement if all its specified parameters match the respective parameters of the DD statement.

- The VTE subsystem matches the DD statement parameters with each INC type rule. For the first matched INC type rule, the DD statement values are matched successively against all the associated EXC type rules in that rules group.

The EXC type dynamic reallocation rule is considered to match a given DD statement if all its specified parameters match the respective parameters of the DD statement.

- If one EXC type rule matches the DD statement, this DD statement is marked for EXclusion.
- If no EXC type rule matches the DD statement, this DD statement is marked for INclusion.

A dynamic reallocation decision is made according to the first matching INC type dynamic reallocation rule.

- As a result of the above rules matching test, each DD statement is marked as INCludable, EXCludable or neither INCludable nor EXCludable. If the DD statement is part of a VOL=REF group, and is marked as INCludable or EXCludable, all related DD statements are marked as INCludable or EXCludable, respectively.

After the reallocation eligibility is set, VTE validates this eligibility consistency:

- If at least one DD statement is marked as never reallocate and INCludable, message HVT297S is issued and the job fails with a JCL error.
- If at least one DD statement is marked as must reallocate and EXCludable, message HVT297S is issued and the job fails with a JCL error.
- If at least one DD statement specifies a volumes list (the VOL=SER parameter) in which part of the tapes are real and other parts are virtual, message HVT292E is issued and the job fails with a JCL error.
- For each DD statement that is marked both INCludable and EXCludable, the EXClude overrides the INClude, the DD statement is marked as EXCludable and message HVT298W is issued.

Re-chain the original UNIT=AFF chains:

- If no previous error has been encountered, UNIT=AFF chains in which part of the DD statements are marked as never reallocate/EXCludable and part are marked as must reallocate/INCludable, are re-chained in two different UNIT=AFF chains.

Perform the actual dynamic reallocation:

- All DD statements marked as **must reallocate** or **INCludable** are reallocated by substituting the UNIT parameter values with the ESOTERIC parameter value of the VDEVICE initialization statement.

Dynamic Reallocation Examples

The examples in this section illustrate the use of dynamic reallocation rules in order to dynamically reallocate real 3490 allocations to VTE virtual devices.

The following conditions are assumed to be true for the examples in this section:

- Volume 000001 is a *known* virtual tape
- Volume 000002 is an *unknown* virtual tape
- The following RULE initialization statements are coded:

Below is an example of dynamic reallocation rules.

```

*
RULE      INC,                               INC (1)
          JOBNAME=M21*,
          UNIT=3490,
          DSN=M20.*.G????V??
          RULE EXC,                           EXC (1)
          PGM=ABCD*,
          STEP=COPY01

*
RULE      INC,                               INC (2)
          JOBNAME=M20*,
          UNIT=3490
          RULE EXC,                           EXC (2)
          PGM=ABCD*,
          STEP=COPY01,
          DSN=M20.TEST

*
RULE      INC,                               INC (3)
          JOBNAME=M22*,
          DATACLAS=DCELCANA
*

```

The above dynamic reallocation rules specify that dynamic reallocation is done for jobs that meet the following criteria:

- Their jobname starts with either M20, M21 or M22.
- Their original UNIT is 3490.
- For jobs whose jobname starts with M21:
 - The allocated data set is a GDG and its name starts with M20.
 - Their PGM does not start with ABCD and their stepname is not COPY01.
- For jobs whose jobname starts with M20:
 - Their PGM does not start with ABCD, their stepname is not COPY01, and the dataset name is not M20.TEST.
- For jobs whose jobname starts with M22:
 - The Data Class is DCELCANA.

In the output display, device number 0500 is a real 3490 device and device number 0571 is a VTE virtual device. The esoteric name of the VTE virtual device is VT90EC.

The dynamic reallocation decision trace has been turned on in order to depict the dynamic reallocation logic – the trace messages are displayed under the HVT995I message.

Example 1

When the following job is submitted:

```
//M21JOB      JOB
//COPY01      EXEC PGM=IEFBR14
//OUT01       DD DISP=(,PASS),
//              DSN=M20.GDG(+1),
//              UNIT=3490,
//              DCB=M20.LIB.JOB
//OUT02       DD DISP=(,PASS),
//              DSN=M20.TEST,
//              UNIT=AFF=OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB   STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I  M21JOB - STARTED
HVT995I  STEP     DDNAME   VOLUME+ R#  A#  M/N+  I/E+  PTH  DSNAME
HVT995I  COPY01   OUT01           51          INC
HVT995I  COPY01   OUT02           01
HVT306I  /PROC     /STEP     /DDNAME  /UNIT  /DATACLAS /STORCLAS /MGMTCLAS ->NEWUNIT
HVT306I  /         /COPY01  /OUT01   /3490  /         /         /         ->VT90EC
*IEF233A M 0571,PRIVAT,SL,M21JOB,COPY01,M20.GDG.G0027V00
*IEF233A M 0500,PRIVAT,SL,M21JOB,COPY01,M20.TEST
M21JOB   .COPY01   .#01; - COMPLETION CODE=0000
IEF234E  K 0571,,PVT,M21JOB
IEF404I  M21JOB - ENDED
$HASP395 M21JOB ENDED
```

OUT01 DD statement will be reallocated due to INC(1).

OUT02 DD statement will not be reallocated because there is no rule matching its DD statement.

The UNIT=AFF chain will be re-chained because of the dynamic reallocation.

OUT01 DD statement will allocate a VTE virtual device while OUT02 DD statement will allocate a 3490 device.

Example 2

When the following job is submitted:

```
//M21JOB      JOB
//COPY01      EXEC PGM=IEFBR14
//OUT01       DD DISP=(,PASS),
```



```
//          DSN=M20.GDG(+1),
//          UNIT=3490,
//          DCB=M20.LIB.JOB
//OUT02     DD DISP=(,PASS),
//          DSN=M20.TEST,
//          UNIT=AFF=OUT01,
//          VOL=REF=*.COPY01. OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB   STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I  M21JOB - STARTED
HVT995I  STEP     DDNAME   VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I  COPY01   OUT01           01 01      INC+
HVT995I  COPY01   OUT02           01 01      INC
HVT306I  /PROC    /STEP     /DDNAME /UNIT   /DATACLAS /STORCLAS/MGMTCLAS ->NEWUNIT
HVT306I  /        /COPY01   /OUT01   /3490   /          /          ->VT90EC
*IEF233A M 0571,PRIVAT,SL,M21JOB,COPY01,M20.GDG.G0027V00
M21JOB   .COPY01 .#01; - COMPLETION CODE=0000
IEF234E  K 0571,,PVT,M21JOB
IEF404I  M21JOB - ENDED
$HASP395 M21JOB   ENDED
```

OUT01 DD statement will be reallocated due to INC(1).

OUT02 DD statement will be automatically reallocated since both DD statements refer to the same tape volume.

The UNIT=AFF chain remains unchanged.

Both OUT01 and OUT02 DD statements will allocate the same VTE virtual device.

Example 3

When the following job is submitted:

```
//M21JOB     JOB
//COPY01     EXEC PGM=IEFBR14
//OUT01      DD DISP=(,PASS),
//          DSN=M20.TEMP.FILE,
//          UNIT=3490,
//          VOL=SER=000001
//OUT02      DD DISP=(,PASS),
//          DSN=M20.TEST,
//          UNIT=AFF=OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB   STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I  M21JOB - STARTED
HVT995I  STEP     DDNAME   VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I  COPY01   OUT01     000001+ 51 MST+
HVT995I  COPY01   OUT02           01
HVT306I  /PROC    /STEP     /DDNAME /UNIT   /DATACLAS /STORCLAS/MGMTCLAS ->NEWUNIT
HVT306I  /        /COPY01   /OUT01   /3490   /          /          ->VT90EC
*IEF233A M 0571,000001,,M21JOB,COPY01,M20.TEMP.FILE
*IEF233A M 0500,PRIVAT,SL,M21JOB,COPY01,M20.TEST
M21JOB   .COPY01 .#01; - COMPLETION CODE=0000
IEF234E  K 0571,000001,PVT,M21JOB
IEF404I  M21JOB - ENDED
$HASP395 M21JOB   ENDED
```

OUT01 DD statement will be reallocated regardless of any dynamic reallocation rule since 000001 is a 'known' virtual tape.

OUT02 DD statement will not be reallocated because there is no rule matching its DD statement.

The UNIT=AFF chain will be re-chained because of the dynamic reallocation.

OUT01 DD statement will allocate a VTE virtual device while OUT02 DD statement will allocate a 3490 device.

Example 4

When the following job is submitted:

```
//M21JOB      JOB
//COPY01     EXEC PGM=IEFBR14
//OUT01      DD DISP=(,PASS),
//           DSN=M20.TEMP.FILE,
//           UNIT=3490,
//           VOL=SER=EC0001,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000,CPRI=E)
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AFF=OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB  STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I M21JOB - STARTED
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I COPY01 OUT01 000001+ 01 NVR!
HVT995I COPY01 OUT02 01
*IEF233A M 0500,000001,,M21JOB,COPY01,M20.TEMP.FILE
M21JOB .COPY01 .#01; - COMPLETION CODE=0000
IEF234E K 0500,000001,PVT,M21JOB
IEF404I M21JOB - ENDED
$HASP395 M21JOB ENDED
```

OUT01 will not be reallocated although 000001 is a 'known' virtual tape since the hyper 'never-reallocate' parameter was specified on it.

OUT02 DD statement will not be reallocated because there is no rule matching its DD statement.

The UNIT=AFF chain remains unchanged.

Both OUT01 and OUT02 DD statements will allocate a 3490 device.

Example 5

When the following job is submitted:

```
//M21JOB      JOB
//COPY01     EXEC PGM=ABCD
//OUT01      DD DISP=(,PASS),
//           DSN=M20.GDG(+1),
//           UNIT=3490,
//           DCB=M20.LIB.JOB
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AFF=OUT01,
//           VOL=REF=*.COPY01.OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB  STARTED - INIT 2 - CLASS A - SYS MM90
IEF403I M21JOB - STARTED
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I COPY01 OUT01 01 01 EXC+
HVT995I COPY01 OUT02 01 01 EXC
*IEF233A M 0500,PRIVAT,SL,M21JOB,COPY01,M20.GDG.G0027V00
M21JOB .COPY01 .#01; - COMPLETION CODE=0000
IEF404I M21JOB - ENDED
$HASP395 M21JOB ENDED
```

OUT01 DD statement will not be reallocated due to INC(1) and EXC(1).

OUT02 DD statement will not be reallocated since both DD statements refer to the same tape volume.

Both OUT01 and OUT02 DD statements will allocate the same 3490 device.

Example 6

When the following job is submitted:

```
//M21JOB      JOB
//COPY01     EXEC PGM=IEFBR14
//OUT01      DD DISP=(,PASS),
//           DSN=M20.GDG(+1),
//           UNIT=3490,
//           VOL=SER=000002,
//           DCB=M20.LIB.JOB
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AF=OUT01,
//           VOL=REF=*.COPY01. OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M21JOB   STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I M21JOB - STARTED
HVT341I *****
HVT297E * REALLOCATION FAILED FOR S.D=COPY01 .OUT01 , RC=01 (NVR/INC) *
HVT297E * REALLOCATION FAILED FOR S.D=COPY01 .OUT02 , RC=01 (NVR/INC) *
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNNAME
HVT995I COPY01 OUT01 000002- 01 01 NVR+ INC+
HVT995I COPY01 OUT02 01 01 NVR INC
HVT294E * JOB FAILED BY 0_RDIREC SERVER DUE TO DYNAMIC REALLOCATION ERROR(S)*
HVT341I *****
M21JOB .COPY01 .#01; - COMPLETION CODE=NOT RUN
IEF404I M21JOB - ENDED
$HASP395 M21JOB ENDED
```

OUT01 DD statement would have been reallocated, due to INC (1). However, volume 000002 is an 'unknown' virtual tape. Due to this conflict, message HVT297E is issued and the job fails with a JCL error.

Example 7

When the following job is submitted:

```
//M20JOB      JOB
//COPY01     EXEC PGM=ABCD
//OUT01      DD DISP=(,PASS),
//           DSN=M20.GDG(+1),
//           UNIT=3490,
//           VOL=SER=000001,
//           DCB=M20.LIB.JOB
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AF=OUT01,
//           VOL=REF=*.COPY01. OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M20JOB   STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I M20JOB - STARTED
HVT298W S.D=COPY01 .OUT01 (INC) SUPP_ED BY S.D=COPY01 .OUT02 (EXC)
HVT341I *****
HVT297E * REALLOCATION FAILED FOR S.D=COPY01 .OUT01 , RC=02 (MST/EXC) *
HVT297E * REALLOCATION FAILED FOR S.D=COPY01 .OUT02 , RC=02 (MST/EXC) *
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNNAME
HVT995I COPY01 OUT01 000001+ 01 01 MST+ EXC
HVT995I COPY01 OUT02 01 01 MST EXC+
HVT294E * JOB FAILED BY 0_RDIREC SERVER DUE TO DYNAMIC REALLOCATION ERROR(S)*
HVT341I *****
M20JOB .COPY01 .#01; - COMPLETION CODE=NOT RUN
IEF404I M20JOB - ENDED
$HASP395 M20JOB ENDED
```

OUT01 DD statement would have been reallocated since volume 000001 is a 'known' virtual tape.

OUT02 DD statement would have been EXcluded due to INC(2) and EXC(2).

Because both DD statements use the same tape volume, message HVT297E is issued and the job fails with a JCL error.

Example 8

When the following job is submitted:

```
//M20JOB      JOB
//COPY01     EXEC PGM=ABCD
//OUT01      DD DISP=(,PASS),
//           DSN=M20.GDG(+1),
//           UNIT=3490,
//           DCB=M20.LIB.JOB
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AFF=OUT01,
//           VOL=REF=*.COPY01. OUT01
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M20JOB   STARTED - INIT 1   - CLASS A - SYS MM90
IEF403I M20JOB - STARTED
HVT298W S.D=COPY01 .OUT01 (INC) SUPP ED BY S.D=COPY01 .OUT02 (EXC)
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I COPY01 OUT01 01 01 EXC
HVT995I COPY01 OUT02 01 01 EXC+
*IEF233A M 0500,PRIVAT,SL,M20JOB,COPY01,M20.GDG.G0027V00
M20JOB .COPY01 .#01; - COMPLETION CODE=0000
IEF404I M20JOB - ENDED
$HASP395 M20JOB ENDED
```

OUT01 DD statement would have been reallocated due to INC(2), but OUT02 DD statement is EXCLUDED due to EXC(2). In this case, message HVT298W is issued and no dynamic reallocation occurs.

Example 9

When the following job is submitted:

```
//M22JOB      JOB
//COPY01     EXEC PGM=IEFBR14
//OUT01      DD DISP=(,PASS),
//           DSN=M20.FILE.TEST,
//           UNIT=3490
//OUT02      DD DISP=(,PASS),
//           DSN=M20.TEST,
//           UNIT=AFF=OUT01
```

And the following Data Class ACS routine is used:

```
PROC DATACLAS
FILTLIST TAPES INCLUDE('3420','3480','TAPE','3490','CART8',
                     'CART9')
FILTLIST M20 INCLUDE(M20.FILE.***)
IF &UNIT=&TAPES AND &DATACLAS='' THEN
  SELECT (&DSN)
  WHEN (&M20) SET &DATACLAS='DCELCANA'
END
END /* END PROC STATEMENT */
```

The following job's log shows the dynamic reallocation results:

```
$HASP373 M22JOB   STARTED - INIT 1   - CLASS A - SYS MM90
IEF403I M22JOB - STARTED - TIME=16.36.13
HVT995I STEP DDNAME VOLUME+ R# A# M/N+ I/E+ PTH DSNAME
HVT995I COPY01 OUT01 51 INC
HVT995I COPY01 OUT02 01
HVT306I /PROC /STEP /DDNAME /UNIT /DATACLAS/STORCLAS/MGMTCLAS ->NEWUNIT
HVT306I / /COPY01 /OUT01 /3490 /DCELCANA/ / ->VT90EC
*IEF233A M 0571,PRIVAT,SL,M22JOB,COPY01,M20.FILE.TEST
```

```

*TMS001 IEF233A M 0571,PRIVAT,SL,M22JOB,COPY01,M20.FILE.TEST
*IEF233A M 0500,PRIVAT,SL,M22JOB,COPY01,M20.TEST
*TMS001 IEF233A M 0500,PRIVAT,SL,M22JOB,COPY01,M20.TEST
M22JOB #01 COPY01 IEFBR14 - COMP CODE=R0000
IEF234E K 0571,,PVT,M22JOB
TMS014 IEF234E K 0571,,PVT,M22JOB
IEF404I M22JOB - ENDED - TIME=16.36.13
$HASP395 M22JOB - ENDED

```

OUT01 DD statement is assigned a Data Class value from the Data Class ACS routine.

OUT01 DD statement will be reallocated due to INC(3).

OUT02 DD statement will not be reallocated because there is no dynamic reallocation rule matching its DD statement.

The UNIT=AFF chain will be re-chained because of the dynamic reallocation.

OUT01 DD statement will allocate a VTE virtual device while OUT02 DD statement will allocate a 3490 device.

Chapter 17. Managing the Virtual Tape Library

This chapter describes the management of the virtual tape library and its activity tracking. It explains the structure of the virtual tape data sets, the VDB and its associated journal file. Special attention is given to the recoverability, availability and serviceability (RAS) of the virtual tape library.

Overview

The key factor in the VTE product data reliability and recoverability is its internal virtual tape datasets and its VDB structure and management. Smart management of these data sets achieves the following goals:

- Eliminate the chance of Sx37 abends. Sx37 abends occur only when there is no more space on the disks eligible for virtual tape data sets allocation (or the VTOC on these disks is full).
- Consistent size of the virtual tape.
- Maximize the used space on each given disk track. This is achieved by dynamic segmentation of the tape blocks written to the disk so that its tracks are fully utilized.
- Minimize the amount of allocated disk space. This is achieved by allocating the virtual tape datasets as small disk extents on one or more disks (as needed).
- Eliminate the need of tape data sets stacking. This is so since no extra disk space is allocated.
- Detect and warn whenever an application attempts to read bad data, that is, the data written when an IPL occurred, while the VTE server was forced down, when a virtual tape disk extent has been restored without proper recovery, and so on.
- Full ability to recover and correlate one or more virtual tape data sets and their corresponding VDB entries as a result of a VDB entry lost, whole VDB lost, virtual tape dataset restore, etc.

Virtual Tape Database - VDB

VTE maintains a database to monitor the virtual tapes and the virtual tape data sets – the VDB. The VDB is a VSAM KSDS file accessed with the virtual tape volume serial number as a key. All the virtual tapes are mapped in the VDB with their status and some statistics.

The first record in the VDB is a control record that relates the VDB to its VTE server and holds synchronization information.

The VDB is accessed when virtual tapes are mounted and unloaded. When a mount for a scratch virtual tape is requested (either from a specific tapepool or from the default tapepool), VTE searches the VDB in order to find an appropriate scratch virtual tape.

In a VTE complex, the VDB is shared among several VTE servers.

DITTO/ESA VSAM Browse			
RBA 0	Key CONTROL	Col 1	Format CHAR
VOLSER	Type KSDS DSNAME HVTT.V130.VDBEC		
0	Control Record	<==5=>.10....5...20....5...30....5...40....5...50....5...60	
256	CONTROL0_RDIRC0101240F20070070A.. 0..V.0101247F11353749....		
512	VBC0006.MM90M20V.VBC0006.M20.EC1005	
768	VBC6001.MM90M20V.VBC6001.M20.EC6003	
1024	VBC6002.	
1280	VBC6003.	
1536	VBC6004.MM10M20V.VBC6004.M20.EC6004	
	VBC6005.MM10M20V.VBC6005.M20.EC6005	
****	End of data		****

Figure 61. Browsing a VDB

VDB-in-Storage

To accelerate the scratch tape selection, the VDB is loaded in memory, and the search process is performed on the VDB-in-Storage. The VDB-in-Storage is also used in the following cases:

- When the VTE subsystem searches for "known" tapes. Refer to [“Dynamic Reallocation Decision Logic” on page 145](#) for more information.
- When the VTE server calculates threshold values associated with the number of virtual tapes in different tapepools. Refer to the TPOOL initialization statement in [“TPOOL” on page 271](#).

When virtual tapes are added, deleted or scratched, or a tapepool threshold interval has expired, the VDB-in-Storage is updated. The VDB is accessed at user-defined intervals to check whether other members of a VTE complex updated it, and if so, the VDB-in-Storage is reloaded.

Virtual Tape Dataset

VTE maintains one multi-volume undefined-record-format disk dataset for each virtual tape it handles. The virtual tape dataset is allocated when a virtual tape is mounted for the first time. The virtual tape dataset remains on disk as long as the virtual tape is 'active'. When the virtual tape is scratched, the virtual tape dataset is deleted from the disks.

The virtual tape dataset contains the original tape blocks, blocks representing tapemarks, and the DVD (Disk Volume Descriptor).

The first block in the virtual tape dataset contains the DVD record. The DVD contains information used to verify and process the virtual tape dataset structure on the disks, the last checkpointed location accessible on the virtual tape and more. The original tape blocks and blocks representing tapemarks follow the DVD.

The original tape blocks may be segmented so that large tape blocks can fit into the various disk type tracks as well as to fill to maximum any track on the allocated virtual tape dataset. Each segment is a separate disk block and is prefixed by some internal VTE management data. A segment may contain the whole or only part of the tape block.

A character indicating which part of the block it contains identifies each segment:

- F** the segment is the first (or only) segment of a block
- M** the segment(s) are the middle segments of a block
- L** the segment is the last segment of a block
- T** the segment is a tapemark block

When the virtual tape is read, all the segments of each block are read and the original block is reconstructed.


```

BROWSE M20V.VEC7051.M20.EC7051 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
"8DVD ....EC7051M20V.VEC7051.M20.EC7051 A.....
.??F.....&.....VOL1EC7051
.??F.....&.....HDR1M20.EC7051 EC705100010001 0082590
.??F.....&.....HDR2F000000008000M20DC0 /S1 P B 00FD84
..?T.....
|.??F.....μ.....&...μ.VOL1HVT300
"8?M..... DO
!%?M..... * //*****
"8?M..... */ /* -- GRANT AUTH TO ISSUE SETCON COMMANDS
!??L..... %SMPVOL%=
"8?F.....V°..... (USCRCTT) DISTLIB(ASAMPLIB) SYSLIB(SAMPLIB)
!%?M..... ENTRY HVTLOD
"8?M.....?^??\^K?\E??\¥??0^?n.-???^??\·??0^??0-.&0^^ 0¶&0^0?0^½.....
..?L.....e.....
..?T.....
.??F.....&.....E0F1M20.EC7051 EC705100010001 0082590
.??F.....&.....E0F2F000000008000M20DC0 /S1 P B 00FD84
..?T.....
..?T.....
***** Bottom of Data *****

```

Figure 62. ISPF Browse of a virtual tape dataset

Virtual Tape Library

The VDB in conjunction with its virtual tapes and the associated virtual tape data sets comprise the virtual tape library. VTE can maintain a Read_Only virtual tape library or a fully Read_Write virtual tape library.

Read_Only Virtual Tape Library

VTE supports Read_Only mode virtual tape library. With this virtual tape library type, certain or all virtual tapes can be protected against update by VTE.

Two types of Read_Only modes are supported. Both types are independent and each one can be used regardless of the other type setting. The modes are:

- Tapepool Read_Only mode.
- VDB Read_Only mode.

With the Read_Only virtual tape library, VTE will:

- Deny any attempt to either rewrite, extend, create or recover a virtual tape that belongs to a Read_Only tapepool.
- Deny any attempt to either rewrite, extend or create any virtual tape if the VDB is in Read_Only mode.
- VTE utilities will scratch, add, delete, remove, vault or recover virtual tapes only if the taken actions do not interfere with the Read_Only mode of the VDB or the associated tapepool.

Tapepool Read_Only mode

With this mode, virtual tapes that belong to a Read_Only mode tapepool can be only read (provided no structural recovery is needed). If the VDB is not in Read_Only mode, the corresponding VDB entry is updated to reflect the virtual tape read status.

Any attempt to modify the virtual tape by means of rewrite, extend, create, scratch add, delete, remove, vault or recovery is rejected.

Virtual tapes from other tapepools can be read, written or modified without any restriction

VBD Read_Only mode

With this mode, all virtual tapes can be only read (regardless their tapepool Read_Only mode). This is so since any action (other than read) implies a VDB update.

Since no VDB update can take place, a virtual tape that is being read does not get its VDB entry updated in order to reflect its status. That means that the virtual tape is not protected from concurrent write from another VTE server in another MVS system in which the Read_Only modes are not set. It is the customer responsibility to ensure no concurrent access is done to this Read_Only mode virtual tapes (unless both accesses are for read purposes).

Any attempt to modify the virtual tape by means of rewrite, extend, create, scratch add, delete, remove, vault or recovery is rejected.

VDB Read_Only mode implies that the journal file is in Read_Only mode as well.

Read_only mode considerations

- When the VDB is placed on Read_Only mode disks and the VDB is set to Read_Only mode and virtual tapes are read, IOS000I followed by IEC331I and IEC251I messages that point to the VDB and the User Catalog the VDB is cataloged into, are to be expected. This is so since although the VDB is opened for input, VSAM tries to update its statistics on the VDB close. These messages should be ignored and can be included in the automation for suppression.
- Although virtual tapes that belong to Read_Only mode tapepools are not overwritten by the VTE server, other external utilities or programs can access the virtual tape datasets in write mode.

This means that the VTE Read_Only mode does not fully protect inadvertent overwrite of virtual tapes and other methods should be employed in order to achieve that protection. An example to this would be to set the disks that store the virtual tape datasets to physically Read_Only mode.

Read_Write Virtual Tape Library

With this type of virtual tape library, both the VDB and the virtual tapes can be read and updated as needed.

Virtual Tapes integrity and verification

This section explains how VTE ensures the integrity of virtual tapes.

Data integrity

In order to make the virtual tape data sets as reliable as possible, the following measures are taken:

- Each segment in a virtual tape dataset has an internal eye catcher.
- All segments of a given original block are linked together by block numbers and segment numbers.
- Each original tape block in the virtual tape dataset is linked to its previous original tape block by its Relative Track Address (TTR).
- The first block in each disk containing the multi-volume virtual tape dataset along with the DVD and the appropriate VDB entry are linked by a unique write level number. This write level number is increased each time the virtual tape is opened for output. It is reset when the virtual tape is scratched.
- When a virtual tape is written, it is check-pointed at user-defined intervals and at certain channel command processes so that after a system crash, the last available data location is well defined.

Internal checkpointing

While processing a virtual tape, the VTE server marks the virtual tape entry in the VDB as 'in-use' for read/write and while writing, it also updates the DVD with checkpoint information.

Internal checkpointing enables VTE to detect the last accessible block in the virtual tape dataset should an IPL (or other VTE server failure) occur. With internal checkpointing, any one of the following events will cause the VTE server to mark the last accessible block in the DVD when writing to a virtual tape:

- A new disk extent has been acquired on a new disk.
- Each tape channel command that causes synchronization between the tape drive and the controlling computer.

- The number of written blocks since the last checkpoint exceeds the CKPINTVL parameter's value in the TPOOL initialization statement.

Should a failure prevent the proper closing of a virtual tape dataset, the HVTURCV utility closes the dataset and sets this block as the last accessible block.

The CKPINTVL parameter value should be set in such a way as to prevent a great deal of data loss, while at the same time using a minimum amount of overhead. When it is defaulted to 0, whole tape datasets on virtual tapes are check-pointed, that is, if a failure occurs while writing the third tape dataset on a virtual tape, the first and second tape data sets are later accessible.

Virtual Tapes verification

A virtual tape is composed of the following components:

- The virtual tape dataset on disk.
- Its appropriate entry in the VDB.

In order for VTE to properly access a virtual tape, the virtual tape dataset structure and its relation to the corresponding VDB entry should be verified. Should any of the verifications fail, a recovery of the virtual tape is required.

The following verifications are done when VTE opens a virtual tape:

- The 'in-use' status. Indicates whether the tape is being read or written.
- All the disks containing the virtual tape dataset extents are present and are in their original collating sequence.
- The virtual tape dataset multi-volume structure has not changed as a result of the dataset being backed-up or restored. Should this happen, the internal structural recovery is automatically attempted.
- All disk extents contain the same write level number.
- The corresponding VDB entry contains the same write level number as its associated virtual tape dataset.

The following verifications are done when VTE accesses a tape block:

- Its block number does not exceed the last written block number.
- Its number matches the number of the requested block.
- Internal blocks segmentation structure is valid.

Whenever the VTE server detects an error in dataset or block verification, it EQCs the job that requested that specific virtual tape with an appropriate message. Thereafter, a virtual tape dataset and VDB recovery should be performed in order to fix the error.

Note: See [Chapter 21, "Handling exceptional events,"](#) on page 189 for more information.

Recovering a Virtual Tape

For a variety of reasons, a virtual tape may need to be recovered. VTE provides a process for recovering virtual tapes.

Reasons for recovering a virtual tape

Virtual tapes recovery is needed when:

- Some virtual tape data sets have been restored. The restored virtual tape data sets may not correspond to their VDB entries. Recovery is required in order to correlate the VDB with the restored virtual tape data sets. In that case, the defer scratch flags and time in the VDB are not changed (see "Deferred Scratch" on page 55). That means that the recovery process does not have any influence on the defer scratch mechanism.
- Virtual tapes are copied to another site using a remote replication product. These virtual tapes are added to the VDB in the remote site using the HVTURCV utility.

- The VTE server encountered an error while verifying a virtual tape dataset and requested recovery.
- Virtual tapes processing was interrupted by an IPL. Recovery is required in order to close the data sets and update the VDB accordingly.
- A virtual tape dataset was manually deleted and no recovery is possible. Recovery is required in order to mark the virtual tape status as 'scratch'. This is done regardless of any deferred scratch condition.
- A VDB entry has been inadvertently removed. Recovery is required in order to recreate the entry. In this case, no deferred scratch flags and time are created. That means that if a recovered virtual tape is scratch in the tape management system, and a previous HVTUSCR run marked this virtual tape to be scratched after a defer-time period, that period is counted again from 0.
- A virtual tape dataset has been recalled and its new disk layout has changed, that is, its extents are organized differently. Recovery is required in order to reset the internal virtual tape dataset pointers. This type of recovery is also done automatically as the virtual tape is accessed.

The recovery process

The recovery process runs as a batch utility and tries to correlate the virtual tape data sets with their corresponding VDB entries. In addition, it closes corrupted virtual tapes (due to IPL) and adjusts internal data in virtual tape data sets that have been moved to different disk device types (on a track by track basis) or that were re-organized during the restore process.

The following types of recovery are performed by VTE:

- Basic recovery
- Structural recovery

In both types, migrated virtual tape data sets are recalled prior to the recovery.

Basic recovery

The basic recovery correlates the virtual tape data sets with the VDB entries. The following describes the steps taken by the basic recovery process:

- The system catalog is searched in order to locate the virtual tape data sets to be recovered.
- An active VDB entry, whose virtual tape dataset is not cataloged, is marked as representing a scratch virtual tape.
- A cataloged virtual tape dataset that is missing its associated VDB entry has a new VDB entry created.
- The virtual tape dataset is allocated and its DVD is read. If there is no DVD, the virtual tape becomes unusable and is marked as such in the VDB.
- The utility accesses all the disks on which the virtual tape dataset resides and verifies that the system catalog reflects all the extents up to the last expected block. If this is found to be false, the virtual tape dataset is recataloged (according to the DVD information).
- Ensure all disk extents are in their original collating sequence and contain the same write level number. If this is found to be false, the virtual tape becomes 'unusable' and is marked as such in the VDB.
- Following the above process and assuming the virtual tape is not 'unusable' or 'scratch', basic recovery is performed by correlating the virtual tape dataset with its associated VDB entry.

Structural recovery

The structural recovery adjusts internal data after structural changes in virtual tape dataset (disk layout), for example, as a result of a virtual tape dataset restore in a different number of extents than it was originally created with.

Structural recovery is always invoked by the VTE server when a virtual tape is opened. That means that recalled or e-Vaulted virtual tapes can be used without further need to run the HVTURCV utility.

Controlling the scratch Virtual Tapes inventory

The VTE server uses scratch virtual tapes from the defined tapepools. As the scratch virtual tapes are used, the inventory of scratch virtual tapes in the tapepool decreases.

When a scratch virtual tape is requested from a tapepool that used all its scratch virtual tapes, the VTE server issues the HVT317W message asking the operator to provide an ad hoc new virtual tape from the specified tapepool:

```
*62 HVT317W $$$DFTL          POOL EXHAUSTED, ENTER VSN FOR
M20INIT /JOB05226, R (RETRY), N (FAIL)
```

In order to minimize the chance of getting out of scratch virtual tapes thus getting these inevitable HVT317W messages, VTE monitors the percentage of active virtual tapes in its tapepools. Should this percentage exceed a pre-defined threshold, the HVT348W warning message is issued at pre-defined intervals:

```
*HVT348W NUMBER OF ACTIVE TAPES EXCEEDS THRESHOLD,  98%
UTILIZATION ($$DFTL )
```

Note: See the THRESH and INTERVAL parameters of the initialization statement, “TPOOL” on page 271, for more information on thresholds.

The user can check the number of active and scratch virtual tapes in the tapepools defined in the HVTPOOLxx member, by issuing the 'F server,DISPLAY=TSP' command. Refer to [Chapter 31, “VTE Operation,”](#) on page 297 for more information.

Preventing getting out of scratch Virtual Tapes

To prevent getting out of scratch virtual tapes, the following can be done:

- Add more virtual tapes to the virtual tape library using the HVTUTAP utility.
- Consider running the daily scratch process more frequently.

Scratching a Virtual Tape

The decision of which virtual tapes to scratch is based on the site’s tape management system, on the MVS catalog or by specific request. The HVTUSCR utility is used to scratch virtual tapes. The utility marks the virtual tapes entries in the VDB as "scratch" and physically deletes the virtual tape data sets from the disk. The actual deletion of virtual tape data sets may be deferred by a predefined number of hours.

When running the HVTUSCR utility in a site that employs a tape management system, HVTUSCR gets a report of active virtual tapes from that tape management system. If no tape management system is employed and the site catalogs the tape data sets in a user catalog, the report is gotten from the catalog using the IGGCSI00 system service. In both cases, HVTUSCR scratches all the virtual tapes that are "active" in the VDB and are not listed in the report.

When a specific tapepool is managed by an EDM (External Data Manager) or it contains eVaulted virtual tapes and the virtual tapes in this tapepool are not defined to the tape management system, this tapepool should be assigned a DEFERBY value of 9999 (in the TPOOL initialization statement). In this case, HVTUSCR utility will never scratch virtual tapes from this tapepool.

Note: For more information on DEFERBY, see “TPOOL” on page 271.

Deferred scratch

Scratching a virtual tape is actually deleting the virtual tape dataset from the disk. Once deleted, the virtual tape is no longer available. In order to enable the reclaiming of virtual tapes, the HVTUSCR utility can defer the actual scratch of virtual tape data sets by a predefined number of hours. During this period (i.e. until the actual scratch occurs), the virtual tape can be used as any other regular virtual tape.

The HVTUSCR utility can reclaim the deferred-scratch virtual tapes, or conversely, it can scratch them regardless any deferred-scratch request.

The number of hours passed from the time the virtual tape is scratch-eligible until it can be physically scratched is called the defer-time.

The defer-time is coded through the DEFERBY parameter value in the tapepool's TPOOL initialization statement. When HVTUSCR runs, each virtual tape is checked whether its defer-time has passed so that the virtual tape scratch is either performed or deferred to the next HVTUSCR utility run.

Scratch of virtual tapes can be done immediately (regardless of any 'defer-scratch' condition) by specifying the FORCE parameter in the HVTUSCR utility initialization statements.

The following should be considered when using the DEFERBY parameter:

- If the DEFERBY parameter is coded for a tapepool, HVTUSCR utility checks for each virtual tape whether its scratch process was previously deferred. If so, the virtual tape will be scratched only if its defer-time has passed.

If a virtual tape scratch was previously deferred, but this virtual tape status is no longer scratch in the tape management system, the virtual tape is automatically reclaimed by HVTUSCR utility.

If specific virtual tapes were previously deferred, they can be explicitly reclaimed.

Each time HVTUSCR utility runs (regardless of the scratching type), all virtual tapes that their scratch was previously deferred and their defer-time has passed are physically scratched.

- If the DEFERBY parameter is not coded for a tapepool, the virtual tapes of this tapepool will be immediately scratched.

VDB integrity and verification

In order to quickly recover a corrupted VDB, the VDB is protected by a journal mechanism. Each time a VDB record is changed by the VTE server or one of its utilities, an appropriate record is written in the journal file. This ensures that the journal file will have the latest VDB changes on record, so that it can be used to forward recover the VDB should it become damaged from the last backup up to its latest update.

The VDB and journal file are synchronized using the timestamp of the last action done on both files. When the VTE server is started or a VTE utility starts to execute, this synchronization is checked. If the timestamps do not match, the VTE server shuts down (or the utility terminates, respectively), with an appropriate message. In that case, a VDB recovery is required (see "VDB Backup and Recovery").

VDB and Journal File synchronization

The following describes the four possible VDB and journal file synchronization statuses detected at VTE server or a utility initialization.

Status 1

The status is normal. The VDB and journal file are synchronized. Processing continues.

Status 2

The following message is issued and the VTE server or utility initialization fails.

```
HVT231S VDB LAST UPD TIME (0107123F/12530393) LOWER THAN JRN
(0107124F/12530393), UJRC REQUIRED
```

This indicates that the VDB was restored from the last backup. VDB recovery from the journal file using the HVTUJRC utility is required in order to synchronize both files (add last updates done to the VDB since its last backup) and bring the VDB to an up-to-date status.

Status 3

The following message is issued and the VTE server or utility initialization fails.

```
HVT227S VDB UNCOVERED PERIOD, LAST VDB
UPDATE=0106124F/12530393 OLDEST JRN
REC=0107113F/20065504
```

This indicates that the VDB was restored from a very old backup and cannot be recovered using the journal file.

The VDB can be recovered in one of the following ways:

- Restore a more current VDB that can be forward recovered by the journal file using the HVTUJRC utility.
- Run the HVTURCV utility in order to rebuild the VDB according to the virtual tape datasets on disk.

Status 4

Due to unexplainable reasons, the journal option was disabled for a period of time. Processing continues because there is nothing to do.

VDB backup and recovery

At times, the virtual tape database (VDB) may need to be backed up or recovered. Reasons for performing these procedures are listed below.

Backing-up the VDB

As part of the general VTE data availability strategy, the VDB should be backed-up frequently, using the IDCAMS REPRO. The VDB can be backed up while VTE is running.

Note: If other backup products are used, they may lock the VDB during the backup process, thus delaying the overall VTE processing.

When journaling is used, the VDB should be backed-up before the journal file is about to become full. This ensures that the journal file will always cover all the changes that were done to the VDB since its last backup.

It is recommend to backup the VDB on a daily basis in order to shorten the process should a backup be required.

A sample job to backup the VDB is found in member UVBK in the SAMPLIB library.

Recovering the VDB

Full VDB recovery is required if the VDB has been restored from backup. Usually, a VDB recovery can be done using the journal file (using the HVTUJRC utility). However, if for some reason the journal file cannot be used to bring the VDB to an up-to-date status, the recovery can be done using the virtual tape data sets on disk (using the HVTURCV utility).

VDB Recovery using the Journal file

Changes to the VDB dataset are continuously recorded in the journal file. This journal file can be used to recover the last VDB changes without having to actually access the virtual tape data sets on disk.

The utility HVTUJRC is used to forward recover the VDB according to an up-to-date journal file. During this process, the journal file is read from the oldest to the newest record. The first journal record whose update timestamp is one hour before the last VDB update timestamp initiates the VDB recovery process. From this point on, all journal records update their respective VDB records. The one-hour overlap is needed to compensate for two CPUs whose clocks are not synchronized and access the same VDB. At the end of the process, both the VDB and the journal files are synchronized.

VDB Recovery without the Journal file

When a journal file is not used or when it is not usable, the VDB recovery should be done according to the virtual tape data sets. This is accomplished by the HVTURCV utility.

This recovery correlates the VDB entries to the status of the associated virtual tape data sets. For further details, see "Recovering a Virtual Tape".

Note: Migrated virtual tape data sets are recalled upon this type of VTE recovery.

Journal File management

The journal file is a sequential wrap-around file in which each new block overwrites the oldest block. That way, the journal file never gets an End-of-File condition. However, a "journal full" condition is expected when the whole journal file has been written once without an intervening VDB backup. If that "journal full" condition is encountered, no virtual tapes can be opened or properly closed by any VTE component.

Planning the Journal File size

The journal file size is automatically calculated according to user parameters that consider all the expected actions done to the VDB in a given time range. Allocation of the journal file is done using the HVTUJFR utility.

Resetting the Journal File

In order to reset the journal file records so that all blocks can be overwritten, a VDB backup should be initiated. Due to the journal special format, special indicators must be set during the backup to tell the journaling mechanism the location of the block that can be overwritten.

UVBK member in the SAMPLIB library contains a sample job that can be used to backup the VDB and reset the journal file. In order not to exceed the overall allowed percentage of used journal records, it is recommended to backup the VDB on a daily basis.

Message HVT223W is periodically displayed if the percentage of used journal file blocks exceeds a pre-defined threshold.

Virtual Tape Library Activity Tracking – VTE User SMF Record

VTE produces a user SMF record for each mounted and used virtual tape. This record contains statistics for the entire virtual tape processing during the period of time it was mounted (that is, from the mount time to the unload time). The statistics are accumulated, regardless of the number of datasets and the number of jobs that accessed the virtual tape (in case it is permanently mounted).

In addition to the accumulated data, the record contains the job name, reader time and reader date (job log identification) of the job that mounted the virtual tape. That way, this record can be associated with the other job's records usually used for SMF statistical reports.

The record mapping is available in the HVTSMF macro in the VTE MAC library.

The user SMF record number is defined by the **RECORD** parameter of the SMF initialization statement in the HVTOPTx member (refer to Chapter 2, "VTE Initialization Statements" of the *IBM System Storage VTF Mainframe Customization and Reference* [GC53- 1190-00]).

Notes:

- VTE can write multiple records in behalf of the same job (if a virtual tape is unloaded between the job's steps). The correct statistics for the job is obtained by accumulating the statistics from all associated records.
- VTE can write one single record in behalf of many jobs that use the same virtual tape (if that virtual tape is permanently mounted). In that case, the statistics are accumulated for the job that first used the virtual tape (even though other jobs used it as well).
- The user record is not written for virtual tapes accessed from within VTE offline utilities.
- The user record is not written if, due to an internal error, the job log identification could not be obtained. In this case, the server issues the HVT115E RC=80 message.

Chapter 18. Managing the diskpools

This chapter describes how VTE allocates virtual tape data sets in diskpools, and how these diskpools are managed. Reconfiguration guidelines for diskpools are mentioned, and backup/restore considerations for virtual tape data sets are provided.

Disk Space Management

This section explains how to manage the space on disks that are used for allocation of virtual tape data sets.

Allocating Space for a New Virtual Tape Dataset

VTE allocates a new virtual tape data set in each of the following situations:

- When a scratch virtual tape is mounted.
- When a specific new (unknown) virtual tape is mounted by specifying VOL=SER in JCL.
- When an "active" virtual tape is re-written from its first label. In this case, the virtual tape data set is deleted and re-allocated.

The virtual tape data set is allocated in equal sized extents on one or more disks contained in a given diskpool. The extent size is set in order to allocate the minimum required space for a given virtual tape data set. The rigorous extent size calculation and virtual tape data set management prevent wasted space while allowing the virtual tape data set to grow up to the value defined in the appropriate TAPECAP parameter of the TPOOL initialization statement.

Note: For more information on the TAPECAP parameter, refer to [“TPOOL” on page 271](#).

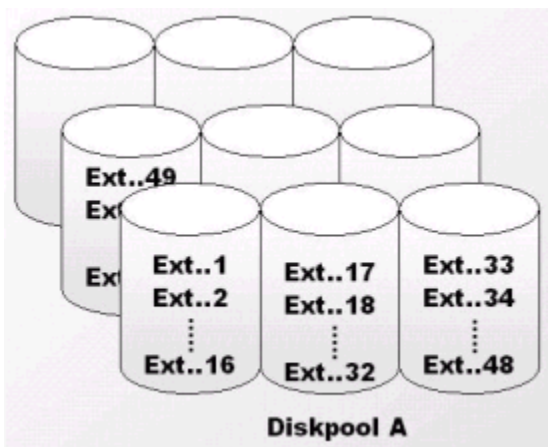


Figure 63. Extents of virtual tape data set on disks

The following formula is implemented in calculating the virtual tape data set space allocation parameters:

$$\text{EXTsize} = C / (D * T * 16)$$

Where:

EXTsize

Specifies the disk extent size in track units.

C

Specifies the tape capacity (as specified in the TAPECAP parameter in the TPOOL initialization statement).

D

Specifies the number of eligible disks in the diskpool minus 1. If the number of eligible disks is 1, this number is not reduced anymore.

- For non-SMS-managed disks, the number of eligible disks is calculated as the minimum of:
 - Number of available disks in the diskpool.
 - The TAPECAP's max_disks value (maximum number of disks to contain one virtual tape data set).

An available non-SMS disk is a non-EAV or EAV disk (regardless of the IGDSMSxx USEEAV setting) that is online, has a mount attribute of STORAGE, and has not converted to SMS.

- For SMS-managed disks, the number of eligible disks is calculated as the minimum of:
 - Number of available disks in the diskpool.
 - The TAPECAP's max_disks value (maximum number of disks to contain one virtual tape data set).
 - Number of candidate volumes in the Storage Group referred to by the diskpool Storage Class parameter.

An available SMS disk is a non-EAV or EAV disk (regardless of the IGDSMSxx USEEAV setting) that is online, belongs to a Storage Group that is either ENABLE, QUIESCE, or QUIESCE_NEW and has a disk status of ENABLE, QUIESCE, or QUIESCE_NEW.

T

Specifies the disk track size.

The following figure illustrates the initialization statements that affect the virtual tape data set allocation:

```
DPOOL    NAME=DISKPOOL1,
          UNIT=(UNIT1),
          THRESH=(85,180,50)
TPPOOL   NAME=TAPEPOOL1,
          DPOOL=DISKPOOL1,
          RANGE1=(EC0001,EC0010),
          RANGE2=(EC0101,EC0110),
          TAPECAP=(610,4),
          CKPINTVL=0,
          DEFERBY=48,
          COMPRESS=(1,2)
```

The virtual tape data set is allocated with primary and secondary space allocations, both equal to the calculated extent size.

Just for illustration, the equivalent JCL SPACE parameter for allocating similar data sets would be:

SPACE+(TRK,(EXTsize,EXTsize),,CONTIG)

During tape processing, extents are allocated as more data is written, according to the calculated extent size. New extents are allocated up to the tape capacity or up to the maximum number of disks defined by the max_disk value. When exhausted, a new virtual tape is mounted. When the virtual tape is unloaded, the amount of space allocated for the virtual tape data set is the amount required to hold the written data. When the virtual tape is further re-opened and more data is written, the extent size to be used for additional extents (if required) is recalculated in order to fit a potential new site disk reconfiguration.

In order to allow extending virtual tapes (by adding additional data sets or extending an existing data set) and still maintain the requested tape capacity, the space allocated for the virtual tape data set is not released at deallocation. This is so in order not lose disk extents. This means that on average, half of the last extent is not utilized. This space is used when the virtual tape is extended.

Recalculating the disk extent size of EAS-Eligible Virtual Tape Datasets

When VTE allocates a new virtual tape dataset and that dataset is EAS-eligible, VTE recalculates the virtual tape dataset disk extent size. This is done in order to fit the EAV disks allocation requirement that each extent must be a multiple of the so-called Multi_Cylinder_Unit (MCU), which for EAV disks is 21 cylinders.

Note: To be considered EAS-eligible, the EATTR=OPT parameter must be specified in the appropriate TPOOL initialization statement and VTE must run on z/OS 1.12 or higher.

A dataset EAS-eligibility condition is sufficient for VTE to recalculate the extent size – the dataset itself does not have to reside on the disk cylinder-managed area. Moreover, the diskpool does not have to contain any EAV disks.

To perform the recalculation, the following steps are carried out:

1. VTE starts by calculating the virtual tape dataset space allocation parameters according to the general formula used for regular virtual tape datasets that are not EAS-eligible .
2. Provided the extent size is not a multiple of the MCU, VTE rounds the extent size to the next multiple of the MCU constant.
3. VTE adjusts the virtual tape capacity to the next greater number, that is, a multiple of the calculated extent size. That way, the allocated virtual tape dataset is fully utilized.

Upon the recalculation completion, the following effects are expected:

- Virtual tapes capacity may be larger than initially specified in the TAPECAP parameter.
- Full virtual tape datasets may span less disks than initially specified in the TAPECAP parameter.

Viewing Virtual Tape Dataset Allocations

Virtual tape data sets information can be viewed via ISPF option 3.2 or option 3.4 (Data Set Information). The following example shows the allocation details for a virtual tape data set allocated under the following conditions:

- The tape capacity is 600MB.
- The virtual tape is fully utilized.
- The diskpool that contains the virtual tape data set is a non-SMS-managed diskpool that contains ten eligible 3390 disks.

```

                                Data Set Information
Command ==>

Data Set Name . . . . : M20V.VEC3002.M20.EC3002

General Data                      Current Allocation
Management class . . . : **None**   Allocated tracks . . : 11,232
Storage class . . . . : **None**   Allocated extents . . : 144
Volume serial . . . . : QA9045+
Device type . . . . . : 3390
Data class . . . . . : **None**
Organization . . . . . : PS
Record format . . . . : U
Record length . . . . : 0
Block size . . . . . : 32760
1st extent tracks . . . : 78
Secondary tracks . . . : 78
Data set name type . . . :
Creation date . . . . : 2008/10/14
Expiration date . . . : ***None***

Current Utilization
Used tracks . . . . . : 11,227
Used extents . . . . : 144

SMS Compressible : NO
Referenced date . . : 2008/10/14
```

Figure 64. Data Set Information screen

The fields on this screen are described below.

Block size (32760)

The block size used when the data set was opened for output. It is not the actual block size used for write—written blocks are similar to the blocks written by the application and may be segmented in order to fill the disk tracks.

1st extent tracks (78)

Number of tracks contained in the first data set extent.

Secondary tracks (78)

Number of tracks to allocate on secondary extents. This number is equal to the first extent size. When this number is multiplied by 16 (to give the number of tracks on one disk) and then by the number of available disks minus 1, the total number of tracks that comprise the virtual tape data set is reached.

Allocated tracks (11232)

Total number of allocated tracks. In our example (where the number of available disks is ten), the number of allocated tracks is $11232 = 78 \times 16 \times (10 - 1)$.

The allocated data set size (in MB) can be roughly calculated using the 'allocated tracks' number. In our example we get $599.85\text{MB} = (11232 \text{ tracks}) \times (56\text{K}) / (1024 \times 1024)$.

Allocated extents (144)

Number of extents on all disks that contain the virtual tape data set. In our example (where the number of available disks is ten), the number of allocated extents is $144 = 16 \times (10 - 1)$.

Used tracks (11227)

Number of tracks that are actually utilized. This number is lower than the 'Allocated tracks' because the virtual tape End-of-Volume is forced when there is no room for an additional tape block. Since the maximum supported tape block size is 256K, and since this block should be segmented (which in turn requires additional gaps and control data), the actual virtual tape End-of-Volume was forced after this number of used tracks.

Pressing <Enter> on the previous Data Set Information display shows all nine disks that contain the virtual tape data set:

```
Volume Information
Command ==>
All allocated volumes:
Number of volumes allocated: 9
QA9045 QA9044 QA9048 QA9046 QA9051
QA9056 QA9057 QA9049 QA9047
```

Figure 65. Volume Information

Viewing EAS-eligible Virtual Tape Dataset Allocations

When allocating the original virtual tape dataset (whose size was 600MB and spanned 9 disks) with extended attributes (EATTR=OPT), the following allocation parameters are used:

```
Data Set Information
Command ==>
Data Set Name . . . . : M20V.VEC3002.M20.EC3002

General Data
Management class . . : VTEMC
Storage class . . . : VTESC
Volume serial . . . : VTE02 +
Device type . . . . : 3390
Data class . . . . : **None**
Organization . . . : PS
Record format . . . : U
Record length . . . : 0
Block size . . . . : 32760
1st extent tracks . . : 315
Secondary tracks . . : 315
Data set name type . . :
SMS Compressible . . : NO
Extended Attributes . : OPT

Current Allocation
Allocated tracks . . : 11,340
Allocated extents . . : 36

Current Utilization
Used tracks . . . . : 11,335
Used extents . . . . : 36

Dates
Creation date . . . : 2018/10/14
Referenced date . . : 2018/10/14
Expiration date . . : ***None***
```

Figure 66. Example of Data Set Information screen with EAS-related fields

The relevant EAS-related fields are the following:

1st extent tracks (315)

Number of tracks contained in the first data set extent. According to the basic formula it should have been 78, but it was rounded up to the next MCU multiple.

Allocated tracks (11,340)

Total number of allocated tracks. According to the basic formula, this should have been 11232, but instead it was rounded up to the next multiple of the extent size. The updated allocated dataset size (in MB) can be roughly calculated using the 'Allocated tracks' number. In this example:

$$605.62\text{MB} = (11340 \text{ tracks}) * (56\text{K}) / (1024 * 1024)$$

Allocated extents (36)

Number of extents on all disks that contain the virtual tape data set. According to the basic formula, it should have been 144 and spanned 9 disks. However, the extent size is larger than originally calculated, so VTE allocates less extents in order to fill the virtual tape ($36 = 11340/315$). Note in that case, the EAS-eligible virtual tape dataset spans only 3 disks (and not 9 as expected).

Extended attributes (OPT)

The virtual tape dataset was allocated with extended attributes that is, it is EAS-eligible. Note that this attribute does not guarantee the dataset resides in the cylinder-managed space of EAV disks. It just indicates that the dataset can optionally reside in the cylinder-managed space and only if other conditions are met (i.e. EAV disks are available, sufficient space exists in the cylinder-managed space, etc.).

Pressing <Enter> on the previous "Data Set Information" screen shows the three disks that contain the virtual tape data set:

```
Volume Information
Command ===>
All allocated volumes:
  Number of volumes allocated: 3
VTE01 VTE02 VTE03 *      *
*      *      *      *
```

Figure 67. Volume Information

Controlling the Virtual Tape Space Allocation

Virtual tape data set space allocation is affected by the specifications of the tape capacity and the number of disks eligible for allocation. The following summarizes the various influences on the VTE allocation algorithm when these specifications are changed.

- **Decreasing the TAPECAP's max_disks value**

TAPECAP's max_disks value should not be decreased. This is so because the virtual tape data set extent size has been calculated already using the previous max_disks value and one or more extents have already been allocated.

Decreasing the max_disks value implies increasing the size of the extents in order to accommodate the entire tape capacity. But the extents that have already been written cannot be increased. So, if an 'active' virtual tape is extended, this may lead to a Sx37 abend before the tape capacity has been reached and a tape End-of-Volume has been enforced.

- **Increasing the TAPECAP's max_disks value**

The new TAPECAP's max_disks value will be effective when allocating scratch tapes or when extending existing virtual tape data sets.

- **Changing the tape capacity value**

A change in the tape capacity value affects the size of the extents on the disk accordingly (when increasing the tape capacity, the extent size will be increased; when decreasing the tape capacity, the extent size will be decreased).

Changing either of these two parameters (tape capacity or number of disks eligible for allocation) affects the amount of wasted space in the last extent (which, on average, is half the extent).

To illustrate this, consider the following example:

- The tape capacity is 600MB.
- The number disks eligible for allocation is 8.
- The disk track size is 56K.

Using the extent size calculation given in "Allocating Space for a New Virtual Tape Dataset" (above), we find that the extent size is 100 tracks and a fully utilized virtual tape allocates 11200 tracks. If the virtual tape is not fully utilized and contains, for instance, only 300MB (5600 tracks), the average waste is 0.89% (in average half extent or 50 tracks out of the 5600 used tracks).

On the other hand, if there were 10 eligible disks, the extent size would become 78 tracks and a fully utilized virtual tape would allocate 11232 tracks. In that case, the average waste would become 0.69% (39 tracks out of the 5616 used tracks).

In a similar way it can be shown that defining a tape capacity that is close to the real used virtual tape size minimizes the average wasted space due to the last partially used extent.

Due to the rigorous block segmentation, the waste that results from the last partially used extent is the only real waste you may encounter.

Handling lack of space on disks eligible for Virtual Tape data set allocations

As previously described, the tape capacity and the max_disks parameter are used to calculate the amount of space required on each disk and the extent size. This is done in order to efficiently utilize disk space (prevent a large amount of wasted space), ensure tape capacity consistency, and prevent Sx37 abends when there is still free space on the diskpool.

However, Sx37 abends may occur under the following conditions:

- Not enough space on the diskpool for a new extent (all disks are full).
- All disks with eligible free space have their VTOCs full.
- Lack of extents on the diskpool.

Due to disk fragmentation, extents are allocated non-contiguously, and each part is considered a separate extent (that is, allocation of one extent may cause creation of three extents summing to the same space value). As a result, the 16-extent limit is reached before all the space for this data set is allocated on the disk. Following this, the next extent allocation is done on the next eligible disk. That way, the maximum number of allowed extents is reached before the tape capacity is reached.

To avoid an Sx37 abend, VTE issues a tape EOV when it reaches the last allocated disk. A new tape is mounted, starting an allocation of another virtual tape data set. In an unbalanced diskpool, the new allocation may be able to access the less occupied disks. However, if the diskpool is full, an Sx37 abend is inevitable.

These Sx37 abends occur on the server only. However, since the job using the associated virtual tape cannot continue, it is EQC-ed by the server. (See Chapter 10, Handling Exceptional Events.)

In order to minimize the chance of filling the disks, thus getting these inevitable Sx37 abends, VTE monitors the percentage of used space on its diskpools. Should this percentage exceed a pre-defined threshold, the HVT347W warning message is issued at predefined intervals.

Note: See the THRESH and INTERVAL parameters of the initialization statement, "[DPOOL](#)" on page 269.

```
HVT347W USED SPACE ON STORGRP SGDORI EXCEEDS THRESHOLD,094%
        UTILIZATION (D_TEST )
```

The user can check the number of eligible disks and the utilization level of esoteric names or Storage Groups associated with diskpools defined in the HVTPOLxx member, by issuing the `F server, DISPLAY=DSP` command.

Note: See [Chapter 31, “VTE Operation,”](#) on page 297 for more information.

Preventing free space errors

To prevent free space errors, the following can be done:

- When free space warning messages are issued, increase the amount of disk space available for virtual tapes or add more disks to the Storage Group or esoteric name displayed in HVT347W message.
- Regularly remove rarely used virtual tape data sets from disk, by de-staging (migrating) them to real tapes.

Virtual tape data set de-stage (migration) and recall

As with any data set, it is important to understand how to back up and restore a virtual data set. These two processes are described below

Virtual Tape data set backup or de-stage

The virtual tape datasets can be backed up or de-staged to real tapes with any standard backup or migration product.

The following should be considered when backing up or de-staging the virtual tape datasets:

- The virtual tape datasets may be multi-volume datasets, so care should be taken to always backup all the data set extents from the disks on which they reside.
- When backing-up a virtual tape data set, all allocated extents should be backed-up (not only the used space). This will prevent data loss should an IPL occur while VTE writes the virtual tape data set and the last TTR has not yet been updated.

Virtual Tape data set restore or recall

A de-staged virtual tape data set is recalled automatically when the associated virtual tape is mounted.

A virtual tape data set can be restored or recalled to an identical or different device type than the one it originally resided on. The restore or recall to a different device type should be done on a track-by-track basis.

After the restore or recall completes, the VTE server validates that the virtual tape data set is complete and up-to-date. The verification includes the so called 'structural recovery', a process that updates the internal virtual tape data set pointers to reflect the correct structure of the restored or recalled virtual tape data set. Only when the verification is successful, the mount succeeds.

Virtual tape datasets should be restored or recalled to disks defined under the diskpool they originally resided on. However, they may be restored or recalled to different disks, provided there is a DPOOL initialization statement defining them. If no diskpool defines these new disks, message HVT387S is issued and the job trying to mount the virtual tape fails.

Recall of a de-staged virtual tape data set is transparent to the VTE server and no manual intervention is required.

If an older version of the virtual tape data set was restored, the HVTURCV utility should be run in order to correlate the up-to-date VDB entry with the virtual tape data set.

When a full disk restore is done and the disk contains extents of multi-volume datasets, these virtual tape datasets should be fully restored. Failing to do this may leave these virtual tape datasets unusable because they may contain mixed write level numbers (old/ new data).

Controlling Virtual Tape data sets de-staging

When diskpools are nearly full, the site can choose to de-stage some of the virtual tape data sets in order to make room for newly created virtual tape data sets.

The best algorithm for selecting which virtual tape data sets are to be de-staged is the LRU (Last-Recently-Used). This means that the virtual tape data sets that were not used for the longest period of time are the best candidates for de-staging.

However, this algorithm proves to be too simplistic.

The virtual tape data sets represent virtual tapes and their usage is solely dictated by the site's tapes policy. Therefore in order to understand the behavior of virtual tape data sets, it is not sufficient to rely only on the usage of disk data sets. Rather, one must analyze the site's tapes usage and behavior. That way, the next virtual tape use (mount or scratch) time may be predicted, hence the next associated virtual tape data set on disk use may be also predicted. As a result, one concludes that virtual tape data sets whose predicted "next use" time is the farthest are the best candidates for de-staging.

Terminology

In this topic, the following terminology is used:

Tape data set

The data set written onto a real or virtual tape. This tape data set has an associated tape data set name.

Virtual tape

The tape entity which is similar to a reel tape or a cartridge.

This virtual tape has an associated volume serial number

Virtual tape data set

The disk data set that contains the data written to the virtual tape.

Character placeholder

An asterisk (*) matches none, one or more successive characters.

A question mark (?) matches exactly one character.

HVTUDST utility overview

Virtual tape data sets can be prioritized for de-stage off disk according to their activity level using the HVTUDST utility. It means that virtual tape data sets whose associated virtual tapes are likely to be mounted or scratched soon, will be last to be de-staged (if at all). Conversely, virtual tape data sets associated with less active virtual tapes will be first to be de-staged.

The collected activity (or history) of all tape data sets is stored in the history file.

The behavior of each tape data set is learned from the information in the relevant record found in the history file so that the future activity (expected mount or scratch date) of the virtual tape that contains that specific tape data set can be predicted.

Besides its collecting function activity, the HVTUDST utility can produce a list of virtual tape data sets sorted by the probability of their associated tape data sets to be either mounted or scratched.

A higher probability means that the virtual tape that contains a given tape data set is likely to be mounted or scratched sooner than a similar virtual tape with a lower such probability.

This capability is derived from the utility capability to analyze the history file and predict the next mount or scratch date of each tape data set.

If a diskpool usage raises beyond a predefined threshold, HVTUDST utility can be used to produce the list of virtual tape data sets that if de-staged, will cause the diskpool usage to fall below the predefined threshold. This list of the best candidates for de-staging can be used as input to the local storage management product in order to let it de-stage the virtual tape data sets and achieve the desired diskpool usage.

HVTUDST does not perform the de-staging by itself.

The following topics give a more detailed description of the HVTUDST functionality.

Generic names and patterns

Tape data set names may be grouped in so-called generic names. This generic name consists basically of the original name in which certain characters are replaced by character placeholders. Each such generic name matches all different tape data set names of tape data sets that in fact contain similar data and have similar activity level.

A good example of different tape data set names and their derived generic name are GDG tape data sets of the same application. The tapes contain similar data and bear the same activity level although their data set names suffix – the generation number – differ.

```
M20.APPL1.G0000V01
M20.APPL1.G0000V02      Three tape data set names
M20.APPL1.G0000V03

M20.APPL1.G????V??      The derived generic name
```

The process of deriving generic names from tape data set names is done by the HVTUDST utility according to user supplied parameters called patterns. In order to define such patterns, the user must identify what tape data sets have identical behavior. By properly defining these patterns, all tape data sets with a similar behavior are grouped under the same generic name.

The most common pattern is the GDG suffix. If the user specifies the GDG suffix as a pattern, all GDG tape data sets will be grouped under their respective base names as generic names.

A less common pattern is a date-dependent qualifier. Such data set names are sometimes used for instance by weekly jobs that create tape data sets whose one of the name qualifiers contains the week sequence number.

The last used user patterns are stored in the history file.

The user defines the pattern through the utility's DPTRN initialization statement.

The following example illustrates two user-defined patterns along with several tape data set names and the resulting generic names:

```
Tape data set names
M20.APPL1.BILL.G0001V00      (1)
M20.APPL1.BILL.G0002V00      (2)
M20.APPL1.BILL.G0003V00      (3)
M20.APPL2.BILL.G0007V00      (4)
M20.APPL2.BILL.G0008V00      (5)
M20.APPL2.BILL.G0009V00      (6)
M20.APPL1.WEEK01.BKP          (7)
M20.APPL1.WEEK02.BKP          (8)
M20.APPL1.WEEK03.BKP          (9)
M20.APPL2.WEEK01.BKP          (10)
M20.APPL2.WEEK02.BKP          (11)
M20.APPL2.WEEK03.BKP          (12)
User patterns
.
.
DPTRN NAME=G????V??,QUAL#=99      (13)
DPTRN NAME=WEEK??,QUAL#=3         (14)
.
.
Resulting generic names
M20.APPL1.BILL.G????V??          (15)
M20.APPL2.BILL.G????V??          (16)
M20.APPL1.WEEK???.BKP             (17)
M20.APPL2.WEEK???.BKP             (18)
```

In the above example:

- User pattern (13) specifies that all tape data sets whose last qualifier is a GDG suffix be grouped under a generic name starting with the base name.

- User pattern (14) specifies that all tape data sets whose third qualifier starts with WEEK followed by any two characters be grouped under the same generic name.
- Tape data sets (1), (2) and (3) are grouped under generic name (15) due to pattern (13).
- Tape data sets (4), (5) and (6) are grouped under generic name (16) due to pattern (13).
- Tape data sets (7), (8) and (9) are grouped under generic name (17) due to pattern (14).
- Tape data sets (10), (11) and (12) are grouped under generic name (18) due to pattern (14).

The History data set record

Based on the previous section conclusion, the statistical data accumulated by the HVTUDST utility should be based on tape data sets behavior. For that, the VTE server records in the VDB the creation date of each virtual tape, each mount date thereafter and the scratch date.

In addition, it associates this virtual tape to the first tape data set on it.

On its daily run, the HVTUDST utility identifies all virtual tapes that had a mount or scratch event since the last utility's run. It then extracts the first tape data set name and looks for a similar data set name in the history file. If a similar tape data set has been encountered in a previous run, this last statistics is appended to that record. If not, a new record is created. See "Tape data sets life cycle" further below in this section for the rules on why only the first tape data set is considered.

Although correct, the above paragraph misses one important aspect.

If we had to rely solely on tape data set names, we will rarely find similar names. See for example GDG tape data sets of the same application – although they contain similar data, their names are never the same.

This is the reason why we use generic names rather than the original tape data set names. With a proper definition of user patterns, one can instruct HVTUDST utility to create a minimal number of generic names – each one covering a large number of tape data sets that share a common behavior. Following that logic, the record that contains the recorded events is led by the generic name that covers all tape data sets that share the common behavior.

This record is called the history data set record (or simply the data set record) and is stored in the history file.

If the user patterns are not properly defined, tape data sets may not be included in a generic name, letting data set records be led by full names rather than generic names.

On its daily run and statistical data accumulation, a data set record is updated at most once for all matching tape data sets. That means that no matter how many tape data sets that match the generic name had mount or scratch events in a given day, the relevant data set is updated only once.

The data set record is composed of the generic name and up to 365 triplets of values. Each such triplet contains the following information:

- Number of days since a tape data set whose name matches the generic name was created.
- Number of scratch events in that day.
- Number of mount events in that day.

The following example depicts the data set record of a group of tape data sets that once created, are MOD-ed (appended data) once a week and scratched after one month. The example shows the data set record after one year of statistics accumulation.

```
M20.APPL1.BILL.G????V??      (1)
00_00_12                      (2)
07_00_12                      (3)
14_00_12                      (4)
21_00_12                      (5)
30_12_00                      (6)
```

In the above example:

- The generic name of all tape data sets that share the common behavior is listed in (1).

- The tape data set is mounted 4 times a month, that is, on the creation day (2), on the 7- th day (3), on the 14-th day (4) and on the 21-st day (5). After one full year, each such event will count to 12 times.
- The tape data set is scratched on the 30-th day (or once a month). After one year, this will count to 12 times.

The history file contents can be listed using the HVTUREP utility.

The History file structure

The history file is a sequential variable-blocked data set that maintains the tape data sets statistics. Basically, the history file contains three types of records:

- The history file control record. It relates the history file to its VTE server and VDB and contains other synchronization information.
- Dataset records. Each generic name along with its related observations are stored in one separate data set record.
- Pattern records. These records contain the last user specified patterns.

```

BROWSE   HVTT.V130.HSTEC                               Line 00000000 Col 001 080
Command ==>                                           Scroll ==> CSR
***** Top of Data *****
CA..O_RDIRECTFRE .....0107336F124723340107324F
D...M20.*.AAAAAAA.BBBBBBBB.CCCCCCCC.A..... Control Record
D...M20.EC7081.....
D...M20.GDG.G????V?? .....±.....
D...M20.GDG1.G????V?? .....±.....
D...M20.GDG2.G????V?? .....±.....
D...M20.JAN???? .....±.....
D...M20.JAN????.AAAAAAA.BBBBBBBB.CCCCCCCC .....?.?.....
D...M20.KIKI.JAN?? .....±.....
D...M20.KIKI.R????Q??KUKU3 .....±.....
D...M20.KUKU.JAN?? .....±.....
D...M20.KUKU.JAN2007 .....?.?.....
D...M20.KUKU1 .....±.....
D...M20.KUKU3.JAN?? .....?.?.....
D...M20.KUKU3.JAN??AAAAAAA.BBBBBBBB.CCCCCCCC .....?.?.....
D...M20.KUKU3.JAN??AAAAAAA.BBBBBBBB.CCCCCCCC.A .....?.?.....
D...M20.N2222022 .....?.?.....
P.....?.....G????V?? .....±.....
P.....N????V?? .....±.....
P.....JAN???? .....±.....
P.....JAN?? .....±.....
P.....M20.*.AAAAAAA.BBBBBBBB.CCCCCCCC.A .....±.....
***** Bottom of Data *****

```

Figure 68. Browsing a history file

The history file is used solely by the HVTUDST and HVTUREP utilities and should not be updated manually. Before the daily update, its contents are copied to the history backup file. Should the utility fail and leave the history file in an unstable state, its contents are automatically restored from the history backup file on the next utility run.

Dataset records consolidation

As seen above, the user patterns are the basic information used by the HVTUDST utility to derive and build the generic names from the tape data set names. On its first run, these generic names are created and the associated data set records are written to the history file.

On each successive daily run and depending on the circumstances, the utility updates existing data set records or (if no generic names can be derived from new tape data set names), new data set records are created.

If after a while it is concluded that several tape data sets not initially included in a user pattern share a common behavior, a new user pattern can be added by the user. When such a new user pattern is detected, the HVTUDST utility scans the whole history file, builds new generic names and consolidates all data set records whose leading names match the new generic name. After the relevant data set records consolidation:

- A new data set record is created in which the leading name is the generic name.
- All mount and scratch events of all existing data set records are inspected and the maximal count for each respective day is updated into the new data set record (that is, corresponding triplets are inspected for the triplet with the maximal values and it is the one inserted in the new record. In addition, if some days are not yet found in the new data set record, new respective triplets are added).
- All existing data set records that participated in the consolidation are deleted.

The following example depicts two data set records that are consolidated due to a new added user pattern.

1-st data set record

M20.APPL1.WEEK01.BKP	(1)
00_00_02	(2)
14_00_01	(3)
30_01_00	(4)

2-nd data set record

M20.APPL1.WEEK02.BKP	(5)
00_00_01	(6)
15_00_03	(7)
30_02_00	(8)

The following user pattern is added

DPTRN NAME=WEEK??.QUAL#=3	(9)
---------------------------	-----

The following new data set record is created

M20.APPL1.WEEK??.BKP	(10)
00_00_02	(11)
14_00_01	(12)
15_00_03	(13)
30_02_00	(14)

The following old data set records are deleted

M20.APPL1.WEEK01.BKP	(15)
M20.APPL1.WEEK02.BKP	(16)

If after a while it is concluded that several tape data sets already included under one generic name do not share a common behavior (that is, the initial considerations for adding the user pattern were wrong), the user pattern can be deleted. When such an existing user pattern is deleted, the HVTUDST utility scans the whole history file and deletes the data set records whose generic names match the deleted user pattern.

From this point on, new statistical data is accumulated for each individual tape data set (unless new user patterns are added in order to include them in other existing data set records).

Tape data sets life cycle

In a production environment, the production tapes have a similar structure (that is, tapes used by the same jobs have the same files on the same labels), and a similar expiration date pattern. Therefore, a tape can be characterized by its first tape data set name and all other tape data sets (on labels higher than 1) can be ignored. Accordingly, what matters is the first tape data set, NOT the virtual tape this tape data set resides on. A virtual tape is scratched according to the latest tape data set expiration date and when reused, it will probably contain another tape data set with another expiration date.

In that production environment, each tape data set has a predefined behavior. After its creation, it can be mounted on the 2nd, 4th, 6th and 8th day and finally scratched on the 9th day. In this case, the tape data set will never be active after the 9th day.

This behavior is called the tape data set life cycle.

If we were to represent the above tape data set in a history data set record, on the 3rd day of the 3rd cycle the data set record would look like in the following example.

```
generic name      (1)
00_00_03         (2)
02_00_03         (3)
04_00_02         (4)
06_00_02         (5)
08_00_02         (6)
09_02_00         (7)
```

In the above example, the creation day (2) and the 2nd day (3) incurred three cycles thus having three mount events while the 4th (4), 6th (5), 8th (6) and the 9th (7) incurred only two cycles, thus having only two scratch or mount events, respectively.

Note that there are cases when tape data sets may be needed (and their associated virtual tapes mounted) at unexpected times. This happens for instance when an ad hoc restore is needed and a manual job is run for that. But as more and more cycles are recorded, the permanent pre-defined mount and scratch events become dominant and reflected as such in the data set records.

Tape data sets activity prediction

Based on the fact that in a production environment tape data sets have a more or less well known life-cycle, the history records can be used to predict when a given tape data set is about to be used (and the associated virtual tape be mounted or scratched). More than this, they can be used to build a comparative scale on which different tape data sets are placed according to their probability of being mounted or scratched in the near future.

To illustrate this, let's look at two data set records representing two groups of tape data sets grouped under generic_name_1 and generic_name_2. Each group has a similar number of members. In addition, generic_name_1 is on its 3rd cycle while generic_name_2 is on its 5th cycle.

Day	Dataset Record generic_name_1	Dataset Record generic_name_2
0	00_00_03	
1		
2		00_00_05
3	03_00_03	
4		
5		
6	06_00_03	04_00_05
7	07_00_01	05_00_01
8		
9	09_00_02	
10		08_00_04
11		
12	12_00_02	12_04_00
13		
14	14_02_00	

In the previous example, if we are at the 8th day, a tape data set that matches generic_name_1 is likely to be used (and its virtual tape mounted) sooner than a tape data set that matches generic_name_2. That means that we are able to predict the relative recorded tape data sets activity.

Tape Datasets grade

Once we know that a tape data set relative activity can be predicted and a comparative scale built, we can assign quantitative values to each tape data set placed on that scale.

That value is called tape data set grade or simply, grade.

The method of grade value calculation ensures that tape data sets with a lower grade are less likely to be used (their virtual tape be mounted or scratched) than tape data sets with a higher grade, thus being the best candidates for de-stage. The grade values are integers greater than 0 and lower than 2147483648 (2^{31}).

The general formula for calculating a tape data set grade is:

$$\text{GRADE} = (A * B * 365 * C) / D$$

Where:

A

Total number of mount events from the current day in the cycle to the scratch day.

If more than one scratch day are present in the cycle, the day with the highest count is considered to be the scratch day.

This value represents the probability of using a tape data set from a given generic name - a higher value implies a higher probability of using that tape data set. Since a higher grade should lower the associated virtual tape data set chance to be de-staged, we place this factor in the numerator.

B

Number of years the initial formula is valid.

This normalization factor is required since the number of mount events grows continuously but the grade value should not be greater than 2147483647 ($2^{31}-1$).

C

Total number of current active different tape data sets that match one generic name.

D

Number of remaining days from the current day in the cycle to the scratch day.

If more than one scratch day are present in the cycle, the day with the highest count is considered to be the scratch day.

This value represents the probability of scratching a tape data set from a given generic name - a lower value implies a higher probability of scratching that tape data set. Since a higher grade should lower the associated virtual tape data set chance to be de-staged (it may be scratched soon), we place this factor in the denominator.

The following restrictions exist for the grade calculation:

- Under no circumstance should the calculated value yield a fraction less than one. In case where $A=B=C=1$ and having $\max(D)=365$, the nominator should be multiplied by a normalization factor of 365.

- The grade value should be lower than 2147483648 (2^{31}).

If $\min(D)=1$ and $\max(A)=365$, then

$$(365 * B * 365 * C) / 1 < 2147483648 \text{ resulting in } B * C < 16119.$$

This initial formula is valid for 10 years ($B=10$ is set into the formula calculation), letting C be less than 1611.

The actual implementation of the above two restrictions is as follows:

- As the 10 years limit approaches, all the history file data set records pass a general normalization process, in which the number of mount and scratch events are halved.
- If the number of tape data sets that match a given generic name exceeds the value of C (1611), only the limit value is taken into the formula regardless the actual C value. This action may cause the grade calculation to become inaccurate, however, a too high number of tape data sets in one generic name may indicate that wrong user patterns have been defined.

If the number of mount events is 0 (B=0), B is set to 1 prior the division and the resulting grade is increased by 1.

The grade calculation method works fine (subject to the two listed restrictions) for tape data sets with at least one recorded cycle and whose relevant factors exist and are valid. The following list shows all exceptional cases where the grade cannot be calculated, letting it default to 2147483646 ($2^{31}-2$), thus making it a bad candidate for de-stage:

- DSN27 of the first tape data set on tape could not be determined.
- Tape creation date missing.
- No matching data set record in history file yet.
- Tape creation date is greater than current date.
- No full cycle yet.
- No scratch day

Even when all required data for grade calculation is available, two special cases exist:

- The scratch day is the current HVTUDST running day. This may happen when HVTUDST runs before HVTUSCR. In that case D = 0 and HVTUDST forces the grade to 2147483646 ($2^{31}-2$) thus making the virtual tape data set a bad candidate for destage.
- No mount events till the scratch day (A=0). In that case let A = 1 and 1 is subtracted from the final calculated grade. This will make the virtual tape data set be a better candidate for de-stage (than a similar virtual tape data set with at least one recorded mount events).

The next topic describes how the grade value can be explicitly set by the user.

Including and excluding Tape Datasets

The user has the option to influence the grade values calculation for certain tape data set through the utility's DRULE initialization statement.

A DRULE initialization statement with an INC parameter causes the specified tape data set to get the lowest possible grade while a DRULE initialization statement with an EXC parameter causes the specified tape data set to get the highest possible grade.

Each tape data set whose grade is to be calculated is filtered through the DRULE initialization statement. The parameters of the first DRULE initialization statement that matches the tape data set are used for the grade calculation and no further DRULE initialization statements are searched for it.

The following example illustrates two DRULE initialization statements and their influence on six tape data sets.

DRULE initialization statements		
.		
DRULE EXC,DSN=M20.APPL1.WEEK??		(1)
DRULE INC,DSN=M20.APPL1.*		(2)
.		
.		
Tape data set names		
M20.APPL1.BILL.G0001V00		(3)
M20.APPL1.WEEK01.BKP		(4)
M20.APPL1.WEEK02.BKP		(5)
M20.APPL2.BILL.G0001V00		(6)
M20.APPL2.WEEK01.BKP		(7)
M20.APPL2.WEEK02.BKP		(8)
Resulting grade		
M20.APPL1.BILL.G0001V00	0	(9)
M20.APPL1.WEEK01.BKP	2147483647	(10)
M20.APPL1.WEEK02.BKP	2147483647	(11)
M20.APPL2.BILL.G0001V00	N/A	(12)
M20.APPL2.WEEK01.BKP	N/A	(13)
M20.APPL2.WEEK02.BKP	N/A	(14)

In the above example, the first DRULE initialization statement that matches tape data set (9) is (2), therefore it gets the lowest possible grade. Tape data sets (10) and (11) are matched by DRULE; (1) therefore, they get the highest possible grade.

Setting the final de-stage prioritization

Once each tape data set grade has been set, the real priority for de-stage can be finalized. The following factors are considered when setting that priority:

- The calculated grade.

According to the method by which grade are calculated, tape data sets with a lower grade are less likely to be used than tape data sets with a higher grade. That means that their associated virtual tape data sets should be first to be de-staged.

- The associated virtual tape data set size.

If two tape data sets are assigned an identical grade, the associated largest virtual tape data set should be first to be de-staged.

Creating the de-stage job

Once the de-stage prioritization logic discussed above has completed, the HVTUDST utility sets the associated virtual tape data sets that reside on different diskpools, in different groups. For each such group of virtual tape data sets, the HVTUDST utility sorts them by the factors discussed in the previous topic and marks each one (in ascending destage prioritization order) until the predefined low threshold is reached. During that process, virtual tape data sets whose associated virtual tapes have been EXCLUDED by the user, are skipped.

The gotten list of virtual tape data sets marked for de-stage is internally sorted by the expected scratch day (number of days until the expected scratch event) and tailored to a job (based on a job skeleton provided in the SAMPLIB library). This last sort ensures that most of the virtual tape data sets that reside on a real tape expire on a similar date, thus minimizing the need for data sets recycling.

The job can be used by the employed storage management product to perform the actual de-stage process. The user defines the storage management product according to which HVTUDST utility will select the skeleton and tailor the job through the utility's DRUN initialization statement.

The following example illustrates the use of the DRUN initialization statement to let a DFSMSHsm migration job skeleton be selected and de-stage job be tailored:

DRUN PROD=HSM

By default, this job is written to a DUMMY data set.

An example of such a tailored job for HSM follows:

```
//M20INSBK JOB (HSM),,MSGCLASS=X,CLASS=A
//*
//* HSM JOB TO MIGRATE VIRTUAL TAPE DATASETS
//*
//MIGRATE EXEC PGM=IKJEFT01,REGION=0K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  PROFILE NOPREFIX
  HMIG M20V.VEC0053.M20.GDG2 ML2 NOWAIT
  HMIG M20V.VEC0054.M20.GDG2 ML2 NOWAIT
```

In the above example, a DFSMSHsm migrating job was tailored by the HVTUDST utility in order to de-stage two virtual tape data sets.

Virtual tape data set compression

VTE is able to conditionally compress its virtual tape data sets written to disk.

There are three compression types that can be selected on a tapepool basis:

Run Length Encoding (RLE)

—A compression technique that compresses repeating characters, such as blanks. The algorithm is very effective for source data sets, reports or printouts. It is less efficient for load modules and other hex type data sets. The compression is done by the CPU (by the special compression hardware assist available with the S/390 and zSeries processors). The implementation is done via the CUSE (Compare Until Substring Equal) machine instruction and is invoked by the VTE server using the CSRCESRV system service.

RLE for Strings (RLEfS)

—A compression technique that compresses repeating strings. The algorithm is able to compress text data as well as binary data. It compresses as much as twice more data than the SMS compression, however, it does not take advantage of the hardware assist. Therefore, it consumes as much as two to three times more CPU. The implementation is done by locating repeated strings, storing the string once and then storing offsets and length of the other occurrences.

Lempel-Ziv

—A dictionary based compression technique. A static (non-adaptive) dictionary is prepared before the compression operation starts and is used throughout the data set to be compressed. The algorithm is very effective for binary data, including load modules and other hex type data sets. Its effectiveness with text type data is dependant on the level of representation of the initial blocks to the rest of the tape data. The compression is done by the CPU (by the special compression hardware assist available with the S/390 and zSeries processors). The implementation is done via the CMPSC (Compression Call) machine instruction and is invoked by the VTE server using the CSRCMPSC system service.

For more information regarding this Lempel-Ziv compression implementation, see "ESA/390 Data Compression - SA22-7208".

VTE implementation of this compression type resembles the SMS tailored compression method. When a virtual tape is created, the VTE server derives a compression dictionary when enough data (60k) is written to the virtual tape data set.

Once derived:

- The dictionary is imbedded within the virtual tape data set blocks.
- Any additional block written to the virtual tape is compressed using that dictionary.

The derived dictionary is not tested for efficiency (that is, it's implied compression ratio is not calculated) and all subsequent written blocks are passed through the compression routines. However, if the compressed block length is larger than the original block length, that block is left uncompressed.

When an existing uncompressed virtual tape has more data added (by means of DISP=MOD or LABEL=2,3,...), and the respective TPOOL indicates that the virtual tape data set should be compressed, a compression dictionary is derived according to the last written blocks and the process performs as if a new virtual tape is created.

The Lempel-Ziv compression type implementation effectiveness is dependent on the type of data:

- In cases where the data set initial blocks (the 60k used to derive the dictionary) do not represent the overall data set blocks contents, the implementation of this compression type may give poor results. For example, DFSMSdss dumps in which the initial blocks contain binary zeroes, do not reflect the real data that follows.
- Non-linear CPU consumption may be observed when small and large data sets with similar data are compressed. This occurs because the dictionary derivation takes a fixed amount of time, regardless of the compressed data set size.

When the virtual tape is read, the derived dictionary is used for expand purposes.

Each compression type can be implemented in three levels of compression – maximum, half, and quarter compression ratio.

A compressed virtual tape data set can be read correctly even though the associated tapepool specifies that its virtual tape data sets should not be compressed.

As with a real tape size, the virtual tape data set size on disk remains constant whether or not the data is compressed. That means that if the data is compressed, the virtual tape can fit more data than the value specified in the TAPECAP parameter of the appropriate tapepool definition.

A given virtual tape data set may be compressed if its associated tapepool specifies so and is subject to further restrictions as described in the next section.

No SMS compression is possible with the VTE virtual tape data sets since they are written to disk using EXCP.

Compression decision

The decision on whether or not to perform compression is based on the COMPRESS parameter value of the TPOOL initialization statement and on other factors, which are described below.

COMPRESS is 0

If the compress parameter in the relevant TPOOL initialization statement is missing or COMPRESS=0, then compression is never done for data written on that tapepool's virtual tapes.

COMPRESS is not 0

If the COMPRESS parameter value in the relevant TPOOL initialization statement is different than 0, compression is performed subject to the following conditions:

- If DCB=TRTCH=COMP is specified on the relevant DD statement, compression is performed.
- If DCB=TRTCH=NOCOMP is specified on the relevant DD statement, compression is not performed.

If no DCB=TRTCH is specified on the relevant DD statement, the COMPACT parameter in the DEVSUPxx member is considered:

- If COMPACT=YES is specified, compression is performed.
- If COMPACT=NO is specified, compression is not performed.

If the COMPACT parameter is not specified or the DEVSUPxx member is missing, compression is performed for all emulations, except 3480. Compression is not performed for a 3480 emulation.

Compression efficiency

Particular blocks are not compressed (regardless of the tapepool's COMPRESS parameter or DCB=TRTCH values) if the length after compression is larger than the original block's length.

Compression candidates

The following data set types are good candidates for RLE compression:

- Report data sets
- Source files data sets

The following data set types are good candidates for Lempel-Ziv and RLEfS compression:

- Binary data sets
- Load modules
- Disk Dumps

Compression ratio and CPU overhead

The following table illustrates the compression ratio and the CPU overhead associated with the various compression operation. The SMS results are shown for illustration purposes only. The table was compiled using the compression results of a full disk dump (Sysres disk including dump data sets) performed by IBM DFSMSdss backup product. The disk contained approximately 5.6 GB of dumped data. The written average blocksize was 55k to 64k.

The used space rank is defined as:

$$(A \times 10) / B$$

Where:

A

The used disk space for a compressed virtual tape.

B

The used disk space for an uncompressed virtual tape.

For illustrative purposes the used space rank value for an uncompressed virtual tape has been chosen to be 10 and all other values for compressed virtual tapes are relative to it.

The CPU rank is defined as:

A / B

Where:

A

The used CPU time for writing using any compression type.

B

The used CPU time for writing without compression.

For illustrative purposes the CPU rank value for an uncompressed virtual tape has been chosen to be one and all other values for compressed virtual tapes are relative to it.

The CPU rank is affected by many variables like the application, the blocksize, the type of data and the CPU model.

The compression ratio for written data is displayed by the HVT023I (tape unload) message and stored in the VTE SMF record.

<i>Table 9. Sample used space and CPU rank comparison table for compressed virtual tape data sets</i>		
Compression TYPE	Used Space Rank	CPU Rank
Uncompressed	10	1
RLE (Level 1)	5.7	5.4
RLE (Level 2)	7.9	3.4
RLE (Level 3)	8.9	2.1
RLEfS (Level 1)	3.7	17.8
RLEfS (Level 2)	6.7	10.5
RLEfS (Level 3)	8.3	5.7
Lempel-Ziv (Level 1)	5.6	4.6
Lempel-Ziv (Level 1)	7.8	3.0
Lempel-Ziv (Level 1)	8.9	2.0
SMS	6.4	6.2

Reconfiguring the Diskpool

From time to time you may be required to modify some of your site's existing policies, introduce new software, hardware, and so on.

Depending on whether your diskpools are SMS-managed or not, the information in the following sections should be considered when modifying the existing software or hardware environment.

SMS considerations

When converting non-SMS-managed disks that contain virtual tape data sets to be SMS-managed, you must convert the entire diskpool containing the virtual tape data sets to be SMS-managed at once. This is necessary because multi-volume data sets cannot be split between SMS-managed and non-SMS-managed disks.

If you fail to convert the entire diskpool at once, the old virtual tape data sets will fail in their allocation process with a dynamic allocation error code of X'021C'. Should this happen, you can scratch the old virtual tapes so that the next time (when they are mounted) the allocation will succeed.

After converting the disks, defining them under an SMS Storage Group and making the necessary Storage Group ACS routine changes so that they can be accessed, you should modify the appropriate diskpool initialization statement(s).

The modification consists of specifying storclas (and optionally, mgmtclas) values on the appropriate UNIT parameter of the DPOOL initialization statements, and removing the old unit value.

For example, the existing DPOOL initialization statement:

```
DPOOL UNIT=(unit),....
```

should be modified to:

```
DPOOL UNIT=(,storclas,mgmtclas),....
```

where unit is the old esoteric name of the non-SMS-managed disks, and storclas/mgmtclas are the new Storage Class and Management Class values, respectively.

Non-SMS considerations

If, for some reason, you decide to modify the reference between existing disks and their respective esoteric names (in the EDT table), the whole diskpool should be contained within the same esoteric name. This is necessary, because the virtual tape data set allocations are always done using the UNIT parameter of the DPOOL initialization statement (for extending purposes). Leaving old virtual tape data sets on disks linked to different esoteric names may cause allocation errors.

If the esoteric name used for virtual tape data set is changed, the unit value of the appropriate UNIT parameter of the DPOOL initialization statement should be changed accordingly.

General considerations

If you decide to copy virtual tape data sets from one disk device type to another (e.g. from 3380 to 3390), the copy should be done on a track-by-track basis. This is because the internal virtual tapes data sets structure depends on the blocks location on the tracks and on their location relative to the data set start.

Chapter 19. Integration with Tape Management Systems

This chapter describes how VTE is integrated to work with different tape management systems.

Overview

Being consistent to real tapes, the VTE virtual tapes may be managed, as part of the whole tape environment, by the existing tape management system employed in the site.

The integration of VTE with the tape management systems achieves the following benefits:

- One integrated management solution for virtual tape libraries and real tape libraries.
- Maintenance is done according to the tape management system; the rules and logic of the tape management system are kept and inherited by VTE.
- Managing VTE does not require retraining of operators who are familiar with the tape management system.
- Natural connection between VTE and other libraries using the tape management system interface.

Identifying the site's Tape Management System

The VTE extended MCS console support enables the VTE server to detect and analyze tape management system's messages issued upon the console. Mount messages issued by MVS or by tape management systems are intercepted by the VTE server, which then mounts the virtual tapes according to the request.

Some of the tape management systems issue special mount messages (in addition to those originally issued by MVS), while other suppress the original MVS mount messages, and issue their own, instead. In any case, VTE uses only the tape management system mount messages, and ignores the MVS mount messages (if not already suppressed by the tape management system). If no tape management system is used, VTE considers the original MVS mount messages. VTE learns about the tape management system mount messages duplication and the need to ignore MVS messages as soon as one of the following events occur:

- VTE server is started and the first mount message is captured.
- The tape management system is started.

If a running tape management system is stopped, VTE resumes the MVS mount messages processing.

For example, when VTE detects for the first time a mount message with a code of TMSxxx, it concludes that CA-1 is in charge and issues the following message:

```
HVT246I CA-1 ACTIVE, MVS MOUNT MESSAGES WILL BE FURTHER IGNORED
```

If the tape management system is deactivated, VTE identifies that situation and starts to consider the MVS mount messages.

```
HVT246I CA-1 INACTIVE, MVS MOUNT MESSAGES WILL BE FURTHER CONSIDERED
```

Tape Management Systems common definitions

In order to integrate VTE and a tape management system, some identical definitions have to be made for both products. For detailed information refer to [Chapter 33, "Implementing VTE and Tape Management Systems,"](#) on page 349.

Adding Tapes

A range of tapes should be dedicated for VTE in the tape management system database. These tapes are added to the VDB using HVTUTAP utility.

Defining Tape pools

If tape pools are implemented by the tape management system, VTE must have a definition of the same tape pools in its tapepools initialization statements for the virtual tape ranges.

That way, VTE will select tapes for scratch mounts from the range expected by the tape management system.

If tapepools are not implemented by the tape management system, the implementation of VTE does not require tapepool implementation. VTE can be implemented with or without tape management tapepools.

Tapepools can be implemented by VTE without changes to the tape management system.

Scratching and Cleaning

When VTE is integrated with the tape management system the scratching and cleaning decisions are done solely by the tape management system as part of the daily cleaning process. VTE synchronizes against the tape management system database and scratches its virtual tapes accordingly.

The synchronization of the VTE VDB and the tape management system database (in terms of which virtual tape is scratch) is done in two steps:

- Run the tape management system cleaning process as usual. The cleaning process identifies tapes valid for scratching, scratches them and makes them available for scratch mount.
- Run the VTE HVTUSCR utility with the appropriate type on the SYNCH initialization statement. The HVTUSCR utility must run as a second step of a job whose first step executed a program to produce a report based on the tape management system database contents.
 - For tape management systems that produce active tape reports (like CA-1 or TLMS), HVTUSCR scratches all active virtual tapes that were not found in the report.
 - For tape management systems that produce general tape reports (like DFSMSrmm), HVTUSCR filters only the active tapes, then scratches all active virtual tapes that were not found in the filtered report.

In sites that use the system catalog to manage tapes, the IDCAMS utility is used to produce an active tapes report, then HVTUSCR scratches all active virtual tapes that were not found in the report.

Chapter 20. Running VTE in JES3 environments

This chapter describes how VTE uses the VTE subsystem and the VTE virtual devices in JES3 environments.

VTE in JES3 environments

All VTE functionality (except PAT for static allocations and automated allocations using SMS Data Class ACS routines) is available in JES3 environments.

Due to the difference that exists between how JES2 and JES3 handle jobs and tape devices, VTE that runs in a JES3 environment uses different approaches and logic in some of its processes.

The following areas are mainly affected by these differences:

- The VTE subsystem
- JES3-managed VTE virtual devices

Dynamic reallocation

Unlike in a JES2 environment where the dynamic reallocation is done in the job's address space once it starts (in the job initiation phase), in a JES3 environment the dynamic reallocation is done for the job during its JES3 Converter/Interpreter C/I phase (either in the global or the C/I FSS address space).

The VTE subsystem is invoked from an IEFUJV SMF exit dynamically activated at VTE server startup (or when the subsystem is turned on by the 'F server,SSNON' operator command). The IEFUJV SMF exit is de-activated at server shutdown (or when the subsystem is turned off by the 'F server,SSNOFF' operator command).

The special IEFUJV SMF exit is added to the system via the MVS dynamic exits facility. The exit's EXITNAME (as defined by MVS) is SYS.IEFUJV. In order to let it be active in the various JES3 subsystems, the following EXITNAMEs are added as well: SYSSTC.IEFUJV and SYSJES3.IEFUJV. The exit module name is similar to the VTE server's SERVERID (as specified in the IDENT initialization statement).

After VTE completes the dynamic reallocation process, the job completes its JES3 Converter/Interpreter C/I phase. It is then passed to the JES3 Converter/Interpreter Prescan phase.

All dynamic reallocation related messages and trace messages are printed on the system log (SYSLOG) and on the 3rd job's JES3 file (JESYSMSG).

VTE virtual devices

One of the most important JES3 features is its ability to control the site's devices. Using the MDS (Main Device Scheduler), JES3 supports the MVS allocation process by satisfying job resource requirements before job execution, thus allowing execution to proceed without allocation delays.

VTE can operate with its virtual devices being either JES3-managed or not, although this support is implemented differently for each type. When the VTE virtual devices are JES3-managed, VTE (in conjunction with JES3 Exit 11, IATUX11, and with the MVS dynamic allocation exit IEFDB401), forces all virtual tape mounts under its control to be deferred. That way, no IAT5210 (or any tape management system related message) is expected for any virtual tape mount, that is, all virtual tapes are mounted by the jobs after they start. Other tapes mount methods remain unaffected.

The mount deferring for JES3-managed devices is done at three points during the normal JES3 job management flow:

- JES3 Converter/Interpreter Prescan phase – Mounts that result from:
 - Non-specific requests for scratch virtual tapes.

- Specific requests for virtual tapes (that is, the DD statement specifies both the UNIT and the VOL=SER parameters).

In these cases, the defer is done by the VTE subsystem invoked by the IEFUJV SMF exit.

- JES3 Converter/Interpreter Postscan phase – Mounts that result from request for virtual tapes that contain cataloged datasets.

In this case, the defer is done by JES3 Exit 11 (IATUX11) that gets control after JES3 has resolved cataloged dataset references.

- Dynamic allocation – Mounts that result from:
 - Either specific or non-specific requests (that is, the dynamic allocation request specifies the UNIT parameter).
 - Request for virtual tapes that contain cataloged datasets.

In these cases, the defer is done by the MVS dynamic allocation exit IEFDB401.

Certain products that dynamically allocate tapes (like OAM) require the tape device type to be one of the IBM standard tape device types. Remembering that the VTE virtual devices are built by the VTE special UIM and their device type is not an IBM standard device type, these products may encounter some problems in their normal flow.

In order to overcome this problem, VTE suppresses the virtual tape mount defer and sets the standard tape device type. The required defer parameter is then restored using JES3 Exit 71, IATUX71, that gets control in order to enable the IAT5210 tape request setup message.

The system defer process should not be confused with the allocation process. The allocations are still done by JES3 and it is the only one that manages these allocations. However, instead of proceeding with the mount before the jobs start, VTE causes the mounts to be deferred until jobs actually open the tape datasets. That way, all job's control blocks are available for VTE inspection.

Chapter 21. Handling exceptional events

This chapter describes the method upon which VTE handles environmental or internal errors.

VTE equipment check simulation

It is well known that real devices encounter physical errors for which they take certain actions. Although VTE virtual devices do not encounter physical errors, some logical errors may affect the server functionality.

The VTE server simulates real tape devices. As such, it receives I/O requests from applications (mostly via standard MVS access methods) and responds with simulated statuses and senses.

The simulation does not consist of just performing I/Os on behalf of jobs. It consists also of simulating problems and acknowledging MVS of that.

One such problem can be a VTE server malfunction. If such a malfunction occurs the job whose virtual tape was served during that problem should be terminated.

VTE initiates jobs termination in the following three cases:

- Data integrity errors:

Data integrity errors are detected by the VTE server's extensive validations on the processed data, virtual tape data sets structure, and so on.

- Operator request:

The operator can request a job termination when the VTE server attempts to avoid a premature termination and needs the operator intervention.

- An internal VTE server problem:

An internal VTE server problem can be an unexpected abend, an internal logic error, and so on.

In all the above cases, the VTE server issues appropriate messages to the server's first JES file (JESMSG LG) and writes them to the VTE server sysout file (HVT SOUT). In addition, messages that are relevant to the terminated job are written to the job's first JES file (JESMSG LG).

The VTE server triggers the job's termination by simulating an Equipment Check (EQC) status on the VTE virtual device whose mounted virtual tape detected the error. When MVS receives such statuses, it terminates the job that uses the involved device.

It must be mentioned that these EQC statuses are just simulated EQCs and no hardware maintenance should be involved in fixing them. The next virtual tape mounted on a VTE virtual device that encounters such an EQC status will function properly.

```
$HASP373 M20DVX STARTED - INIT 1 - CLASS A - SYS MM90
IEF403I M20DVX - STARTED - TIME=14.31.06
IEF233A M 0571,EC6000,,M20DVX,S1,M20.FIRST.TAPE.FILE
HVT341I * *****
HVT311E * ALLOC ERROR, R15=04, ERROR=X'1708',X'0002',X'0000FD6',DSN=ELC10.VEC60
HVT998I * IKJ56228I DATA SET ELC10.VEC6000.M20.FIRST.TAPE.FILE NOT IN CATALOG OR
HVT342E * M20DVX /JOB07156 I/O FAILED BY 0_RDIREC SERVER - EQC GENERATED
HVT341I * *****
IOS000I 0571,49,EQC,07,0E00,,*,M20DVX
1040202C400000200000(0000000000000000)0070(00000000)878F1E7114720000
IEC512I I/O ERR 0571,,M20DVX,S1,M20.FIRST.TAPE.FILE
IEC502E R 0571,,M20DVX,S1
IEC145I 413-08,IFG0194K,M20DVX,S1,OUT01,0571,,M20.FIRST.TAPE.FILE
IEA995I SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=413 REASON CODE=00000008
TIME=14.31.06 SEQ=00750 CPU=0000 ASID=0016
PSW AT TIME OF ERROR 075C1000 00DB6448 ILC 2 INTC 0D
NO ACTIVE MODULE FOUND
NAME=UNKNOWN
DATA AT PSW 00DB6442 - 41003ABA 0A0D4DE0 3A5A5820
GPR 0-3 00DB662C A4413000 00006ADC 00DB5B72
GPR 4-7 008C7410 008C77A4 008C774C 008C77A4
```

```
GPR 8-11 008C776C 008C6AD8 008C6AE9 00DB9834
GPR 12-15 80DB9908 008C76D0 00DB5CD4 00000008
END OF SYMPTOM DUMP
IEF450I M20DVX S1 - ABEND=S413 U0000 REASON=00000008
      TIME=14.31.07
IEF404I M20DVX - ENDED - TIME=14.31.07
$HASP395 M20DVX ENDED
```

In the above example, the VTE server could not allocate the virtual tape data set for virtual tape volume serial number EC6000. It propagated the error messages (HVT311E and HVT998I), appended the HVT342E message and wrote them on the job's JESMSG LG file. The VTE server then simulated an Equipment Check status on VTE virtual device 0571. This EQC status led MVS to abend the job with a S413-08 abend.

Note: As most of the errors occur when a virtual tape is mounted, most times MVS abends the job with a S413-08. This is not a real open error, its just the way MVS terminates jobs that encounter EQC errors during the mount process.

Chapter 22. e-Vault - VTE electronic vaulting

This chapter describes the unique VTE electronic vaulting process.

Traditional vault concept

The traditional vault concept was introduced by data centers in order to ensure that certain vital data stored on tapes is put aside and protected in a safe place. Practically, this is achieved by moving this set of tapes to a remote location or off-site storage location. Occasionally, when the data center vital data is updated, a new set of tapes is sent to the off-site storage location and the old set is returned to the local location.

The traditional vault process consists of two components:

- The logical component:

This component consists of the tape management system process. The tape management system updates the current location for each tape in its database according to the so-called "vault patterns". Reports are usually produced to notify operators of which tapes are to be moved to each of the off-site storage locations.

- The physical component:

This component consists of a physical method of transportation such as trucks and planes. This is often nick-named TAM or "Truck Access Method". The tapes reported to be in an off-site storage location are loaded and transported by this physical method of transportation.

The traditional vault concept is known as an error-prone process that involves manual intervention. Great delays are imposed on a data center's normal processing when some vital data is lost and the off-site storage tapes should be returned. This delay can be critical to the data center business continuance.

VTE e-Vault concept

VTE introduces the concept of electronic vaulting (or e-Vault). This concept is based on the fact that the VTE virtual tape data sets reside on disks and the virtual tape data sets can be moved between these disks without manual intervention. The decision about which virtual tapes are to be moved is still done by the local tape management system.

By using this new concept, VTE completely alleviates the problems that arise from the physical nature of the traditional vault. VTE solves the problems associated with the traditional vault process by fully eliminating the vault physical component. Remote locations containing disks connected to the data center's disks by proper links can be used as off-site storage locations. These disks will store the vital data (originally written to tapes and 'TAM-ed' to remote off-site storage locations). By doing so:

- The whole vault process is automated and quick.
- The e-Vaulted data is returned much faster, should it be required by the data center.
- The e-Vaulted data is immediately available in the off-site storage location (provided it runs an MVS and VTE).

While none of the scenarios described below require any change to the customer's JCL, programs or data creation techniques, certain procedures unique to shops recovery goals must be developed on a case by case basis that incorporate this unique VTE e-Vault technology. By contracting IBM personnel it can be done such that it is a complete turnkey solution.

e-Vault types

VTE supports three types of e-Vault: Type3, Type4, and Type5.

Note that one VTE server can simultaneously use one or more types of e-Vault.

The three types of e-Vault are described below:

- FTP Type3 e-Vault

This type of e-Vault moves the virtual tape data sets to a remote off-site storage location using standard FTP links. The remote off site storage location consists of a data center that runs any operating system in conjunction with an FTP server.

- FTP Type4 e-Vault

This type of e-Vault moves the virtual tape data sets to a remote off-site storage location using standard FTP links. The remote off site storage location is required to run MVS.

- CTC Type5 e-Vault

This type of e-Vault moves the virtual tape data sets to a remote off-site storage location on the Cloud. VTE must be installed and available to accomplish this task.

FTP Type3 e-Vault

This type of VTE e-Vault uses existing FTP links between MVS and any platform equipped with a sufficient quantity of disk space to continuously contain all e-Vaulted virtual tape data sets. The tape management system indicates the virtual tapes whose virtual tape data sets are e-Vaulted off-site.

The e-Vaulted data exists at the remote off-site storage location until it is needed at the local location or no longer required and remote-scratched – the tape management system controls this.

The data at the remote off-site storage location cannot be used until it is returned to the local location.

If the local location should need an e-Vaulted virtual tape, VTE transparently moves it back.

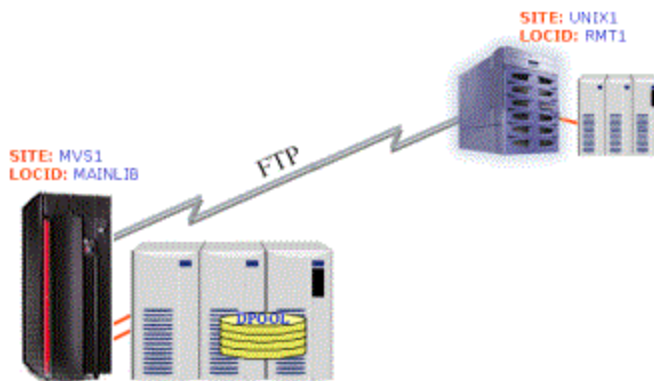


Figure 69. A typical Type3 e-Vault configuration

Configuration

The following components are required in a Type3 e-Vault environment:

- Any platform (Unix, Windows, etc..) with sufficient disk space to contain all the e-Vaulted virtual tape data sets.
- The above platform should have an active FTP server.
- Appropriate initialization statements in the VTE server.

Process

When VTE moves a virtual tape to a remote off-site storage location using Type3 e-Vault, the virtual tape dataset is FTP-ed to a remote off-site storage location managed by any platform.

The actual FTP is done by a new started task started by the VTE server. This started task executes the HVTUFTP utility.

When the FTP process completes, the virtual tape dataset is deleted from the original diskpool. Following that process:

- The virtual tape dataset exists on the remote off-site storage location disks.

- The virtual tape is marked 'vaulted' in the VDB.

Advantages

- Simple process
- The data is stored on low-cost disks.

Considerations

- Inability to return an e-Vaulted virtual tape when the FTP link or platform are down.
- The data on the remote platform is not usable unless returned to an MVS platform.

FTP Type4 e-Vault

This type of VTE e-Vault is a bit more sophisticated than a Type3 e-Vault. The e-Vaulted virtual tape data sets are FTP-ed to a remote off-site storage location. Once safely there, they are deleted from the source diskpool thus mimicking exactly the manual off-site vaulting process. A host running MVS and VTE (and eventually a tape management system as the source -- if the virtual tapes are to be accessed) is required at the remote off-site storage location. The VTE e-Vaulted virtual tapes are available for use at the remote off-site storage location (DR testing, interchange, and so on).

The e-Vaulted data exists at the remote off-site storage location until it is needed at the local location or no longer required and remote-scratched – the tape management system controls this.

If the local location should need an e-Vaulted virtual tape, VTE moves it back to the local location reversing the process discussed here.

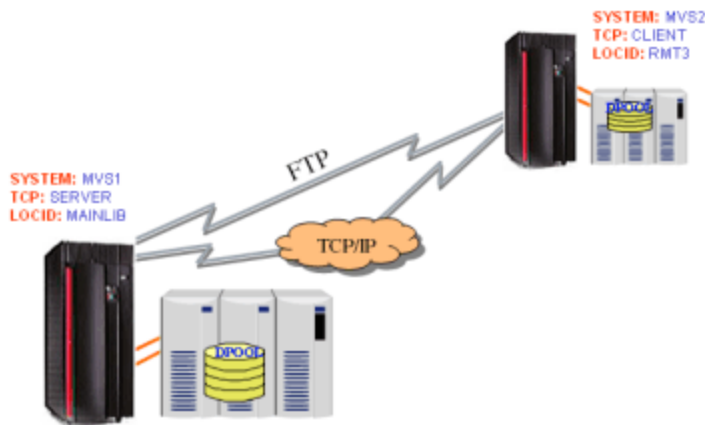


Figure 70. A typical Type4 e-Vault configuration

Configuration

The following components are required in a Type4 e-Vault environment:

- An MVS remote off-site storage location with sufficient disk space to contain the off-site data. The required disk space size is as large as the total size of the e-Vaulted virtual tape data sets.
- The above MVS should have an active FTP server.
- Running MVS and VTE at the remote off-site storage location.
- TCP/IP connection between the local and remote off-site storage location MVS's.
- Appropriate initialization statements in the VTE server.

Process

When VTE moves a virtual tape to a remote off-site storage location using Type4 e-Vault, the virtual tape dataset is FTP-ed to a remote off-site storage location managed by an MVS platform. The actual FTP is done by a started task started by the VTE server. This started task executes the HVTUFTP utility.

When the FTP process completes, the virtual tape dataset is deleted from the original diskpool. Following that process:

- The virtual tape dataset exists on the remote off-site storage location disks.
- The virtual tape is marked 'vaulted' in the local VDB and 'safe' in the remote VDB.

Advantage

The virtual tapes are immediately available on the remote off-site storage location, which is helpful should a disaster recovery situation occur.

Considerations

- More complicated process than Type3 e-Vault.
- The remote off-site storage location is required to run MVS and VTE.
- Inability to return an e-Vaulted virtual tape when the TCP/IP connection or FTP link are down.

CTC Type5 e-Vault

Type5 e-Vault is similar to the Type3 e-Vault, except for the off-site storage location type. While Type3 e-Vault uses FTP to copy files to an off-site storage locations, Type5 e-Vault uses Cloud Tape Connector (CTC), which in turn uses TCP/IP, to copy files to Cloud off-site storage locations.

The tape management system indicates the virtual tapes whose virtual tape data sets are e-Vaulted off-site. The e-Vaulted data exists at the remote off-site storage location on the Cloud until it is needed at the local location or no longer required and remote-scratched – the tape management system controls this.

The data at the remote off-site storage location cannot be used until it is returned to the local location. If the local location should need an e-Vaulted virtual tape, VTE transparently moves it back.

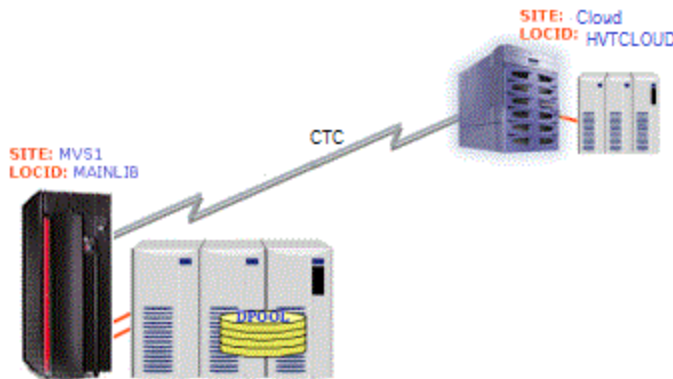


Figure 71. A typical Type5 e-Vault configuration

Configuration

The following components are required in a Type5 e-Vault environment:

- Any Cloud servers (S3, ICOS, FTP, etc..) with sufficient disk space to contain all the e-Vaulted virtual tape data sets and correctly defined in Cloud Tape Connector (CTC).
- Appropriate initialization statements in the VTE server.
- You must define a valid OMVS segment for the userid assigned to the VTE started task.

Process

When VTE moves a virtual tape to a remote off-site storage location on the Cloud using Type5 e-Vault, the virtual tape dataset is sent to a remote off-site storage location on Cloud managed by Cloud Tape Connector (CTC).

The actual sending is done by a new started task started by the VTE server. This started task executes the HVTUUTC utility.

When this process completes, the virtual tape dataset is deleted from the original diskpool. Following that process:

- The virtual tape dataset exists on the remote off-site storage location on the Cloud.

- The virtual tape is marked 'vaulted' in the VDB.

Advantages

- Simple process
- The data is stored on Clouds.

Considerations

- Inability to return an e-Vaulted virtual tape when the CTC connection to remote servers is down.
- The data on the remote platform is not usable unless returned to an MVS platform.

Additional e-Vault features

The following additional features are available with VTE e-Vault:

Multiple e-Vault off-site storage locations

Tape management systems support more than one remote off-site storage location. That means that the tape management system can decide to move one set of tapes to one offsite storage location and another set of tapes to another off-site storage location.

VTE can handle this situation and if configured properly, it can e-Vault these virtual tapes to their proper off-site storage locations.

VTE can support all types of e-Vault off-site storage locations at the same time. That is, a VTE e-Vault complex can contain a mix of type3, type4, and type5 e-Vault off-site storage locations and each virtual tape will be processed according to its off-site storage location definition.

Remote search

If a virtual tape is currently e-Vaulted in an off-site storage location and the virtual tape is to be scratched, the virtual tape is not returned. Rather, it is 'remote-scratched', that is deleted from the remote off-site storage location and marked 'scratch' in the local location.

Implicit return

When an e-Vaulted virtual tape mount is attempted, an implicit return of that virtual tape is automatically started. When the return is completed, the mount process is resumed and the job continues its normal process.

e-Vault process flow

The following section describes in more detail what are the various phases a virtual tape passes between the local location and its associated remote off-site storage location.

The e-Vault process is basically composed of two main phases:

- Phase 1 – Mark a virtual tape as 'to be vaulted'/'to be returned'/'remote scratch'.
- Phase 2 – Actual vault/return/scratch.

Phase 1

This phase is responsible for marking the 'to be vaulted'/'to be returned'/'remote scratch' status in the VDB. This phase is similar to all e-Vault types.

The initial decision on whether a given virtual tape is to be e-Vaulted (along with its vault location), returned or scratched is done by the tape management system. Various tape management systems produce reports for the vault or scratch information. The VTE HVTUSCR utility scans the appropriate produced report, and for each virtual tape it sets internal flags as follows:

- If the virtual tape is at the local location and the tape management system indicates that it should be e-Vaulted, the virtual tape is marked as 'to be vaulted' and the vault location is stored in the VDB.

- If the virtual tape is fully e-Vaulted and the tape management system indicates that it should be located at the local location, it is marked as 'to be returned'.
- If the virtual tape is at the local location and the tape management system indicates that it is scratch, the virtual tape is scratched (subject to its defer-scratch conditions).
- If the virtual tape is fully e-Vaulted and the tape management system indicates that it is scratch, the virtual tape is marked for 'remote scratch' (subject to its defer-scratch conditions).
- In any other case, the VDB entry is not changed.

Note that the HVTUSCR execution does both the regular scratch process and the e-Vault phase 1 process.

Phase 2

The actual vault/return/scratch process is done by the VTE server. The virtual tape data sets are moved to the remote off-site storage location as follows:

- The virtual tape data sets are FTP-ed to the remote off-site storage location by one or more server subtasks which in turn start one or more started tasks to perform the actual FTP (each such subtask starts one started task). The subtask waits for the started task to complete before it returns control to the VTE server.

This way, the regular VTE work is not affected.

Type3 e-Vault processing

The e-Vault process varies slightly depending on the type of processing that needs to be performed. This section describes Type3 e-Vault processing.

Below are sample statements that define one off-site storage location:

```
VLTOWN      OWNLOCID=MAINLIB,
             MAXTASK=4
VLTRMT      LOCID=RMT1,
             TYPE=3,
             FTPIP=192.114.172.19,
             PATH='/pub/incoming',
             NETRC=(HVT.V300.SHVTSAMP,HVTNETX)
```

In the above figure:

- The local location is named MAINLIB. Its VTE server uses four subtasks to FTP virtual tape data sets to the remote off-site storage location.
- The off-site storage location is named RMT1. Its IP address is 192.114.172.19 and the data sets are to be stored under the /pub/incoming directory. The FTP user and password are listed in member HVTNETX of the HVT.V300.SHVTSAMP partitioned data set.

When a virtual tape whose status is 'to be returned' is encountered, one of the free subtasks gets control of the virtual tape data set.

Following that selection, three steps are performed:

- The subtask in charge of that virtual tape data set starts a started task that executes the HVTUFTP utility. That utility FTP's the virtual tape data set back from the remote offsite storage location identified by the FTPIP and NETRC parameters to the local location. If a similar data set exists already, it is replaced.
- The 'vaulted' indication is removed from the VDB entry of this virtual tape.
- A started task that executes the HVTUFTP utility is started by the server. That utility deletes the virtual tape data set from the remote off-site storage location. If HVTUFTP fails to delete the remote data set, the virtual tape data set is still considered as successfully returned.

If a system crash occurs after step 2, the virtual tape data set on the remote off-site storage location will not be deleted, however, it will be reused when the same virtual tape will be e-Vaulted again.

Type4 e-Vault processing

The e-Vault process varies slightly depending on the type of processing that needs to be performed. This section describes Type4 e-Vault processing.

Type4 e-Vault process uses a two-layer logic for moving a virtual tape data set between the local location and the off-site storage location:

- A logical layer controls data move between the two locations. The logical layer is implemented using TCP/IP connection between the local location MVS and each one of the off-site storage locations MVS.
- A physical layer that actually FTPs virtual tape data sets from/to the local VTE diskpool from/to the off-site remote location VTE diskpool. The physical layer is implemented by several independent subtasks that start independent started tasks that actually invoke FTP to copy the virtual tape data sets.

The two layers are independent - the logical layer just initiates the physical layer different phases start.

The logical layer

The logical layer is implemented by a VTE subtask that uses a TCP/IP protocol to communicate with other VTE servers. The type4 e-Vault algorithm requires that the local location VTE runs a TCP/IP server application and each one of the off-site storage locations run a VTE TCP/IP client application.

The following figures show the logical layer initialization statements for the local and a remote off-site storage location.

```
TCP      TYPE=SERVER,
         SERVERIP=168.159.150.26,
         PORT=6160
```

```
TCP      TYPE=CLIENT,
         SERVERIP=168.159.150.26,
         PORT=6160
```

The physical layer

The physical layer consists of the hardware infrastructure required to move virtual tape data sets between sites. This layer is implemented by standard FTP links and several independent subtasks that run under the VTE server and start independent started task that invoke the above FTP services.

Below are sample initialization statements in the local location that define itself and one off-site storage locations:

```
VL TOWN  OWNLOCID=MAINLIB,
         MAXTASK=4
VL TRMT  LOCID=RMT3,
         TYPE=4,
         FTPIP=168.159.150.28,
         NETRC=(HVT.V300.SHVTSAMP,HVTNET28)
```

In the above figure:

- The local location is named MAINLIB. Its VTE server uses four subtasks to FTP virtual tape data sets to the remote off-site storage location.
- The off-site storage location is named RMT3. Its IP address is 168.159.150.28. The FTP user and password are listed in member HVTNET28 of the HVT.V300.SHVTSAMP partitioned data set.

Below are sample initialization statements in the RMT6 remote off-site storage location that define itself and the local location:

```
VL TOWN  OWNLOCID=RMT6,
         MAXTASK=3,
VL TRMT  LOCID=MAINLIB,
```

```
TYPE=4,  
FTPIP=168.159.150.26,  
NETRC=(HVT.V300.SHVTSAMP,HVTNET26)
```

In the above figure:

- The OWNLOCID parameter identifies the off-site storage location. It uses three subtasks to FTP virtual tape data sets to the local location.
- The remote location name is MAINLIB (which is the actual local site). Its IP address is 168.159.150.26. The FTP user and password are listed in member HVTNET26 of the HVT.V300.SHVTSAMP partitioned data set.

How a Virtual Tape moves between Type4 e-Vault locations

The virtual tape move between type4 e-Vault locations can be viewed as a four-phase move:

- Handshake
- The local location move
- The remote off-site storage location recovery
- Final disposition

If a system crash occurs during any of the above phases, that phase will resume as soon as the system is up.

Handshake

When the local location VTE server is started, it accepts connections from the remote offsite storage locations VTE servers. Whenever a remote off-site storage location VTE server is started, it establishes the connection with the local VTE server. Following this, both partners verify that the FTPIP address, the e-Vault type definitions and the OWNLOCID/LOCID definitions match. After the handshake succeeds, the e-Vault process can start.

The local location move

The VTE server e-Vault's subtasks periodically scan the VDB. Whenever a virtual tape whose status is "to be vaulted" is encountered, one of the free subtasks gets control of the virtual tape data set. By doing so, it selects the proper off-site storage location to which this virtual tape is to be copied along with the FTP IP address and its characteristics.

Following that selection, five steps are performed:

- The VDB entry that represents this virtual tape is marked as "obtain extent size".
- A query is initiated to the remote site in order to obtain the system parameters (extent size, unit, unit count, etc.) the FTP will use in order to allocate the virtual tape data set on the remote site.
- The subtask in charge of that virtual tape data set starts a started task that executes the HVTUFTP utility and passes to it the remote site parameters. HVTUFTP invokes the site's FTP to allocate and copy the virtual tape data set to the remote off-site storage location identified by the FTPIP and NETRC parameters. If a similar data set exists already, it is replaced. Upon completion, the VDB entry that represents this virtual tape is marked as 'FTP-ed'.
- A request is initiated to the remote off-site storage location to perform a virtual tape recovery for the virtual tape just FTP-ed.
- The subtask relinquishes control of the virtual tape.

The remote off-site storage location recovery

Upon receiving the request to recover a FTP-ed virtual tape, the following steps are taken:

- The remote off-site storage location VTE server recovers the virtual tape. Upon completion, the VDB entry that represents this virtual tape is marked as "recovered after FTP".

- A request to delete the original virtual tape data set is sent to the local location.

Final disposition

- After the local site deletes the original virtual tape data set, the VDB entry that represents this virtual tape is marked as "vaulted".
- A request is initiated to the remote off-site storage location to mark the virtual tape as "safe".

Virtual Tape return

An e-Vaulted virtual tape return can be initiated by two components:

- An explicit request made by the tape management system.
- A mount request for an e-Vaulted virtual tape performed by a job.

In both cases, the virtual tape is marked as "to be returned". The VTE server TCP/IP subtasks periodically scan the VDB. Whenever a virtual tape whose status is "to be returned" is encountered, a notification is sent to the remote off-site storage location to mark this virtual tape as 'to be vaulted' to the local location.

The virtual tape is marked as "safe" in the local location. From here on, the return process is similar to the vault process, but in a reverse order. When the return process ends, the virtual tape becomes a regular virtual tape in the local location.

Type5 e-Vault processing

The e-Vault process varies slightly depending on the type of processing that needs to be performed. This section describes Type5 e-Vault processing.

Below are sample statements that define one off-site storage location on the Cloud:

```
VLTOWN OWNLOCID=MAINLIB,
MAXTASK=4
VLTRMT LOCID=HVTCLLOUD,
TYPE=5
```

In the above sample statements, notice the following:

- The local location is named MAINLIB.
- The associated VTE server uses four subtasks to FTP virtual tape data sets to the remote off-site storage location.
- The off-site storage location is named HVTCLLOUD. This is a Cloud Tape Connector cloud name.

Steps for copying a "to be vaulted" virtual tape

The VTE server e-Vault's subtasks periodically scan the VDB. Whenever a virtual tape whose status is "to be vaulted" is encountered, one of the free subtasks gets control of the virtual tape data set. By doing so, it selects the proper off-site storage location to which this virtual tape is to be copied along with the Cloud name.

After that selection, the following steps are performed:

1. A started task that executes the HVTUCTC utility is started by the server. This utility uses Cloud Tape Connector to send the virtual tape data set to the remote off-site storage location on the Cloud server identified by the LOCID parameter. The LOCID value should be identical to the relevant CTC Cloud name.
2. The subtask that started the started task is suspended until HVTUCTC utility completes.
3. If HVTUCTC ended OK, the VDB entry that represents this virtual tape is marked as 'vaulted'.
4. The original virtual tape data set is deleted from the diskpool.

Note: If a system crash occurs after step 3, the original virtual tape data set will not be deleted. However, on the virtual tape return, this original virtual tape data set will be reused.

If a system crash occurs in any other step, the process starts all over again (as soon as the system is up).

Steps for resolving a "returned" virtual tape

When a virtual tape whose status is "to be returned" is encountered, one of the free subtasks gets control of the virtual tape data set.

After that selection, the following steps are performed:

1. The subtask in charge of that virtual tape data set starts a started task that executes the HVTUCTC utility. This utility uses Cloud Tape Connector(CTC), which in turn uses TCP/IP, to retrieve the virtual tape data set back from the remote (off-site) storage location on the Cloud identified by the LOCID parameter (local location). If a similar data set exists already, it is replaced.
2. The "vaulted" status is removed from the VDB entry of this virtual tape.
3. A started task that executes the HVTUCTC utility is started by the server. This utility deletes the virtual tape data set from the remote (off-site) storage location on the Cloud. If HVTUCTC fails to delete the remote data set, the virtual tape data set is still considered as successfully returned.

Note: If a system crash occurs after step 2, the virtual tape data set on the remote (off-site) storage location will not be deleted. However, it will be reused when the same virtual tape is e-Vaulted again.

Recovery considerations for type3 e-Vault

The following section applies to a type3 e-Vault only and describes various recovery options when a system crash occurs.

If a system crash occurs in any of the e-Vault phases, that phase is retried as soon as the crashed system is up. That means that the whole process is fully recoverable and no manual actions are to be taken by operation. However, there are situations that require manual intervention for a recovery. These situations are described below.

A virtual tape being e-Vaulted remains in an interim status.

In this case, the virtual tape is unusable. That situation can happen when a virtual tape is being e-Vaulted and the remote off-site storage location is down, the FTP links are down, etc. The scenario is also true for a virtual tape being returned to the local location.

To move virtual tapes to a stable (that is 'local' or 'vaulted') status, the HVTURCV utility should be run for the specified virtual tape. Thereafter, when the connection is restored, the e-Vault process can be started from the beginning.

The e-Vaulted virtual tape e-Vault indicators are lost.

When these indicators, like the off-site storage location name or e-Vault status, are lost, the VTE server is not able to return the virtual tapes or change the tapes to "scratch" remotely. Refer to the example scenarios below.

Scenario #1: The local site lost the VDB along with the journal file.

To restore the e-Vault indicators and enable the virtual tapes return, the following steps should be performed:

1. Run the HVTUTAP utility to add the lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTUSCR utility with the SYNC TYPE=VFTP option in order to restore the e-Vault indicators.
3. The virtual tapes can be either returned or remote scratched.
4. If returned, run the HVTURCV utility to make them usable.

Scenario #2: The local site is totally lost and the production should be resumed in a DR site.

The DR site needs to return a virtual tape from the remote off-site storage location. However, the VDB found at the DR site may not reflect the proper virtual tape status because of one of the following reasons:

- The VDB was copied to the DR site before the relevant virtual tape was created in the local site and e-Vaulted. In that case, the virtual tape is marked as scratch.
- The VDB was copied to the DR site after the relevant virtual tape was created in the local site, but before it was e-Vaulted. In that case, the virtual tape is marked as active.

To restore the e-Vault indicators, the following steps should be performed:

1. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
2. The virtual tape can be either returned or scratched remotely.
3. If returned, run the HVTURCV utility to make it usable.

Other e-Vault status

Any other e-Vault status is invalid and HVTUSCR will not be able to restore the e-Vault indicators.

Summary

To summarize, here are the four points to consider when you need to recover an e-Vaulted virtual tape:

System crash, FTP link down (but no VDB lost)

The recovery is automatic as soon as the system is up again and the links are restored.

Long non-stable interim status (due to the links going down)

Run the HVTURCV utility.

VDB Lost

Perform these steps:

1. Run the HVTUTAP utility to add lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
3. The virtual tapes can be either returned or remote scratched.
4. If returned, run the HVTURCV utility to make them usable.

Resume at a DR site

Perform these steps:

1. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
2. The virtual tape can be either returned or remote scratched.
3. If returned, run the HVTURCV utility to make it usable.

Recovery considerations for type4 e-Vault

The following section applies to type4 e-Vault only.

If a system crash occurs in any of the e-Vault phases, (either in the local MVS or remote MVS), that phase is retried as soon as the crashed system is up. That means that the whole process (either in the local or remote off-site storage locations) is fully recoverable and no manual actions are to be taken by operation. However, there are situations that require manual intervention for a recovery. These situations are described below.

A virtual tape being e-Vaulted remains in an interim status.

In this case, the virtual tape is unusable. That situation can happen when a virtual tape is being e-Vaulted and the remote off-site storage MVS is down for a long time, the FTP links are down, the TCP/IP connection is down, etc. The above scenario is true also for a virtual tape being returned to the local location.

In order to move virtual tapes to a stable (that is 'local', 'vaulted' or 'safe') status, the HVTURCV utility should be run in both the local and the remote off-site storage locations. Thereafter, when the connection is restored, the e-Vault (or return process) can be started from the beginning. Note that if the HVTURCV

utility is run only in one location, the virtual tape may still remain in an unpredictable status (that is, it is still unusable).

The VDB is restored from an older copy and no journal is available.

In this case, the HVTURCV utility is run in order to rebuild the VDB entries according to the virtual tape data sets on disk. Note that in order to synchronize the virtual tape e-Vault status, the HVTURCV utility should be run in both the local and the remote off-site storage locations. However, since HVTURCV does not consider the virtual tape current vault location from the tape management system, the correct tape locations may be lost. Therefore, after a regular recovery (using HVTURCV) is performed, a second recovery should be performed using the HVTUSCR utility. This utility will restore all these entries to their right e-Vault status and location according to their status in the tape management system. Note that as with the eVault initiation process, this second recovery should be run only in the local location.

The VDB is lost, but the virtual tapes exist on disk.

In this case, you will need to run three different utilities, as described below:

1. Run the HVTUTAP utility to add the lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTURCV utility to recover the virtual tapes and make them usable.
3. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to complete the recovery process, restore the e-Vault indicators, and mark the tapes as replicated (that is, exist on both the local and remote off-site storage location).

Summary

To summarize, here are the three points to consider when you need to recover an eVaulted virtual tape:

System crash, FTP link down, TCP/IP down (but no VDB lost):

The recovery is automatic as soon as the system is up again and the links are restored.

Long non-stable interim status (due to the links going down):

Run HVTURCV in both sites.

VDB lost:

Perform these steps:

1. Run HVTUTAP.
2. Run HVTURCV in both sites.
3. Run HVTUSCR in the local site only.

Recovery considerations for type5 e-Vault

The following section applies to a type5 e-Vault only and describes various recovery options when a system crash occurs.

If a system crash occurs in any of the e-Vault phases, that phase is retried as soon as the crashed system is up. That means that the whole process is fully recoverable and no manual actions are to be taken by operation. However, there are situations that require manual intervention for a recovery. These situations are described below.

A virtual tape being e-Vaulted remains in an interim status.

In this case, the virtual tape is unusable. That situation can happen when a virtual tape is being e-Vaulted and the remote off-site storage location is down, Cloud Tape Connector is down, etc. The scenario is also true for a virtual tape being returned to the local location.

To move virtual tapes to a stable (that is 'local' or 'vaulted') status, the HVTURCV utility should be run for the specified virtual tape. Thereafter, when the connection is restored, the e-Vault process can be started from the beginning.

The e-Vaulted virtual tape e-Vault indicators are lost.

When these indicators, like the off-site storage location name or e-Vault status, are lost, the VTE server is not able to return the virtual tapes or change the tapes to "scratch" remotely. Refer to the example scenarios below.

Scenario #1: The local site lost the VDB along with the journal file.

In order to restore the e-Vault indicators and enable the virtual tapes return, the following steps should be performed:

1. Run the HVTUTAP utility to add the lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTUSCR utility with the SYNC TYPE=VFTP option in order to restore the e-Vault indicators.
3. The virtual tapes can be either returned or remote scratched.
4. If returned, run the HVTURCV utility to make them usable.

Scenario #2: The local site is totally lost and the production should be resumed in a DR site.

The DR site needs to return a virtual tape from the remote off-site storage location. However, the VDB found at the DR site may not reflect the proper virtual tape status because of one of the following reasons:

- The VDB was copied to the DR site before the relevant virtual tape was created in the local site and e-Vaulted. In that case, the virtual tape is marked as scratch.
- The VDB was copied to the DR site after the relevant virtual tape was created in the local site, but before it was e-Vaulted. In that case, the virtual tape is marked as active.

To restore the e-Vault indicators, the following steps should be performed:

1. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
2. The virtual tape can be either returned or scratched remotely.
3. If returned, run the HVTURCV utility to make it usable.

Other e-Vault status

Any other e-Vault status is invalid and HVTUSCR will not be able to restore the e-Vault indicators.

Summary

To summarize, here are the four points to consider when you need to recover an e-Vaulted virtual tape:

System crash, Cloud Tape Connector (CTC) link down (but no VDB lost)

The recovery is automatic as soon as the system is up again and the links are restored.

Long non-stable interim status (due to the links going down)

Run the HVTURCV utility.

VDB Lost

Perform these steps:

1. Run the HVTUTAP utility to add lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
3. The virtual tapes can be either returned or remote scratched.
4. If returned, run the HVTURCV utility to make them usable.

Resume at a DR site

Perform these steps:

1. Run the HVTUSCR utility with the SYNC TYPE=VFTP option to restore the e-Vault indicators.
2. The virtual tape can be either returned or remote scratched.
3. If returned, run the HVTURCV utility to make it usable.

Implementation samples

This section provides an example of how to implement an e-Vault complex, which uses multiple locations.

Implementation sample for Type3, Type4, and Type5 e-Vault Complex

The following section describes a sample e-Vault complex in which the local location is connected to three remote off-site storage locations—one type3, one type4, and one type5.

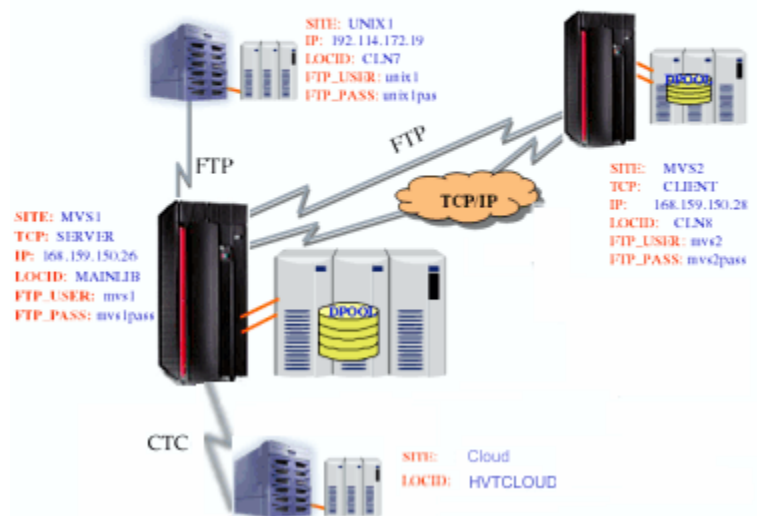


Figure 72. An e-Vault complex with one type3, one type4, and one type5 e-Vault location

The following notes are for the samples below:

- The code examples below contain only the e-Vault relevant VTE initialization statements. Other initialization statements are not shown.
- These examples are for explanatory purposes only. Actual implementation may require different parameters.

VTE initialization statements for location MAINLIB

The HVTOPTx member in location MAINLIB is as follows:

```

OPTIONS DYR=ENABLE,
JRN=DISABLE,
TCP=ENABLE,          TCP/IP IS ENABLED
VLT=ENABLE           E-VAULT IS ENABLED
TCP TYPE=SERVER,    <--- RUNS AS A TCP/IP SERVER
SERVERIP=168.159.150.26,
PORT=6160

```

The HVTVLTx member in location MAINLIB is as follows:

```

VLTOWN MAXTASK=4,          4 TASKS TO FTP DATA
OWNLOCID=MAINLIB          LOCAL LOCATION IS MAINLIB
VLTRMT LOCID=CLN7,         REMOTE1 LOCATION IS CLN7
TYPE=3,                   THIS IS A TYPE3 E-VAULT
FTPIP=192.114.172.19,     CLN7 LOCATION IP ADDRESS
PATH='/pub/incoming',
NETRC=(HVT.V300.SHVTSAMP,HVTNETX) FTP LOGIN PARAMETERS
VLTRMT LOCID=CLN8,         REMOTE2 LOCATION IS CLN8
TYPE=4,                   THIS IS A TYPE4 E-VAULT
FTPIP=168.159.150.28,     CLN8 LOCATION IP ADDRESS

```



```

NETRC=(HVT.V300.SHVTSAMP,HVTNET28) FTP LOGIN PARAMETERS
VLTRMT LOCID=HVTCLLOUD,          REMOTE3 LOCATION IS HVTCLLOUD
      TYPE=5                      THIS IS A TYPE5 E-VAULT

```

The HVTNET28 member in location MAINLIB is as follows:

```

machine 168.159.150.028 login mvs2      password mvs2pass

```

The HVTNETX member in location MAINLIB is as follows:

```

machine 192.114.172.019 login unix1     password unix1pas

```

VTE initialization statements for location CLN8

The HVTOPTxx member in location CLN8 is as follows:

```

OPTIONS DYR=ENABLE,
        JRN=DISABLE,
        TCP=ENABLE,          TCP/IP IS ENABLED
        VLT=ENABLE          E-VAULT IS ENABLED
TCP      TYPE=CLIENT,        <--- RUNS AS A TCP/IP CLIENT
        SERVERIP=168.159.150.28,
        PORT=6160

```

The HVTVLTxx member in location CLN8 is as follows:

```

VLTOWN MAXTASK=4,           4 TASKS TO FTP DATA
      OWNLOCID=CLN8         LOCAL LOCATION IS CLN8
VLTRMT LOCID=MAINLIB,       REMOTE1 LOCATION IS MAINLIB
      TYPE=4,              THIS IS A TYPE4 E-VAULT
      FTPIP=168.159.150.26, MAINLIB LOCATION IP ADDRESS
      NETRC=(HVT.V300.SHVTSAMP,HVTNET26)FTP LOGIN PARAMETERS

```

The HVTNET26 member in location CLN8 is as follows:

```

machine 168.159.150.026 login mvs1      password mvs1pass

```

Chapter 23. Virtual Tape Replication

The replication feature is a method by which the VTE server copies virtual tape datasets to remote locations thus maintaining two copies of the same virtual tape dataset.

The replication feature should not be confused with mirroring. While mirroring requires that every written tape block is to be written on the mirror site before the I/O completion is posted, the replication process copies full virtual tape datasets only after their associated virtual tapes are unloaded.

The replication process uses the type3 and type5 e-Vault infrastructures and is associated with one or more tapepools. These tapepools are replication-eligible only if the following three conditions apply:

1. The VLT option is ENABLE-d in the server.
2. The tapepool contain the LOCID parameter.
3. The LOCID parameter value specifies a type3/type5 e-Vault location.

A virtual tape is replication-eligible if it belongs to a tapepool that is replication-eligible.

Replicating tapes

When a virtual tape is replication-eligible, upon unload (and provided it has been written to at least one block), the replication process will start, subject to the server replication status which is based on either of the following:

- RPL=ENABLE | DISABLE parameter on the OPTIONS initialization statement
- RPLON | RPOFF operator command

The whole virtual tape dataset is copied to the remote off-site storage location specified in the associated tapepool and the local copy remains on disk.

Note: The virtual tape is not eligible for use as long as it waits to be replicated. Any attempt to use it leads to the HVT413W message asking the operator whether to wait or cancel the waiting job.

When the virtual tape is scratched, both the local and the remote copies are deleted.

If local copy is deleted

If the local copy is inadvertently deleted:

1. An attempt to access the virtual tape will cause the remote copy to be automatically returned and used. Upon completion, and provided it has been modified, it will be replicated and replace the remote copy.
2. HVTUSCR utility can be used to return the remote copy.

In both cases the remote copy is not deleted.

If the local copy exists, HVTUSCR utility will NOT cause it to be replicated nor the remote copy to be returned.

Processing a DR case

In a DR case, when virtual tape datasets are copied from the remote off-site storage location to the local location after using the HVTUSCR utility with TYPE=VFTP option, the remote copy is not deleted.

Recovery considerations

If the VDB is lost but the virtual tapes exist on disk, perform the following steps:

1. Run the HVTUTAP utility to add the lost virtual tapes to the VDB as scratch virtual tapes.
2. Run the HVTURCV utility to recover the virtual tapes and make them usable.

3. Run the HVTUSCR utility with the SYNC TYPE=VFTP option in order to complete the recovery process, restore the e-Vault indicators and mark the tapes as replicated (that is, exist on both the local and remote off-site storage location).

Chapter 24. zIIP Processing

The VTE zIIP support allows most of the VTE CPU intensive work to be zIIP-eligible, that is, eligible to be dispatched on (or offloaded to) zIIP processor(s). This intensive CPU work includes general data moves, all types of compression and other internal tasks (like addresses translation).

This capability is enabled provided the following is true:

- The CPC machine supports zIIP.
- The system level supports zIIP.
- The host has at least one online zIIP processor.
- No zIIP processors are available, but the z/OS capacity planning projection for zIIP processor(s) is enabled (PROJECTCPU=YES in IEAOPTxx member), in which case the projected CPU is accumulated as zIIP on CP time.

When the zIIP processor(s) are available and all scheduled work can be dispatched on them, as much as 97% of the VTE server work is offloaded from the CP processor(s).

Note: Due to the special processing required by the zIIP processor(s) (i.e. managing enclave SRBs), the total consumed CPU (i.e. CP + zIIP) slightly increases.

The following table depicts an estimation of work percentage that is offloaded from the CP processor(s) to the zIIP processor(s) when virtual tapes are written using the various compression algorithms.

Note: The estimated offloaded work results depend on the compression ratio and were obtained using a mix of full disk backup jobs and various application jobs.

Table 10. Compression algorithms and CPU savings	
Compression Type	CPU % Savings
(0,0)	6
(1,1)	76
(2,1)	97
(3,1)	62

The read results vary and depend on the compression ratio results. The better the compression ratio is, the more percentage of workload is offloaded to the zIIP processor(s).

The amount of offloaded work is displayed for each accessed virtual tape in the HVT023I message and is updated in the VTE SMF record.

Note: The zIIP support is not available for VTE offline utilities (like HVTUV2V) or to utilities that use the VTE Access Method (VAM) to access virtual tapes.

Chapter 25. Set percent of zIIP-eligible work to be offloaded

The MVS dispatcher can dispatch all or only a part of the zIIP-eligible work to zIIP processor(s). The amount of work to be actually offloaded to zIIP processor(s) can be dynamically changed.

The following actions are taken when the percent of zIIP-eligible work to be offloaded is changed:

- The server checks whether the operating system level supports zIIP and the host has at least one online zIIP processor.
 - If positive, the percent of zIIP-eligible work to be offloaded is changed and HVT029I message displays the updated value.
 - Otherwise, the percent of zIIP-eligible work to be offloaded is unchanged and HVT029I message displays the old values.

The following section describes how the VTE server workload is affected by changing the amount of work to be actually offloaded to zIIP processor(s).

- A virtual tape percent of zIIP-eligible work to be offloaded is set at mount time.
- Changing the amount of work to be offloaded to zIIP processor(s) affects only virtual tapes mounted after the change was done.

The percent of zIIP-eligible work to be offloaded can be changed by issuing the following operator command on the system console:

```
F server,ZIIP%=percent
```

Where *percent* is an integer from 1 to 100.

Chapter 26. Electronic Migration (e-Mig)

The Electronic Migration (e-Mig) feature is a method by which the VTE server migrates selected virtual tape datasets off the managed diskpools.

The selection of the virtual tape datasets to be migrated is based on the VTE capability to predict virtual tape activity, which is implemented using the HVTUDST utility. The HVTUDST utility predicts virtual tape activity and selects the virtual tape datasets that are the best candidates for migration (to free space on the VTE diskpools, thus meeting user-defined thresholds).

Highlights of electronic migration (e-Mig)

Highlights of the e-Mig feature are the following:

Provides a tapeless solution for migrating virtual tape datasets.

Electronic migration (e-Mig) is based on FTP services enabling the VTE server to migrate virtual tape datasets from the MVS platform to any FTP server with assigned disk storage. The destination FTP server can be either another MVS platform or an open system platform.

By doing so, electronic migration relieves several constraints related to migrating datasets using common migration products:

- Recycling

Since no tapes are involved, the e-Mig feature eliminates the need to recycle real large migration tape volumes in order to free space on them. This saves time and host cycles.

- Stacking

Since no tapes are involved, the e-Mig feature eliminates the need to stack migrated datasets on large tape volumes in order to better utilize them. This also exposes the customer to a situation where loss of one migration tape causes the loss of many migrated datasets. The e-Mig feature migrates each dataset independently, not depending on or affecting the migration of other datasets.

- Concatenated Tapes

Since no tapes are involved, the e-Mig feature eliminates the grief caused by chains of migrated tape volumes, which can lead to a situation where a loss of one migration tape affects the other migration tapes that are in the same chain.

Uses low cost disk storage arrays.

Using the e-Mig feature, the VTE server migrates virtual tape datasets from the mainframe's high-end storage to open system low-end storage, which reduces storage costs.

Electronic migration is a complete VTE feature for its diskpool management.

The e-Mig feature provides the VTE server with full control of the migration process from start to end. Using this feature:

- VTE is the engine that moves the virtual tape datasets.
- VTE maintains the complete information about migrated virtual tape datasets.
- VTE is responsible to recall migrated virtual tape datasets upon request.

The e-Mig feature is a fully automated and transparent process to users and applications. It performs all common functions of migration, recall and scratch similar to other external disk manager systems.

How electronic migration works

When VTE diskpools become full, you may want to migrate virtual tape datasets to real tapes, thus freeing space for more datasets. The migration process can be controlled and performed by any available external disk management product (i.e. DFSMSHsm, FDRABR or CA-Disk).

VTE e-Mig (electronic migration) extends the migration support by letting VTE perform the actual migration off its diskpools to other MVS or open system platforms. The actual migration process is done using standard FTP services and is built on top of the VTE e-Vault feature. Virtual tape datasets are migrated and recalled or scratched upon request with no manual intervention. The whole process fully resembles the operation performed by external storage management products, where it is transparent and fully automated to users and applications.

The e-Mig selection process is based on the HVTUDST utility, which selects the best virtual tape dataset candidates eligible for migration.

HVTUDST utility is used to control the utilization level of the VTE diskpools. It is responsible for prioritizing the migration of virtual tape datasets and provides the input for any external storage management product or e-Mig process that performs the actual migration and recall operations. See [“HVTUDST – Set De-Stage Priority” on page 317](#).

To use e-Mig, the e-Vault feature should be activated as described in [Chapter 22, “e-Vault - VTE electronic vaulting,” on page 191](#).

The following steps are required in order to migrate virtual tape datasets using e-Mig:

1. Activate the e-Vault feature (type3, type4, or type5 e-Vault) as described in [Chapter 22, “e-Vault - VTE electronic vaulting,” on page 191](#). As part of this activity a new remote location should be defined to act as the e-Mig destination for migrated virtual tape datasets.
2. Activate and run the HVTUDST utility on a daily basis to accumulate history data regarding virtual tapes activity.
3. Run the HVTUDST utility to produce a list of the best virtual tape dataset candidates that are eligible for migration. When the utility is customized to work in conjunction with e-Mig (that is, PROD=EMG is defined in the utility's DRUN initialization statement), it will use this list to tailor and build a migration job that, once submitted, will initiate the actual migration process.
4. Check the migration job. This job is based on the HVTUSCR utility that is used to actually update the VDB regarding the migration eligibility of the virtual tape datasets.
5. Once submitted, the migration job will initiate the migration process of the selected virtual tape datasets to the remote location defined in step 1.

Chapter 27. VAM Access Method

VAM (Virtual Access Method) provides a means by which customers are able to access virtual tapes without the VTE server services. By invoking the necessary VAM services, the invoker can read, write and create new virtual tapes. Virtual tapes created using VAM services can be accessed later by applications as regular tapes using the VTE server. Virtual tapes created by applications using the VTE server can be accessed using the VAM services.

User applications that invoke VAM services to access virtual tapes (read, write or create) are associated with a VTE environment. That is, the user application that invokes VAM services must specify the same STEPLIB, VDB, journal file and the same initialization statements as the associated VTE server. It is suggested therefore that in order to run the user application, the customer uses the VTE procedure where he provides the appropriate SERVICE symbolic parameter value (that is, if the user application program name is HVTJVM then SERVICE=JVM).

Virtual tapes created or updated using VAM services can be accessed by the associated VTE server and vice versa. No special indication exists on what environment created or recently accessed the virtual tape.

Virtual tapes accessed using VAM services inherit all the characteristics of similar virtual tapes used by the VTE server. That means that no matter under what environment the virtual tape is accessed, its tape capacity, tapepool, diskpool and other characteristics are the same.

The VAM Access Method allows customers to read or write blocks of data. However VAM does not replace the standard access methods (QSAM or BSAM) tape management. VAM does not build nor check tape labels during OPEN. In addition, VAM does not perform the EOVS functions usually performed by these access methods (write trailer and header labels or switch tapes). That means that the VAM services invoker should perform all related tasks, like writing and checking the labels and switching tapes.

The user application can access up to 256 virtual tapes simultaneously. However only one virtual tape can be accessed by a given task. That means that if more than one virtual tape is to be accessed simultaneously, the user application should attach one or more subtasks and each one of them should access a different virtual tape. Once a given subtask closes the access to a given virtual tape, it can open an access to another virtual tape.

Basic functions

The VAM Access Method consists of the HVTVM callable macro with 7 different functions. The functions are grouped as follows:

- INIT, TERM – Initiate and terminate the VAM access method environment.
- OPEN, CLOSE – Open and close a virtual tape access.
- READ – Read one block from the virtual tape.
- WRITE, WRITETM – Write one block on the virtual tape.

Environment

The requirements for the environment for using VAM are described below.

Authorization

The minimum authorization requirements are the following:

- The user application must be APF authorized.
- The user application can be in either "problem" mode or "supervisor" mode. VAM does not modify the initial mode setting.

AMODE

The user application must be in 31-bit addressing mode.

Control parameters

The parameters can reside in any addressable area.

Programming requirements

The user application must provide an ESTAE environment. No ESTAE is provided by the VAM Access Method.

Restrictions

The restrictions for using VAM are described below:

- VTE SMF recording is not performed for the accessed virtual tapes.
- Virtual tapes accessed using VAM services are not recorded by the local TMS. Special attention should be taken when the HVTUSCR utility SYNCs the VDB contents with the TMC contents (since these virtual tapes are not recorded in the TMC and can therefore go scratch).
- Virtual tapes created using VAM have TEMPORAR as their last level qualifier dataset name on disk.
- The Tape pool Rules (TRULE) feature does not consider the DSN parameter when accessing virtual tapes using VAM services.
- The CPU usage reported by HVT023I message contains only the CPU used by VAM. Other task CPU is not reported by it.
- Message HVT250I (tape eligible for compression) is not issued even though all types of compression are available for virtual tapes accessed by the VAM Access Method.

Passing Parameters via the JCL PARM Parameter

User applications that invoke VAM services must specify SYS=*xx* in the PARM parameter of the JCL EXEC statement (where the *xx* value is similar to the *xx* value of the associated VTE server).

Additional parameters can be passed to the user application via the JCL PARM parameter as follows:

- Up to four strings separated by commas can be specified in the JCL PARM parameter.
- The strings can contain any character except '=' (equal sign).
- The strings can be of any length, however the total PARM parameter length should not exceed 100.
- All strings must precede the SYS=*xx* parameter. The SYS=*xx* parameter that follows is mandatory.
- When the user application starts execution, the whole PARM field length and address is pointed by register 1.

The example below shows how to specify two additional parameters in the JCL PARM parameter:

```
//VAM EXEC PGM=HVTJVAM1, PARM='AA,BB,SYS=00'
```

VAM Registers

Information about the use of registers for Virtual Access Method (VAM) is described in this section. The requirements are provided for input, output, and access VAM registers

Input register information

Before issuing the HVTVAM macro, the caller must ensure that the following GPRs contain the specified information:

Table 11. Input register	
Register	Contents
13	Address of a standard 18-word save area

Output register information

When control returns to the caller, the GPRs contain the following information:

Table 12. Output register	
Register	Contents
0-1	Used as work registers by the system
2 -13	Unchanged
14	Used as work register by the system
15	Return code

Access registers

When control returns to the caller, the access registers (ARs) contain:

Table 13. Access registers	
Register	Contents
0 - 15	Unchanged

VAM Syntax and Parameters

The VAM access method has a label command and various parameters that can be assigned with it. The syntax and the parameters are described below.

Syntax

```
label    HVTVAM function,
        HANDLE=handle
        [, BUFF=buff, BUFFLEN=bufflen]
        [, VOL=vol, SIZE=size]
```

Parameters

Each of the parameters is described below, along with the possible values for each parameter.

function

Indicates the function you want HVTVAM to perform. Possible values are the following:

INIT

Requests that the VAM Access Method environment is to be initialized. That invocation should be done once at the user application start. The environment will exist until terminated by the associated TERM invocation.

Note: No two consecutive INIT requests should be done without an intervening TERM request.

TERM

Requests that the VAM Access Method environment is to be terminated. That invocation should be done once at the user application end and after all accesses to virtual tapes have completed.

The TERM request must be invoked by the same task that invoked the INIT request.

OPEN

Requests to open a virtual tape access. Only one virtual tape can be opened simultaneously by one task. If more than one virtual tape is to be opened simultaneously, a multi-task user application should be designed.

A second virtual tape access can be opened by the same task once the previous access has been closed.

The OPEN request does not build a tape header label. After the OPEN completion, the virtual tape is positioned at the Begin-of-Tape location and is marked in the associated VDB as in-use.

CLOSE

Requests to close a virtual tape access. The CLOSE request does not build a tape trailer label. After the CLOSE completion the virtual tape in-use indicator is removed from the VDB.

WRITE

Requests to write a block of data at the current position on the virtual tape. After the write completion the virtual tape positioning is advanced by one block.

WRITEM

Requests to write a tapemark at the current position on the virtual tape. After the write completion, the virtual tape positioning is advanced by one block.

READ

Requests to read a block of data from the current position on the virtual tape. After the write completion, the virtual tape positioning is advanced by one block.

,HANDLE=handle

RX-type address or register (2) - (12).

Specifies the address of a 256 bytes area to be used by the VAM Access Method as a handle between its various invocations. The handle should be obtained by the user application and must be aligned on a fullword boundary. The user application should store the handle length in the first four bytes of that handle before the INIT invocation.

The handle should be passed unmodified to all subsequent VAM Access Method invocations.

,VOL=vol

RX-type address or register (2) - (12).

Specifies the address of a 6-byte area field that contains the volume serial number of a virtual tape to be opened or blanks.

- If the area is not blank, a specific tape mount request is simulated and that specific virtual tape is opened.
- If the area is blank, a non-specific tape mount request is simulated and that scratch virtual tape is opened. In that case, the opened virtual tape volume serial number is returned in the *vol* field.

This parameter is required only for OPEN requests. For other requests it is ignored.

,SIZE=size

RX-type address or register (2) - (12).

Specifies the address of a fullword area in which the OPEN request will return the size of the associated opened virtual tape dataset in unit of tracks.

The size value is returned only if all of the following conditions are true:

- The OPEN request is for a specific virtual tape.
- The virtual tape is not scratch.
- The virtual tape exists in the VDB.

In all other cases the returned value is 0.

This parameter is required only for OPEN requests. For other requests it is ignored.

,BUFF=buff

RX-type address or register (2) - (12).

Specifies the address of an area field that:

- Contains the data to be written (if the user application request is WRITE).
- Receives the read data (if the user application request is READ).

This parameter is required only for READ and WRITE requests. For other requests it is ignored.

,BUFFLEN=bufflen

RX-type address or register (2) - (12).

Specifies the length of the buffer area.

- If the request is WRITE, a block of that length will be written. The *bufflen* value should not exceed x'3FFFF' (or 256K).
- If the request is READ, that is the maximum data length passed to the user application upon the read completion.

If the read block is larger than that value, the *bufflen* size will not be exceeded. The *bufflen* value should not exceed x'3FFFF' (or 256K).

Upon the read completion, the *bufflen* area contains the length of the read buffer.

This parameter is required only for READ and WRITE requests. For other requests it is ignored.

Return Codes

This section lists all of the return codes that could be generated based on the *function* performed by HVTVM.

INIT

When control is returned from HVTVM INIT, register 15 contains one of the following decimal return codes:

Table 14. INIT return codes	
Value	Description
0	OK
12	Initialization statements error
16	Wrong VDB/Journal files
20	Handle too short - should not call 'TERM'.
36	Application not APF authorized - should not call 'TERM'.
44	Two consecutive 'INIT' requests without an intervening 'TERM' request.

TERM

When control is returned from HVTVM TERM, register 15 contains one of the following decimal return codes:

Table 15. TERM return codes	
Value	Description
0	OK
20	Bad handle, SA03 abend may further occur.
28	One or more tapes are still open.

OPEN

When control is returned from HVTVM OPEN, register 15 contains one of the following decimal return codes:

Table 16. OPEN return codes	
Value	Description
0	OK
12	Internal error (HVTxxxy messages issued on job log).
20	Bad handle, SA03 abend may further occur.

Table 16. OPEN return codes (continued)	
Value	Description
24	More than 256 opened virtual tapes.
28	Two or more consecutive OPENs without an intervening CLOSE.
48	Open rejected by EXIT006.
52	VAM device UCB could not be obtained.

CLOSE

When control is returned from HVTVAM CLOSE, register 15 contains one of the following decimal return codes:

Table 17. CLOSE return codes	
Value	Description
0	OK
12	Internal error (HVTxxxx messages issued on job log).
20	Bad handle, SA03 abend may further occur.
24	Associated control blocks that represent the pseudo virtual device not found.

READ

When control is returned from HVTVAM READ, register 15 contains one of the following decimal return codes:

Table 18. READ return codes	
Value	Description
0	OK
4	Tapemark detected, returned <i>bufflen</i> is 0.
8	EOD during read, returned <i>bufflen</i> is 0.
12	Internal error (HVTxxxx messages issued on job log).
20	Bad handle, SA03 abend may further occur.
24	Associated control blocks that represent the pseudo virtual device not found.
32	Output buffer too small - partial data moved.
40	Data expand failed (incl. output buffer too small). No data moved.

WRITE

When control is returned from HVTVAM WRITE, register 15 contains one of the following decimal return codes:

Table 19. WRITE return codes	
Value	Description
0	OK
4	EOV during write. The current block has been written but the user application should CLOSE the virtual tape.
12	Internal error (HVTxxxx messages issued on job log).

Table 19. WRITE return codes (continued)	
Value	Description
20	Bad handle, SA03 abend may further occur.
24	Associated control blocks that represent the pseudo virtual device not found.
32	Input buffer exceeds maximum supported size of 256K.

WRITETM

When control is returned from HVTVAM WRITETM, register 15 contains one of the following decimal return codes:

Table 20. WRITETM return codes	
Value	Description
0	OK
4	EOV during write. The current tapemark has been written but the user application should CLOSE the virtual tape.
12	Internal error (HVTxxxy messages issued on job log).
20	Bad handle, SA03 abend may further occur.
24	Associated control blocks that represent the pseudo virtual device not found.

Examples

Two examples can be found in the VTE SAMPLIB to help you understand how to use the VAM access method.

HVTJVAM1

An example of copying a real tape to a scratch virtual tape using the VAM Access Method can be found in member HVTJVAM1 in the VTE SAMPLIB library.

HVTJVAM2

An example of using the VAM Access Method to access more than one virtual tape concurrently can be found in member HVTJVAM2 in the VTE SAMPLIB library.

List form of HVTVAM and parameter

The list form of HVTVAM assigns the correct amount of storage for a HVTVAM parameter list.

The list form is written as follows:

```
label    HVTVAM MF=L
```

Where MF=L specifies the list form of the HVTVAM macro.

Execute form of HVTVAM and parameter

A remote control parameter list is used in, and can be modified by, the execute form of the HVTVAM macro. The parameter list can be generated by the list form of the macro.

The execute form is written as follows:

```
label    HVTVAM list,MF=(E,ctl-addr)
```

Where:

list

Specifies the parameter list described under the macro syntax topic.

,MF=(E,*cntl-addr*)

RX-type address or register (2) - (12).

Specifies the execute form, which uses a remote parameter list. The *cntl-addr* variable specifies the address of the remote parameter list, created by a list generated by the list form of the macro.

Chapter 28. The VTE Dialog Customization

This material describes how to customize the dialog, meaning the ISPF panels, so you can access and maneuver through them easily in your environment.

The VTE Dialog basic installation

The VTE Dialog basic installation is accomplished once the PTF for APAR PI50621 and subsequent APARs are applied and the basic installation steps described in the PTF cover letters are completed. At this point, the VTE Dialog is installed and disabled. In order to enable it, the following customization steps should be followed.

The VTE Dialog customization steps

In order to enable the VTE Dialog, the following customization steps should be performed.

TSO setup

Steps for preparing to use VTE in a TSO environment

1. **Copy the VTE Dialog EXEC to the SYSEXEC concatenation**

- a. Edit member HVTEXEC in the SAMPLIB library. It contains the VTE Dialog EXEC.

Substitute the #-prefixed parameters as required, then save the member.

- b. Edit member HVTJCEXE in the SAMPLIB library.

Substitute the #-prefixed parameters as required, then submit the job. The job should end with a return code of 0.

2. **TSO logon procedure APF considerations**

- a. Add the VTE load library to the STEPLIB concatenation of the TSO logon procedure.
b. Make sure that all load libraries concatenated to the STEPLIB concatenation of the TSO logon procedure are APF-Authorized.

3. **Define authorized programs to TSO**

- a. Edit the active IKJTSOxx member in the system PARMLIB and add module HVTTAPF to the following two lists:

AUTHPGM

The list of authorized programs.

AUTHTSF

The list of APF-authorized programs that may be called through the TSO service facility.

- b. Refresh the above lists by issuing the following operator command:

```
T IKJTSO=xx
```

where xx is the member suffix.

- c. Please refer to the example (below) of a tailored IKJTSOxx member:

```
AUTHPGM NAMES(          /* AUTHORIZED PROGRAMS      */ +
  PGM1             /*          */ +
  PGM2             /*          */ +
  HVTTAPF )        /*          */ +
                  /*          */ +
                  /*          */ +
                  /*          */ +
  AUTHTSF NAMES(      /* PROGRAMS TO BE AUTHORIZED */ +
                  /* WHEN CALLED THROUGH THE  */ +
```

```

PGM3          /* TSO SERVICE FACILITY.    */ +
PGM4          /*                          */ +
HVTTAPF )     /*                          */ +
              /*                          */
              /*                          */

```

ISPF setup

To use the product, you will want to add the VTE Dialog (panels) to ISPF, and understand how to move around the panels.

Steps for setting up the panels in ISPF

1. Add the VTE Dialog option to the ISPF Primary Option Menu panel

- a. Edit the ISPF Primary Option Menu panel.
- b. Add the following line in the)BODY section:

```
xx,VTE Dialog      Virtual Tape Emulation
```

where xx is any selection identifier used to access the VTE Dialog.

- c. Add the following line in the)PROC section in order to assign a new value to the &ZSEL variable:

```
xx, 'CMD(yyyyyyyy)'
```

where xx is the previously used selection identifier and yyyyyyyy is the VTE EXEC which was previously copied to the SYSEXEC concatenation.

- d. Please refer to next figure for a partial example of a tailored ISPF Primary Option Menu panel:

```

)BODY CMD(ZCMD)
0  Settings      Terminal and user parameters
1  View          Display source data or listings

11 Workplace    ISPF Object/Action Workplace
xx VTE Dialog    Virtual Tape Emulation
. . .
)PROC
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
0, 'PGM(ISPISM) SCRNAME(SETTINGS)'
1, 'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW)'
. . .
11, 'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
xx, 'CMD(yyyyyyyy)'
. . .
)END

```

Figure 73. ISPF Primary Option Menu panel containing the VTE Dialog

2. Enable the LEFT and RIGHT commands

- a. Enter ISPF option 3.9 (Command Table Utility) and select Application ID HVT.
- b. In the newly created HVTcmds table:
 - Type E (Edit) on the first line.
 - On the Extended Command Entry panel, type the following:

```

Verb   . . . LEFT
Trunc  . . . 0
Action . . . PASSTHRU

```

- Press Enter and exit the panel.
- c. Above process is to be followed also for Verb=RIGHT.
 - d. Please refer to next figure for an example of the ISPF Extended Command Entry panel:

Extended Command Entry

Command ==>

Make changes to the command and select Update to update the entry or Cancel to ignore the changes.

Verb . . . LEFT
 Trunc . . . 0
 Action . . PASSTHRU

Description	Update	Cancel
-------------	--------	--------

Figure 74. ISPF Extended Command Entry panel containing the VTE dialog

3. Enable the PF Keys definitions display

If the PF Keys definition display on the VTE Dialog panels is not needed, skip this step.

- a. Edit HVTDLGxx member in the SAMPLIB library and locate the DLGPFK initialization statement.
- b. On the DLGPFK initialization statement, set the PFSHOW parameter value to ENABLE:

```
DLG=ENABLE
```

The HVTDLGxx member initialization statements are described in the [“HVTDLGxx – VTE Dialog initialization statements”](#) on page 226 topic.

- c. If the VTE server is active, refresh the member by using the following operator command:

```
'F server,DLG=xx'
```

- d. Use the PFSHOW ISPF command to display the PF Keys definitions.

Note that due to limited panels size, displaying the PF Keys may overlay useful VTE Dialog data.

VTE server resources modification authorization

Requests that result in modification of VTE server resources or managed entities are protected by an internal VTE Dialog authorization process. By default, the VTE Dialog provides no authorization to perform any modification.

Perform the following steps to change the default:

1. Edit HVTDLGxx member in the SAMPLIB library and locate the DLGAUTH initialization statement(s).
2. Tailor the DLGAUTH initialization statement(s) as needed.

The HVTDLGxx member initialization statements are described in the [“HVTDLGxx – VTE Dialog initialization statements”](#) on page 226 topic.

3. If the VTE server is active, refresh the member by using the following operator command:

```
'F server,DLG=xx'
```

Enable the VTE Dialog

Completing a few short steps will enable you to use the VTE Dialog.

Steps to complete to enable the use of the VTE Dialog

1. Set the HVTDLGxx member name default suffix

- a. Edit HVTSYSxx member in the SAMPLIB library and locate the DLG=xx parameter. This parameter specifies the HVTDLGxx member that contains the VTE Dialog initialization statements.
- b. Set the xx parameter to a two-character member suffix to be appended to HVTDLG to form the name of the HVTDLGxx member that will be used.

2. Enable the VTE Dialog

- a. Edit HVTOPxxx member in the SAMPLIB library and locate the OPTIONS initialization statement.
- b. On the OPTIONS initialization statement, set the DLG parameter value to ENABLE:

```
DLG=ENABLE
```

For immediate activation of the VTE Dialog, enter the following operator command:

```
F server,DLGON
```

HVTDLGxx – VTE Dialog initialization statements

The HVTDLGxx utility is used to define and initialize the ISPF panels and authorities.

Purpose of HVTDLGxx member

The HVTDLGxx member:

- Defines the definitions that controls the VTE Dialog PF Keys display.
- Defines the VTE Dialog authorization definitions.

The following rules apply to the VTE Dialog authorization process:

- Prior to performing any request that results in a modification to VTE resources or managed entities, VTE Dialog invokes the authorization process and the request is rejected should the user miss the right authorization.
- All users can access the VTE Dialog Primary Application Selection panel.
- All users can watch and inspect the VTE resources and managed entities.
- Any modification to VTE resources or managed entities is subject to the VTE Dialog authorization process:
 - The authorization process successively scans all DLGAUTH initialization statements with a matching userid and considers it to be the matching DLGAUTH initialization statement. If no matching userid, the request is rejected.
 - The authorization process scans the matching DLGAUTH initialization statement for an application identical to the invoked application. If no matching application, the request is rejected.
 - The authorization process scans the matching application for a request identical to the invoked request. If no matching request, the request is rejected.
- Any modification attempt to VTE resources or managed entities (either successful or not), is audited in the VTE server log.

The HVTDLGxx member contains:

- None or one DLGPfK initialization statement.
- None, one, or as many as 50 DLGAUTH initialization statements.

Syntax and parameters for HVTDLBxx member

The following figure displays the HVTDLGxx member syntax:

```
DLGPfK  PFSHOW={ENABLE|DISABLE}
DLGAUTH USERID=userid
        [,TA=( [ONLINE] [,OFFLINE] [,KILL] ) )
        [,TC=(SAVEC)]
        [,TL=( [ADDDPL] [,DELDPL] [,UPDDPL] [,ADDTPL] [,DELTPL] [,UPDTPL] ) )
        [,TV=( [SCRATCH] [,RECLAIM] [,VAULT] [,RECOVER]
              [,ADDTAP] [,DELTAP] [,UPDTAP] ) )
```

This list shows the syntax of the DLGPfK keyword and parameters that can be used in the HVTDLGxx member.

DLGPFK

The DLGPFK initialization statement defines the PF Keys definitions display options. The parameters for PFSHOW={ENABLE | DISABLE} are described below.

ENABLE

Specifies that VTE Dialog users can use the PFSHOW ISPF command to display or hide the PF Keys definitions.

DISABLE

Specifies that the PF Keys definitions will not be displayed on the VTE Dialog panels. In addition, the PFSHOW ISPF command is disabled. DISABLE is the default.

This list shows the syntax of the DLGAUTH keyword and the parameters that can be used in the HVTDLGxx member.

DLGAUTH

The DLGAUTH initialization statement defines the series of authorized requests a VTE Dialog user is authorized to perform.

USERID=userid

This parameter specifies the TSO userid of the VTE Dialog user whose authorization is defined in that initialization statement.

The *userid* can be a mask or a 1- to 8-character alphanumeric or national character string. The first character must be alphabetic or a national character.

TA=([ONLINE],[OFFLINE],[KILL])

Specifies the list of Units application's requests the user is authorized to perform.

- If the TA parameter is missing, none of the Units application's requests is authorized.
- If the TA parameter is provided, the list should contain at least one request.
- If more than one request is specified, they can be specified in any order.

ONLINE

The O (ONLINE) Line Command is authorized.

OFFLINE

The F (OFFLINE) Line Command is authorized.

KILL

The K (KILL) Line Command is authorized.

TC=(SAVEC)

Specifies the list of Configuration application's requests the user is authorized to perform.

- If the TC parameter is missing, none of the Configuration application's requests is authorized.
- If the TC parameter is provided, the list should contain at least one request.
- If more than one request is specified, they can be specified in any order.

SAVEC

The SAVE Primary Command is authorized.

TL=([ADDPL],[DELDPL],[UPDDPL] [,ADDTPL],[DELTPL],[UPDTPL])

Specifies the list of Pools application's requests the user is authorized to perform.

- If the TL parameter is missing, none of the Pools application's requests is authorized.
- If the TL parameter is provided, the list should contain at least one request.
- If more than one request is specified, they can be specified in any order.

ADDPL

The SAVE Primary Command for adding a new diskpool is authorized.

DELDPL

The D (DELETE) Line Command for deleting an existing diskpool is authorized.

UPDDPL

The SAVE Primary Command for updating an existing diskpool is authorized.

ADDDTL

The SAVE Primary Command for adding a new tapepool is authorized.

DELDTL

The D (DELETE) Line Command for deleting an existing tapepool is authorized.

UPDTPL

The SAVE Primary Command for updating an existing tapepool is authorized.

**TV=([SCRATCH],[RECLAIM],[VAULT],[RECOVER],[ADDTAP]
[,DELTAP],[UPDTAP])**

Specifies the list of Virtual Tapes application's requests the user is authorized to perform.

- If the TV parameter is missing, none of the Virtual Tapes application's requests is authorized.
- If the TV parameter is provided, the list should contain at least one request.
- If more than one request is specified, they can be specified in any order.

SCRATCH

The C (SYNC) Line Command for scratching an active virtual tape is authorized.

RECLAIM

The C (SYNC) Line Command for reclaiming a deferred-scratch virtual tape is authorized.

VAULT

The C (SYNC) Line Command for marking a virtual tape 'to be vaulted' or 'to be returned' is authorized.

RECOVER

The R (RECOVER) Line Command is authorized.

ADDTAP

The SAVE Primary Command for adding a new virtual tape is authorized.

DELTAP

The D (DELETE) Line Command for deleting an existing virtual tape is authorized.

UPDTAP

The SAVE Primary Command for updating an existing virtual tape is authorized.

Chapter 29. VTE Customization

This chapter describes the customization process of Virtual Tape Emulation (VTE) Version 2.1. The instructions should be followed step-by-step in order to ensure an easy and smooth customization process.

Introduction

Before processing any of the installation steps described in this chapter, make sure that all installation steps included in the *Program Directory* for this product are completed successfully. Only after successful completion of installation steps in the *Program Directory* can you continue with the customization steps in this chapter.

The customization steps included in this chapter are based on the steps in the *Program Directory*. It is assumed that the Virtual Tape Emulation (VTE) product (also referred to as "VTFM") is SMP/E APPLY-ed and ACCEPT-ed, and both the distribution and target libraries are allocated and contain the required data.

Customization steps

Several steps are involved in customizing this product for your environment. Be sure to work through the steps carefully to ensure the best result.

Step 1: Hardware configuration

Two major hardware types must be made available to VTE for proper functioning. These hardware types are real disks and virtual tape devices.

About this task

The real disks can be of any type, model and vendor. They are defined to MVS using a standard HCD process (without any special requirement) and are later managed by any standard available tool.

The customization process assumes the disks are already defined and available.

The virtual tape devices definition requires special attention.

Follow the steps below to define the new VTE virtual devices.

Note: If the VTE UIM has been installed in VTE release 2.1, skip ahead to step 2 (below), "Define the VTE virtual devices to MVS using HCD".

If you intend to use 3590/3592 virtual devices, continue with the following steps in order to install the new UIM.

Procedure

1. Install the VTE virtual devices UIM.

The VTE virtual devices simulate IBM compatible 3480, 3490, 3590, and 3592 devices. The installation of the VTE virtual devices is done the same way as if they were real hardware devices.

The VTE virtual devices are introduced to the hardware configuration with a User Unit Information Module (UIM). Some of the hardware configuration parameters are supplied as defaults with the VTE default UIM. If the default values conflict with your site's configuration you may change them through VTE EXIT002.

The default UIM number is 254 and it supplies 6 simulations – 2 for 3480 devices, 2 for 3490 devices and 2 for 3590 or 3592 devices, as follows:

Simulation	GENERIC	DEVUNIT	UCBTBYT1	PREFVAL
3480	VT3480GN	VT3480DU	39	9013
3480	VT3480G2	VT3480D2	40	9014
3490	VT3490GN	VT3490DU	41	9015
3490	VT3490G2	VT3490D2	42	9016
3590/3592	VT3590GN	VT3590DU	43	9017
3590/3592	VT3590G2	VT3590D2	44	9018

Note that 3592 devices are simulated as 3590 devices.

a) Define the VTE virtual devices characteristics.

Make sure that the UIM default definitions do not conflict with other UIMs or devices in your system.

If the supplied UIM defaults conflict with your installation, modify them using VTE EXIT002.

For information on EXIT002 and its installation, refer to "EXIT002" on page 154.

b) Copy the UIM modules to SYS1.NUCLEUS.

Edit member HVTJCUIM in the SAMPLIB library.

It contains a job that copies the UIM modules to SYS1.NUCLEUS.

Substitute the #-prefixed parameters as required, then submit the job.

The job should end with a return code of 0.

2. Define the VTE virtual devices to MVS using HCD.

Note: If the VTE virtual devices have been installed from a previous VTE version and you don't intend to use 3590/3592 virtual devices, you may continue to use them. In this case, skip to "[Step 2: Setup Steps](#)" on page 235. If you intend to use 3590/3592 virtual devices, continue with the following steps.

In order to illustrate the HCD process, the following values are assumed in this step:

Table 21. Installation of 32 3490 virtual devices.

Parameter	Value
Generic name	VT3490G2
Device unit	VT3490D2
Esoteric	VT3490ES
Number of devices	32
First device number	0570

a) Initialize the HCD environment.

1) Enter the Hardware Configuration Definition (HCD) panels.

2) It is recommended to save the current production I/O Definition File (option 6.2 from the main Hardware Configuration panel).

3) Select the work I/O Definition File in the *I/O Definition File* field (you may press PF4 on that field to get a list of the existing IODFs).

b) Define the VTE virtual devices esoteric name.

The esoteric name defined in this section will be used to address the VTE virtual devices.

- On the *Hardware Configuration* panel, select option 1 (*Define, modify, or view configuration data*):

```

Command ==> -----
                                Hardware Configuration

Select one of the following.

1  1. Define, modify, or view configuration data
    2. Activate or process configuration data
    3. Print or compare configuration data
    4. Create or view graphical configuration report
    5. Migrate configuration data
    6. Maintain I/O definition files
    7. Query supported hardware and installed UIMs
    8. Getting started with this dialog
    9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYSJENNA.IODF05.WORK'          +

```

- In the *Define, Modify, or View Configuration Data* panel, select option 1 (*Operating system configurations*):

```

                                Define, Modify, or View Configuration Data

Select type of objects to define, modify, or view data.

1_ 1. Operating system configurations
    consoles
    system-defined generics
    EDTs
    esoterics
    user-modified generics
2. Switches
    ports
    switch configurations
    port matrix
3. Processors
    channel subsystems
    partitions
    channel paths
4. Control units
5. I/O devices

```

- In the *Operating System Configuration List* panel type S by each Config ID where the VTE virtual devices will be used:

```

Goto  Backup  Query  Help
-----
                                Operating System Configuration List      Row 1 of 4
Command ==> -----Scroll ==> CSR

Select one or more operating system configurations, then press Enter. To add,
use F11.

/ Config. ID  Type  Description
_ JJ01       MVS   z/OS V1R5
s JJ02       MVS   z/OS V1R7
_ MM11       MVS   OS/390 V2R10
***** Bottom of data *****

```

- In the *EDT List Panel*, select the Eligible Device Table (EDT) where the VTE virtual devices will be defined by typing S on the requested EDT line:

```

                                EDT List
      Goto  Backup  Query  Help
-----
                                Row 1 of 1
Command ==> _____ Scroll ==> CSR
Select one or more EDTs, then press Enter. To add, use F11.
Configuration ID . : JJ02          z/OS V1R7
/ EDT Last Update By      Description
_ J0  2007-07-09  M24      _____
***** Bottom of data *****

```

- In the *Esoteric List* panel, press PF11 to add new esoteric name.
- In the *Add Esoteric* panel, fill the esoteric name for the VTE virtual devices:

```

                                Add Esoteric

Specify the following values.

Esoteric name . . . VT3490ES
VIO eligible . . . . No      (Yes or No)
Token . . . . . ____

```

- *No device defined* will be prompted in the *State* field of this esoteric name. This esoteric name will be used later to group the VTE virtual devices.

```

                                Esoteric List
      Goto  Filter  Backup  Query  Help
-----
                                Row 12 of 46
Command ==> _____ Scroll ==> CSR
Select one or more esoterics, then press Enter. To add, use F11.
Configuration ID . : JJ02          z/OS V1R7
EDT identifier . . : J0

/ Esoteric  VIO    Token  State
- VTSWIZ    No     ____
- VT3490ES No     ____      No device defined
- VT3590    No     ____
- VT3590EC  No     ____
- VT3590TS  No     ____
- VT80DT    No     ____
- VT80EC    No     ____
- VT80QA1   No     ____
- VT90CS    No     ____
- VT90DT    No     ____
- VT90EC    No     ____

```

- Exit the *Esoteric List* panel to return to the *Define, modify, or view configuration data* panel.
- c) Add and connect the VTE virtual devices to the system.
- In the *Define, Modify, or View Configuration Data* panel select option 5 (I/O devices):

```

Define, Modify, or View Configuration Data

Select type of objects to define, modify, or view data.

5_ 1. Operating system configurations
    consoles
    system-defined generics
    EDTs
    esoterics
    user-modified generics
2. Switches
    ports
    switch configurations
    port matrix
3. Processors
    channel subsystems
    partitions
    channel paths
4. Control units
5. I/O devices

```

- In the *I/O Device List* panel press PF11 to add new devices. In the Device Number field, fill in the first device number for the VTE virtual devices.

In the *Number Of Devices* field, fill in the number of devices that you want to add to the system.

In the *Device Type* field, fill in the device unit type.

You may press PF4 on this field to receive a list of the device types known to the system, and then select the requested VTE device type from the list:

```

Add Device

Specify or revise the following values.

Device number . . . . . 0570 (0000 - FFFF)
Number of devices . . . . . 32
Device type . . . . . VT3490D2_____ +

Serial number . . . . . _____
Description . . . . . Virtual Devices_____

Volume serial number . . . . . _____ (for DASD)

Connected to CUs . . _____ +

```

- In the *Define devices to Operating System Configuration* panel select the systems you want the VTE virtual devices to be connected to by typing S on the requested *Config ID* line:

```

Define Device to Operating System Configuration
Command ==> _____ Row 1 of 4
Scroll ==> CSR

Select OSs to connect or disconnect devices, then press Enter.

Device number . : 0570          Number of devices : 32
Device type . . : VT3490D2

/ Config. ID  Type  SS Description Defined
- JJ01       MVS    z/OS V1R5
s JJ02       MVS    z/OS V1R7
- JJ03       MVS    z/OS V1R7
- MM11       MVS    OS/390 V2R10
***** Bottom of data *****

```

- This panel is optional. Type YES in the LOCANY Parameter/Feature if you want the virtual devices UCBs to reside above the line.

```

Define Device Parameters / Features
Row 1 of 6
Command ==> _____ Scroll ==> CSR

Specify or revise the values below.

Configuration ID . : JJ02          z/OS V1R7
Device number . . : 0570          Number of devices : 32
Device type . . . : VT3490D2

Parameter/
Feature Value +      R Description
OFFLINE No           DEVICE CONSIDERED ONLINE OR OFFLINE AT IPL
DYNAMIC Yes          DEVICE SUPPORTS DYNAMIC CONFIGURATION
LOCANY Yes            UCB CAN RESIDE IN 31 BIT STORAGE
ALTCTRL No           SEPARATE PHYSICAL CONTROL UNIT PATH
DATACONV No          DATA CONVERSION
DUALDENS No          DUAL DENSITY
***** Bottom of data *****

```

- In the *Assign/Unassign Device to Esoteric* panel, assign the esoteric name you defined previously for the VTE virtual devices by typing YES on the *Assigned* field of the requested esoteric line.

Press Enter to end the selection:

```

Assign/Unassign Device to Esoteric
Row 12 of 46
Command ==> _____ Scroll ==> CSR

Specify Yes to assign or No to unassign. To view devices already
assigned to esoteric, select and press Enter.

Configuration ID : JJ02          z/OS V1R7
Device number . : 0570          Number of devices : 32
Device type . . : VT3490D2      Generic . . . . . : VT3490G2

/ EDT.Esoteric Assigned Starting Number Number of Devices
- J0.VTSWIZ No -----
- J0.VT3490ES YES -----
- J0.VT3590 No -----
- J0.VT3590EC No -----
- J0.VT3590TS No -----
- J0.VT80DT No -----
- J0.VT80EC No -----
- J0.VT80QA1 No -----
- J0.VT90CS No -----
- J0.VT90DT No -----
- J0.VT90EC No -----

```

- The optional I/O device list panel is displayed. It shows the new VTE devices.

Press Enter to return to *Define Device to Operating System Configuration* panel.

```

Goto Filter Backup Query Help
-----
I/O Device List Row 1 of 102 More: >
Command ==> _____ Scroll ==> CSR

Select one or more devices, then press Enter. To add, use F11.

-----Device----- --#--- -----Control Unit Numbers + -----
/ Number Type +      CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0100,16 3380      1  4 0100 -----
- 0110,8 3380      1  4 0100 -----
- 0118,5 3380      1  4 0100 -----
- 011D,227 3380     1  4 0100 -----
- 0200,256 3390     1  4 0200 -----
- 0300,256 3390     1  4 0300 -----
- 0500,2 3490      1  4 0500 -----
- 0510,4 3590      1  1 0510 -----
- 0520,8 3590      1  4 0520 -----
- 0570,32 VT3490D2 1  -----
- 0700,36 3390      1  4 0700 -----
- 0724,42 3390      1  4 0700 -----
- 074E,10 3390      1  4 0700 -----

```

- Exit to the main *Hardware Configuration* panel.

d) Create a production IODF

- 1) In the *Hardware Configuration* panel, select option 2 (*Activate or Process Configuration Data*).
- 2) In the *Activate or Process Configuration Data* panel, select option 1 (*Build Production I/O Definition File*).
- 3) Specify the name of the new IODF and the volume serial on which your IODF files reside.
- 4) In the *Continue Using as Current IODF* field, select option 2 (*The new production IODF specified above*).

e) Build the IOCDS.

- 1) In the *Activate or Process Configuration Data* panel, select option 2 (*Build IOCDS*).
- 2) Select the processor where the devices are installed.
- 3) In the *IOCDS List* panel, type U by the IOCDS you wish to update.
- 4) In the *Build IOCDS* panel, press Enter to submit a job that updates the IOCDS.
- 5) Exit HCD.
- 6) Check the console for any replies to confirm the IOCDS change and make sure the IOCDS creation ended with a return code of 0.

f) Update your system with the new IODF and IOCDS values so these values will be used during the next IPL and Power-On Reset.

g) Activate the new I/O configuration.

Issue the following operator command from the system console where "xx" is the suffix of the newly created IODF.

```
ACTIVATE IODF=xx,SOFT=NOVALIDATE
```

The following messages are expected after successful completion of the new I/O configuration activation.

```
ACTIVATE IODF=05,SOFT=NOVALIDATE
IOS502I I/O CONFIGURATION CHANGED 412
INVOKER = *MASTER*
NEW IODF = SYSJENNA.IODF05
EDT REBUILT, NEW EDT ID = J0
NOTE = 0100,SOFTWARE-ONLY CHANGE
COMPID=SC1C3
IOS501I ACTIVATE CLEANUP COMPLETE
DEVICE(S) ADDED
0570-058F.

IOS500I ACTIVATE RESULTS 421
ACTIVATE COMPLETED SUCCESSFULLY
NOTE = 0100,SOFTWARE-ONLY CHANGE
COMPID=SC1C3
```

Step 2: Setup Steps

This section describes the setup steps you need to complete.

Procedure

1. Copy the VTE procedure to the system PROCLIB library.

Edit member HVTPROC in the SAMPLIB library. It contains the VTE procedure. Substitute the #-prefixed parameters as required then save the member.

Edit member HVTJCPRC in the SAMPLIB library. Substitute the #-prefixed parameters as required then submit the job.

The job should end with a return code of 0.

2. Define a user catalog to contain the virtual tape data sets.

Note: If you are an existing customer of a previous VTE version, see "Customize the VTE server basic options" (below).

Edit member HVTJUCAT in the SAMPLIB library. Substitute the #-prefixed parameters as required, then submit the job.

The job should end with a return code of 0.

3. Customize the VTE server basic options.

The VTE basic options are located in the HVTOPTxx member in the SAMPLIB library. Substitute the #-prefixed parameters as required then save the member.

For a description of the basic options, refer to ["HVTOPTxx – Basic Options" on page 256](#).

4. Customize the VTE tapepool and diskpool initialization statements.

The VTE tapepools and diskpools initialization statements are located in the HVTPOLxx member in the SAMPLIB library. Substitute the #-prefixed parameters as required then save the member.

For a description of the tapepool and diskpool initialization statements, refer to ["HVTPOLxx – Tapepools and Diskpools" on page 268](#).

5. APF authorize the VTE load library.

Follow the steps below to APF authorize the VTE load library.

a) Dynamically add the VTE load library to the APF list.

Issue the following operator command from the system console in order to APF authorize the VTE library:

```
SETPROG APF,ADD,DSNAME=steplib,VOLUME=prodvol
```

Where *steplib* is the name of the VTE load library and *prodvol* is the disk volume serial number where the VTE load library resides.

b) Add the VTE load library to the APF list

Enter the appropriate PROGxx member in the SYS1.PARMLIB library and add the VTE load library to the APF list.

6. Allocate and format the VTE database files.

Note: If you are an existing customer of a previous VTE version, see "Allocate the history files" (below).

a) Allocate and format the journal file.

Edit member HVTUJFR in the SAMPLIB library. It contains a job that deletes the existing VTE journal file (if any), defines a new journal file and initializes it. Substitute the #-prefixed parameters as required then save the member.

Calculate the journal file size and customize the utility initialization statements according to the explanation in ["HVTUJFR – Journal File Format" on page 322](#).

Optionally, uncomment the SMS parameters and submit the job.

The job should end with a return code of 0.

b) Allocate and format the VDB.

Edit member HVTUVSI in the SAMPLIB library. Substitute the #-prefixed parameters as required then save the member. Note that the job's first step is a dummy step that never executes.

Calculate the VDB size according to the explanation in ["HVTUVSI – VDB Initialization" on page 347](#).

Optionally, uncomment the SMS parameters and submit the job.

The job should end with a return code of 0.

7. Allocate the history files.

If you do not intend to use the HVTUDST utility, skip to "Optional apply of EXIT001" (below).

a) Allocate the history and the history backup files.

Edit member HVTUHST in the SAMPLIB library. It contains a job that allocates the history file and the history backup file. The history backup file name is similar to the history file suffixed by '.BKP'.

Substitute the #-prefixed parameters as required, then submit the job.

The job should end with a return code of 0.

b) Enable HVTUDST utility to write the migration job.

Edit member HVTUDST in the SAMPLIB library. It contains a sample job that executes the HVTUDST utility.

HVTMIG DD statement in that sample job defines a DUMMY dataset.

In order for the migration job to be created on that DD statement, define it as a non-DUMMY dataset (internal reader, production library or other).

Note that the defined DCB parameters should not be altered or removed.

c) Enable HVTUREP utility to list the history file contents.

Edit member HVTUREP in the SAMPLIB library. It contains a sample job that executes the HVTUREP utility. Uncomment the JCL statements that define the HVTHST DD statement and save the member.

8. Optional apply of EXIT001

Note: If you do not need to use EXIT001, skip to "Optional apply of EXIT004" (below).

A dummy EXIT001 is supplied with VTE. If this customization step is skipped, the dummy exit will be invoked.

For information on EXIT001 and its customization, refer to [“EXIT001” on page 361](#).

9. Optional apply of EXIT004

Note: If you do not need to use EXIT004, skip to [“Security Requirements and Definitions”](#).

A dummy EXIT004 is supplied with VTE. If this customization step is skipped, the dummy exit will be invoked.

For information on EXIT004 and its customization, refer to [“EXIT004” on page 365](#).

Step 3: Security Requirements and Definitions

This step describes the security requirements of VTE.

Procedure

1. General security requirements

The following security authorizations are required (regardless of the employed security product):

- ALTER authorization should be given to the VTE server (and all its associated offline utilities) for:
 - The virtual tape data sets.
 - The VDB and the journal file.
 - The history file.
- The offline utilities should be submitted by users with sufficient authorization as explained above.
- Individuals that may need to browse the virtual tape data sets, the VDB or the journal file (for problem determination purposes) should be given READ authorization for:
 - The virtual tape data sets.
 - The VDB and the journal file.
 - The history file.

2. Security definitions in a RACF environment

Note: If you do not employ RACF, skip to “[Step 4: System Settings](#)” on page 238.

This step provides detailed explanations and sample definitions for users that employ RACF and wish to allow RACF to manage the security aspects of VTE. You may choose to apply the sample 'as is', modify it to fit your special needs or just ignore it and make your own definitions that fit your site policy.

This step should be performed by a privileged user.

Edit member HVTJRACF in the SAMPLIB library. It contains a job that:

- Defines three profiles to protect the virtual tape data sets, the VDB and the journal file.
- Defines a profile for the VTE started task.
- Grants VTE an ALTER authorization to the virtual tape data sets, the VDB and the journal file.
- Grants VTE authorization to issue MODIFY operator commands, authorization to issue VARY commands to its devices and authorization to issue MONITOR commands.
- You can give VTE full authorizations to the protected files by defining VTE started task as a TRUSTED or PRIVILEGED started task. To do so, uncomment the following command:

```
'RALT STARTED (hvtproc.hvtproc) STDATA(TRUSTED(YES))'
```

- For JES3 users, uncomment the following command:

```
'PE JES3.VARY.DEV CLASS(OPERCMDS) ID(vdsnq1) ACCESS(CONTROL)'
```

Substitute the #-prefixed parameters as required, verify that the definitions fits your site's policy, then submit the job.

The maximum permitted return code of the job is 4.

Note: The user that runs the VTE offline utilities should have ALTER authorization to the VDSNQ1 profile.

Step 4: System Settings

This section describes the steps for assigning your system settings.

Procedure

1. Define the VTE subsystem.

This step is optional. If the subsystem is not pre-defined, VTE will dynamically define it during its initialization.

Define the subsystem name in the appropriate IEFSSNxx member in SYS1.PARMLIB. The subsystem name should be identical to the SUBSYS parameter specified in the HVTOPTxx member. The definition will be effective after the next IPL.

In the meantime, VTE will define it dynamically during its initialization.

2. Global Resource Serialization considerations

The disks on which the VDB and the journal file reside are subject to frequent RESERVEs. This may block other systems that share the same disks from accessing them.

The resource names are as follows:

- QNAME - V_TAPE
- RNAME - The name of the VDB or journal file, respectively, padded with blanks. Its total length is 44 characters.

In addition, you may expect contentions (or even deadlocks) on the SYSIGGV2 QNAME.

If a Global Resource Serialization product is employed, it is necessary to convert these RESERVEs.

For instance:

- GRS customers should place the resources in the RESERVE Conversion RNL, not in the SYSTEMS Exclusion RNL. For example:

```
RNLDEF RNL(CON) TYPE(GENERIC)
        QNAME('V_TAPE')
RNLDEF RNL(CON) TYPE(GENERIC)
        QNAME(SYSIGGV2)
```

- MIM customers should define the following QNAME statements:

```
QNAME=SYSIGGV2      GDIF=YES,SCOPE=RESERVES,
                    EXEMPT=NO,
                    TRACE=NONE,ECMF=NO,
                    RPTAFTER=000,RPTCYCLE=060
QNAME=V_TAPE        GDIF=YES,SCOPE=SYSTEMS,EXEMPT=NO,
                    TRACE=NONE,ECMF=NO,
                    RPTAFTER=000,RPTCYCLE=060
```

3. Allow VTE user SMF record writing

This step is optional and should be carried out only if the VTE user SMF record is to be written (that is, OPTIONS SMF=ENABLE parameter is specified in the HVTOPTxx member and the VTE user SMF record number is not 0).

a) IEFU83 and IEFU84 SMF exits

If IEFU83 or IEFU84 SMF exits are active, make sure they do not suppress the write of the VTE user SMF record.

b) SMF record number

The VTE user SMF record number should be included in the appropriate SYS or SUBSYS statement in the SMFPRMxx member.

When finished, the SMFPRMxx member should be refreshed using the following operator command:

```
T SMF=xx
```

where xx is the SMFPRMxx member suffix.

4. Performance considerations

Users that run in WLM *compatibility* mode should assign the VTE started task a PGN number of a performance group assigned to batch jobs that use tapes.

Users that run in WLM *goal* mode should assign the VTE started task a Service Class assigned to batch jobs that use tapes.

This way other applications will compete fairly with the VTE started task over the available system resources.

5. SMS ACS routines considerations

The following paragraphs detail what SMS ACS routines modifications should be made to allow VTE to function properly in SMS-managed and non-SMS-managed environments.

a) SMS considerations for the VTE back-end components

The information below describes the actions that are relevant to the VTE back-end components, which includes the virtual tape data sets on disk management.

With one exception (when at least one diskpool is SMS-managed), no SMS ACS modifications are required to allow VTE to function properly in both SMS-managed and non-SMS-managed environments. VTE allocates the virtual tape data sets on SMS-managed or non-SMS-managed disks, according to the definitions in the appropriate DPOOL initialization statements, respectively, and **no** ACS routine(s) statements should be coded in order to modify its decisions.

The following information is to be considered **only** if your existing ACS routines interfere with the VTE virtual tape data sets.

- Ensure that no data class is assigned to virtual tape data sets.

Assuming the virtual tape data sets high-level qualifier is CC, this is done by coding the following statements at the very beginning of your DATACLAS ACS routine.

```
PROC DATACLAS
IF &HLQ='CC' THEN
DO
SET &DATACLAS=' '
EXIT
END
```

- Ensure that the management class set by VTE is not modified.

Assuming the virtual tape data sets high-level qualifier is CC, this is done by coding the following statements at the very beginning of your MGMTCLAS ACS routine.

```
PROC MGMTCLAS
IF &HLQ='CC' THEN
DO
EXIT
END
```

- Ensure that the storage class set by VTE is not modified.

Assuming the virtual tape data sets high-level qualifier is CC, this is done by coding the following statements at the very beginning of your STORCLAS ACS routine.

```
PROC STORCLAS
IF &HLQ='CC' THEN
DO
EXIT
END
```

- If at least one diskpool is SMS-managed, you should relate the Storage Group that contains the SMS-managed disks dedicated to VTE to the storage class assigned to the virtual tape dataset.

Assuming the storage class dedicated to the VTE virtual tape data sets is CCSTOR and the storage group is CCSTG, this is done by coding the following statements in your STORGRP ACS routine.

```
PROC STORGRP
*
*
IF &STORCLAS='CCSTOR' THEN
DO
SET &STORGRP = 'CCSTG'
END
```

- If at least one diskpool is SMS-managed, you should make sure that your SMS Base Configuration specifications specify the proper default device geometry.
 - If the default unit is 3390, the bytes/track value should be 56664 and the tracks/cylinder value should be 15.
 - If the default unit is 3380, the bytes/track value should be 47476 and the tracks/cylinder value should be 15.

You should pay attention to this because SMS converts all space requests from tracks to MB and this conversion is done according to the default geometry definition. If wrong default values are specified, the allocation values are adjusted by SMS and wrong extents sizes will be allocated resulting in possible Sx37 abends.

b) SMS considerations for the VTE front-end components

This paragraph details the actions relevant to the VTE front-end components. In this context, these components include the VTE virtual devices allocation.

- If your GDG real tape allocations lack the pattern DSCB specification (counting on the fact that a data class value presence compensates for that), and you use the VTE subsystem to dynamically reallocate real device allocations to VTE virtual devices, then code the following statements in your DATACLAS ACS routine. It is assumed that DCNULL data class is a null data class.

```
PROC DATACLAS
.
.
SELECT
  WHEN ((&DSTYPE EQ 'GDS' ) AND
        (&ACSENVIR EQ 'ALLOC') AND
        (&DATACLAS EQ ' ' ))
    SET &DATACLAS EQ 'DCNULL '
END
```

6. Non-SMS diskpool considerations

This step is optional and should be performed only if at least one diskpool is non-SMSmanaged.

The disks that comprise the non-SMS diskpool should have a mount attribute of STORAGE. That means that if any other application allocates new data sets using UNIT=SYSALLDA or UNIT=3390, these data sets may be allocated on the disks that belong to the VTE diskpool.

MVS does not provide a standard method to avoid this. If your site has applications that allocate disk data sets using UNIT=SYSALLDA or UNIT=3390, the following methods can be implemented to overcome the problem:

- Consider allowing VTE to use SMS-managed disks.
- Restrict the use of SYSALLDA or 3390 as unit names.
- Use a third-party disk management product or DADSM exit to restrict the VTE disks to VTE use only.

7. RACF considerations

a) "ERASE-ON-SCRATCH" is active

Check if the "ERASE-ON-SCRATCH" option is active in your RACF configuration by issuing the following command:

```
SETR LIST
```

If it is active, make sure it is not applicable to the profile protecting the virtual tape data sets on disk. Otherwise, each virtual tape dataset associated with a virtual tape being scratched by the HVTUSCR utility is overwritten by data prior to its deletion. This, in turn, causes high and undesirable overhead to the HVTUSCR utility.

b) "ERASE-ON-SCRATCH" is not active

If 'ERASE-ON-SCRATCH IS INACTIVE' is displayed, no additional actions are required.

8. MONITOR JOB NAMES considerations

On its startup, the VTE server issues one of the following MVS operator commands (depending on the MVS release):

```
MN JOB NAMES,T
```

or

```
SETCON MN, JOB NAMES=(ON,LOG), T=ON
```

The above operator command is required by the dynamic reallocation feature and it causes certain MVS messages to be issued whenever a job or started task starts or ends (IEF403I/IEF404I) or when a TSO user logs on or off (IEF125I/IEF126I).

If the above MVS operator command or its associated MVS messages are suppressed, no dynamic reallocation will occur. If your site is somehow affected by these messages, you should consider modifying your operation product accordingly.

Step 5: JES3 Additional Customization Steps

This step describes the additional steps required for the VTE customization in a JES3 complex.

About this task

The steps should be performed in all local processors that will run VTE, as well as in the global processor (whether or not it runs VTE), unless otherwise specified.

Note: If you are not a JES3 user, skip to [“Step 6: e-Vault Installation”](#) on page 246.

Note: If the JES3 support has been installed from a previous VTE version and you use the same hardware configuration values, skip to "Customize JES3 Exit71 (IATUX71)" in this section.

Procedure

1. Revise the VTE virtual devices definitions to MVS

The VTE virtual devices should be defined to MVS as described in [“Step 1: Hardware configuration”](#) on page 229. Each processor (local(s) and global) running VTE should define them.

If the global processor does not run VTE, a set of VTE virtual devices should be defined to it as well.

Each processor may define up to 256 VTE virtual devices, but the total number of VTE virtual devices in the complex may exceed that number. In any case, it is mandatory that the esoteric and generic names of the VTE virtual devices in all processors are identical. VTE does not check this on its initialization, however if this requirement is not observed, invalid dynamic reallocation results will occur.

In the following example, a total number of 32 VTE virtual devices are defined for the whole complex and they are shared between two processors:

Processor	Esoteric	Generic	Device Numbers
MM81	VT3490ES	VT3490G2	0570-058F
MM90	VT3490ES	VT3490G2	0570-058F

In the following example, each processors uses its own 32 VTE virtual devices:

Processor	Esoteric	Generic	Device Numbers
MM81	VT3490ES	VT3490G2	0570-058F
MM90	VT3490ES	VT3490G2	0670-068F

2. Define the VTE virtual devices to JES3

Note: If the VTE virtual devices have been defined from a previous VTE version, you may continue to use them. In this case, skip to the next step, "Enable IEFUJV exit", in this section.

You must decide whether or not the VTE virtual devices will be JES3-managed. Basically, if the real devices are JES3-managed, it is recommended to define the VTE virtual devices as JES3-managed as well. This is so since if some dynamic reallocation occurs and the original real device was referenced by `//*MAIN SETUP` statements, the job may fail if the new device is not JES3-managed as well.

Note: If you decide that your VTE virtual devices are not JES3-managed, skip to "Customize JES3 Exit11 (IATUX11)" below.

All VTE virtual devices defined under the VTE server(s) esoteric name should be defined in the JES3 initialization member. If only part of them are defined, unpredictable results are experienced.

a) Define the SETNAME statement

The SETNAME statement defines the user-assigned names along with the device type names associated with the JES3-managed VTE virtual devices. In our definitions, the NAMES parameter specifies the esoteric and the generic names of the VTE virtual devices.

For example:

```
SETNAME, XTYPE=VT3490ES,  
        NAMES=(VT3490ES, VT3490G2)
```

Note: All the processors that belong to the JES3 complex must use the same esoteric and generic names for the VTE virtual devices.

b) Define the DEVICE statement

The DEVICE statement defines devices that JES3 can use. According to the definitions made to MVS, the DEVICE statements should reflect the devices availability to JES3.

In the following example, a total number of 32 VTE virtual devices are defined for the whole complex and they are shared between two processors:

```
DEVICE, XTYPE=VT3490ES,  
        XUNIT=(0570, MM81, , ON, 0570, MM90, , ON),  
        NUMDEV=32
```

In the following example, each processor uses its own 32 VTE virtual devices:

```
DEVICE, XTYPE=VT3490ES,  
        XUNIT=(0570, MM81, , ON),  
        NUMDEV=32  
DEVICE, XTYPE=VT3490ES,  
        XUNIT=(0670, MM90, , ON),  
        NUMDEV=32
```

Note:

- For JES3 versions lower than 2.9 the NUMDEV parameter is not supported. In this case, the DEVICE statement should be replicated for all devices in the defined range.
- For JES3 versions lower than 2.9, warm start with refresh is required in order to add the DEVICE statements.
- For JES3 version 2.9 and up, hot start with refresh is required in order to add the DEVICE statements.

3. Enable IEFUJV exit

Note: If IEFUJV exit is already defined in your system, skip to *step 5.5 - "Customize JES3 Exit11 (IATUX11)"* on page 19.

Code the IEFUJV exit as an exit that is to receive control during SMF processing.

- Edit member SMFPRMxx in SYS1.PARMLIB.
- Code the IEFUJV exit in the EXITS sub-parameter of the SYS statement.

Example:

```
SYS(TYPE(14, 15, 21, 30(4, 5), 65, 66, 74, 240, 241),  
    NOINTERVAL, NODETAIL,  
    EXITS(IEFACTRT, IEFUTL, IEFU29, IEFUJV))
```

If the SUBSYS(STC...) statement is defined and it contains the EXITS parameter, the EXITS parameter must contain the IEFUJV exit as one of its sub-parameters.

Example:

```
SUBSYS(STC,  
    EXITS(IEFU29, IEFU84, IEFUJP, IEFUJV, IEFUS0, IEFACRT))
```

If the SUBSYS(JES3...) statement is defined and it contains the EXITS parameter, the EXITS parameter must contain the IEFUJV exit as one of its sub-parameters.

Example:

```
SUBSYS(JES3,  
      EXITS(IEFU29,IEFU84,IEFUJP,IEFUJV,IEFUS0,IEFACTRT))
```

- Save the member. Activate it by entering the following operator command:

```
T SMF=xx
```

Where xx is the two-character SMFPRMxx member name suffix.

4. Allow SMS-managed tape drives dynamic reallocation

If your existing real tape devices are not SMS-managed, or you don't want to reallocate their allocations, skip to "Customize JES3 Exit11 (IATUX11)" below.

The Storage Class ACS routine should not assign any Storage Class value to the virtual device allocations.

Assuming the virtual devices esoteric name is VT3490ES, the following statements should be added at the very beginning of your Storage Class ACS routine.

```
PROC STORCLAS  
IF &UNIT='VT3490ES' THEN  
  DO  
    SET &STORCLAS=' '  
  EXIT  
END
```

5. Customize JES3 Exit11 (IATUX11)

Note:

- If you decide that your VTE virtual devices are not JES3-managed, skip to [“Step 6: e-Vault Installation”](#) on page 246.
- If JES3 Exit 11 has been installed from a previous VTE version, you may continue to use it. In this case, skip to "Customize JES3 Exit71 (IATUX71)" further down in this section.

VTE provides a JES3 Exit11 (IATUX11) that should be installed in all the processors comprising the JES3 complex. If JES3 STEPLIB is shared among the processors, install it only once.

The exit customization process consists of two steps:

- IATUX11 customization
- IATUX11 installation

a) IATUX11 customization

If the defaults values of the VTE virtual devices generic name are used, skip to "IATUX11 installation" (below).

Edit member HVTJ3X11 in the VTE SAMPLIB library. This member contains the modification done in behalf of VTE to the original IATUX11 exit supplied by IBM with FMID HJS7720 (JES3 z/OS version 1.7).

The modifications are in IEBUPDTE format and the member's sequence numbers in columns 73-80 correspond to that JES3 version. If you use a different JES3 version, make sure to adjust the sequence numbers accordingly.

Label UTAB in the VTE supplied HVTJ3X11 member contains the list of generic names of the VTE virtual devices. This list contains the names originally supplied with HVTX002 exit. You must add there the generic names you added to HVTX002. When adding, pay attention to the sequence numbers in columns 73-80. Note that there is no need to code the VTE virtual devices esoteric names in the UTAB table.

Save the member and exit it.

If you already employ IATUX11, you need to merge your existing exit with the VTE-supplied exit.

b) IATUX11 installation

IATUX11 is installed using USERMOD HVTU003.

- Edit member HVTU003 in the VTE SAMPLIB library. Change the FMID value, the CSI and the target zone names to fit your JES3 environment, then submit the job. The job should end with a return code of 0.
- Make the exit active by either of the following methods:
 - Refresh the exit to JES3 by issuing the following command:

```
*F X,MODULE=IATUX11,REFRESH
```

This command is accepted only if APAR OW40634 has been applied, and only if IATUX11 does not run in a C/I FSS address space (that is, JES3 itself interprets the JCL streams).

- If the exit cannot be refreshed, a JES3 hot start is required.

6. Customize JES3 Exit71 (IATUX71)

Note:

- If you decide that your VTE virtual devices are not JES3-managed, skip to [“Step 6: e-Vault Installation” on page 246](#).
- If JES3 Exit 71 has been installed from a previous VTE version, you may continue to use it. In this case, skip to [“Step 6: e-Vault Installation” on page 246](#).
- If you do not use OAM to write objects onto virtual tape media, skip to [“Step 6: e-Vault Installation” on page 246](#).

VTE provides a JES3 Exit71 (IATUX71) that should be installed in all the processors comprising the JES3 complex. If JES3 STEPLIB is shared among the processors, install it only once.

The exit customization process consists of two steps:

- IATUX71 customization
- IATUX71 installation

a) IATUX71 customization

Edit member HVTJ3X71 in the VTE SAMPLIB library. This member contains the modification done on behalf of VTE to the original IATUX71 exit supplied by IBM with FMID HJS7720 (JES3 z/OS version 1.7).

The modifications are in IEBUPDTE format and the member's sequence numbers in columns 73-80 correspond to that JES3 version. If you use a different JES3 version, make sure to adjust the sequence numbers accordingly.

Save the member and exit it.

If you already employ IATUX71, you need to merge your existing exit with the VTE-supplied exit.

b) IATUX71 installation

IATUX71 is installed using USERMOD HVTU005.

- Edit member HVTU005 in the VTE SAMPLIB library.
 - Change the FMID value, the CSI and the target zone names to fit your JES3 environment, then submit the job. The job should end with a return code of 0.
- Make the exit active by either of the following methods:
 - Refresh the exit to JES3 by issuing the following command:

```
*F X,MODULE=IATUX71,REFRESH
```

This command is accepted only if APAR OW40634 has been applied.

- If the exit cannot be refreshed, a JES3 hot start is required.

Step 6: e-Vault Installation

This step describes the prerequisites and installation steps required for implementing e-Vault.

About this task

Note: If you do not intend to use the e-Vault feature, skip to [“Step 7: Third-Party Product Prerequisites and Coexistence”](#) on page 246.

If you decide to use the e-Vault feature at a later time, you can proceed with this installation step even after VTE has already been fully implemented.

Procedure

1. FTP Type3 e-Vault Installation

This step describes the prerequisites and installation steps required for implementing type3 e-Vault.

Note: If you do not intend to use type3 e-Vault, skip to "FTP Type4 e-Vault Installation" (below).

Edit the VTE procedure and increase the SYSUT1 DD statement allocation size so that it will be able to contain one full virtual tape data set.

It is recommended to increase the SPACE parameter and to add a UNIT count parameter:

```
//SYSUT1 DD UNIT=(SYSALLDA,16),  
//          SPACE=(TRK,(6900,6900)),  
//          DISP=(,DELETE)
```

2. FTP Type4 e-Vault Installation

This step describes the prerequisites and installation steps required for implementing type4 e-Vault.

Note: If you do not intend to use type4 e-Vault, see "FTP Type5 e-Vault" below.

When virtual tape data sets are allocated or extended by the VTE server, their last allocated extent is not RLSE-d. On the other side, FTP defaults to release the last extent of its allocated output data sets.

In order to ensure a consistent allocation logic, (that is, not to let FTP release the last allocated extent), IBM's APAR PQ42563 should be applied. That APAR is available for z/OS systems only. On lower MVS releases, the virtual tape last allocated extent will be RLSE-d.

3. FTP Type5 e-Vault Installation

This step describes the prerequisites and installation steps required for implementing type5 e-Vault.

Note: If you do not intend to use type5 e-Vault, refer to [“Step 7: Third-Party Product Prerequisites and Coexistence”](#) on page 246.

It is necessary to define a valid OMVS segment for the userid assigned to the VTE started task.

Edit the VTE procedure and increase the SYSUT1 DD statement allocation size so that it will be able to contain one full virtual tape data set.

It is recommended to increase the SPACE parameter and to add a UNIT count parameter:

```
//SYSUT1 DD UNIT=(SYSALLDA,16),  
//          SPACE=(TRK,(6900,6900)),  
//          DISP=(,DELETE)
```

Step 7: Third-Party Product Prerequisites and Coexistence

The following step describes the considerations that should be taken when third-party products are employed.

Procedure

1. CA-7 from CA

CA-7 customers should revise the ICMDSECT options set in CA-7 ICMDSECT module. VTE depends on the original settings of the job's Reader-Date as initially set by the system. If for some reasons ICMDSECT options let CA-7 make use of the high order byte of the Reader-Date, (offset x'07' in ICMDSECT module equal x'40') you should contact IBM Customer Support before VTE activation.

2. JOB/SCAN from DSSI

The JOB/SCAN Control Tables specify details on how JOB/SCAN processes jobs at your data center. The 'JOB/SCAN How to Install Guide' from DSSI gives information on how the Control Tables can be modified.

Control Table AA associates unit names with device codes that indicate the type of device. This table can be altered to override device codes and define unit values as appropriate.

In order to let JOB/SCAN recognize the VTE virtual devices, locate the Control Table group-level statement in JSOPT02 in the PARMLIB library.

For VTE, the "AA" Control Table should be updated as follows:

- Column 6 and 7 contain AA.
- Column 8 indicates the VTE virtual devices generic name.
- Column 21 indicates the device code.

'T' is for tape. You should code a 'T' for VTE virtual devices.

```
COLS> -----1-----2-----3-----4-----5--
00089      AAVENDOR          K
00090      AAVT3480GN        T   the new virtual device
00091      AATPCART          Z
00092      AAVault           Z
```

3. PROSMS and STOPx37 from BMC

PROSMS and STOPx37 should not handle VTE virtual tape data sets. Under normal circumstances, VTE never experiences Sx37 abends. In order to be ignored by the above products, VTE procedure contains already the following DD statement:

```
//PROIGN DD DUMMY
```

4. ACC and SRS from DTS Software

ACC and SRS should not handle VTE virtual tape data sets. Under normal circumstances, VTE never experiences Sx37 abends. In addition, it manages its diskpools efficiently and maximizes the extents usage.

In order to be ignored by the above products, VTE procedure contains already the following DD statement:

```
//ACCIGN DD DUMMY
```

5. REXX library from IBM

If you intend to use the Lempel-Ziv compression algorithm in order to compress virtual tape data sets, the IBM library for SAA REXX/370 Release 3 (FMID HWJ9130) or the IBM library for z/Series Release 4 (FMID HWJ9140) must be purchased from IBM and installed.

Upon completion, the library must be APF authorized and concatenated to the VTE procedure STEPLIB. Failure to do so will result in the server's inability to compress virtual tape data sets using the Lempel-Ziv algorithm.

The library is usually named *prefix*.SEAGLPA and contains the EAGRTLIB module.

Note: The provided VTE procedure contains the concatenated library in a STEPLIB commented DD statement.

6. FDRABR from INNOVATION

INNOVATION's FDRABR V5.4.x and up are fully compatible with VTE. FDRABR V5.3.x users should apply zap-id P-53.5196, supplied by INNOVATION, to enable FDRABR to work correctly when new extents are allocated for a VTE virtual tape dataset.

The following should be considered when FDRABR is used to migrate (archive) virtual tape data sets from VTE diskpools to real tapes:

- If the VTE diskpool is non-SMS-managed, local rules in the FDRABR Restore Allocation List may cause portions of the virtual tape data sets to be recalled to disks that belong to different esoteric names.

This may cause the following messages to be issued by the VTE server when it allocates these data sets:

```
HVT311E ALLOC      ERROR, R15=04, ERROR=X'0214',X'0000',
                  X'00000000',DSN=M20V.VEC9500.M20.EC9500
HVT998I IKJ56241I  DATA SET M20V.VEC9500.M20.EC9500 NOT
                  ALLOCATED
HVT998I IKJ56241I  NO UNIT AVAILABLE
```

In case you have rules in the FDRABR Restore Allocation List, which will misdirect virtual tape data sets, you should add ALLOCATE rules, so that VTE virtual tape data sets are restored to the same set of volumes from which they were archived. This is done by making the VOL= disk volume list similar to the associated NVOL= disk volume list.

For example:

```
ALLOCATE DSN=CC.** ,VOL=(vol1a,vol1b,...),NVOL=(vol1a,vol1b,...)
```

In the above ALLOCATE rule, CC is the high-level qualifier of the VTE virtual tape data set, and *vol1x* are disks that belong to the same esoteric name and contain VTE virtual tape data sets. If one of the disks is not eligible for allocation, another disk from the same list will be selected.

- The FDRABR archive job should include the RECALL=YES parameter.
- The FDRABR archive job should include the MIGRAT=YES parameter (or this should be the product default).
- The SYNRECAL procedure should be properly customized (since it is used to recall virtual tape data sets).
- The VTE started task should have the appropriate authority to issue the MVS start command to the SYNRECAL procedure.

7. CA-Disk from CA

If CA-Disk is used to migrate virtual tape data sets to real tapes, then:

- The ARCHIVE statement of the CA-Disk migrating job should include the DISP=RECAT parameter (otherwise, the server will not be able to locate its virtual tape data sets for recall purposes).
- The ARCHIVE statement of the CA-Disk migrating job should include the EXPDT=99365 parameter (for permanent retention).
- The DMSAR procedure should be properly customized (since it is used to recall virtual tape data sets).
- The VTE started task should have the appropriate authority to issue the MVS start command to the DMSAR procedure.
- The recall delay time for the Auto-Restore function should be increased to 600 seconds. The default value of 120 seconds is too small and causes virtual tape data sets recall to fail.

To increase it:

- Enter DMS.Vvrm.DMSASM(ADSUMODC) member, modify the DELAY value accordingly and save the member.

Note: The member description is wrong and the DELAY value applies to ALL Auto-Restores performed through the DMSAR procedure.

- Enter DMS.Vvrm.INSTALL library and apply USERMODC.

Step 8: Start VTE

This section outlines how to start this product, VTE.

About this task

Note: Before proceeding with this step, make sure that the new hardware configuration has been activated successfully.

Procedure

1. Start the VTE server.

Start the VTE server by issuing the following operator command:

```
S hvtproc
```

Where *hvtproc* is the VTE procedure name.

Verify that the startup was successful by checking that message HVT009I was issued.

If you are a JES3 user, and use VTE only in a JES3 local processor, you should start the VTE server in the JES3 global processor as well.

2. Vary online the VTE virtual devices.

Vary the VTE devices online by issuing the following operator command:

```
V cuu1-cuu2,ONLINE
```

Where *cuu1* and *cuu2* are a range of VTE device numbers.

Check that message IEE302I was issued.

JES3 users, whose VTE virtual devices are JES3-managed, should issue the proper JES3 VARY operator command and check that message IAT5510 was issued.

Step 9: Run the IVP jobs

Once customization is completed, you should run two IVP jobs that perform basic tests to some of the VTE components.

About this task

Note that the IVP jobs do not intend to test the VTE interface to the tape management system.

Procedure

1. Run the first Installation Verification Program.

Edit member HVTJIVPA in the SAMPLIB library. Substitute the #-prefixed parameters as required and submit it.

The job tests the following VTE components:

- HVTUTAP utility (to add a new virtual tape to VDB).
- Scratch selection and writing data on a 'known' non-specific scratch virtual tape.
- Writing data on the second label of the previously selected scratch virtual tape.
- Writing data on a 'unknown' non-specific virtual tape. Due to the lack of scratch virtual tapes in the VDB, you will be requested to reply with a new scratch virtual tape volume serial number to the HVT317W message.
 - In order to continue, the new virtual tape to be added should be IVP002.

- You will be requested to confirm the addition of a new tape to your tape management system.
 - You will be requested to reply "U" to IEC704A message.
 - Copying all tape data sets back to disk.
 - Comparing the original data sets with the data sets just copied back and forth. The return code from the compare steps (STEP09-STEP11) should be 0.
 - HVTUREP utility (to report the virtual tapes status in the VDB).
2. Run the second Installation Verification Program.
- Edit member HVTJIVPB in the SAMPLIB library. Substitute the #-prefixed parameters as required and submit it.
- The job tests the following VTE components:
- HVTUSCR utility (to scratch volumes IVP001-IVP002).
 - HVTUTAP utility (to delete volumes IVP001-IVP002).

Step 10: Add Virtual Tapes to the VDB

If you postpone the full implementation of VTE and want to "play around", you need to add some virtual tapes to the VDB.

Procedure

1. Add virtual tapes to the default tapepool.

To add virtual tapes to the default tapepool, edit member HVTUTAP in the SAMPLIB library. It contains a job that adds virtual tapes to the VDB.

Note: Substitute the #-prefixed parameters as required then save the member. It is recommended to use the provided virtual tapepools without modification and, in a later stage when fully implementing VTE, to add more virtual tapes.

Submit the job. The job should end with a return code of 0.

2. Add virtual tapes to specific tapepools to VTE and to the tape management system.

If you want to fully implement VTE, refer to [Chapter 33, "Implementing VTE and Tape Management Systems,"](#) on page 349, for a full description of how to define virtual tapes and tapepools to VTE and the specific tape management system in use at your site.

Chapter 30. VTE Initialization Statements

This chapter provides a description of the VTE initialization statements. It describes the initialization statements members, syntax, parameters, and values.

The Initialization Statements Residence

Different members are used for the initialization statements. These members are described in this section.

The VTE initialization statements are supplied in separate members that reside in the partitioned data set pointed to by the HVTSIN DD statement of the VTE procedure. The members names are composed of a six-character (fixed) string (that identifies the member type) suffixed by a two-character, modifiable, alphanumeric string. That way, multiple versions of initialization statements can be managed.

During initialization, VTE reads the HVTSYSxx member to find out from which members to read its initialization statements. The default HVTSYSxx is pointed to from the VTE procedure. At the VTE server startup, the operator can supply a different two-character, alphanumeric string to be appended to HVTSYSxx, so that a different HVTSYSxx member is used.

The members and their respective initialization statements are listed in the following table:

Table 22. VTE initialization statements members				
Member name	Description	Initialization Statements	Dynamically refreshable	Optional/Mandatory
HVTDLGxx	Contains the statements related to the VTE Dialog.	DLGPFK DLGAUTH	Y	O
HVTMCSxx	Contains the definitions required by VTE to detect WTO messages on the operator console.	MCS	N	M
HVTOPTxx	Contains the VTE server basic options.	IDENT JRN OPTIONS PAT PERFORM SMF SUBSYS TCP VDEVICE VDB VDSN	N	M

Table 22. VTE initialization statements members (continued)				
Member name	Description	Initialization Statements	Dynamically refreshable	Optional/Mandatory
HVTPOLxx	Contains the tapepools and diskpools definitions.	DPOOL TPOOL	Y	M
HVTPRLxx	Contains the PAT rules.	PRULE PFILT PPTRN	Y	O
HVTRULxx	Contains the dynamic reallocation rules.	RULE RFILT RPTRN	Y	O
HVTSYSxx	Contains all other accessed members default suffixes.	SYS	N	M
HVTTRLxx	Contains the tapepool assignment definitions.	TRULE	Y	O
HVTVLTxx	Contains the VTE e-Vault complex definitions.	VLTOWN VLTRMT	N	O

Initialization Statements Syntax

Initialization statements are supplied on one or more 80-byte records. A VTE initialization statement consists of a set of such records.

Each VTE initialization statement is logically divided into the following fields. The fields are coded in free form, can start on any column, and are separated by at least one blank.

```
name [pos1][,pos2][,kwd1=val1][,kwd2=val2] [comment]
```

- Name field:

The name field uniquely identifies the particular initialization statement. It can start on any position in the record and must be followed by at least one blank.

- Parameters field:

The parameter field contains parameters separated by commas. It follows the name field and must be preceded and followed by at least one blank.

The parameters in this field are positional or keyword parameters. All positional parameters must precede all keyword parameters. Keyword parameters must follow the positional parameters.

- A positional parameter consists of characters that must be coded, as shown, in the particular parameter or variable information. If a positional parameter is omitted, a comma should be coded to indicate the omitted parameter.
- A keyword parameter consists of characters coded, as shown, in the particular parameter followed by an equal sign (=). The equal sign is followed by either characters that must be coded, as shown, or variable information. If the variable information contains embedded blanks, it must be enclosed by

quotes. The keyword parameters can be coded in any order in the parameter field and must follow the positional parameters (if any).

A positional parameter or a keyword parameter value can consist of a list of positional or keyword sub-parameters. Such lists should be enclosed in parentheses.

- Comment field:

The comment field follows the parameter field and must be preceded by at least one blank. Comments may be coded even though the parameter field is continued on a subsequent record.

When the total length of the fields on an initialization statement exceeds 80 columns, the parameters field (including sub-parameter lists) can be continued on subsequent records. The last parameter on the continued record should be followed by a comma and the next parameter is coded on the continuation record on any column. The name field should not be coded on the continuation record.

An initialization statement starting with an asterisk (*) is considered a comment and is ignored.

HVTMCSxx – Extended MCS Console Support

The HVTMCSxx member contains all the definitions required by VTE to detect messages on the operator console.

The HVTMCSxx member must contain at least one and as many as 80 MCS initialization statements.

The following figure displays the HVTMCSxx member syntax:

```
MCS      MSGID=msg_id,
          DO={COMMAND|REPLY|
             CA1ACT|CA1INACT|
             TLMACT|TLMINACT|
             ENDMCS|WTL }
          [,WTL={YES|NO}]
          [,VALIDAT1='%%n=value']
          [,VALIDAT2='%%n=value']
          [,VALIDAT3='%%n=value']
          [,VALIDAT4='%%n=value']
          [,DLM1=d1m1]
          [,DLM2=d1m2]
          [,DLM3=d1m3]
          [,TEXT=text]
          [,EXITMCS={YES|NO}]
```

MCS

The MCS initialization statement defines a series of actions (for example, operator commands or replies) to be issued automatically when a given event occurs in the system.

Such events are operations usually initiated or handled by real 3480/3490/3590/3592, which should now be handled by the VTE server.

An event is identified by a message appearing on the console. If more than one event should be handled, the MCS initialization statement should be repeated.

MSGID=msg_id

msg_id

Specifies a string identifying a particular text. Once this text starts in positions 1-8 of the message on the operator console, VTE handles it.

Value range: A 1 to 16 alphanumeric character string.

Default: None.

DO={COMMAND | REPLY | CA1ACT | CA1INACT | TLMACT | TLMINACT | ENDMCS | WTL}

Specifies the action to be performed when the above event occurs.

COMMAND

Issue an operator command.

REPLY	Reply to the message comprising the event that occurred.
CA1ACT	Notify VTE that CA-1 is active in the system.
CA1INACT	Notify VTE that CA-1 is not active in the system.
TLMACT	Notify VTE that TLMS is active in the system.
TLMINACT	Notify VTE that TLMS is not active in the system.
ENDMCS	Deactivate the VTE MCS subtask
WTL	Writes the message comprising the event on the server log.
	Value range: N/A.
	Default: None.

WTL={YES|NO}

Specifies whether the reply of the issued command is to be written on the server log. This option is applicable only when DO=COMMAND. If specified for other DO changes, it is ignored.

YES	The reply is written on the server log.
NO	The reply is not written on the server log.
	Value range: N/A.
	Default: None.

VALIDATn='%%n=value'

Specifies that validation of the *n*-th word of the message comprising the event that triggered the performed action, should be performed. This validation is done against an internal value. If the validation fails, the DO clause is not performed. If more than one type of validation is required, more VALIDATn parameters can be specified (up to 4). The *n* suffix of the VALIDAT literal specifies the parameter sequence in the initialization statement. If more than one VALIDATn parameters are specified, all the validations must succeed in order for the DO clause to be performed.

%%n	Substituted by the <i>n</i> -th word of the message comprising the event that triggered the performed action.
	See the DLMn parameter description (below) for delimiter considerations.
value	One of the following:
	FLECUU N/A.
	FLEJOBNM None.
	FLEASID
	MCTISRV
	FLEJOBID
	~string

DLMn

Consecutive words in the message, comprising the event that triggered the performed action, are separated by standard delimiters. The following are considered to be standard delimiters between two consecutive words:

- Any consecutive number of blanks
- One comma (,)

Note: Two adjacent commas designate a null word.

The above delimiters are used for extracting the n -th word. If more delimiters are required, the $DLMn$ parameter defines additional delimiters. If more than one additional delimiter is required, more $DLMn$ parameters can be specified (up to 3).

The n suffix of the DLM literal specifies the parameter sequence in the initialization statement.

Value range: A 1 character string.

Default: None.

TEXT=*text*

text

Specifies the text to be contained in the command or reply. The text string may contain the following imbedded special symbols:

%%SERVSTC Substituted by the VTE server's STC name.

%%SERVID Substituted by the VTE server's implicitly or explicitly specified identifier.

%% n Substituted by the n -th word of the message comprising the event that triggered the performed action.

See the $DLMn$ parameter description (above) for delimiters considerations.

%%DFLTP Substituted by the default tapepool name. This tapepool name is characterized by its RANGE1=(0,0) parameter in the TPOOL initialization statement.

Value range: A 1 to 52 alphanumeric character string. The text may contain blanks, in which case it should be enclosed in quotes.

Default: None.

EXITMCS={YES|NO}

Specifies whether after processing the current msg_id according to the DO clause, MCS initialization statements that follow are to be processed as well. This offers the possibility to take several actions to one given detected msg_id.

YES The MCS processing exits after the current msg_id process ends.

NO MCS processing resumes with the next MCS initialization statement.

Value range: N/A.

Default: None.

HVTOPTxx – Basic Options

The HVTOPTxx member defines the basic VTE options.

The following figure displays the HVTOPTxx member syntax:

IDENT	SERVERID=server_id
JRN	THRESH=percent
OPTIONS	DLG={ENABLE DISABLE}, DYN={ENABLE DISABLE}, JRN={ENABLE DISABLE}, PAT=({ENABLE DISABLE},{ENABLE DISABLE}), RPL={ENABLE DISABLE}, SMF={ENABLE DISABLE}, SSN={ENABLE DISABLE}, TCP={ENABLE DISABLE}, VLT={ENABLE DISABLE}, ZIIP={ENABLE DISABLE},
PAT	INCLUDE={PRULE SHR CPRI}, ALIASPF=aliaspf
PERFORM	MAXTASK=max_task [,VDBACC=interval] [,CPU%=percent] [,ZIIP%=percent]
SMF	RECORD=smfrec#
SUBSYS	NAME=subsys_name
TCP	TYPE={SERVER CLIENT}, SERVERIP=ip.ip.ip.ip, PORT=port [,TCP=tcpname]
VDB	RMODE={RW R0}
VDEVICE	ESOTERIC=hvtes, VCHPID=[(,vchpid1[,vchpid2[,vchpid3 [,vchpid4]][])] [,SIMULATE={3590 3592}] [,VARYON=count] [,VSTATS={YES NO}]
VDSN	QUAL1=vdsnq1, QUAL2={%VOL TDSN }, QUAL3={%VOL TDSN qual3}

IDENT

The IDENT initialization statement defines the symbolic name (identifier) of VTE.

SERVERID=server_id

<i>server_id</i>	This name specifies the VTE identifier. This identifier is used to relate the VTE server and offline utilities to the database files. It is also used in various displays and messages and is used by the VTE server to ensure no other similar VTE server(s) are started simultaneously in the same system.
Value range:	A 1 to 8 alphanumeric character string.
Default:	None.

JRN

The JRN initialization statement defines the VDB journaling options.

THRESH=percent

<i>percent</i>	Specifies the overall percentage of used records in the journal file over which a warning message will be issued on each new written record. If the VDB journaling is disabled, this parameter is checked for syntax and is further ignored.
Value range:	Integer from 50 to 99.
Default:	None.

OPTIONS

The OPTIONS initialization statement defines those options that are to be handled (enabled) or ignored (disabled) by VTE. When used with the VTE server, some of the options may be enabled while the VTE server is running.

DLG={ENABLE|DISABLE}

Specifies whether the VTE Dialog is to be enabled or disabled on server's startup.

ENABLE	Specifies that the VTE Dialog is enabled on VTE server's startup.
DISABLE	Specifies that the VTE Dialog is disabled on VTE server's startup.
Value range:	N/A
Default:	None

DYR={ENABLE|DISABLE}

Specifies whether the dynamic reallocation is to be enabled or disabled on VTE server's startup.

ENABLE	Specifies that the dynamic reallocation feature is enabled on VTE server's startup.
DISABLE	Specifies that the dynamic reallocation feature is disabled on VTE server's startup.
Value range:	N/A
Default:	None

JRN={ENABLE|DISABLE}

Specifies whether the VDB journaling is enabled or disabled.

ENABLE	Specifies that the VDB journaling is enabled.
DISABLE	Specifies that the VDB journaling is disabled.
Value range:	N/A
Default:	None

PAT=({ENABLE|DISABLE},{ENABLE|DISABLE})

Specifies that the dynamic reallocation feature is disabled on VTE server's startup. Specifies whether Parallel Access Tape (PAT) is enabled or disabled on VTE server's startup.

The parameter has two sub-parameters. The first sub-parameter is related to batch allocations, while the second sub-parameter is related to dynamic allocations. Any combination of ENABLE|DISABLE values in the two sub-parameters is acceptable.

ENABLE	Specifies that PAT is enabled on VTE server's startup. For the first sub-parameter, the PAT feature is enabled for batch jobs only. For the second sub-parameter, the PAT feature is enabled for dynamic allocation only.
DISABLE	Specifies that PAT is disabled on VTE server's startup. For the first sub-parameter, the PAT feature is disabled for batch jobs only. For the second sub-parameter, the PAT feature is disabled for dynamic allocation only.
Value range:	N/A

Default: None

RPL={ENABLE|DISABLE}

Specifies whether the replication status is to be enabled or disabled on VTE server's startup.

ENABLE	<p>Specifies that the replication feature status is to be enabled on VTE server's startup.</p> <p>When enabled, the replication process for replication-eligible virtual tapes starts immediately upon the virtual tape unload.</p> <p>Note: The replication feature status is unconditionally set to DISABLE if the VLT option is disabled.</p>
DISABLE	<p>Specifies that the replication feature status is to be disabled on VTE server's startup.</p> <p>When disabled, the replication process for replication-eligible virtual tapes is delayed until the replication feature status is ENABLE-d (via the RPLON operator command).</p> <p>Value range: N/A</p> <p>Default: None</p>

SMF={ENABLE|DISABLE}

Specifies whether the VTE user SMF record generation is enabled or disabled.

ENABLE	Specifies that the VTE user SMF record generation is enabled.
DISABLE	Specifies that the VTE user SMF record generation is disabled.
Value range:	N/A
Default:	None

SSN={ENABLE|DISABLE}

Specifies whether the VTE subsystem is to be enabled or disabled on VTE server's startup.

The VTE subsystem is required to perform dynamic reallocation, to enable PAT, and to perform type3 and type4 e-Vault.

ENABLE	Specifies that the VTE subsystem is enabled on VTE server's startup.
DISABLE	Specifies that the VTE subsystem is disabled on VTE server's startup.
Value range:	N/A
Default:	None

TCP={ENABLE|DISABLE}

Specifies whether the TCP/IP connectivity is enabled or disabled. The TCP/IP connectivity is required by the VTE server if the e-Vault option is enabled and one or more remote off-site storage locations are type4 e-Vault locations.

ENABLE	Specifies that the TCP/IP connectivity is enabled.
DISABLE	Specifies that the TCP/IP connectivity is disabled.
Value range:	N/A
Default:	None

VLT={ENABLE|DISABLE}

Specifies whether the e-Vault option is enabled or disabled.

ENABLE	Specifies that the e-Vault option is enabled.
DISABLE	Specifies that the e-Vault option is disabled.
Value range:	N/A
Default:	None

The TCP/IP connectivity is required as well if the e-Vault option is enabled and one or more remote off-site storage locations are type4 e-Vault locations.

ZIIP={ENABLE|DISABLE}

Specifies whether the internal zIIP support is to be enabled or disabled on VTE server's startup.

ENABLE	<p>Specifies that the internal zIIP support is to be enabled on VTE server's startup.</p> <p>If set to ENABLE, but the host system does not have zIIP capabilities, it will be automatically set to DISABLE even though PROJECTCPU=YES is specified in IEAOPTxx.</p>
DISABLE	<p>Specifies that the internal zIIP support is to be disabled on VTE server's startup.</p> <p>Value range: N/A</p> <p>Default: None</p>

PAT

The PAT initialization statement defines the method upon which DD statements are marked as PAT-Eligible and the PAT alias volumes that are used by PAT.

ALIASPFX=aliaspfx

<i>aliaspfx</i>	<p>Specifies the range of PAT alias volumes that PAT uses when satisfying a mount request of a PAT-Eligible DD statement. The range starts with the 2 characters defined in this parameter followed by an integer from 0000 to 9999.</p> <p>For example, if the alias prefix is 9\$, the range of PAT alias volumes is 9\$0000 to 9\$9999.</p> <p>If the VTE server is part of a VTE complex and the SYSZVOLS resource is a global (SYSTEMS) resource, the <i>aliaspfx</i> value should be different in each server.</p> <p>Value range: 2 alphanumeric or national (\$, #, @) character string.</p> <p>Default: None.</p>
-----------------	--

INCLUDE={PRULE|SHR|CPRI}

This parameter specifies the method upon which DD statements are marked as PAT-Eligible.

PRULE	Specifies that DD statements are considered to be PAT-Eligible according to the PAT Rules.
SHR	Specifies that DD statements defining a disposition of DISP=SHR are considered to be PAT-Eligible.

CPRI	Specifies that DD statements defining a DCB of DCB=CPRI=S are considered to be PAT-Eligible.
Value range:	N/A.
Default:	None.

PERFORM

The PERFORM initialization statement specifies parameters that let the VTE server control some of its resources use.

CPU%=*percent*

The VTE server CPU% utilization can be limited to a specific percentage. The VTE server CPU% utilization will be kept below that limit, regardless of the workload or the overall CPU% utilization in the system.

It is highly recommended to use the standard SRM or WLM for VTE tuning and ignore this parameter.

<i>percent</i>	Specifies the CPU% utilization, which the server should not exceed.
	The percent value should be normalized. That is, if your system has more than one CPU (and the total CPU% utilization goes beyond 100%), the percent value reflects the normalized CPU% utilization (e.g. total CPU% divided by number of online CPU's). That way, the CPU% utilization as seen by the server will never go beyond 100%.
Value range:	Integer from 0 to 99. If CPU% is not specified or a value of 0 is specified, no CPU% utilization restrictions apply, and the server will use all resources made available to it by the system.
Default:	0.

MAXTASK=*max_tasks*

The VTE server uses a series of subtasks (I/O subtasks) to serve I/O requests that are issued by jobs using virtual tapes. On one hand, each such subtask uses server resources (CPU and storage), and on the other hand, it provides useful work for the served jobs. This parameter provides the possibility to set the trade-off between the main storage and the CPU used by the server for that purpose.

max_tasks

Specifies the number of I/O subtasks that are allowed to be active concurrently, use server CPU and storage and provide useful work for the served jobs.

- As the parameter value is set lower, less storage is used, but more CPU is spent for the concurrency synchronization.
- As the parameter value is set higher, more storage is used and less CPU is spent for the concurrency synchronization.
- If the parameter value is set sufficiently high (that is, equal to the number of defined devices), the concurrency algorithm is skipped so no CPU is spent for it at all. In this case, the maximum CPU usage efficiency is gained (an estimated decrease of 5% to 7% in the overall server CPU usage).

According to the above and put in other words, CPU usage reduction can be gained at the cost of more used storage.

See “[VTE Server Main Storage Usage](#)” on page 371 to calculate the storage usage as a function on the other server defined resources.

Assuming VLT=DISABLE, if the site generated 128 devices, the number of concurrent tasks is 64 and 32 devices are active, the server will use approximately 76MB.

If the number of concurrent tasks is set to its maximum (in this case, 128), the server will use approximately 114MB. In this case, the CPU usage will drop by 5% to 7%.

This parameter is not associated with the concurrent number of virtual devices currently served by the VTE server. If the *max_task* value is lower than the number of devices and I/Os are issued at once by more devices, the excess I/Os will be queued.

Value range: Integer from 16 to 256 or 0.

If greater than the number of generated devices or 0, it is adjusted to the number of generated devices.

The recommended value is 0 in which case the maximum CPU usage reduction is gained.

Default: None.

VDBACC=interval

VTE periodically accesses the VDB to check whether a reload of its contents is needed (as a result of adding or deleting new virtual tapes, scratching virtual tapes, etc.).

The frequency at which the VDB is accessed for VDB-In-Storage update can be tuned by this parameter.

A low VDBACC value enables VTE to faster recognize any change to the VDB contents. However, it imposes a great overhead on the VDB, overhead that may lead to high contention and even to delays in getting virtual tapes mounted.

A high VDBACC value minimizes the risk of experiencing contention on the VDB, however, changes to the VDB contents will be recorded slower. The expected impact may be that new added virtual tapes will be recognized by the VTE subsystem only after the VDBACC interval has expired.

The VDBACC value default of 90 seconds means that, on average, 45 seconds will pass from the time new virtual tapes are added to the VDB until the VTE subsystem recognizes them or from the time virtual tapes are scratched until they will be mounted as a result of a scratch mount request.

interval

Specifies the interval (in seconds) in which the VTE server accesses the VDB for testing whether a VDB-In-Storage update is needed.

Value range: Integer from 10 to 600.
 Default: 90.

ZIIP%=percent

The MVS dispatcher can dispatch all or only a part of the zIIP-eligible work to zIIP processor(s). That optional parameter specifies how much of that work will be dispatched on zIIP processor(s).

If ZIIP=DISABLE, the parameter is checked for valid syntax and is further ignored.

<i>percent</i>	Specifies the percent of zIIP-eligible work to be actually offloaded to zIIP processor(s). If no special reasons exist, it is recommended to specify a value of 100 or to omit that parameter, thus letting the VTE server make maximum use of the zIIP processor(s).	
	Value range:	Integer from 1 to 100
	Default:	100

SMF

The SMF initialization statement defines the VTE user SMF record definitions.

RECORD=smfrec#

smfrec# Specifies the VTE user SMF record number.

Value range: Integer from 128 to 255 or 0. If the specified value is 0, no VTE user SMF record is written (regardless of the SMF parameter value on the OPTIONS initialization statement).

Default: 0.

SUBSYS

The SUBSYS initialization statement defines the VTE subsystem name.

NAME=subsys_name

subsys_name Specifies the name of the VTE subsystem. This name should be unique (not used by any other MVS subsystem). It may or may not be defined in SYS1.PARMLIB(IEFSSNxx). If not defined, the VTE server will dynamically define the VTE subsystem and use it. If defined and not yet in use, the VTE server will use it. Otherwise, the VTE server will fail in its initialization process.

Value range: A 1 to 4 alphanumeric character string. The first character must range from G to Z.

Default: None.

TCP

The TCP initialization statement defines the TCP/IP connection properties of the VTE server. This connection is required by the type4 e-Vault feature.

TYPE={SERVER|CLIENT}

Specifies whether VTE runs as a TCP/IP server or TCP/IP client. Conceptually, the TCP/IP server waits passively for requests from one or more clients, processes the request, and returns the result to the client that originated the request. On the other side, the TCP/IP clients initiate service requests to the TCP/IP server.

The type4 e-Vault implementation requires that the VTE server that resides at the local location be defined as a TCP/IP server. VTE servers that reside at the remote off-site storage locations should be defined as TCP/IP clients.

SERVER	The VTE server runs in the local location and is defined as a TCP/IP server.
CLIENT	The VTE server runs in a remote off-site storage location and is defined as a TCP/IP client.
	Value range: N/A
	Default: None.

SERVERIP=*ip.ip.ip.ip*

<i>ip.ip.ip.ip</i>	Specifies the dotted decimal IP address of the MVS that runs the VTE TCP/IP server. This parameter should be specified in each one of the VTE servers that run as TCP/IP clients. The VTE server that runs the TCP/IP server just checks the initialization statement's syntax, but it further ignores it.
	Value range: Each <i>ip</i> is an integer from 0 to 255.
	Default: None.

PORT=*port*

<i>port</i>	Specifies the port number that will bind to the sockets used by the VTE TCP/IP server and clients. The port number should be identical in the server and all its clients. This port number should not be used by any other TCP/IP application in the MVS system that runs the VTE TCP/IP server.
	Value range: Integer value from 5001 to 65535.
	Default: None.

TCPNAME=*tcpname*

<i>tcpname</i>	Specifies the name of the TCP/IP address space the server is to connect to.
	If there is only one TCP/IP address space and the parameter is not specified, the system derives a value from the TCP/IP configuration file.
	Value range: A 1 to 8 alphanumeric or national (\$, #, @) characters string.
	Default: The TCPIP address space name as derived from the TCP/IP configuration file.

VDB

The VDB initialization statement defines the VDB characteristics.

VDB={RW|RO}

Specifies the read/write mode of the VDB.

RW	With this mode, the VDB can be updated as needed by the server or associated offline utilities.
----	---

RO

With this mode, the VDB can be updated as needed by the server or associated offline utilities. With this mode, all virtual tapes can be only read (regardless of the tapepool Read_Only mode). This is so since any action (other than read) implies a VDB update.

Since no VDB update can take place, a virtual tape that is being read does not get its VDB entry updated in order to reflect its status. That means that the virtual tape is not protected from concurrent write from another VTE server in another LPAR in which the Read_Only modes are not set. It is the customer's responsibility to ensure no concurrent access is done to these Read_Only mode virtual tapes (unless both accesses are for read purposes).

Any attempt to modify the virtual tape by means of rewrite, extend, create, scratch, add, delete, remove, vault or recovery is rejected.

VDB Read_Only mode implies that the journal file is in Read_Only mode as well.

When the VDB is placed on Read_Only mode disks and the VDB is set to Read_Only mode and virtual tapes are read, IOS000I followed by IEC331I and IEC251I messages that point to the VDB and the User Catalog the VDB is cataloged into, are to be expected. This is so because, although the VDB is opened for input, VSAM tries to update the file statistics on the file's close. These messages should be ignored and can be included in the automation for suppression.

Value range: N/A.

Default: None.

VDEVICE

The VDEVICE initialization statement defines the VTE virtual devices to be used by the VTE server.

ESOTERIC=*hvtes*

hvtes

The VTE virtual devices esoteric name. This name should be identical to the name you defined for VTE in the HCD during the installation process. Either this esoteric name (or the generic name defined for VTE virtual devices in the HCD) should be coded on the UNIT parameter of DD statements that use VTE virtual devices.

Value range: 1 to 8-character alphanumeric string.

Default: None

VCHPID=[([vchpid1[,vchpid2[,vchpid3[,vchpid4]]]])]

vchpid

Specifies virtual (existing or non-existing) channel path-ids used by VTE virtual devices.

The following rules apply to the virtual CHPIDs support:

- As many as four virtual CHPIDs are supported. By attaching the virtual control unit to more than one virtual CHPID, the VTE server will continue to function properly even when a CHPID attached to its control unit is varied offline.
- Virtual CHPIDs connected to the VTE virtual control unit can consist of real (existing) channels, virtual (nonexisting) channels or both.

For non-existing channel, it is mandatory that the output of the Display Channel Path status will indicate "CHP=xx DOES NOT EXIST". Any other description can cause unpredictable results. For example, in the following display, non-existing CHPID 60 can be used:

```
D M=CHP(60)
IEE174I 18.19.33 DISPLAY M 963
CHPID 60: TYPE=00, DESC=UNKNOWN, OFFLINE
DEVICE STATUS FOR CHANNEL PATH 60
CHP=60 DOES NOT EXIST
***** SYMBOL EXPLANATIONS*****
+ ONLINE @ PATH NOT VALIDATED - OFFLINE DOES NOT EXIST
* PHYSICALLY ONLINE $ PATH NOT OPERATIONAL
```

vchpid

(continued)

- Although not recommended, the VTE control unit and its attached CHPIDs can be defined in HCD. The recommended method is to define them in the VCHPID parameter (and omit them from the HCD).
- If the control unit and CHPIDs are defined in HCD, they override any VCHPID parameter. In that case, the VCHPID parameter is ignored and HVT038W message is issued.
- If the control unit and CHPIDs are not defined in HCD, the VTE server will use the values supplied on the VCHPID parameter.
- If two or more servers run in a single MVS system, each VTE server should define different VCHPID values (even when the virtual devices are defined in HCD and the VCHPID values are ignored).
- Non-existing CHPIDs defined through the VCHPID parameter should not be taken offline (by means of the 'CF CHP(xx),OFFLINE' operator command). If taken offline, the server will have to be bounced in order to vary them online. Existing CHPIDs can be taken offline and online by operator commands.
- Non-existing CHPIDs defined through the VCHPID parameter should not be taken offline (by means of the 'CF CHP(xx),OFFLINE' operator command). If taken offline, the server will have to be bounced in order to vary them online. Existing CHPIDs can be taken offline and online by operator commands. When more than one existing CHPID are defined through the VCHPID parameter (and not through HCD), they should not be taken offline at once (since, in this case, MVS does not check for online virtual devices attached to these CHPIDs).
- On its startup, the VTE server varies online only non-existing CHPIDs. Existing CHPIDs should be online or brought online prior to the server startup by using the 'CF CHP(xx),ONLINE' operator command.
- The 'D M=DEV(xxxx)' operator command does not display real information for non-existing CHPIDs and should be ignored.
- On server startup, HVT037I message lists the final used CHPIDs.

Value range: Any vchpid value is a 2 hexadecimal digit from 00 to FF.

If only one vchpid value is specified, the parentheses can be omitted.

Default: N/A.

SIMULATE={3590|3592}

Specifies the simulated device type. This parameter is applicable only for virtual devices generated as 3590. If specified for other generated device types it is checked for valid syntax and is further ignored.

For further information on virtual devices types and media, refer to [“Simulated virtual device types” on page 125](#).

Note that the virtual tape capacity is always set by the TAPECAP parameter value on the TPOOL initialization statement (regardless of the simulated device type).

3590	Specifies that the generated VTE 3590 virtual devices do not simulate other devices. Virtual tapes from all tapepools are of MEDIA3.
3592	Specifies that the generated VTE 3590 virtual devices simulate 3592 virtual devices. Virtual tapes from all tapepools are of MEDIA5.

Value range:	N/A
Default:	3590

VSTATS={YES|NO}

Specifies whether media statistics data (read/written bytes, media capacity) is to be provided by VTE when the Read Buffered Log and Medium Sense tape channel commands are posted.

This parameter aims to maintain VTE backward compatibility after APAR PI77753 apply as described below.

YES	Specifies that the media statistics data (read/written bytes, media capacity) is provided by VTE when the Read Buffered Log and Medium Sense tape channel commands are posted.
NO	Specifies that the media statistics data (read/written bytes, media capacity) is not provided by VTE when the Read Buffered Log and Medium Sense tape channel commands are posted.

VDSN

The VDSN initialization statement defines the name structure of the virtual tape data sets.

QUAL1=*vdsnq1*

<i>vdsnq1</i>	Specifies the high-level qualifier to be assigned to the virtual tape data set's name. This high-level qualifier should be unique throughout the system. No other system component should allocate and catalog data sets having this value as the high-level qualifier for a data set's name. If this condition is not observed, the VTE recovery procedure will not be able to properly locate the cataloged virtual tape data sets and will not be able to recover any VTE virtual tape.
---------------	--

Value range:	A one- to eight- alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).
--------------	--

Default:	None
----------	------

QUAL2={%VOL|%TDSN}

Specifies the second qualifier to be assigned to the virtual tape data set name.

%VOL	The virtual tape volume serial number preceded by 'V'.
%TDSN	The data set name of the first data set allocated on the virtual tape. Unless the application modifies this name before the data set is opened, this name is the name of the first tape data set written to the virtual tape. If needed, this name is truncated so that the total virtual tape data set name does not exceed 44 characters.

Note: If the first tape data set name can not be determined, the value is resolved to TEMPORAR.

Value range:	N/A
--------------	-----

Default:	None.
----------	-------

QUAL3={%VOL|%TDSN/*qual3*}

Specifies the third qualifier to be assigned to the virtual tapes data set name.

%VOL	The virtual tape volume serial number preceded by 'V'.
------	--

<code>%TDSN</code>	<p>The data set name of the first data set allocated on the virtual tape. Unless the application modifies this name before the data set is opened, this name is the name of the first tape data set written to the virtual tape. If needed, this name is truncated so that the total virtual tape data set name does not exceed 44 characters.</p> <p>If this tape data set is a GDG data set, the GDG suffix (that is, GnnnnVnn) is truncated so as not to make the virtual tape data set a GDG-like data set.</p> <p>Note: If the first tape data set name can not be determined, the value is resolved to TEMPORAR.</p>
<code>qual3</code>	<p>Specifies a literal to be used as the third virtual tapes data set name qualifier.</p> <p>Value range: If QUAL3=qual3, qual3 is a 1 to 8 alphanumeric or national (\$,#,@) characters string. The first character must be alphabetic or national (\$,#,@).</p> <p>Default: None.</p>

Note:

- At least QUAL2 or QUAL3 should be set to %VOL.
- If virtual tapes are moved between type4 e-Vault locations, all locations must specify the %VOL qualifier as the same qualifier (either 2nd or 3rd qualifier).

HVTPOLxx – Tape pools and Diskpools

The HVTPOLxx member defines the VTE tapepools and diskpools infrastructure.

The HVTPOLxx member must contain:

- At least one and as many as 50 DPOOL initialization statements.
- At least one and as many as 50 TPOOL initialization statements.

The following figure displays the HVTPOLxx member syntax:

```

DPOOL      NAME=dpool_name,
           UNIT=[(]unit[,storclas[,mgmtclas]][]],
           THRESH=(percent,seconds,[low%],[volmask])
TPOOL      NAME=tpool_name,
           DPOOL=dpool_name,
           TAPECAP=(megabytes,max_disks),
           THRESH=(percent,seconds),
           RANGE1={ (0,0) | (volser1,volser2) }
           [,RANGE2=(volser1,volser2)]
           [,RANGE3=(volser1,volser2)]
           [,RANGE4=(volser1,volser2)]
           [,RANGE5=(volser1,volser2)]
           [,RANGE6=(volser1,volser2)]
           [,RANGE7=(volser1,volser2)]
           [,CKPINTVL=io_writes#]
           [,DSNTYPE={BASIC|LARGE}]
           [,EATTR={OPT|NO}]
           [,LOCID=locid]
           [,COMPRESS=[(]type[,level]][]]]
           [,DEFERBY=hours]
           [,MTYPE={STANDARD|ENHANCED}]
           [,RMODE={RW|RO}]

```


DPOOL

The DPOOL initialization statement defines one diskpool (that is, one group of disks eligible for virtual tape data sets allocation). Usually this diskpool contains disks grouped under one esoteric name or one Storage Group.

NAME=*dpool_name*

<i>dpool_name</i>	Specifies the diskpool name. This name is later referenced by one or more tapepool(s) definition(s).
Value range:	1- to 16-character alphanumeric string.
Default:	None

UNIT=[(*unit*[,*storclas*[,*mgmtclas*]])]

This parameter specifies the disks (either SMS-managed or non-SMS-managed) that are eligible for virtual tape data sets allocations in this diskpool. If EAV disks are included in either the esoteric name or Storage Group defined by this diskpool, the PARMLIB IGDSMSxx member should specify USEEAV (YES). Failure to do so may lead to unexpected Sx37 abends when creating or extending virtual tapes.

Rules for UNIT sub-parameters co-existence and allocation decision:

- Either the *unit* or *storclas* sub-parameters should be specified. The *mgmtclas* subparameter is optional.
- Virtual tape datasets will be allocated on non-SMS-managed disks if the *unit* subparameter is specified and the *storclas* sub-parameter is not specified. The *mgmtclas* sub-parameter (if specified) is ignored.
- Virtual tape datasets will be allocated on SMS-managed disks if the *storclas* (and optionally, the *mgmtclas*) sub-parameter is specified. The *unit* sub-parameter (if specified) is ignored. The Data Class ACS routine should **not** assign any Data Class value to the virtual tape data sets.
- The Storage Class ACS routine should not remove the *storclas* value assigned by this parameter.

unit Specifies the esoteric name of a group of non-SMS-managed disks. This esoteric name should be defined in the EDT table (via HCD). The disks in this group should be of the same type (for example, only 3380 or 3390) and should have a mount attribute of STORAGE.

Value range: A one- to eight-character alphanumeric string

Default: None. If not specified, the *storclas* (and optionally, the *mgmtclas*) sub-parameter should be specified.

If only the *unit* value is specified, the parentheses can be omitted.

storclas Specifies the name of a valid Storage Class. Adjust the appropriate Storage Group ACS routine to relate this Storage Class with the Storage Group assigned to contain the virtual tape data sets.

The following restrictions apply to this Storage Class:

- Guaranteed Space=N

Value range: A one- to eight-character alphanumeric or national (\$, #, @) string. The first character must be alphabetic or national (\$, #, @).

Default: None. If not specified, the *unit* subparameter should be specified.

<i>mgmtclas</i>	Specifies the name of a valid Management Class. The following restrictions apply to this Management Class: <ul style="list-style-type: none"> • Expire After Days Non-Usage=NOLIMIT • Expire After Date/Days=NOLIMIT • Partial Release=N Value range: A one- to eight-character alphanumeric or national (\$, #, @) string. The first character must be alphabetic or national (\$, #, @). Default: None
-----------------	---

THRESH=(*percent*,*seconds*,[*low%*],[*volmask*])

This parameter specifies the conditions under which warning messages will be issued when the used space on the disks defined under the UNIT parameter exceeds a predefined threshold.

Additionally, it sets the low threshold target for the HVTUDST utility. The HVTUDST utility creates a list of virtual tape data sets to be de-staged, such that the threshold defined by the *low%* parameter for this diskpool is reached.

If an optional positional parameter (which is not the last) is omitted, a comma should be coded to indicate its position.

<i>percent</i>	Specifies the overall percentage of used space in the esoteric name or Storage Group associated with the current diskpool over which the HVT347W message will be issued. Value range: Integer from 50 to 99. Default: None
<i>seconds</i>	Specifies the interval (in seconds) on which the used space recalculation is done and HVT347W is re-issued. The recommended value is 180. Value range: Integer from 1 to 1800. Default: None
<i>low%</i>	Specifies the percentage of used space in the esoteric name or Storage Group associated with the current diskpool, under which the HVTUDST utility stops selecting virtual tape data sets to be de-staged. If "1" is specified, HVTUDST considers it to be the lowest possible value and selects all virtual tape datasets. Value range: Integer from 1 to 100. Default: 100 (No virtual tape data sets are selected).
<i>volmask</i>	Specifies the group of disks to be included in the used space calculation. If the parameter is omitted, all disks in the diskpool participate in the calculation. Otherwise, only the disks whose volser match the volmask value participate in the calculation. The <i>volmask</i> parameter is effective for non-SMS-managed diskpools only. If specified on an SMS-managed diskpool, it is ignored. The <i>volmask</i> parameter accepts mask characters as character placeholders. Value range: A one- to six-character alphanumeric or national (\$, #, @) string.

Default: *

TPOOL

The TPOOL initialization statement defines one tapepool, that is, one or more ranges of virtual tapes grouped under one common set of characteristics. If a tape management system is employed, the VTE tapepools along with their respective ranges should be identical to the tape management system tapepool definitions.

Additional tapepools may be defined and can be referenced by the VTE tapepool rules (TRULE initialization statements).

Each tapepool should define (in addition to other common characteristics) the diskpool it is associated with. This way, all virtual tape data sets belonging to a given tapepool reside on the disks defined under the associated diskpool.

One TPOOL initialization statement must designate the default tapepool, by having its associated TPOOL's RANGE1 defined as RANGE1=(0,0).

NAME=*tpool_name*

Specifies the name of the tapepool.

<i>tpool_name</i>	Specifies the tapepool name.
Value range:	1- to 16-character alphanumeric string.
Default:	None

DPOOL=*dpool_name*

Specifies the name of the diskpool.

<i>dpool_name</i>	Specifies the associated diskpool name defined under a DPOOL initialization statement.
Value range:	1- to 16-character alphanumeric string.
Default:	None

TAPECAP=(*megabytes,max_disks*)

Defines the virtual tape size capacity and the maximum number of disks allowed for a virtual tape dataset.

The TAPECAP parameter defines the virtual tape size capacity, as well as its associated virtual tape dataset layout. Please refer to the "Allocating Space for a New Virtual Tape Dataset" topic in ["Disk Space Management"](#) on [page 165](#) for a description on how the TAPECAP values affect virtual tape datasets allocation.

<i>megabytes</i>	Specifies the maximum capacity (in MB) of one virtual tape in this tapepool. When an output opened virtual tape volume reaches this limit, an End-of-Volume is forced, and a new scratch virtual tape mount is automatically requested.
------------------	---

If the associated virtual tape dataset is allocated with extended attributes (EATTR=OPT), the final capacity may be greater than specified by this parameter, regardless of whether or not the dataset resides on EAV disks.

If virtual tapes from this tapepool are moved between type4 e-Vault locations, the tapepools that store the e-Vaulted virtual tapes in all of these locations should specify a similar or higher *megabytes* value.

Value range:	Any positive integer
Default:	None

<i>max_disks</i>	Specifies the maximum number of disks that one given virtual tape dataset in this tapepool (representing one VTE virtual tape) may reside on. This value is used for calculating the allocated virtual tape dataset extent size, as well as the number of disks this virtual tape dataset is allowed to allocate. If the associated virtual tape dataset is allocated with extended attributes (EATTR=OPT), the final number of disks may be lower than specified by this parameter, regardless of whether or not the dataset resides on EAV disks. Value range: Integer from 1 to 16 Default: None
------------------	---

THRESH=(percent,seconds)

Specifies the threshold for warning messages to be issued.

This parameter specifies the conditions under which warning messages will be issued when the number of active virtual tapes defined in the tapepool exceeds a predefined threshold.

<i>percent</i>	Specifies the overall percentage of active virtual tapes in the tapepool over which the HVT348W message will be issued. Value range: Integer from 50 to 99 Default: None
<i>seconds</i>	Specifies the interval (in seconds) on which the active virtual tapes percentage recalculation is done and HVT348W is reissued. The recommended value is 1200. Value range: Integer from 180 to 1800 Default: None

Notes:

1. The data, according to which the threshold values are calculated, is extracted from the VDB-In-Storage.
2. In order for the VDB-In-Storage to be accurate, it is updated as soon as a threshold interval expires (after which message HVT354I is expected).
3. When multiple TPOOL initialization statements are defined (with associated THRESH parameters), only the shortest threshold interval will trigger the VDB-In-Storage update. All other TPOOLS (with a longer threshold interval) do not trigger its update. Rather, they benefit from the update done earlier as a result of the shortest threshold interval expiration.
4. If the VDB is shared in a VTE complex, the VTE servers running in other LPARs sharing the same VDB will not trigger their respective VDB-In-Storage update as a result of the current LPAR threshold interval expiration.

RANGEn=(volser1,volser2)

Specifies a range of tapepools.

The *RANGE_n* parameter informs the VTE server about the tapepools/tapes relationship. The *RANGE_n* definition does not define new virtual tapes in the VDB, nor does it hardcode the tapepool name in the existing virtual tape entries.

The ranges of a particular tapepool must be identical to the tapepool ranges that are already defined in the tape management system. If wrong ranges are defined, VTE may present a scratch virtual tape from the wrong tapepool. In this case, the result is that the tape management system rejects the virtual tape.

As many as seven *RANGE*s can be specified (in any order). All *RANGE_n* (except *RANGE1*) are optional. The *n* suffix of the *RANGE* literal specifies the parameter sequence in the initialization statement.

volser1 Specifies the first virtual tape volume serial number from one of the tapepool's ranges.

Value range: A 1- to 6-character alphanumeric string

Default: None

volser2 Specifies the last virtual tape volume serial number from one of the tapepool's ranges. *volser2* should not be lower than *volser1*.

Value range: A 1- to 6-character alphanumeric string

Default: None

CKPINTVL=*io_writes*#

Specifies when to establish a checkpoint on a virtual tape.

In order to be able to recover a virtual tape (whose written process has been suddenly interrupted) up to a known point, VTE checkpoints the written blocks. That checkpoint indicates the last accessible block on the virtual tape.

io_writes# Specifies the number of write operations (from the last taken checkpoint) on a virtual tape from this tapepool after which a checkpoint is taken by the VTE server.

In addition, a checkpoint is automatically taken:

- Whenever a new virtual tape dataset extent is allocated on a new disk.
- On each tape channel command that causes synchronization between the tape drive and the controlling computer, specifically:
 - Synchronize
 - Write_Tape_Mark
 - Erase_Gap
 - Locate_Block
 - Rewind/Rewind_Unload
 - Data_Security_Erase
 - Any Space command (Forward_Space_Block, Forward_Space_File, etc.)
 - Any Read command that follows a Write command

If 0 is specified, checkpoints are taken only at the events specified in paragraphs (1) and (2) above. Each checkpoint performs 4 I/Os on the virtual tape, so specifying a low value implies a high I/O rate. The recommended value is 0, thus checkpointing on disks switch or on the selected tape channel commands specified above.

Value range: Integer from 0 to 999999

Default: 0

COMPRESS=([type[,level]])

Specifies whether or not to compress virtual tape data sets whose virtual tapes belong to this tapepool.

If compressed, it specifies the level of compression, "1" being the most powerful compression using the highest CPU resources.

<i>type</i>	Specifies the type of compression to be applied on the virtual tape dataset:	
0		Specifies that no compression is to be done for virtual tape data sets whose virtual tapes belong to this tapepool.
1		Compress using the 'Run Length Encoding' (RLE) algorithm.
2		Compress using the RLEfS type algorithm.
3		Compress using the RLEfS type algorithm. Compress using the Lempel-Ziv type algorithm. To use this compression type, the IBM REXX library must be purchased from IBM and installed. For further information, refer to "REXX library from IBM" in "Step 7: Third-Party Product Prerequisites and Coexistence" on page 246.
Value range:	N/A	
	If level is not specified, the parentheses can be omitted.	
Default:	0	
	For further information on compression, refer to Chapter 18, "Managing the diskpools," on page 165.	
<i>level</i>	Specifies the compression level to be applied on the virtual tape dataset:	
1		Compress the data so that a maximum compression ratio is achieved.
		Accordingly, a high amount of CPU is consumed when using this compression level.
2		Compress the data so that the compression ratio is approximately half of that achieved by level 1. The CPU consumption decreases accordingly.
3		Compress the data so that the compression ratio is approximately one quarter of that achieved by level 1. The CPU consumption decreases accordingly.
Value range:	Integer from 1 to 3	
Default:	2	

DEFERBY=hours

Specifies a time delay for virtual tapes to be scratched.

The DEFERBY parameter specifies if and by how much time virtual tapes scratching should be delayed by HVTUSCR utility.

hours

Specifies the number of hours to pass from the time HVTUSCR first encountered a scratch-eligible virtual tape in this tapepool until the virtual tape can be actually scratched.

The virtual tape scratch is not done automatically when the deferred time expires, but on the next run of the HVTUSCR utility.

The following special values exist:

0	No defer-scratch occurs and the HVTUSCR utility immediately scratches any scratch-eligible virtual tape from this tapepool.
9999	The HVTUSCR will never scratch virtual tapes from this tapepool. Use this value when: <ul style="list-style-type: none">• The virtual tapes from this tapepool are managed by an EDM and you did not define them to the local tape management system.• This site is used as a remote off-site storage location for type4 e-Vault virtual tapes and the virtual tapes are not defined to the local tape management system.

Value range: Integer from 0 to 1800 or 9999

Default: 0

DSNTYPE={BASIC|LARGE}

VTE allocates disk sequential datasets to store the virtual tapes data. This parameter specifies whether the allocated dataset is a Basic_Format_Dataset, whose total number of tracks per disk is limited to 65536, or a Large_Format_Dataset, whose size is limited only by the current disk's size.

Specifies the format of the dataset to be allocated.

BASIC Specifies that newly allocated virtual tape datasets on disks that are associated with virtual tapes in the tapepool will be Basic_Format_Datasets.

LARGE Specifies that newly allocated virtual tape datasets on disks that are associated with virtual tapes in the tapepool can be Large_Format_Datasets.

Special consideration should be given to 3480/3490 tape devices, which are limited to a capacity of 4M blocks. If the tape blocksize is not large enough and a 3480/3490 tape device is used, a tape EOVS will be forced before the full capacity is reached on the associated virtual tape dataset, even if DSNTYPE=LARGE is specified.

It is recommended to use a 3590 tape device whenever DSNTYPE=LARGE is specified.

If at least one of the following conditions is true, the VTE server will ignore the LARGE parameter and will allocate Basic_Format_Datasets:

- The system level is lower than z/OS 1.7.
- The calculated dataset extent size is lower than 4096 tracks.

Value range: N/A

Default: BASIC

EATTR={OPT|NO}

VTE allocates disk sequential datasets to store the data on virtual tapes. This parameter specifies whether the allocated dataset can support extended attributes, which are EAS-eligible and can be allocated on the cylinder-managed space of EAV disks.

Specifies whether extended attributes are supported.

NO	Specifies that new allocated virtual tape datasets on disks associated with virtual tapes in the tapepool will not be EAS-eligible and cannot be allocated on the cylinder-managed space of EAV disks.
OPT	Specifies that new allocated virtual tape datasets on disks associated with virtual tapes in the tapepool are EAS-eligible and can optionally be allocated on the cylinder-managed space of EAV disks..

Value range: N/A

Default: NO

Notes:

1. If the system level is lower than z/OS release 1.12, the VTE server will ignore the OPT parameter and will not assign extended attributes to the datasets.
2. If the datasets are shared by other systems, all systems must be z/OS release 1.12 or higher.
3. The extended attribute does not guarantee the dataset resides in the cylinder-managed space of EAV disks. It just indicates that the dataset can optionally reside in the cylinder-managed space and only if other conditions are met (i.e. EAV disks are available, sufficient space exists in the cylinder-managed space, etc.).
4. If other products access these datasets, such as a disk manager product or FTP, make sure the products support datasets allocated on cylinder-managed space.
5. VTE does not check if the associated diskpool contains EAV disks before assigning extended attributes to the allocated datasets.
6. Do not code EATTR=OPT if the associated diskpool does not have EAV disks. The extents size calculation of EAS-eligible datasets is not compatible with regular (not EAS-eligible) datasets, and different extent sizes may cause disk space fragmentation.

Please refer to the "Recalculating the disk extent size of EAS-Eligible Virtual Tape Datasets" topic in [Chapter 18, "Managing the diskpools," on page 165](#).

LOCID=locid

The LOCID parameter indicates that this tapepool is replication-eligible.

Upon unload, all new or modified virtual tapes from this tapepool are marked as replication-eligible. Thereafter, the e-Vault process will replicate them to the remote off-site storage location.

locid	Specifies the type3 or type5 remote off-site storage location to which virtual tapes from this tapepool will be replicated. This value must match the locid value of one of the VLTRMT initialization statements in the HVTVLTxx member.
-------	--

If the VLT option is disabled (VLT=DISABLE), the following is true:

- This parameter is checked for syntax and is further ignored.
- Unloaded virtual tapes will not be marked as replication-eligible.

Value range: A 1- to 8-character alphanumeric string

Default: STANDARD

MTYPE={STANDARD|ENHANCED}

Specifies the media type of the virtual tape volume.

This parameter is applicable for 3490 simulation only. If specified for other device simulation, it is ignored.

For further information on virtual device types and media, refer to the [“Simulated virtual device types”](#) on page 125.

STANDARD	Virtual tapes from this tapepool simulate standard 3490 (MEDIA1) cartridges.
----------	--

ENHANCED	Virtual tapes from this tapepool simulate enhanced capacity 3490 (MEDIA2) cartridges.
----------	---

Note that the virtual tape capacity is always set by the TAPECAP parameter value (regardless of the media type). This means that on an output enhanced capacity virtual tape can be written the number of megabytes specified in the TAPECAP parameter (after which an EOVS will be encountered), regardless of its media type. Setting a tapepool to contain enhanced capacity virtual tapes is not useful by itself. This option is of use only in cases when (for compatibility reasons) certain applications require the virtual tape to simulate an enhanced capacity cartridge.

Value range: N/A

Default: STANDARD

RMODE={RW|RO}

Specifies the read/write mode of this tapepool.

Specifies either Read_Write (RW) or Read_Only (RO) for this tapepool.

RO	Virtual tapes from this tapepool are set to regular Read_Write mode. For Read_Write mode, the VTE server will accept any action on the virtual tapes that belong to this tapepool.
----	--

RW	Virtual tapes from this tapepool are set to Read_Only mode.
----	---

With this mode, virtual tapes that belong to this tapepool can only be read. If the VDB is not in Read_Only mode, the corresponding VDB entry is updated to reflect the virtual tape read status.

Any attempt to modify the virtual tape by means of rewrite, extend, create, scratch, add, delete, remove, vault or recovery is rejected.

Value range: N/A

Default: RW

HVTPRLxx – PAT Rules

The HVTPRLxx member contains definitions that enable VTE to dynamically mark specific DD statements as PAT-Eligible according to certain criteria. For information on PAT rules and logic, refer to [“Virtual Tape Sharing”](#) on page 137.

The VTE PAT rules consist of series of INCLUDE and EXCLUDE PRULE initialization statements that specify whether the characteristics of a given tape allocation make it PAT-Eligible.

Each of the above initialization statements contains a series of one or more keyword parameters. The values of these parameters are matched against the corresponding DD statement parameters, after which a DD statement is marked as PAT-Eligible.

Using filter lists, it is possible to specify more than one value to each of the PRULE initialization statements keyword parameters.

PAT rule filtlist name (rather than a static value) to each one of the PRULE initialization statements keyword parameters. In turn, this PAT rule filtlist can contain as many as 64 different values and each one of them is matched against the corresponding DD statement parameters.

This option is achieved by coding a The name, coded on a PRULE initialization statement parameter value which refers to the PAT rule filtlist, is defined via the PFILT (PAT rule Filtlist) initialization statement. The PPTRN (PAT rule patterns) initialization statement contains the values associated with a PFILT initialization statement.

The HVTPLxx member must contain:

- None, one, and as many as 80 PRULE initialization statements.
- None, one, and as many as 20 PFILT initialization statements.
- Each PFILT block may define as many PPTRN initialization statements as needed, in order to define up to 64 patterns.

The following figure displays the HVTPLxx member syntax:

```
PFILT      NAME=pfilt_name
PPTRN      PVALUE=[(]pattern1[,pattern2[,...
            [,pattern20]])]
PRULE      {INC|EXC}
            [,JOBNAME={[-]job_name|&pfilt_name}]
            [,STEP={[-]step_name|&pfilt_name}]
            [,PGM={[-]program_name|&pfilt_name}]
            [,DSN={[-]dataset_name|&pfilt_name}]
            [,UNIT={[-]unit_name|&pfilt_name}]
            [,DISP={[-]disp|&pfilt_name}]
            [,DATACLAS={[-]dataclas|&pfilt_name}]
            [,STORCLAS={[-]storclas|&pfilt_name}]
            [,MGMTCLAS={[-]mgmtclas|&pfilt_name}]
```

The following rules apply to the PFILT initialization statements:

- At least one PFILT initialization statement must precede the PPTRN initialization statements.
- The PFILT initialization statement along with all its adjacent PPTRN initialization statements forms one PFILT block. The PFILT block ends either when a new PFILT initialization statement is encountered or when no more PFILT initialization statements exist.
- The PFILT initialization statement may be referred to by a PRULE initialization statement, however, this is not mandatory. If not referred to, it is ignored
- The total number of patterns in a PFILT block must not exceed 64.
- The PFILT block can be defined before or after the PRULE initialization statement that refers it.

The following rules apply to the PPTRN initialization statements:

- All PPTRN initialization statements belong to the PFILT block that immediately precedes it.
- Each PPTRN initialization statement can contain one or as many as 20 patterns. If more than 20 patterns are to be coded, more PPTRN initialization statements must be defined.
- The total number of patterns (accumulated in all PPTRN initialization statements that belong to the PFILT block) must not exceed 64.
- Each pattern can accept mask characters as character placeholders. The following mask rules apply when using these character placeholders:
 - An asterisk (*) matches none, one or more successive characters.
 - A question mark (?) matches exactly one character.

The following rules apply to the PRULE initialization statements:

- The first PRULE initialization statement must be an INC type statement.

- Each PRULE initialization statement parameter value may be preceded by a NOT (^ hex 5F) sign. In this case, the compared values match if they are not equal. The NOT sign is mutually exclusive with the ampersand sign.
- Each PRULE initialization statement parameter value may be preceded by an ampersand (& hex 50) sign. In this case, the value represents a reference to a PFILT initialization statement with an identical NAME value. The ampersand sign is mutually exclusive with the NOT sign.
- If a PRULE initialization statement parameter value refers to a PFILT block, the DD statement value is successively matched against all patterns that belong to the PFILT block. If one pattern matches, the compared values match.
- If a PRULE initialization statement parameter value is missing from the statement, it is not checked against the corresponding DD statement value and it is considered to match.
- If a PRULE initialization statement parameter value is not a reference to a PFILT initialization statement, it can accept mask characters as character placeholders. The following mask rules apply when using these character placeholders:
 - An asterisk (*) matches none, one or more successive characters.
 - A question mark (?) matches exactly one character.

PFILT

The PFILT (PAT rule filtlist) initialization statement starts a new PFILT block. This PFILT block name should be specified in a PRULE initialization statement parameter in order to refer it.

NAME=pfilt_name

<i>pfilt_name</i>	Specifies the PFILT block name. This name may be referred by one or more PRULE initialization statement parameter values.
Value range:	A one- to eight-character alphanumeric string.
Default:	None

PPTRN

The PPTRN (PAT rule Patterns) initialization statement lists all patterns associated with the immediately preceding PFILT initialization statement to form the PFILT block.

Up to 20 patterns can be specified in one PPTRN initialization statement. If more patterns are to be specified, more PPTRN initialization statements can be defined.

The maximum number of patterns in one PFILT block is 64.

PVALUE=[(]pattern1[,pattern2[,...[,pattern20]])]

<i>pattern</i>	Specifies the patterns that are associated with the PFILT block. One PVALUE parameter must contain one or as many as 20 patterns. If only one pattern is specified, the parentheses can be omitted.
	In order to achieve the maximum number of allowed 64 patterns, more PPTRN initialization statements can be defined.
Value range:	Each pattern is a 1- to 44-character alphanumeric string. This string accepts mask characters as character placeholders.
Default:	*.

PRULE

INC|EXC

Specifies that the current PRULE initialization statement is an INC or EXC type PRULE initialization statement.

Value range: N/A
Default: None

JOBNAME=*job_name*

job_name

Specifies the job names whose allocations are subject to PAT.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default: *.

STEP=*step_name*

step_name

Specifies the step names whose allocations are subject to PAT.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default: *.

PGM=*program_name*

program_name

Specifies the job step program names whose allocations are subject to PAT.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default: *.

DSN=*dataset_name*

dataset_name

Specifies the dataset names that, if contained on a DD statement, cause the DD statement to be PAT-Eligible.

Value range: 1 to 44 alphanumeric character string, including periods. Each qualifier between the periods can be 1 to 8 alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default: *.

UNIT=*unit_name*

unit_name

Specifies the original unit (esoteric name, generic name or device type in EBCDIC) that, if coded in the UNIT parameter of a DD statement, this DD statement is subject to PAT. If the DD statement contains UNIT=AFF, the matched UNIT value is the one in the DD statement that this DD statement is AFFed to.

Value range: A one- to eight-character alphanumeric string.

Default: *.

DISP=disp

disp

Specifies certain DISP values that if contained on a DD statement, this DD statement becomes PAT-Eligible.

No validity check is done on the DISP parameter value.

Therefore, any character string which is an invalid DISP value will cause the DD statement not to be PAT-Eligible.

SHR If the DD status disposition is SHR, this DD statement is PAT-Eligible.

OLD If the DD status disposition is OLD, this DD statement is PAT-Eligible.

NEW If the DD status disposition is NEW, this DD statement is PAT-Eligible.

Value range: 8 alphanumeric or national (\$, #, @) characters.

Default: *.

DATACLAS=dataclas

dataclas

Specifies a Data Class that, if contained on a DD statement, causes the DD statement allocations to be subject to PAT.

The *dataclas* value can be coded in the DD statement in the respective ACS routine.

Value range: One to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: *.

STORCLAS=storclas

storclas

Specifies a Storage Class that, if contained on a DD statement, causes the DD statement allocations to be subject to dynamic reallocation.

The *storclas* value can be coded in the DD statement in the respective ACS routine.

Value range: A one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: *.

MGMTCLAS=mgmtclas

mgmtclas

Specifies a Management Class that, if contained in a DD statement, causes the DD statement allocations to be subject to PAT.

The *mgmtclas* value can be coded in the DD statement in the respective ACS routine.

Value range: A one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: *.

HVTRULxx – Dynamic Reallocation Rules

The HVTRULxx member contains definitions that enable VTE to dynamically reallocate specific DD statements according to certain criteria. For information on the dynamic reallocation rules, logic, and examples, refer to *Chapter 5, Dynamic Reallocation Logic* in the *IBM System Storage VTF Mainframe User's Guide [GC53-1191-00]*

The VTE dynamic reallocation rules consist of a series of INclude and EXclude RULE initialization statements that specify whether the characteristics of a given allocated dataset make it eligible for dynamic reallocation.

Each of the above initialization statements contains a series of one or more keyword parameters. The values of these parameters are matched against the corresponding DD statement parameters, after which a dynamic reallocation may occur.

Using filter lists, more than one value can be specified for each of the RULE initialization statements keyword parameters. This is done by coding a rule filtlist name (rather than a static value) to each one of the RULE initialization statements keyword parameters. In turn, this rule filtlist can contain as many as 64 different values and each one of them is matched against the corresponding DD statement parameters.

The name, coded on a RULE initialization statement parameter value which refers to the rule filtlist, is defined via the RFILT (Rule Filtlist) initialization statement. The RPTRN (Rule Patterns) initialization statement contains the values associated with an RFILT initialization statement.

The HVTRULxx member must contain:

- None, one, and as many as 80 RULE initialization statements.
- None, one, and as many as 20 RFILT initialization statements.
- Each RFILT block may define as many RPTRN initialization statements as needed, in order to define up to 64 patterns.

The following figure displays the HVTRULxx member syntax:

```
RFILT      NAME=rfilt_name
RPTRN      RVALUE=[(]pattern1[,pattern2[,...
              [,pattern20][)]

RULE        {INC|EXC}
              [,JOBNAME={[-]job_name|&rfilt_name}]
              [,STEP={[-]step_name|&rfilt_name}]
              [,PGM={[-]program_name|&rfilt_name}]
              [,DSN={[-]dataset_name|&rfilt_name}]
              [,UNIT={[-]unit_name|&rfilt_name}]
              [,VOL={[-]vol_ser|&rfilt_name}]
              [,DATACLAS={[-]dataclas|&rfilt_name}]
              [,STORCLAS={[-]storclas|&rfilt_name}]
              [,MGMTCLAS={[-]mgmtclas|&rfilt_name}]
              [,EXPDT=expdt]
```

The following rules apply to the RFILT initialization statements:

- At least one RFILT initialization statement must precede the RPTRN initialization statements.
- The RFILT initialization statement, along with all its adjacent RPTRN initialization statements, forms one RFILT block. The RFILT block ends either when a new RFILT initialization statement is encountered or when no more RFILT initialization statements exist.
- The RFILT initialization statement may be referred to by a RULE initialization statement, however, this is not mandatory. If not referred to, it is ignored.
- The total number of patterns in an RFILT block must not exceed 64.
- The RFILT block can be defined before or after the RULE initialization statement that refers it.

The following rules apply to the RPTRN initialization statements:

- All RPTRN initialization statements belong to the RFILT block that immediately precedes it.

- Each RPTRN initialization statement can contain one or as many as 20 patterns. If more than 20 patterns are to be coded, more RPTRN initialization statement must be defined.
- The total number of patterns (accumulated in all RPTRN initialization statements that belong to the RFILT block) must not exceed 64.
- Each pattern can accept mask characters as character placeholders. The following mask rules apply when using these character placeholders:
 - An asterisk (*) matches none, one or more successive characters.
 - A question mark (?) matches exactly one character.

The following rules apply to the RULE initialization statements:

- The first RULE initialization statement must be an INC type statement.
- Each RULE initialization statement parameter value may be preceded by a NOT (^ hex 5F) sign. In this case, the compared values match if they are not equal. The NOT sign is mutually exclusive with the ampersand sign.
- Each RULE initialization statement parameter value may be preceded by an ampersand (& hex 50) sign. In this case, the value represents a reference to a RFILT initialization statement with an identical NAME value. The ampersand sign is mutually exclusive with the NOT sign.
- If a RULE initialization statement parameter value refers to an RFILT block, the DD statement value is successively matched against all patterns that belong to the RFILT block. If one pattern matches, the compared values match.
- If a RULE initialization statement parameter value is missing from the statement, it is not checked against the corresponding DD statement value and it is considered to match.
- If a RULE initialization statement parameter value is not a reference to an RFILT initialization statement, it can accept mask characters as character placeholders. The following mask rules apply when using character placeholders:
 - An asterisk (*) matches none, one or more successive characters.
 - A question mark (?) matches exactly one character.

RFILT

The RFILT (rule filist) initialization statement starts a new RFILT block. This RFILT block name should be specified in a RULE initialization statement parameter in order to refer it.

NAME=*rfilt_name*

<i>rfilt_name</i>	Specifies the RFILT block name. This name may be referred to by one or more RULE initialization statement parameter values.
Value range:	A one- to eight-character alphanumeric string.
Default:	None

RPTRN

The RPTRN (Rule Patterns) initialization statement lists all patterns associated with the immediately preceding RFILT initialization statement to form the RFILT block.

Up to 20 patterns can be specified in one RPTRN initialization statement. If more patterns are to be specified, more RPTRN initialization statements can be defined.

The maximum number of patterns in one RFILT block is 64.

RVALUE=[(*pattern1* [,*pattern2* [,... [,*pattern20*]])]

<i>pattern</i>	Specifies the patterns that are associated with the RFILT block. One RVALUE parameter must contain one or as many as 20 patterns. In order to achieve the maximum number of allowed 64 patterns, more RPTRN initialization statements can be defined.
----------------	---

Value range:	Each pattern is a 1 to 44 alphanumeric character string. This string accepts mask characters as character placeholders.
Default:	*.

RULE

The RULE initialization statement defines an include or exclude rule for dynamic reallocation.

DATACLAS=*dataclas*

<i>dataclas</i>	Specifies a Data Class that, if contained on a DD statement, causes the DD statement allocations to be subject to dynamic reallocation.
	In JES2 environment:
	<ul style="list-style-type: none"> The <i>dataclas</i> value can be coded either in the DD statement or in the respective ACS routine.
	In JES3 environment:
	<ul style="list-style-type: none"> The <i>dataclas</i> value can be coded only in the DD statement. <i>dataclas</i> values set in ACS routines are ignored by the VTE subsystem.
Value range:	A one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).
Default:	*.

DSN=*dataset_name*

<i>dataset_name</i>	Specifies the dataset names that, if contained on a DD statement, cause the DD statement to be subject to dynamic reallocation.
Value range:	A 1 to 44 alphanumeric character string, including periods. Each qualifier between the periods can be one to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default:	*.

EXPDT=*expdt*

<i>expdt</i>	Specifies an EXPDT value that, if contained on a DD statement, causes the DD statement to be subject to dynamic reallocation.
Value range:	A numeric value of the form YYYY/DDD where YYYY is a four-digit year number (1900 through 2155) and DDD is a three-digit day number (000 through 366). This value should be coded as a four-digit year number (or mask) regardless of whether the DD statement specifies a two-digit year number (YYDDD) or a four-digit year number (YYYY/DDD).
Default:	*

INC|EXC

	Specifies that the current RULE initialization statement is an INC or EXC type RULE initialization statement.
Value range:	N/A.
Default:	None.

JOBNAME=job_name*job_name*

Specifies the job names whose allocations are subject to dynamic reallocation.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).

Default: *.

MGMTCLAS=mgmtclas*mgmtclas*

Specifies a Management Class that, if contained in a DD statement, causes the DD statement allocations to be subject to dynamic reallocation.

In JES2 environment:

- The *mgmtclas* value can be coded either in the DD statement or in the respective ACS routine.

In JES3 environment:

- The *mgmtclas* value can be coded only in the DD statement. *mgmtclas* values set in ACS routines are ignored by the VTE subsystem.

Value range: A one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: *.

PGM=program_name*program_name*

Specifies the job step program names whose allocations are subject to dynamic reallocation.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).

Default: *.

STEP=step_name*step_name*

Specifies the step names whose allocations are subject to dynamic reallocation.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).

Default: *.

STORCLAS=storclas*storclas*

Specifies a Storage Class that, if contained on a DD statement, causes the DD statement allocations to be subject to dynamic reallocation.

In JES2 environment:

- The *storclas* value can be coded either in the DD statement or in the respective ACS routine.

In JES3 environment:

- The *storclas* value can be coded only in the DD statement. *storclas* values set in ACS routines are ignored by the VTE subsystem.

Value range: A one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: *.

UNIT=unit_name*unit_name*

Specifies the original unit (esoteric name, generic name or device type in EBCDIC) that, if coded in the UNIT parameter of a DD statement, this DD statement is subject to dynamic reallocation. If the DD statement contains UNIT=AFF, the matched UNIT value is the one in the DD statement that this DD statement is AFF-ed to.

Value range: A one- to eight-character alphanumeric string.

Default: *.

VOL=vol_ser*vol_ser*

Specifies a specific volume serial number that, if contained on a DD statement, causes the DD statement to be subject to dynamic reallocation.

Note: Note that only the first volser specified on the DD statement is checked against the *vol_ser* value.

Value range: One to six alphanumeric or national (\$, #, @) characters.

Default: *.

HVTSYSxx – Startup Selection

The HVTSYSxx member contains the suffixes of all other members that contain the VTE initialization statements.

The HVTSYSxx member must contain one SYS initialization statement.

The following figure displays the HVTSYSxx member syntax:

```
SYS      DLG=xx
         MCS=xx,
         OPT=xx,
         POL=xx,
         PRL=xx,
         RUL=xx,
         TRL=xx,
         VLT=xx
```

SYS

The SYS initialization statement contains the two-character member suffix to be appended to the other members that contain VTE initialization statements.

MCS=xx

This parameter specifies the HVTMCSxx member that contains the VTE extended MCS console initialization statements.

xx	Two-character member suffix to be appended to HVTMCS to form the name of the HVTMCSxx member that will be used.
Value range:	A two-character alphanumeric string
Default:	None

OPT=xx

This parameter specifies the HVTOPTxx member that contains the VTE basic initialization statements.

xx	Two-character member suffix to be appended to HVTOPT to form the name of the HVTOPTxx member that will be used.
Value range:	A two-character alphanumeric string
Default:	None.

POL=xx

This parameter specifies the HVTPOLxx member that contains the VTE tapepools and diskpools definitions.

xx	Two-character member suffix to be appended to HVTPOL to form the name of the HVTPOLxx member that will be used.
Value range:	A two-character alphanumeric string
Default:	None.

PRL=xx

This parameter specifies the HVTPRLxx member that contains the VTE PAT rules definitions.

xx	Two-character member suffix to be appended to HVTPRL to form the name of the HVTPRLxx member that will be used.
Value range:	A two-character alphanumeric string
Default:	None

RUL=xx

This parameter specifies the HVTRULxx member that contains the VTE dynamic reallocation rules definitions.

xx	Two-character member suffix to be appended to HVTRUL to form the name of the HVTRULxx member that will be used.
Value range:	A two-character alphanumeric string
Default:	None

TRL=xx

This parameter specifies the HVTTRLxx member that contains the VTE tapepool rules definitions.

xx Two-character member suffix to be appended to HVTTRL to form the name of the HVTTRLxx member that will be used.

Value range: A two-character alphanumeric string.

Default: None

VLT=xx

This parameter specifies the HVTVLTxx member that contains the VTE e-Vault local and remote off-site storage locations definitions.

xx Two-character member suffix to be appended to HVTVLT to form the name of the HVTVLTxx member that will be used.

Value range: A two-character alphanumeric string

Default: None

DLG=xx

This parameter specifies the HVTDLGxx member that contains the VTE Dialog initialization statements.

xx Two-character member suffix to be appended to HVTDLG to form the name of the HVTDLGxx member that will be used.

Value range: A two-character alphanumeric string

Default: None

HVTTRLxx – Tapepool Rules

The HVTTRLxx member contains definitions that enables VTE to assign tapepools to scratch mount requests (even though the virtual tapes tapepool is not handled by the tape management system tape pools definitions). For information on the tapepool rules use, refer to *Chapter 3, Implementation* in the *IBM System Storage VTF Mainframe User's Guide [GC53-1191-00]*.

The following rules apply to the TRULE initialization statements:

- The very first TRULE initialization statement must be an INC type statement.
- Each TRULE initialization statement value may be preceded by a NOT (^ hex 5F) sign. In this case, the compared values match if they are **not** equal.
- If a TRULE initialization statement parameter value is missing from the statement, it is not checked against the appropriate DD statement value and it is considered to match.
- The TRULE initialization statement parameter values (except TPOOL) accept mask characters as character placeholders. The following mask rules apply when using these character placeholders:
 - An asterisk (*) matches none, one or more successive characters.
 - A question mark (?) matches exactly one character.

The HVTTRLxx member may contain 0 - 80 TRULE initialization statements.

The following figure displays the HVTTRLxx member syntax:

```
TRULE      {INC|EXC},  
           TPOOL=tpool_name  
           [,JOBNAME=job_name]  
           [,DSN=dataset_name]  
           [,PGM=program_name]
```

INC|EXC

Specifies that the current PRULE initialization statement is an INC or EXC type PRULE initialization statement.

Value range: N/A
Default: None

TPOOL=*tpool_name*

tpool_name Specifies the tapepool to be assigned to the virtual tape. Although this parameter can be specified on both INC and EXC type TRULE initialization statements, its value is ignored for EXC type TRULE initialization statement.

Value range: 1- to 16-character alphanumeric string.
Default: None

JOBNAME=*job_name*

job_name Specifies the jobnames whose allocated virtual tapes are to be assigned a tapepool.

Value range: One to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).
Default: *.

DSN=*dataset_name*

dataset_name Specifies the dataset name that if allocated on a virtual tape, that virtual tape is to be assigned a tapepool.

Value range: A 1- to 44-character alphanumeric string, including periods. Each qualifier between the periods can be a one to eight alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).
Default: *.

PGM=*program_name*

program_name Specifies the step program names whose allocated virtual tapes are to be assigned a tapepool.

Value range: One to eight alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).
Default: *.

HVTVLTxx – e-Vault complex

The HVTVLTxx member defines the VTE e-Vault complex definitions.

For more information on e-Vault logic and definitions, refer to [Chapter 22, “e-Vault - VTE electronic vaulting,”](#) on page 191.

The HVTVLTxx member must contain:

- One VLTOWN initialization statement.

- At least one and as many as 10 VLTRMT initialization statements.

The following figure displays the HVTVLTxx member syntax:

VL TOWN	MAXTASK=[(]max_tasks_g[,max_tasks_r[)], OWNLOCID=locid [,TCPNAME=tcpname] [,XFERHLQ=temp_hlq] [,VLTSUS=Y/N] [,VLTRET=nnnn]
VL TRMT	LOCID=locid, TYPE=3,FTPIP=ip.ip.ip.ip, NETRC=(dsname,library,member), [,PATH=path] TYPE=4,FTPIP=ip.ip.ip.ip, NETRC=(dsname,member) TYPE=5,FTPIP=ip.ip.ip.ip, NETRC=(dsname,library,member), [,PATH=path] [,VSID=vsid] }

VL TOWN

The VL TOWN initialization statement defines the e-Vault local location characteristics. This initialization statement (in conjunction with the VLTRMT initialization statement(s)) defines the VTE e-Vault complex.

MAXTASK=[(]max_tasks_g[,max_tasks_r[)]

max_tasks_g Specifies the number of VTE subtasks that simultaneously control the execution of up to max_tasks HVTUFTP or HVTUCTC utilities. These utilities are general purpose utilities, that is, can perform 'to be vaulted', 'to be returned' or 'to be remote-scratched' requests from/to the local location to/from a type3, type4 or type5 remote off-site storage location.

Note: If only the *max_tasks_g* parameter is specified, the parentheses can be omitted.

Value range: An integer from 1 to 16

Default: None

max_tasks_r Specifies the number of VTE subtasks included in the *max_tasks_g* parameter that are dedicated to perform 'to be returned' and 'to be remote-scratched' requests only. This parameter is useful when an ad hoc recall or remote-scratch are requested and all subtasks are busy with vaulting tapes. If *max_tasks_r* is not 0, all recall or remote-scratch requests will be executed ahead of the line by means of these dedicated subtasks. If no 'to be returned' or 'to be remote-scratched' requests are pending, these subtasks are idle.

Value range: An integer from 0 to one less than *max_tasks_g*

Default: 0

OWNLOCID=locid

Specifies the local vault location name. This value should be similar to the local vault location name specified in the tape management system. VTE uses this value to identify the virtual tapes that are to be vaulted to a remote off-site storage location or that are to be returned from an off-site storage location.

Value range: A one- to eight-character alphanumeric string or blanks (surrounded by quotes).

Default: None

TCPNAME=tcpname

tcpname

Specifies the name of the TCP on the local host to which the FTP client should connect. The parameter should not be specified if the system is not configured for multiple TCPIP stacks.

This parameter only applies to type3 and type4 e-Vaults. For a type4 e-Vault, the TCPNAME parameter on the TCP initialization statement should be specified as well. If TCPNAME is specified for other e-Vault types, it is checked for valid syntax and is further ignored.

Value range: A one- to eight-character alphanumeric string or national character string (\$, #, @).

Default: None

VLTRT=nnnn

nnnn

The VTE server controls the interval at which VTE retries failing attempts to e-Vault virtual tapes. This value can be later modified by a VTE operator command.

The *nnnn* variable specifies the interval, in minutes, at which VTE retries the e-Vault of virtual tapes whose previous e-Vault process failed. Follow these guidelines:

- The count is stopped while the e-Vault processes activity is suspended.
- Since the count starts after a full scan of all e-Vault eligible virtual tapes completes, the actual expected interval between successive retries may be longer than *nnnn*.

Value range: Integer from 1 to 1440

Default: 5

VLTSUS=Y/N

Specifies whether the e-Vault processes activity is to be suspended on VTE server's startup. This option can be later modified by a VTE operator command.

- Y – Specifies that the e-Vault processes activity is to be suspended on VTE server's startup. The e-Vault eligible tapes remain in e-Vault eligible status.
- N – Specifies that the e-Vault processes activity is not to be suspended on VTE server's startup. The e-Vault eligible tapes will be e-Vaulted subject to other relevant factors, like remote off-site storage location status or retry conditions.

Value range: N/A

Default: N

VLTRMT

The VLTRMT initialization statement defines one remote off-site storage location (type3, type4, or type5) characteristics. If more than one remote off-site storage location participates in the VTE e-Vault complex, VLTRMT initialization statement should be repeated.

Note: Type4 e-Vault remote off-site storage locations see the local location as its remote off-site storage location. Therefore, care should be taken when defining this initialization statement since its definitions differ according to the location in which it is defined.

LOCID=locid

locid

Specifies the remote off-site storage location name. This value should be identical to the remote off-site storage location name specified in the tape management system. VTE uses this value to identify the virtual tapes that are to be e-Vaulted to that remote off-site storage location or that are to be returned from that off-site storage location, along with the FTP characteristics.

If TYPE=4:

In the local location, the *locid* value is the value known to the tape management system as the remote off-site storage location name.

In the remote off-site storage location, the *locid* value is the name of the local location.

If TYPE=5:

In addition to being identical to the remote off-site storage location name specified in the tape management system, the *locid* value should be identical to the relevant Cloud Tape Connector cloud name, that is, to either:

- The CLOUD_NAME parameter value specified in the Cloud Tape Connector DEFINE_CLOUD statement,
- or
- The "Cld Name" value in the Cloud Tape Connector Parmlib Options menu, Option 3, Cloud Server Display panel.

Value range: A one- to eight-character alphanumeric string or blanks (surrounded by quotes).

Default: None

TYPE={3|4|5}

Specifies the type of e-Vault this remote off-site storage location supports.

3 Specifies that the remote off-site storage location is of type 3 ("type3").

When TYPE=3:

- FTPIP and NETRC parameters must be specified.
- PATH parameter is optional.

4 Specifies that the remote off-site storage location is of type 4 ("type4").

When TYPE=4:

- FTPIP and NETRC parameters must be specified.
- PATH parameter (if specified), is checked for valid syntax, then ignored.

5	Specifies that the remote off-site storage location is of type 5 ("type5"). When TYPE=5: <ul style="list-style-type: none"> • FTPIP, NETRC or PATH parameters (if specified), are checked for valid syntax, then ignored. • Note: During initialization, VTE does not check whether IBM Cloud Tape Connector for z/OS that is required by type5 is available. Value range: N/A Default: None
---	---

FTPIP=*ip.ip.ip.ip*

FTPIP parameter is meaningful for type3 and type4 e-Vault.

ip.ip.ip.ip	<p>This parameter specifies the dotted-decimal IP address of the remote off-site storage location host that runs an FTP server.</p> <p>This parameter should be specified in each one of the VTE servers that participate in the VTE type3 and type4 e-Vault complex, namely:</p> <ul style="list-style-type: none"> • The local location VTE server should specify the IP address of all remote off-site storage locations. • Each remote off-site storage location should specify the IP address of the local location. • For type3 e-Vault, the parameter should be specified only in the local location VTE server. <p>Note: The value must be identical to the MACHINE value in the NETRC data set.</p> <p>Validation is not done on startup, and if a mismatch is detected, the external FTP service will fail.</p> Value range: Each <i>ip</i> is an integer from 1 to 255. Default: None
-------------	---

NETRC=(*dsname,member*)

NETRC parameter is meaningful for type3 and type4 e-Vault.

This parameter contains the remote off-site storage location's FTP specifications (like user and password). The reason this parameter is used (rather than direct specification of the user and password values) is for security reasons - this way these values can be stored in a protected dataset accessible only by the HVTUFTP utility and the FTP program.

dsname	<p>Specifies the partitioned dataset name which in conjunction with the following member contains the remote off-site storage location's FTP specifications.</p> Value range: A 1- to 44-character alphanumeric string, including periods. Each qualifier between the periods can be 1 to 8 alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @). Default: None
--------	---

member

Specifies the member which in conjunction with the previous partitioned dataset, contains the remote off-site storage location's FTP specifications.

Following is a sample member that contains the FTP specification:

```
MACHINE ftpip LOGIN user PASSWORD password
```

In the above sample the uppercase values should be typed as shown. The lowercase values are case sensitive:

ftpip

Specifies the remote off-site storage location dotted-decimal IP address. This address is a standard four-qualifiers IP address in which each qualifier is a number from 1 to 255.

Note:

- The value must be identical to the FTPIP value. Validation is not done on startup, and if a mismatch is detected, the external FTP service will fail.
- Each qualifier **must** be a three-digit number padded with leading zeroes.

user

Specifies the FTP user.

password

Specifies the FTP user's password.

PATH=*path*

PATH parameter is meaningful only for type3 e-Vault. If specified for other e-Vault types, it is checked for valid syntax and is further ignored.

This parameter specifies the path in a Hierarchical File System in a non-MVS platform that stores the virtual tape data sets. If the data sets are to be stored in the Root directory, the path parameter need not be specified.

path

Specifies the full path (including slashes or backslashes) of a directory in a Hierarchical File System used to store the virtual tape dataset.

Value range: 1 to 40-character alphanumeric string (including slashes).

Note: The string is case sensitive.

Default:

None

VSID=*vsid*

If VTE operates in a VTE complex, the e-Vault processes of vault, return and remote-scratch can be directed to a specific system in the complex.

vsid

Specifies the system identifier (SID) of the system whose active VTE server should perform the actual e-Vault function. This parameter is meaningful for VTE complexes only and should not be specified for VTE servers that do not share a VTE complex.

If the VSID parameter value specifies a system identifier which is not the local system identifier, the other system notification will be delayed by the VDBACC parameter value specified on the PERFORM initialization statement. Therefore, decreasing the value of the VDBACC parameter on both servers should be considered.

No validation is performed on the specified *vsid* value. Failure to specify a valid and existing system identifier may cause the virtual tape to remain in an interim e-Vault status. Should this happen, the virtual tape should be recovered using the HVTURCV utility in order to move it back to a stable e-Vault status, i.e. 'local', 'vaulted' or 'safe'.

Value range: 1 to 4-character alphanumeric string.

Default: Local system identifier

Chapter 31. VTE Operation

This chapter explains how VTE is started and stopped, and details all its operator commands along with their parameters and output displays.

Starting and Stopping VTE

The VTE server acts as a started task that remains active from system activation through system shutdown.

The VTE Procedure

The VTE server and its offline utilities are controlled by one procedure located in the system PROCLIB concatenation.

The procedure activates either the server or the offline utilities according to a preset symbolic parameter.

The following figure illustrates the VTE procedure member in the system PROCLIB:

```
//HVT300EC PROC SERVICE=SRV,
//          SYS=00
//*
//*****
//* LICENSED MATERIALS - PROPERTY OF IBM *
//* SYSTEM STORAGE VTF MAINFRAME *
//* 5655-V43 *
//* (C) COPYRIGHT IBM CORPORATION 2000, 2008. ALL RIGHTS RESERVED.
//*
//* US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
//* DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
//*
//* PROCEDURE:
//*   HVTPROC
//*
//* FUNCTION:
//*   VTF Mainframe procedure sample
//*
//* NOTE:
//*   Arrows "<==" point to lines which are recognized as
//*   requiring customization. Parameters and keywords needing
//*   to be customized are entered in lower case to make them
//*   easier to find. All jcl must be in upper case before
//*   submitting the job to avoid jcl errors. Specific items
//*   needing customization include:
//*
//*   Change the following lowercase variables to values
//*   suitable for your installation.
//*
//*   #hvtproc - The VTF Mainframe procedure name. The
//*               recommended default would be 'HVT300'.
//*   #tgthlq - The high level qualifier for the target
//*               libraries.
//*   #vdbname - The VDB file dataset name.
//*   #jrnnam - The journal file dataset name.
//*
//* ACTIVITY:
//*
//*****
```

```
//*
//HVT300EC EXEC PGM=HVT&SERVICE,      <== PROCEDURE NAME
//          PARM='SYS=&SYS',
//          TIME=1440,
//          REGION=0M
//*
//STEPLIB DD DSN=HVTT.V300.SHVTLOAD,    <== TGT LIB HLQ
//          DISP=SHR
//*          DD DSN=SYS1.V1R3M0.SEAGLPA,  SAA REXX/370 RELEASE 3
//*          DISP=SHR
//*          DD DSN=FAN140.SEAGLPA,      REXX FOR Z/OS 1.4
```

```

//*          DISP=SHR
//*----- SYSOUT DATASETS -----
//HVTLIST DD  SYSOUT=*,          LIST INITIALIZATION STATS
//          OUTLIM=16777215
//HVTOUT DD  SYSOUT=*,          SERVER MESSAGES
//          OUTLIM=16777215
//HVTTRC DD  SYSOUT=*,          INTERNAL TRACE (TRCON)
//          OUTLIM=16777215
//HVTGTF DD  SYSOUT=*,          VIRTUAL GTF TRACE (GTFON)
//          OUTLIM=16777215
//SYSTSPRT DD SYSOUT=*,          DICT BUILD FOR TYPE3 COMPRESS
//          OUTLIM=16777215
//SYSABEND DD SYSOUT=*,
//          OUTLIM=16777215
//SYSPRINT DD SYSOUT=*,
//          OUTLIM=16777215
//*----- INITIALIZATION STATEMENTS -----
//HVT SIN DD  DSN=HVTT.V300.SHVTSAMP, <== TGT LIB HLQ
//          DISP=SHR
//HVTUTIL DD  DSN=NULLFILE

```

```

//*----- DATABASE DATASETS -----
//HVTVDB DD  DSN=HVTT.V300.VDBEC, <== VDB FILE DATASET NAME
//          DISP=SHR
//HVTJRN DD  DSN=HVTT.V300.JRNEC, <== JOURNAL FILE DATASET NAME
//          DISP=SHR
//PROIGN DD  DUMMY
//ACCIGN DD  DUMMY
//*----- SORT DATASETS -----
//SORTWK01 DD UNIT=SYSALLDA,SPACE=(TRK,(150,150)),
//          DISP=(,DELETE)
//SORTWK02 DD UNIT=SYSALLDA,SPACE=(TRK,(150,150)),
//          DISP=(,DELETE)
//SORTIN DD  UNIT=SYSALLDA,SPACE=(TRK,(180,180)),
//          DISP=(,DELETE)
//SORTOUT DD UNIT=SYSALLDA,SPACE=(TRK,(180,180)),
//          DISP=(,DELETE)
//SYSOUT DD  SYSOUT=*
//*----- WORK DATASETS -----
//SYSUT1 DD  UNIT=SYSALLDA,SPACE=(TRK,(180,180)),
//          DISP=(,DELETE)
//SYSUT2 DD  UNIT=SYSALLDA,SPACE=(TRK,(180,180)),
//          DISP=(,DELETE)
//*----- PROSMS, STOPX37, ACC -----

```

Procedure Symbolic Parameters Definitions

&SERVICE	For the VTE server, &SERVICE expands to SRV, making HVTSRV the server main program name. For utilities, it expands to the appropriate utility program name suffix.
&SYS	Defines the default HVTSYSxx member name suffix.

JCL Definitions:

STEPLIB	Defines the VTE APF authorized load library.
HVTLIST	Defines a sysout dataset on which the initialization statements (either static or refreshable) are printed. If any error is detected during the initialization statements processing, the error messages are printed on this dataset.
HVTSOUT	Defines a sysout dataset on which messages showing the VTE server or utilities process are printed.
HVTTRC	Defines a sysout dataset on which trace messages are printed (only when the trace facility is turned on).
HVTGTF	Defines a sysout dataset on which GTF-like messages are printed (only when the GTF facility is turned on).

SYSTSPRT	Defines a sysout dataset on which dictionary building messages are printed (only for type3 compress).
HVTSIN	Defines a partitioned dataset in which the members that contain the initialization statements are located. The partitioned dataset is a fixed block dataset with LRECL=80.
HVTUTIL	Defines an optional sequential (or instream) dataset that contains utility initialization statements. This dataset is a fixed block dataset with LRECL=80.
HVTVDB	Defines the VTE virtual tape database.
HVTJRN	Defines the VTE journal file.
SORTWK01	Defines the sort utility work dataset.
SORTWK02	Defines the sort utility work dataset.
SORTIN	Defines the sort utility input dataset.
SORTOUT	Defines the sort utility output dataset.
SYSUT1	Defines work dataset.
SYSUT2	Defines work dataset.
SYSOUT	Defines the sort utility sysout dataset.
PROIGN	Indicates that PROSMS and STOPx37 products should ignore VTE.
ACCIGN	Indicates that ACC and SRS products should ignore VTE.

Starting VTE

The VTE server is started with a START operator command from the system console.

```
S server[,SYS=suffix]
```

server The member name of the VTE procedure in the system PROCLIB.

suffix The suffix of the HVTSYSxx member name in the VTE SAMPLIB library.

```
S HVT300EC
$HASP100 HVT300EC ON STCINRDR
IEF695I START HVT300EC WITH JOBNAME HVT300EC IS ASSIGNED TO USER STCUSER
, GROUP STCGRP
$HASP373 HVT300EC STARTED
IEF403I HVT300EC - STARTED - TIME=17.18.34
HVT421I USING MEMBER HVTSYSEC FROM HVTT.V300.SAMPLIB
HVT421I USING MEMBER HVTHVTEC FROM HVTT.V300.SAMPLIB
HVT421I USING MEMBER HVTMCSEC FROM HVTT.V300.SAMPLIB
HVT421I USING MEMBER HVTVLTEC FROM HVTT.V300.SAMPLIB
HVT001I VTF Mainframe VER 3.0.0 SERVER 0_RDIREC INITIALIZATION STARTED
HVT266I ENF      MANAGER ADD      WAS OK, RC=00 (X'88677058')
HVT007I 0_RDIREC SERVER OBTAINED X'005AC0' BYTES AT X'01A9B540' (SP245)
HVT266I PURGER  MANAGER ADD      WAS OK, RC=00 (X'886765A0')
HVT008I 0_RDIREC SERVER INTERFACE ESTABLISHED AT X'80BE7360'
HVT354I 0_RDIREC SERVER VDB-IN-STORAGE UPDATE COMPLETED
HVT400I SERVER 0_RDIREC TCP/IP ACCEPTING CONNECTIONS ON PORT=06160
IEA630I OPERATOR EC16JJ02 NOW ACTIVE,  SYSTEM=JJ02      , LU=0_RDIREC
HVT421I USING MEMBER HVTPOLEC FROM HVTT.V300.SAMPLIB
HVT304I POL      REFRESH WAS OK, RC=00
HVT421I USING MEMBER HVTTTRLEC FROM HVTT.V300.SAMPLIB
HVT304I TRL      REFRESH WAS OK, RC=00
HVT301I SUBSYSTEM EC16 SSVT -> X'00BFD1A8', EXECUTOR -> X'88669890'
SETCON MN,JOBNAMES=(ON,LOG),T=ON
D OPDATA,MONITOR - INTERNALLY ISSUED SETCON MONITOR RESPONSE.
HVT260I IEFDB401      EXIT ADD      WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 ACTIVATION  WAS OK, RC=00
HVT019I SSN IS NOW ENABLED
CNZ1100I 19.38.40 MONITOR DISPLAY 301
SPACE=OFF  DSNAME=ON  TIMESTAMP=ON
```

```
MSGTYPE  SETCON MN  NUMBER OF RECEIVERS
JOBNAME  ON,LOG      1 CONSOLE
SESS     OFF         0 CONSOLES
STATUS   OFF         0 CONSOLES
```

```
HVT421I USING MEMBER HVTRULEC FROM HVTT.V300.SAMPLIB
HVT304I RUL REFRESH WAS OK, RC=00
HVT019I DYN IS NOW ENABLED
HVT421I USING MEMBER HVTPRLEC FROM HVTT.V300.SAMPLIB
HVT304I PRL REFRESH WAS OK, RC=00
HVT019I PAT IS NOW ENABLED
HVT039I HW: TYPE=2086,MODEL=410 ,SERIAL=0AE43B SW: NAME=z/OS ,VERSION=01.07.00
HVT037I 0_RDIREC SERVER: CUU=0570,ES=VT90EC ,GN=VT3490G2,DT=X'45048081',CHP=(FB)
HVT009I INITIALIZATION OF 0_RDIREC SERVER SUCCESSFULLY COMPLETED
V 0570-057F,ONLINE
D U,,,0570,00016,L=Z VARY RANGE DISPLAY
IEE457I 17.40.07 UNIT STATUS 924
UNIT TYPE STATUS VOLSER VOLSTATE
0570 VT34 0-NRD /REMOV
0571 VT34 0-NRD /REMOV
0572 VT34 0-NRD /REMOV
0573 VT34 0-NRD /REMOV
0574 VT34 0-NRD /REMOV
0575 VT34 0-NRD /REMOV
0576 VT34 0-NRD /REMOV
0577 VT34 0-NRD /REMOV
0578 VT34 0-NRD /REMOV
0579 VT34 0-NRD /REMOV
057A VT34 0-NRD /REMOV
057B VT34 0-NRD /REMOV
057C VT34 0-NRD /REMOV
057D VT34 0-NRD /REMOV
057E VT34 0-NRD /REMOV
057F VT34 0-NRD /REMOV
HVT015I 0_RDIREC SERVER DISKPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
F HVT300EC.HVT300EC,D=DSP
HVT432I ES=SYSALLDA VOL#= 129 TRKS= 9758_K USED= 39% DP00L=D_SYSALLDA
HVT432I ES=VTSWIZ VOL#= 15 TRKS= 583_K USED= 37% DP00L=D_VTSWIZ
HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DP00L=D_DORI
HVT432I ES=VTSQVP VOL#= 7 TRKS= 272_K USED= 42% DP00L=D_VTSQVP
HVT432I ES=VTSQA VOL#= 19 TRKS= 929_K USED= 40% DP00L=D_VTSQA
HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DP00L=D_VTSDSMS
HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DP00L=D_TEST
HVT432I ES=VAULT VOL#= 8 TRKS= 391_K USED= 29% DP00L=D_VAULT
HVT432I ES=VAULT VOL#= 8 TRKS= 391_K USED= 29% LOCID=CLN4
HVT432I SG=SGWIZ VOL#= 2 TRKS= 97_K USED= 40% LOCID=CLN5
HVT015I 0_RDIREC SERVER TAPEPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
F HVT300EC.HVT300EC,D=TSP
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_SYSALLDA
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_VTSWIZ
HVT433I TAPES= 1 ACT= 0( 0%) SCR= 1 TP00L=T_VTSQVP
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_VTSQA
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_VTSDSMS
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_DORI
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TP00L=T_VAULT
HVT433I TAPES= 55 ACT= 30( 54%) SCR= 25 TP00L=$$$FLT
HVT434I TOTAL: TAPES= 56 ACT= 30( 53%) SCR= 26
HVT354I 0_RDIREC SERVER VDB-IN-STORAGE UPDATE COMPLETED
```

Stopping VTE

The VTE server may be stopped while no jobs are using any of the VTE virtual devices or unconditionally while virtual tapes are being processed.

Normal Shutdown of VTE

The VTE server is normally shut down with a STOP operator command from the system console.

```
P server
```

The STOP command ensures that there are no jobs using the VTE virtual devices before shutting down the server. 'V cuu,OFFLINE' operator commands are issued to ensure that no new jobs are started while the server is shutting down. If jobs are using VTE virtual devices, the shutdown will pend until the job steps that allocated to the VTE virtual devices complete.

If the VTE virtual devices remain in a pending offline status while no jobs are using them, start the DEALLOC procedure by issuing the 'S DEALLOC' operator command. This will awake the MVS allocation mechanism. Once awoken, it will take all pending offline devices to offline status. As all VTE virtual devices are offline, the server shuts down.

```
P HVT300EC
HVT343W 0_RDIREC SERVER STOP PENDING. ENTER 'S DEALLOC' OPERATOR COMMAN
TO LET MVS VARY DEVICES OFFLINE
V 0570-057F,OFFLINE
D U,,,0570,00016,L=Z VARY RANGE DISPLAY
IEE457I 17.46.26 UNIT STATUS 962
UNIT TYPE STATUS VOLSER VOLSTATE
0570 VT34 F-NRD /REMOV
0571 VT34 F-NRD /REMOV
0572 VT34 F-NRD /REMOV
0573 VT34 F-NRD /REMOV
0574 VT34 F-NRD /REMOV
0575 VT34 F-NRD /REMOV
0576 VT34 F-NRD /REMOV
0577 VT34 F-NRD /REMOV
0578 VT34 F-NRD /REMOV
0579 VT34 F-NRD /REMOV
057A VT34 F-NRD /REMOV
057B VT34 F-NRD /REMOV
057C VT34 F-NRD /REMOV
057D VT34 F-NRD /REMOV
057E VT34 F-NRD /REMOV
057F VT34 F-NRD /REMOV
HVT013W 0_RDIREC SERVER SHUTTING DOWN
HVT266I PURGER MANAGER DELETE WAS OK, RC=00 (X'886765A0')
HVT260I IEFDB401 EXIT DELETE WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 DEACTIVATION WAS OK, RC=00
HVT025I 0_RDIREC SERVER MCS CONSOLE TERMINATING
IEA631I OPERATOR EC16JJ02 NOW INACTIVE, SYSTEM=JJ02 , LU=0_RDIREC
HVT266I ENF MANAGER DELETE WAS OK, RC=00 (X'88677058')
IEF404I HVT300EC - ENDED - TIME=17.46.35
$HASP395 HVT300EC ENDED
```

Unconditional Shutdown of VTE

VTE server can be shut down unconditionally with a MODIFY operator command from the system console.

```
F server,FORCE
```

When VTE is shut down unconditionally, it does not vary its devices offline, nor it waits for the running jobs to complete. Executing I/Os are completed with IOE status (I/O error). This option should be used with care since it leaves all currently running virtual tapes in an unpredictable status.

```
F HVT300EC,FORCE
HVT344I 0_RDIREC SERVER FORCE STOP ACCEPTED
HVT013W 0_RDIREC SERVER SHUTTING DOWN
HVT266I PURGER MANAGER DELETE WAS OK, RC=00 (X'886765A0')
HVT260I IEFDB401 EXIT DELETE WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 DEACTIVATION WAS OK, RC=00
BPXP018I THREAD 0A53A250000000001, IN PROCESS 16777231, ENDED 065
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000, AND REASON CODE 00000000.
HVT025I 0_RDIREC SERVER MCS CONSOLE TERMINATING
HVT266I ENF MANAGER DELETE WAS OK, RC=00 (X'88677058')
IEE501I CONSOLE EC16JJ02 FAILED, REASON=ABTERM. ALL ALTERNATES
UNAVAILABLE, CONSOLE IS NOT SWITCHED
IEA631I OPERATOR EC16JJ02 NOW INACTIVE, SYSTEM=JJ02 , LU=0_RDIREC
IEF404I HVT300EC - ENDED - TIME=17.49.52
$HASP395 HVT300EC ENDED
```

Cancelling VTE

VTE server can be cancelled with a CANCEL operator command from the system console. When cancelling the VTE server, it follows the same path as when it is shut down unconditionally. This option should be used with care since it leaves all currently running virtual tapes in an unpredictable status.

```
C HVT300EC
IEA989I SLIP TRAP ID=X222 MATCHED. JOBNAME=HVT300EC, ASID=003C.
```

```

HVT012W 0_RDIREC SERVER CAPTURED A S222-U0000 ABEND, CLEANUP IN PROGRESS
HVT013W 0_RDIREC SERVER SHUTTING DOWN
IEE301I HVT300EC CANCEL COMMAND ACCEPTED
HVT266I PURGER MANAGER DELETE WAS OK, RC=00 (X'886765A0')
HVT260I IEFDB401 EXIT DELETE WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 DEACTIVATION WAS OK, RC=00
HVT266I ENF MANAGER DELETE WAS OK, RC=00 (X'88677058')
IEA989I SLIP TRAP ID=X13E MATCHED. JOBNAME=HVT300EC, ASID=003C.
BPXP018I THREAD 0A53A25000000001, IN PROCESS 50331663, ENDED 159
WITHOUT BEING UNDOUBBED WITH COMPLETION CODE 00222000, AND REASON CODE 00000000.
BPXP018I THREAD 0A53934000000000, IN PROCESS 50331663, ENDED 160
WITHOUT BEING UNDOUBBED WITH COMPLETION CODE 40222000, AND REASON CODE 00000000.
IEE501I CONSOLE EC16JJ02 FAILED, REASON=ABTERM. ALL ALTERNATES
UNAVAILABLE, CONSOLE IS NOT SWITCHED
IEA631I OPERATOR EC16JJ02 NOW INACTIVE, SYSTEM=JJ02, LU=0_RDIREC
IEF450I HVT300EC HVT300EC - ABEND=S222 U0000 REASON=00000000 163
TIME=17.56.19
$HASP395 HVT300EC ENDED

```



Attention:

Under no circumstances should the server be MVS FORCE-d. Such a FORCE will leave the I/O interface in an unstable state and may lead to an IPL.

Operator Commands

VTE provides a series of operator commands that let the system operator enable/disable features, refresh selected server parameters, set/reset traces, display activity and resources, and minimal server tuning. The commands are entered as started task MODIFY operator commands.

The following table summarizes the available VTE operator commands:

Category	Command	Description
Enable/Disable selected features	{DLGON DLGOFF}	Turning On/Off the VTE Dialog option
	{DYRON DYROFF}	Turning On/Off the dynamic reallocation
	{PATON PATOFF}	Turning On/Off PAT (batch and dynamic allocations)
	{PTBON PTBOFF}	Turning On/Off PAT (batch allocations only)
	{PTDON PTDOFF}	Turning On/Off PAT (dynamic allocations only)
	{RPLON RPLOFF}	Turning On/Off the replication feature status
	{SSNON SSNOFF}	Turning On/Off the VTE subsystem
	{VLTSUS=Y N}	Suspend/Resume e-Vault processing activity
	{ZIIPON ZIIPOFF}	Turning On/Off the internal ZIIP support

Category	Command	Description
Refresh selected server parameters	DLG=member suffix	Refresh VTE Dialog definitions
	POL=member suffix	Refresh definitions for diskpools and tapepools
	PRL=member suffix	Refresh PAT rules
	RUL=member suffix	Refresh dynamic reallocation rules
	TRL=member suffix	Refresh tapepool rules
Trace VTE operation	{ALCON= <i>jobname</i> ALCOFF}	Dynamic allocation trace
	{ALEON ALEOFF}	Access list entries trace
	{CCPON CCPOFF}	CCW summary trace
	{GTFON GTFOFF}	GTF-like trace
	{TCPON TCPOFF}	TCP/IP flow trace
	{TRCON TRCOFF}	Internal flow trace
	{VLTON VLTOFF}	e-Vault virtual tape status change trace
	{WTOON= <i>jobname</i> WTOOFF}	Dynamic reallocation decision trace
Display activity and resources	DISPLAY=A	Display virtual devices activity
	DISPLAY=DSP	Display diskpools utilization
	DISPLAY=TSP	Display tapepools utilization
Server tuning	CPU%= <i>nn</i>	Limit server CPU utilization
	VLRET= <i>nnnn</i>	Interval at which e-Vault processes are retried
	ZIIP%= <i>nn</i>	Percent of zIIP-eligible work to be offloaded
Server management	KILL,{CUU= <i>cuu</i> TCB= <i>tcb</i> }	Terminate an I/O subtask and free its resources

Enable/Disable Selected Features

Depending on your environment, you may want to enable or disable some features, such as the VTE subsystem and the Parallel Access Tape (PAT) feature. The methods for doing that vary depending upon whether batch or dynamic allocation/reallocation is affected.

Turning On/Off the VTE subsystem

The VTE subsystem can be activated/deactivated by turning it on and off.

When the VTE subsystem is turned off, dynamic reallocation, PAT, and type3 and type4 e-Vault are disabled. When the VTE subsystem is turned on, the dynamic reallocation and PAT are not enabled and they need to be explicitly enabled.

Even if the VTE subsystem was disabled on server startup, it can be turned on at anytime.

The VTE subsystem can be turned on and off by issuing the following operator commands on the system console:

```
F server,{SSNON|SSNOFF}
```

When you issue the SSNON or SSNOFF command, several messages are displayed on the screen. The example below shows the messages that display when you issue these commands.

```
F HVT300EC,SSNON
HVT301I SUBSYSTEM EC16 SSVT -> X'00BFD1A8', EXECUTOR -> X'88664890'
SETCON MN,JOBNAMES=(ON,LOG),T=ON
HVT260I IEFDB401          EXIT ADD      WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 ACTIVATION WAS OK, RC=00
D OPDATA,MONITOR - INTERNALLY ISSUED SETCON MONITOR RESPONSE.
HVT019I SSN IS NOW ENABLED
CNZ1100I 19.39.24 MONITOR DISPLAY 343
SPACE=OFF DSNNAME=ON  TIMESTAMP=ON
MSGTYPE   SETCON MN  NUMBER OF RECEIVERS
JOBNAMES   ON,LOG           1 CONSOLE
SESS       OFF              0 CONSOLES
STATUS     OFF              0 CONSOLES
.
.
.
F HVT300EC,SSNOFF
HVT260I IEFDB401          EXIT DELETE WAS OK, RC=X'00', RS=X'0000'
HVT303I SUBSYSTEM EC16 DEACTIVATION WAS OK, RC=00
HVT019I SSN IS NOW DISABLED
HVT019I DYN IS NOW DISABLED
HVT019I PAT IS NOW DISABLED
```

Turning On/Off the VTE Dialog option

The VTE Dialog can be enabled/disabled by turning it on and off. When the VTE Dialog is enabled, the VTE Dialog definitions in member HVTDLGxx are automatically refreshed.

The HVTDLGxx member used to read the definitions is the one that was used on the last definitions refresh. However, if the server is bounced, the definitions member (HVTDLGxx) to be used is the one pointed to by the appropriate HVTSYSxx member.

If the VTE Dialog was disabled on the server startup, it can be turned on at any time. The VTE Dialog option can be turned on and off by issuing the following operator command on the system console:

```
F server,{DLGON|DLGOFF}
```

An example is shown below:

```
+-----+
| F HVT300EC,DLGON
| HVT421I USING MEMBER HVTDLGEC FROM HVTT.V300.SHVTSAMP
| HVT304I DLG REFRESH WAS OK, RC=00
| HVT019I DLG IS NOW ENABLED
| .
| F HVT300EC,DLGOFF
| HVT019I DLG IS NOW DISABLED
+-----+
```

Turning On/Off the VTE dynamic reallocation

VTE dynamic reallocation can be enabled/disabled by turning it on and off.

When dynamic reallocation is enabled, the dynamic reallocation rules are automatically refreshed. The HVTRULxx member used to read the dynamic reallocation rules is the one that was used on the last dynamic reallocation rules refresh. However, if the server is bounced, the dynamic reallocation rules member (HVTRULxx) to be used is the one pointed to by the appropriate HVTSYSxx member.

In order to turn on the VTE dynamic reallocation, the VTE subsystem must be already enabled.

Even if the dynamic reallocation was disabled on server startup, it can be turned on at anytime, provided the VTE subsystem is active.

Dynamic reallocation can be turned on and off by issuing the following operator command on the system console:

```
F server,{DYRON|DYROFF}
```

When you issue the DRYON or DRYOFF command, several messages are displayed on the screen. The example below shows the messages that display when you issue these commands.

```
F HVT300EC,DYRON
HVT421I USING MEMBER HVTRULEC FROM HVTT.V300.SAMPLIB
HVT304I RUL REFRESH WAS OK, RC=00
HVT019I Dyr IS NOW ENABLED
.
.
.
F HVT300EC,DYROFF
HVT019I Dyr IS NOW DISABLED
```

Turning On/Off PAT (batch and dynamic allocations)

VTE PAT (for batch and dynamic allocations) can be enabled/disabled by turning it on and off.

When PAT (for batch and dynamic allocations) is enabled, the PAT rules are automatically refreshed. The HVTPRLxx member used to read the PAT rules is the one that was used on the last PAT rules refresh. However, if the server is bounced, the PAT rules member (HVTPRLxx) to be used is the one pointed to by the appropriate HVTSYSxx member.

In order to turn on the VTE PAT (for batch and dynamic allocations), the VTE subsystem must be already enabled.

Even if the PAT (for batch and dynamic allocations) was disabled on server startup, it can be turned on at anytime, provided the VTE subsystem is active.

PAT (for batch and dynamic allocations) can be turned on and off by issuing the following operator command on the system console:

```
F server,{PATON|PATOFF}
```

When you issue the PATON or PATOFF command, several messages are displayed on the screen. The example below shows the messages that display when you issue these commands.

```
F HVT300EC,PATON
HVT421I USING MEMBER HVTPRLEC FROM HVTT.V300.SAMPLIB
HVT304I PRL REFRESH WAS OK, RC=00
HVT019I PAT IS NOW ENABLED
.
.
.
F HVT300EC,PATOFF
HVT019I PAT IS NOW DISABLED
```

Turning On/Off PAT (batch allocations only)

VTE PAT (for batch allocations) can be enabled/disabled by turning it on and off. Turning it on or off does not influence PAT for dynamic allocations.

When PAT (for batch allocations) is enabled, the PAT rules are automatically refreshed. The HVTPRLxx member used to read the PAT rules is the one that was used on the last PAT rules refresh. However, if the server is bounced, the PAT rules member (HVTPRLxx) to be used is the one pointed to by the appropriate HVTSYSxx member.

In order to turn on the VTE PAT (for batch allocations), the VTE subsystem must be already enabled.

Even if the PAT (for batch allocations) was disabled on server startup, it can be turned on at anytime, provided the VTE subsystem is active.

PAT (for batch allocations) can be turned on and off by issuing the following operator command on the system console:

```
F server,{PTBON|PTBOFF}
```

When you issue the PTBON or PTBOFF command, several messages are displayed on the screen. The example below shows the messages that display when you issue these commands.

```
F HVT300EC,PTBON
HVT421I USING MEMBER HVTPRLEC FROM HVTT.V300.SAMPLIB
HVT304I PRL REFRESH WAS OK, RC=00
HVT019I PTB IS NOW ENABLED
.
.
.
F HVT300EC,PTBOFF
HVT019I PTB IS NOW DISABLED
```

Turning On/Off PAT (dynamic allocations only)

VTE PAT (for dynamic allocations) can be enabled/disabled by turning it on and off. Turning it on or off does not influence PAT for batch allocations.

When PAT (for dynamic allocations) is enabled, the PAT rules are automatically refreshed. The HVTPRLxx member used to read the PAT rules is the one that was used on the last PAT rules refresh. However, if the server is bounced, the PAT rules member (HVTPRLxx) to be used is the one pointed to by the appropriate HVTSYSxx member.

In order to turn on the VTE PAT (for dynamic allocations), the VTE subsystem must be already enabled.

Even if the PAT (for dynamic allocations) was disabled on server startup, it can be turned on at anytime, provided the VTE subsystem is active.

PAT (for dynamic allocations) can be turned on and off by issuing the following operator command on the system console:

```
F server,{PTDON|PTDOFF}
```

When you issue the PTDON or PTDOFF command, several messages are displayed on the screen. The example below shows the messages that display when you issue these commands.

```
F HVT300EC,PTDON
HVT421I USING MEMBER HVTPRLEC FROM HVTT.V300.SAMPLIB
HVT304I PRL REFRESH WAS OK, RC=00
HVT019I PTD IS NOW ENABLED
.
.
.
F HVT300EC,PTDOFF
HVT019I PTD IS NOW DISABLED
```

Turning On/Off the replication feature status

The replication feature status can be enabled/disabled by turning it on and off.

- When the replication feature status is turned on, the replication process starts to replicate virtual tapes pending for replication.
- When the replication feature status is turned off, the replication process stops to select new virtual tapes pending for replication.

Virtual tapes being replicated continue the process till full completion.

If the VLT option is disabled, the replication feature status cannot be turned on and will remain disabled. The replication feature status can be turned on and off by issuing the following operator command on the system console:

```
F server,{RPLON|RPOFF}
```

The resulting messages are shown below:

```
+-----+
| F HVT300EC,RPLON
| HVT019I RPL   IS NOW ENABLED
|
| F HVT300EC,RPOFF
| HVT019I RPL   IS NOW DISABLED
+-----+
```

Suspend or Resume e-Vault processing activity

The e-Vault processes activity can be suspended and later resumed.

The following actions are taken when the e-Vault processing activity is suspended:

- Currently active e-Vault processes continue normally.
- New e-Vault processes are not started and e-Vault eligible virtual tapes remain in e-Vault eligible status.
- The retry count is stopped. It will resume from the suspended value once the e-Vault processes activity is resumed.

The following actions are taken when the e-Vault processes activity is resumed:

- The e-Vault processes are started and e-Vault eligible virtual tapes are processed as needed.
- Retry of previously failed attempts to e-Vault virtual tapes is resumed.

This operator command is valid only if e-Vault is enabled (VLT=ENABLE in HVTOPTxx member).

The e-Vault processing activity of e-Vault eligible virtual tapes can be suspended and resumed by issuing the following operator command on the system console:

```
F server,VLTSUS={Y|N}
```

The resulting messages are shown below:

```
+-----+
| F HVT300EC,VLTSUS=Y
| HVT030I VLTSUS   VAULT OPTION SET TO Y
|
|
| F HVT300EC,VLTSUS=N
| HVT030I VLTSUS   VAULT OPTION SET TO N
+-----+
```

Setting the interval at which e-Vault processes are retried

The frequency by which VTE retries to e-Vault failing e-Vaulted virtual tapes can be modified by changing the interval between retries.

The following actions are taken when the e-Vault retries interval is changed:

1. Waiting retries whose new interval has not yet expired will continue to wait for the new interval expiration.
2. Waiting retries whose new interval has expired will retry as soon as other e-Vault retry conditions are met.
3. The interval is changed even though the e-Vault processes are suspended.

This operator command is valid only if e-Vault is enabled (VLT=ENABLE in HVTOPTxx member).

The interval at which VTE retries failing attempts to e-Vault virtual tapes can be modified by issuing the following operator command on the system console:

```
F server, VLTRET=nnnn
```

where *nnnn* is an integer from 1 to 1440.

When you issue this command, you will see this result:

```
+-----+
| F HVT300EC, VLTRET=60          |
| HVT030I VLTRET  VAULT OPTION SET TO 60 |
+-----+
```

Turning On/Off the internal zIIP support

The internal zIIP support can be enabled/disabled by turning it on and off. The following actions are taken when the internal zIIP support is turned on:

- If the internal zIIP support is already enabled, no action occurs.
- If the internal zIIP support is disabled, the server checks whether the operating system level supports zIIP and the host has at least one online zIIP processor.
 - If positive, the internal zIIP support is enabled.
 - Otherwise, it remains disabled.

In both cases HVT044I message is issued detailing the internal zIIP support status.

The following section describes how the VTE server workload is affected by turning the internal zIIP support on and off:

- A virtual tape zIIP-eligibility is set at mount time only.
- When the internal zIIP support is enabled, it affects only virtual tapes mounted after the internal zIIP support is enabled.
- Exceptions to that rule are virtual tapes that were already mounted while the internal zIIP support was enabled and then became zIIP-eligible. For these virtual tapes, the setting of the internal zIIP support on and off is effective immediately.

Even if the internal zIIP support was disabled on server startup, it can be turned on at anytime (provided the operating system supports zIIP and the host has at least one online zIIP processor). The internal zIIP support can be turned on and off by issuing the following operator command on the system console:

```
F server, {ZIIPON|ZIIPOFF}
```

An example is shown below.

```
+-----+
| F HVT300EC, ZIIPON              |
| HVT044I ZIIP=( SUPPORTED, ENABLED ), |
|           O_CPU= 4, Z_CPU= 1, ZIIP%=100 |
| HVT019I ZIIP  IS NOW ENABLED        |
| .                                     |
| .                                     |
| F HVT300EC, ZIIPOFF             |
| HVT019I ZIIP  IS NOW DISABLED       |
+-----+
```

Setting the percentage of zIIP-eligible work to be offloaded

The MVS dispatcher can dispatch all or only a part of the zIIP-eligible work to zIIP processor(s). The amount of work to be actually offloaded to zIIP processor(s) can be dynamically changed.

The following actions are taken when the percent of zIIP-eligible work to be offloaded is changed:

- If the zIIP support is enabled and the host has at least one online zIIP processor, the percent of zIIP-eligible work to be offloaded is changed and the HVT029I message displays the updated value.
- If the zIIP support is enabled due to the PROJECTCPU=YES specification, but the host does not have any online zIIP processors, the following occurs:
 1. The percent of zIIP-eligible work to be offloaded is unchanged (100% of the work is directed to CP and reported as zIIP on CP).
 2. The HVT029I message displays the old values.
- If the zIIP support is disabled, the percent of zIIP-eligible work to be offloaded is unchanged and the HVT029I message displays the old values.

The VTF server workload is affected by changing the amount of work to be offloaded to zIIP processor(s). This is a percent amount that is set at mount time. Changing the amount of work to be offloaded to zIIP processor(s) affects only virtual tapes mounted after the new percent was specified.

The percent of zIIP-eligible work to be offloaded can be changed by issuing the following operator command on the system console:

```
F server,ZIIP%=percent
```

Where *percent* is an integer from 1 to 100.

Refresh Selected Server Parameters

Certain VTE parameters can be refreshed without having to bounce the VTE server.

The following parameters can be dynamically refreshed:

- Tapepools and diskpools definitions
- Dynamic reallocation rules
- PAT rules
- Tapepool rules.

Refresh Tapepools and Diskpools Definitions

The tapepools and diskpools definitions can be refreshed by performing the following steps:

- Update the HVTPOLxx member currently in use or create a new HVTPOLxx member with valid tapepool and diskpool definitions.
- Type the following operator command on the system console:

```
F server,POL=xx
```

xx The HVTPOLxx member suffix that contains the valid tapepool and diskpool definitions.

After the command completes, the new tapepool and diskpool definitions become effective.

```
F HVT300EC,POL=EC
HVT421I USING MEMBER HVTPOLEC FROM HVTT.V300.SAMPLIB
HVT304I POL      REFRESH WAS OK, RC=00
HVT015I 0_RDIREC SERVER TAPEPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
HVT015I 0_RDIREC SERVER DISKPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
```

Refresh Dynamic Reallocation Rules

The dynamic reallocation rules can be refreshed by performing the following steps:

- Update the HVTRULxx member currently in use or create a new HVTRULxx member with valid dynamic reallocation rules.
- Type the following operator command on the system console:

```
F server,RUL=xx
```

xx The suffix of the HVTRULxx member that contains the valid dynamic reallocation rules.

After the command completes, the new dynamic reallocation rules become effective.

```
F HVT300EC,RUL=EC
HVT421I USING MEMBER HVTRULEC FROM HVTT.V300.SAMPLIB
HVT304I RUL        REFRESH WAS OK, RC=00
```

Refresh PAT Rules

The PAT rules definitions can be refreshed by performing the following steps:

- Update the HVTPLxx member currently in use or create a new HVTPLxx member with valid PAT rules definitions.
- Type the following operator command on the system console:

```
F server,PRL=xx
```

xx The HVTPLxx member suffix that contains the valid PAT rules definitions.

After the command completes, the new PAT rules definitions become effective.

```
F HVT300EC,PRL=EC
HVT421I USING MEMBER HVTPLxx FROM HVTT.V300.SAMPLIB
HVT304I PRL        REFRESH WAS OK, RC=00
```

Refresh Tapepool Rules

The tapepool rules definitions can be refreshed by performing the following steps:

- Update the HVTTRLxx member currently in use or create a new HVTTRLxx member with valid tapepool rules definitions.
- Type the following operator command on the system console:

```
F server,TRL=xx
```

xx The HVTTRLxx member suffix that contains the valid tapepool rules definitions.

After the command completes, the new tapepool rules definitions become effective.

```
F HVT300EC,TRL=EC
HVT421I USING MEMBER HVTTRLxx FROM HVTT.V300.SAMPLIB
HVT304I TRL        REFRESH WAS OK, RC=00
```

Refresh VTE Dialog definitions

The VTE Dialog definitions can be refreshed by performing the following steps:

1. Update the HVTDLGxx member currently in use or create a new HVTDLGxx member with valid VTE Dialog definitions.
2. Type the following operator command on the system console:

```
F server,DLG=xx
```

Where xx is the suffix of the HVTDLGxx member that contains the valid definitions.

After the command completes, the new definitions become effective.

```
+-----+
| F HVT300EC,DLG=EC
| HVT421I USING MEMBER HVTDLGEC FROM HVTT.V300.SHVTSAMP
| HVT304I DLG REFRESH WAS OK, RC=00
+-----+
```

Trace VTE Operation

VTE contains a number of built-in trace facilities that enable the operator or IBM Customer Support to diagnose problems should they occur. In addition, several trace types enable the VTE administrator to set and test its statements (or rules).

Internal Flow Trace

The internal flow trace enables to follow the path the server followed as it processed a given task. Before turning on the internal flow trace (TRCON), the zIIP support should be turned off. Failing to do so may cause a VTF server S0F8 abend.

```
F server,ZIIP0FF
F server,TRCON
```

After the trace is obtained, turn off the internal flow trace (TRCOFF) and turn on the zIIP support.

```
F server,TRCOFF
F server,ZIIPON
```

The internal flow trace can be turned on and off by typing the following operator commands on the system console:

```
F server,{TRCON|TRCOFF}
```

The trace is printed on the dataset pointed to by HVTTRC DDname.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

GTF-Like Trace

The GTF-like trace shows the CCWs along with the data and channel status of the I/Os processed by the server.

The GTF-like trace can be turned on and off by typing the following operator command on the system console:

```
F server,{GTFON|GTOFF}
```

The trace is printed on the dataset pointed to by HVTGTF DDname.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

Note that since the VTE devices are virtual devices, the standard IBM GTF trace cannot trace them.

CCW Summary Trace

The CCW summary trace shows the CCWs processed by the server along with the originating ASID and TCB.

The CCW summary trace can be turned on and off by typing the following operator command on the system console:

```
F server,{CCPON|CCPOFF}
```

The trace is displayed on the system console only.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

Access List Entries Trace

The Access List entries trace shows the VTE server added and deleted ALETs (Access List Entry Token) along with the STOKENs (Space Token) and the connected ASIDs.

The Access List entries trace can be turned on and off by typing the following operator command on the system console:

```
F server,{ALEON|ALEOFF}
```

The trace is displayed on the system console only.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

TCP/IP Flow Trace

The TCP/IP flow trace shows the data flow between the VTE TCP/IP server and its currently connected VTE TCP/IP client(s). The data flow exists only when the TCP/IP option is enabled and at least one type4 e-Vault location is defined in the VTE initialization statements.

The TCP/IP flow trace can be turned on and off by typing the following operator command on the system console:

```
F server,{TCPON|TCPOFF}
```

The trace is displayed on the system console only.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

e-Vault Virtual Tape Status Change Trace

The e-Vault virtual tape status change trace shows the various statuses an e-Vaulted virtual tape passes from the time the e-Vault process starts until it ends. In order to get a full picture of the process, the trace should be turned on on both local and remote off-site storage locations. The status symbol meaning is similar to the one described in the HVTUREP utility's REP001 report under the 'VD' heading. (See [“HVTUREP – Database Reports” on page 326](#) for details.)

The e-Vault virtual tapes status change trace can be turned on and off by typing the following operator command on the system console:

```
F server,{VLTON|VLTOFF}
```

The trace is displayed on the system console only.

This trace imposes a great overhead and should be used only as directed by IBM Customer Support.

Dynamic Reallocation Decision Trace

The dynamic reallocation decision trace shows the detailed status of each DD statement in the job prior to the VTE dynamic reallocation decision.

For more information about the display, see the explanation for message HVT995I in the *IBM Virtual Tape Emulation User Guide*.

The dynamic reallocation decision trace can be turned on and off by typing the following operator command on the system console:

```
F server,{WTOON=jobname|WTOFF}
```

The jobname value may contain mask characters as characters place holders. Each job that matches the requested jobname is traced. To trace other jobnames, reissue the operator command with a new jobname value.

The trace is displayed on the system console only, accompanied by message HVT995I.

This trace can be used in order to check why the server's dynamic reallocation decisions are not as expected.

Dynamic Allocation Trace

The dynamic allocation trace shows selected options used by a certain dynamic allocation or deallocation request. The dynamic allocation trace can be turned on and off by typing the following operator command on the system console:

```
F server,{ALCON=jobname|ALCOFF}
```

The jobname value may contain mask characters as character place holders. Each job that matches the requested jobname is traced. To trace other jobnames, reissue the operator command with a new jobname value.

The trace is displayed on the system console only.

Display Activity and Resources

VTE monitors and displays upon request its resources utilization.

According to this, the site automation product can be set to take appropriate actions.

The following resources are monitored:

- Virtual devices activity
- Diskpools utilization
- Tape pools utilization.

Display Virtual Devices Activity

The VTE virtual device activity, and the virtual tapes that are mounted, can be displayed using the following command:

```
F server, {D|DISPLAY}={A|ACTIVE}
```

After typing the above operator command on the system console, message HVT430I is issued for every VTE virtual device on which virtual tapes are currently mounted. The message details the current VTE virtual device status, as well as various counters associated with jobs that use them.

```
F HVT300EC,D=A
HVT430I UNIT VOLUME(ALIAS ) JOBNAME /JOBID R/W KB_IN KB_OUT
HVT430I 0570 EC0193(EC0193) M20INIT4/JOB00140 W 00000000 00007983
HVT430I 0574 EC0192(EC0192) M20INIT3/JOB00139 R 00014764 00000000
HVT430I 057B EC0191(EC0191) M20INIT2/JOB00138 W 00000000 00009217
```

Display Diskpool Utilization

VTE monitors the number of eligible disks, number of tracks, and percentage of used space on Esoteric names or Storage Groups associated with diskpools defined in the HVTPOLxx member.

The following operator command can be used to display this data:

```
F server,{D|DISPLAY}={DSP|DSPACE}
```

After typing the above operator command on the system console, message HVT432I is issued for every diskpool defined in the HVTPOLxx member. The displayed data is extracted from fields that were updated on the last threshold interval associated with each diskpool.

```
F HVT300EC,D=DSP
HVT432I ES=SYSALLDA VOL#= 129 TRKS= 9758_K USED= 40% DP00L=D_SYSALLDA
HVT432I ES=VTSWIZ VOL#= 15 TRKS= 583_K USED= 37% DP00L=D_VTSWIZ
```

```

HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DPOOL=D_DORI
HVT432I ES=VTSDVP VOL#= 7 TRKS= 272_K USED= 42% DPOOL=D_VTSDVP
HVT432I ES=VTSQA VOL#= 19 TRKS= 929_K USED= 41% DPOOL=D_VTSQA
HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DPOOL=D_VTSDSMS
HVT432I SG=SGDORI VOL#= 9 TRKS= 430_K USED= 56% DPOOL=D_TEST
HVT432I ES=VAULT VOL#= 8 TRKS= 391_K USED= 29% DPOOL=D_VAULT
HVT432I ES=VAULT VOL#= 8 TRKS= 391_K USED= 29% LOCID=CLN4
HVT432I SG=SGWIZ VOL#= 2 TRKS= 97_K USED= 40% LOCID=CLN5

```

Display Tapepools Utilization

VTE monitors the total number of virtual tapes and the number of active and scratch virtual tapes in each defined tapepool.

The following operator command can be used to display this data:

```
F server,{D|DISPLAY}={TSP|TSPACE}
```

After typing the above operator command on the system console, message HVT433I is issued for every tapepool defined in the HVTPOOLxx member. After all tapepools are displayed, message HVT434I gives the total summary for all displayed tapepools.

The displayed data is extracted from fields that were last updated on the expiration of the lowest threshold interval of all defined tapepools. That means delays may be encountered between tapepool capacity or content change and the updated HVT433I message data display.

```

F HVT300EC,DISPLAY=TSP
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_SYSALLDA
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_VTSWIZ
HVT433I TAPES= 1 ACT= 0( 0%) SCR= 1 TPOOL=T_VTSDVP
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_VTSQA
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_VTSDSMS
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_DORI
HVT433I TAPES= 0 ACT= 0( 0%) SCR= 0 TPOOL=T_VAULT
HVT433I TAPES= 55 ACT= 33( 60%) SCR= 22 TPOOL=$$DFLT
HVT434I TOTAL: TAPES= 56 ACT= 33( 58%) SCR= 23

```

Server Tuning

VTE provides some manual means by which its resources utilization can be controlled.

Limit Server CPU Utilization

The VTE server CPU% utilization can be limited to a specific percentage. The VTE server CPU% utilization will be kept below that limit, regardless of the workload or the overall CPU% utilization in the system.

The following should be noted regarding the CPU% utilization of the server:

- It is highly recommended to use the standard SRM or WLM for VTE tuning and ignore this parameter.
- The CPU% utilization of the server can be viewed by performance monitors (i.e. TMON). SDSF uses some internal normalization factors and does not show the absolute CPU% consumption of jobs.
- Under no circumstances can this option let the server consume more resource than available by the physical configuration, nor more resources than allowed by SRM or WLM.
- An internal mechanism continuously samples the server CPU% utilization. This mechanism needs a minimal number of samples to calculate the CPU% utilization after which it iteratively adjusts the internal CPU% consumption of the server. Due to this iterative process the server may utilize a higher CPU% than allowed by the customer predefined value. This higher utilization lasts approximately 15 to 20 seconds, after which it stabilizes around the user supplied value.

As said above, in the following cases the server may utilize a higher CPU% than allowed by the customer predefined value:

- The server was idle for a while, then a great bulk of work is to be performed (e.g. many jobs are submitted at once).

- The user dynamically decreased the CPU% while the server already has a CPU% limitation. In that case, the previously used CPU% remains in effect for a while.

In any case, the CPU% utilization fluctuates around the user defined value and may be higher or lower by a few percents.

The following operator command can be used to dynamically modify the CPU% value so that the server CPU% utilization will not exceed it:

```
F server,CPU%=percent
```

where *percent* is the CPU% utilization, which the server should not exceed. The percent value should be normalized. That is, if your system has more than one CPU (and the total CPU% utilization goes beyond 100%), the percent value reflects the normalized CPU% utilization (e.g. total CPU% divided by number of online CPU's). That way, the CPU% utilization as seen by the server will never go beyond 100%.

Server Management

VTE provides some manual means by which its internal resources can be managed by the operator.

Terminate an I/O Subtask and Free its Resources

Usually when a job using virtual tapes is to be terminated, the operator cancels it. As part of the cancel process, MVS purges all pending I/Os prior to letting the job write the trailer label and finally terminate.

In cases when the VTE server delays the tape I/O completion (as a result of internal errors or other delays), the job will not get cancelled and messages like HVT265W (Purge Pending) and HVT314W (Access to VDB Delayed) will be repeatedly issued.

The most common cause of such delays is when the user catalog that contains the virtual tape datasets or the user catalog that contains the VDB or the VDB itself are backed-up by VTE on virtual tapes. In this case, deadlocks in which resources like SYSZTIOT, SYSIGGV2 or V_TAPE are involved.

Such deadlocks can be relieved by letting the server relinquish control of the currently performed tape I/O and send an EQC sense. One method is to just cancel the VTE server. The other method is to kill the VTE server I/O subtask that currently serves the I/O and free its resources. Upon termination, the server I/O subtask will EQC the job letting its recovery routine handle the failure.

The following operator command can be used to kill a VTE server I/O subtask:

```
F server,KILL,{CUU=cuu|TCB=tcb}
```

After typing the above operator command on the system console, message HVT437I is issued to confirm the subtask termination.

If the command fails, message HVT436I is issued.

The command has two formats:

- `F server,KILL,CUU=cuu`

This format will kill the server I/O subtask according to the current served virtual device number. The *cuu* value is the three or four hexadecimal digits virtual device number. If a three-digit value is used, a zero is prefixed to the entered value.

- `F server,KILL,TCB=tcb`

This format will kill the server I/O subtask according to its TCB address. The *tcb* value is a six or eight hexadecimal digits value and represents a TCB in the server address space. If a six-digit value is used, two zeroes are prefixed to the entered value.

This command format will kill only server I/O subtasks. Attempts to kill other server subtasks will be ignored.

```
F HVT300EC,KILL,CUU=8A6
HVT437I VOLUME=EC1003 CUU=08A6 TCB=X'008CC250' (M20BDG4 /JOB08058) KILLED
```

```
HVT342E M20BDG4 /JOB08058 I/O FAILED BY 0_RDIREC SERVER - EQC GENERATED  
IEC215I 714-0C,IFG0200Z,M20BDG4,C3,OUT1,08A6,EC1003,M20.EC1003
```

Note:

- If the TCB address derived from the command does not represent a really stuck server I/O subtask, another server I/O subtask may be killed leading to another virtual device being affected.
- After the KILL command completion, the affected virtual device may remain in an unstable state. Should this happen, the device and/or its path should be varied offline and online.

Operator Commands Restrictions

The VTE virtual devices fully simulate IBM-compatible cartridge devices and tape volumes. As such, system operator commands (with some exceptions) can be used for these devices. System operator commands (like DISPLAY) can show the VTE virtual devices status while VARY system operator commands can change their status.

Unsupported System Operator Commands

Due to the special name of VTE virtual devices, some of the system operator commands are not supported.

The following system operator commands are not supported and special actions are taken by the server when they are encountered:

- **UNLOAD cuu**

The manual unload of a virtual tape is not supported. When an UNLOAD system operator command involving a VTE virtual device is encountered, message HVT293W is issued and the command is suppressed. HVT293W message description lists a special procedure to be followed when a virtual tape has been manually mounted and should be unloaded.

Note that under normal circumstances, no manual unload of virtual tapes is needed (as MVS always unloads the tapes at job end or when the device is needed for another mount).

- **SWAP cuu,cuu**

Swap of a virtual tape from one VTE virtual device to another or from a virtual device to a real device or vice versa is not supported. When a SWAP system operator command involving a VTE virtual device is issued, the system replies with this message:

```
IEE380I cuu DEVICE TYPE INVALID
```

and the SWAP is not executed.

- **DS P,cuu**

VTE uses virtual CHPIDs and virtual paths for its VTE virtual devices. As such, the system cannot display the path status of VTE virtual devices using the 'DEVSERV PATH' system operator command.

When a 'DEVSERV PATH' system operator command involving a VTE virtual device is issued, the system replies with this message:

```
IEE459I DEVICE NOT SUPPORTED BY DEVSERV PATHS COMMAND
```

Note that the other forms of the DEVSERV system operator commands are fully supported.

- **MOUNT cuu,VOL=(labtyp,volser) (JES3 only)**

Manual mount of virtual tapes is not supported in JES3 environments. This restriction is caused by the fact that VTE processes the virtual tapes mounts only when the job that needs them has created its address space. When a manual mount of a virtual tape is requested, JES3 itself processes the request. In that case, the VTE server will not react to this mount, leaving the VTE virtual device in an AC (in use by a setup job) status. A Cancel Setup (*C S, jobid) of the MOUNT job is required in order to free the VTE virtual device.

Chapter 32. VTE Offline Utilities

This chapter describes the offline VTE utilities used to manage the VTE virtual tape library.

The input initialization statements for each utility provided in this chapter are in addition to the statements defined in the initialization statements section of this book. VTE initialization statements are concatenated before the utility input statements.

HVTUCTC – CTC and a Virtual Tape data set

The HVTUCTC utility is used to start and control an VTE session with a type5 e-Vault remote off-site storage location.

This utility, which is an alias for the HVTUFTP utility, runs as a started task and is activated solely by the VTE server when a virtual tape is to be e-Vaulted using a type5 e-Vault. This type is used for cloud transactions. Starting this utility manually has no effect; it will not find the proper environment to run under and will terminate. The started task procedure is similar to the procedure used to start the server or any other utility.

Since this utility is NOT a stand-alone utility, no sample is provided for it in the VTE SAMPLIB library.

HVTUDST – Set De-Stage Priority

HVTUDST utility can be used to control the utilization level of the VTE diskpools. It is responsible for prioritizing the virtual tape data sets de-stage and providing the input for any storage management product that performs the migration and recall operations. However, it does not perform the migration of the virtual tape data sets by itself.

Run daily, the HVTUDST utility records the activity level of tape data sets on site. The activity level of a tape dataset consists of the number of mount and scratch events of the virtual tape it resides on, for any given day.

As more information is collected about the activity level of tape data sets, the HVTUDST utility acquires more knowledge about the life cycle of the associated virtual tapes.

Over time, the HVTUDST utility collects enough information so that it can predict the probability of a mount or a scratch operation in the future of each virtual tape. Based on this information, virtual tape data sets whose associated virtual tapes have a high activity level will have a higher priority to remain on disk than those with a lower activity level. The later will be prioritized to be de-staged off disk.

At the end of the daily process, the HVTUDST utility creates a list of virtual tape data sets that should be de-staged so that the utilization level on the diskpool reaches a pre-defined threshold. This list is embedded into a migration job (according to the local storage management product) that can be submitted for execution.

The VTE server constantly monitors the utilization level of its diskpools. If that utilization level exceeds a predefined threshold, warning messages are issued. The local automation product can intercept these messages and initiate the HVTUDST utility execution to create the migration job.

For further information on the utility logic and parameters, refer to the *IBM System Storage VTF Mainframe User's Guide [GC53-1191-00], Chapter 7, Managing the Diskpools*

A sample job to run HVTUDST can be found in member HVTUDST in the VTE SAMPLIB library.

Input

- One DRUN initialization statement.
- None, one and as many as 1000 DRULE initialization statements.
- At least one and as many as 1000 DPTRN initialization statements.

DRUN	PROD={HSM ABR DMS EMG}
	[,WAIT=nn]
DRULE	{INC EXC},
	DSN=dsn
DPTRN	NAME=pattern
	[,QUAL#=qual#]

DRUN

The DRUN initialization statement identifies the external storage management product or electronic migration for which a migration job is created. The migration job contains the list of virtual tape data sets to be migrated.

PROD={HSM|ABR|DMS|EMG}

The PROD parameter determines the storage management product for which the migration job is created.

HSM

Use this option when the local storage management product is DFSMSHsm.

When using this option, HVTUDST uses member HVTSKHSM in the VTE SAMPLIB library to create a migration job.

The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSN keyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

ABR

Use this option when the local storage management product is FDR/ABR.

When using this option, HVTUDST uses member HVTSKABR in the VTE SAMPLIB library to create a migration job.

The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSN keyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

DMS

Use this option when the local storage management product is CA-Disk.

When using this option, HVTUDST uses member HVTSKDMS in the VTE SAMPLIB library to create a migration job.

The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSN keyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

Value range: N/A

Default: None

EMG

Use this option when VTE is to use its built-in FTP services and use the electronic migration (e-Mig) function to migrate the virtual tape data sets.

When using this option, the HVTUDST utility uses member HVTSKEMG in the VTE SAMPLIB library to create a migration job. The HVTSKEMG member is a skeleton in which:

1. One selected virtual tape volume serial number replaces the %VOLSER keyword.
2. The text lines between the ") REPEATSTART" and ") REPEATEND" statements are repeated for each virtual tape volume.

Note that the above example contains the LOCID name of the remote location that is to receive the migrated virtual tape data sets.

WAIT=*nn*

By default, once the migration job is prepared, HVTUDST does not monitor it. Instead, HVTUDST finishes executing upon completion of its own tasks.

When the WAIT parameter is specified and its value is greater than 0, HVTUDST will not complete immediately and will wait until all datasets specified in the migration job are indeed migrated.

nn

Specifies the number of minutes HVTUDST will wait before issuing the HVT454W message asking the operator whether or not to extend the monitoring period for another *nn* minutes.

During that interval, if all datasets were migrated, HVTUDST will complete execution regardless of the *nn* value.

If *nn* is 0, no monitoring is performed and HVTUDST completes execution regardless of the specified status of the datasets.

Value range: Integer from 0 to 1440.

Default: 0

DRULE

The DRULE initialization statement identifies tape data set names for which the associated virtual tape data set will be INCLUDED or EXCLUDED from the de-stage process regardless the calculated grade.

The following rules apply to the DRULE initialization statements:

- Each dsname value is matched against all DRULE initialization statements starting with the first. When a DRULE initialization statement matches, the associated tape data set is INCLUDED or EXCLUDED respectively, and no further statements are scanned.
- If the diskpool low threshold (the 3rd positional parameter of the respective DPOOL initialization statement) is reached and more tape data sets should be INCLUDED, they are **not** INCLUDED.

INC|EXC

Specifies that the current PRULE initialization statement is an INC or EXC type PRULE initialization statement.

Value range: N/A.

Default: None.

DSN=*dsn*

dsn

Specifies the tape data set name whose associated virtual tape data set is to be INCluded or EXCluded from the de-stage process. This tape data set name should be the first on the virtual tape otherwise it is ignored.

The DSN parameter value accepts mask characters as character placeholders. The following mask rules apply when using these character placeholders:

- An asterisk (*) matches none, one or more successive characters.
- A question mark (?) matches exactly one character.

Value range: A 1 to 44 alphanumeric character string, including periods. Each qualifier between the periods can be a 1 through 8 alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).

Default: None.

DPTRN

The DPTRN initialization statement defines patterns according to which the HVTUDST utility derives generic names from tape data set names.

The order of the DPTRN initialization statements is important as the HVTUDST utility considers the first pattern that matches the virtual tape data set name. Once a pattern matches a tape data set name and a generic name is derived, no more DPTRN initialization statements are scanned.

NAME=*name*

name

Specifies the name portion of the pattern definition.

According to the QUAL# parameter value, the NAME value can represent one data set name qualifier or a full data set name (including character placeholders).

The NAME parameter value accepts mask characters as characters placeholders. The following mask rules apply when using these characters placeholders:

- An asterisk (*) matches none, one or more successive characters.
- A question mark (?) matches exactly one character.

Value range: A 1 to 44 alphanumeric character string, including periods. The first character must be alphabetic or national (\$, #, @).

Default: N/A.

QUAL#=*qual#*

qual#

Specifies the qualifier-level portion of the pattern definition. When the specified pattern in the current DPTRN initialization statement matches the specified qualifier level in a given data set name, a generic name can be derived.

Two special values can be specified:

0	The NAME parameter value is considered to represent a full tape data set name (including character placeholders).
99	The last qualifier within the data set name.
Value range:	Integer from 0 to 99.
Default:	0.

Return Codes

The following return codes are returned by HVTUDST:

0	No errors encountered, at least one virtual tape dataset found to be eligible for de-staging.
4	No errors encountered, but one of the following conditions were encountered: <ul style="list-style-type: none"> • The history file was not updated with new observations, last utility run was less than one day ago. • At least one duplicate DPOOL was encountered and ignored. • One of the following: <ul style="list-style-type: none"> – At least one VDB entry does not have a corresponding virtual tape dataset on disk. – At least one virtual tape dataset does not have a corresponding entry in the VDB. – At least one VDB entry and the associated virtual tape dataset on disk do not match. – At least one VDB entry that misses the virtual tape creation date or full tape dataset name was encountered. – Migration job not created, HVT452W message issued.
8	Severe error, utility terminated.
12	Utility abended or cancelled. If the history file was being updated while the abend occurred, on the next utility run it will be restored from the internal backup with no operator intervention.

HVTUFTP – FTP a Virtual Tape Data set

The HVTUFTP utility is used to start and control an FTP session with a type3 or type4 e-Vault remote off-site storage location. HVTUCTC utility (an HVTUFTP utility alias) is used to start and control an IBM Cloud Tape Connector (CTC) session with a type5 e-Vault remote off-site storage location.

These utilities run as started tasks and are activated solely by the VTE server when a virtual tape is to be e-Vaulted using any of the supported types. Starting these utilities manually has no effect; they will not find the proper environment to run under and will terminate. The started task procedures are similar to the procedure used to start the server or any other utility.

Since these utilities are not stand-alone utilities, no samples are provided for them in the VTE SAMPLIB library.

Note: To start and control an FTP session with a type5 (cloud) e-Vault remote off-site storage location, you must the HVTUCTC utility. See [“HVTUCTC – CTC and a Virtual Tape data set” on page 317](#)

Input

No input initialization statements are required by HVTUFTP utility.

Upon starting, the utility receives the following external parameters from the server:

- The volume serial number to be e-Vaulted is passed as the procedure identifier in the format of Vvolser.

- The suffix of the HVTSYSxx member name in the VTE SAMPLIB library is passed as a symbolic parameter.
- The VTE subsystem is accessed in order to retrieve internal server parameters.

Return Codes

The following return codes are returned by HVTUFTP:

0	OK. The utility's sysout is purged. For any other return code, the sysout is not purged in order to let the operator check and possibly correct the problem. Note: On completion of a HVTUFTP or HVTUCTC utility run, VTE purges all of the previous utility sysouts that handled the same virtual tape and leaves only the last one. Doing that prevents the accumulation of sysouts of failed utility runs.
4	VTE server went down while executing or the proper server environment could not be found.
8	Severe error, utility terminated.
12	Utility abended or cancelled.

HVTUJFR – Journal File Format

The HVTUJFR utility can be used to allocate and pre-format the VTE wrap-around journal file. The utility is used during VTE installation or when the journal file is lost or damaged and a new journal file should be allocated.

The VTE server should be stopped while running this utility. This is necessary since the utility scratches the old journal file before it allocates the new one.

A sample job to run HVTUJFR can be found in member HVTUJFR in the VTE SAMPLIB library.

Planning the Journal File Size

The journal file size is automatically calculated according to user parameters that consider all the expected manipulations done to the VDB in a given time range.

The following formula is used to calculate the size of the journal:

$$\text{blk\#} = ((\text{days} * \text{mntsday}) * 4 + \text{utils}) / 16$$

<i>blk#</i>	The number of blocks required for the file.
<i>days</i>	The number of days between two consecutive VDB backups.
<i>mntsday</i>	The number of virtual tape mounts per day.
<i>utils</i>	The number of virtual tapes manipulated by VTE offline utilities during the days period.

Note: The journal file record size is 384 bytes.

The allocation is done in blocks. Each block contains 16 records (plus a 16 byte block header), and its size is 6160 bytes.

Input

The input to the HVTUJFR utility consists of one FORMAT initialization statement.

FORMAT	JRNNAME=jrnnname, DAYS=days, MNTSDAY=mntsday,
--------	---

```

UTIL=utils,
UNIT=unit
[,STORCLAS=storclas]
[,MGMTCLAS=mgmtclas]
[,DATACLAS=dataclas]
[,UNIT=unit]
[,VOL=volser]

```

FORMAT

The FORMAT initialization statement defines the parameters upon which the journal file will be allocated and formatted.

A valid combination of STORCLAS, MGMTCLAS, UNIT and VOL optional parameters is required to form a valid allocation request.

It is highly recommended to allocate the journal file and the VDB on different volumes.

JRNNAME=*jrnnname*

<i>jrnnname</i>	Specifies the full name of the journal file to allocate and format.
Value range:	1 to 44-character alphanumeric string, including periods. Each qualifier between the periods can be a 1 to 8-character alphanumeric or national (\$, #, @) characters string. The first character must be alphabetic or national (\$, #, @).
Default:	None

DAYS=*days*

<i>days</i>	Specifies the number of days between 2 consecutive backups of the VDB.
Value range:	Integer between 2 to 10
Default:	None

MNTSDAY=*mntsday*

<i>mntsday</i>	Specifies the expected number of virtual tape mounts per day.
Value range:	Integer between 100 to 9999
Default:	None

UTIL=*utils*

<i>utils</i>	Specifies the expected number of tapes manipulated by all the VTE offline utilities (like HVTUTAP, HVTUSCR, etc.), in the period given by the days value.
Value range:	Integer between 100 to 9999
Default:	None

STORCLAS=*storclas*

<i>storclas</i>	Specifies the Storage Class name associated with the newly allocated journal file.
Value range:	1 to 8-character alphanumeric string.
Default:	None

MGMTCLAS=*mgmtclas*

mgmtclas Specifies the Management Class name associated with the newly allocated journal file.

Value range: 1 to 8-character alphanumeric string.

Default: None.

DATACLAS=*dataclas*

dataclas Specifies the Data Class name associated with the newly allocated journal file.

Value range: 1 to 8-character alphanumeric string.

Default: None

UNIT=*unit*

unit Specifies the disk unitname on which the utility will allocate and format the journal file.

Value range: 1 to 8-character alphanumeric string.

Default: None.

VOL=*volser*

volser Specifies the disk volume serial number on which the utility will allocate and format the journal file.

Value range: 1 to 8-character alphanumeric string.

Default: None

Return Codes

The following return codes are returned by HVTUJFR:

0	OK
8	Severe error, utility terminated.

HVTUJRC – VDB Recovery

The HVTUJRC utility can be used to forward recover the VDB according to the journal file. The utility will fail if the VDB is set to Read_Only mode.

During this recovery process, the journal file is read from the oldest to the newest record. The first journal record whose update timestamp is one hour before the last VDB update timestamp initiates the VDB recovery process. From this point on, all journal records update their respective VDB records. The one hour overlap is needed to compensate for two CPUs whose clocks are not synchronized and access the same VDB. At the end of the process, the VDB is up-to-date according to the journal file.

Note: If, prior to using HVTUJFR utility to recover the VDB, you defined a new VDB (using the HVTUVSI job) and restored its contents from a backup, make sure that the IDCAMS REPRO's statement contains the REPLACE parameter. This is so since the newly defined VDB already contains a control record with the current date that should be replaced with the control record from the restored VDB.

A sample job to run HVTUJRC can be found in member HVTUJRC in the VTE SAMPLIB library.

Input

No input initialization statements are required by the HVTURCV utility.

Return Codes

The following return codes are returned by HVTUJRC:

0	OK
4	Journal file is empty.
8	Severe error, utility terminated.

HVTURCV – Virtual Tapes Recovery

This section describes the HVTURCV utility which recovers virtual tapes.

The HVTURCV utility can be used to:

- Recover one or more VDB entries according to the virtual tape data set on disk after virtual tape data sets have been manually managed (restored, deleted, etc.), or the virtual tapes have not been properly closed.
- Recover the whole VDB according to the virtual tape data sets on disk (after the VDB was inadvertently deleted, destroyed etc.) and the journal file is not usable.
- Reset the e-Vault status of a virtual tape that is stuck in the e-Vault process (due to MVS, TCP/IP, FTP failure or internal VTE problems).

The utility will not change a virtual tape status if the tapepool the virtual tape belongs to or the VDB are in Read_Only mode.

The following virtual tape conditions are expected after HVTURCV has been run:

- Every recovered virtual tape that exists in the VDB but does not have a corresponding virtual tape data set on disk is scratched. The DEFERBY parameter of the virtual tape tapepool's TPOOL statement is ignored.
- A new VDB entry is built for every recovered virtual tape data set that does not have a corresponding entry in the VDB. The tape status is 'active' and it may be synchronized with the tape management system using the HVTUSCR utility. If the virtual tape is marked scratch in the tape management system database, its DEFERBY period will be counted again from 0.
- Every recovered VDB entry that has a corresponding virtual tape data set on disk is checked for consistency and, if needed, fixed according to the disk data. The deferred scratch period count remains.
- In cases (2) and (3) above, if the virtual tape data set on disk is found to be corrupted, the corresponding VDB entry is marked as representing an 'unusable' virtual tape.
- A virtual tape found in an interim e-Vault status is reset to one of the following states:
 - 'local'
 - 'vaulted'
 - 'safe'

When the environmental problem that caused the virtual tape found in an interim e-Vault status is fixed, the e-Vault process can be started again.

Note: Migrated virtual tape data sets whose VDB entries are recovered will be recalled.

A sample job to run HVTURCV can be found in member HVTURCV in the VTE SAMPLIB library.

Input

No input initialization statements are required by the HVTURCV utility.

Return Codes

The following return codes are returned by HVTURCV:

0	Either all requested tapes were successfully recovered or no virtual tape was found in the requested range.
4	At least one virtual tape was not recovered because of an error in the process or at least one virtual tape has been marked as 'unusable'.
8	Severe error, utility terminated.
12	Utility abended or cancelled.

HVTUREP – Database Reports

The HVTUREP utility can be used to print reports describing the contents of VTE databases.

The following reports are supported:

- REP001

This report lists the virtual tapes status in the VDB.

Additionally, some informative messages about the number of virtual tapes found under certain categories (i.e. 'scratch', 'active', 'in-use', etc.) in the entire VDB and selected volume ranges are displayed.

- REP002

This report lists the data set records contents from the history file.

A sample job to run HVTUREP can be found in member HVTUREP in the VTE SAMPLIB library.

Input

The input to HVTUREP utility consists of one REPORT initialization statement.

```
REPORT          REP={REP001|REP002},LINECT=linect
                [,VOLSER=volser]
                [,ONLYSCR={YES|NO}]
                [,ONLYNSCR={YES|NO}]
                [,ONLYINU={YES|NO}]
                [,ONLYSL={YES|NO}]
                [,ONLYERR={YES|NO}]
```

REPORT

The REPORT statement identifies this request as a report request.

REP={REP001|REP002}

Specifies the report to be produced.

REP001 Extracts virtual tapes status from the VDB and prints it.

REP002 Extracts the history file contents and prints it.

Value range: N/A

Default: None

LINECT=*linect*

linect Specifies the number of lines to be printed on one report page.

Value range: Integer from 5 to 88

Default: None

ONLYSCR={YES|NO}

This parameter applies to REP001 report only.

YES	Specifies to include only scratch virtual tapes.
NO	Specifies to include virtual tapes regardless of their scratch status.
	Value range: N/A
	Default: NO

ONLYNSCR={YES|NO}

This parameter applies to REP001 report only.

YES	Specifies to include only non-scratch virtual tapes.
NO	Specifies to include virtual tapes regardless of their scratch status.
	Value range: N/A
	Default: NO

ONLYINU={YES|NO}

This parameter applies to REP001 report only.

YES	Specifies to include only virtual tapes that are either being read or written.
NO	Specifies to include virtual tapes regardless of their read or write status.
	Value range: N/A
	Default: NO

ONLYSL={YES|NO}

This parameter applies to REP001 report only.

YES	Specifies to include only standard label virtual tapes.
NO	Specifies to include virtual tapes regardless of their label type.
	Value range: N/A
	Default: NO

ONLYERR={YES|NO}

This parameter applies to REP001 report only.

YES	Specifies to include only virtual tapes that are marked 'unusable' by the recovery utility process (HVTURCV).
NO	Specifies to include virtual tapes regardless of their usability status.
	Value range: N/A
	Default: NO

VOLSER=volser

This parameter applies to REP001 report only.

volser

Specifies the virtual tapes volume serial number that should be included in the report.

The volser value accepts mask characters as character placeholders.

The following mask rules apply when using these character placeholders:

- An asterisk (*) matches none, one or more successive characters.
- A question mark (?) matches exactly one character.

Value range: A 1 to 6 alphanumeric character string.

Default: *.

Report Output

Two different types of reports can be generated and printed. REP001 provides detailed information about the virtual tapes, while REP002 provides information on generic data sets and the number of scratch and mount events that occurred on a day.

REP001 Report

The following information is printed by REP001 report:

```
20.34.30 HVT019I RPL   IS NOW ENABLED                                *HVTPOL
*
20.34.31 HVT304I POL   REFRESH WAS OK, RC=00                        *HVTRFR  *
20.34.31 HVT200I VTF Mainframe VER 3.0.0 UTILITY HVTUREP  STARTED  *HVTUREP  *

VOLUME  SCR      LOCID___(VD)L LB U R SID  ALC_M*  JOBNNAME  JOBJID  DATE    TIME    VIRTUAL TAPE  DATASET
-----  -
DM0001  YES      *MAIN*      SL      JJ02  00082  VTFTEST1  JOB00346  2019.205  16:38:47  VTYH.VDM0001.VTF.T1
DM0002  NO        LINUX      (V )L  SL      JJ02  00005  VTFTEST2  JOB00347  2019.205  16:39:52  VTYH.VDM0002.VTF.T2
DM0003  NO        *MAIN*      SL      JJ02  00005  VTFTEST3  JOB00348  2019.205  16:40:40  VTYH.VDM0003.VTF.T2
DM0004  NO        CLN2      (V )  SL      JJ02  00005  H#1121    JOB00351  2019.205  16:41:44  VTYH.VDM0004.VTF.T5
DM0005  NO(0017H) *MAIN*      SL      JJ02  00099  D#VT11    JOB00380  2019.205  17:45:08  VTYH.VDM0005.VTF.T6
DM0006  NO        CLN2      (B )  SL      JJ02  00049  H#1122    JOB00389  2019.205  18:18:30  VTYH.VDM0006.VTF.T7
DM0007  NO        *MAIN*      SL R    JJ02  00049  H#1123    JOB00390  2019.205  18:18:43  VTYH.VDM0007.VTF.T3
DM0008  NO        *MAIN*      SL W    JJ02  00049  H#1123    JOB00390  2019.205  18:18:43  VTYH.VDM0008.VTF.T8
DM0009  YES      *MAIN*      SL
DM0010  YES      *MAIN*      SL

18.20.37 HVT202I ENTRIES IN VDB = 10      ENTRIES IN LIST = 10      *HVTUREP *
18.20.37 HVT202I SCRATCH IN VDB = 2        SCRATCH IN LIST = 2        *HVTUREP *
18.20.37 HVT202I ACTIVE  IN VDB = 8        ACTIVE IN LIST = 8        *HVTUREP *
18.20.37 HVT202I ERROR   IN VDB = 0        ERROR IN LIST = 0        *HVTUREP *
18.20.37 HVT202I WRITE   IN VDB = 1        WRITE IN LIST = 1        *HVTUREP *
18.20.37 HVT202I READ    IN VDB = 1        READ IN LIST = 1        *HVTUREP *
18.20.37 HVT202I S_LABEL IN VDB = 10       S_LABEL IN LIST = 10     *HVTUREP *
18.20.37 HVT202I ALLOC_GB IN VDB = 0        ALLOC_GB IN LIST = 0     *HVTUREP *
18.20.37 HVT202I USED_GB IN VDB = 0        USED_GB IN LIST = 0      *HVTUREP *
18.20.37 HVT201I UTILITY HVTUREP ENDED, RC=00      *HVTUREP *
```

The report lists the following fields:

VOLUME	The volume serial number of the listed virtual tape.	
SCR	YES	The virtual tape is scratch.
	YES-	The virtual tape bears the 'Mount-On- Specific-Request-Only' attribute and is not eligible for scratch mount requests.
	NO(nnnnH)	<p>The virtual tape is not scratch.</p> <p>If (nnnnH) appears after the NO literal, the virtual tape is in deferred-scratch status. The nnnn value represents the number of hours until it will be eligible for scratch.</p> <p>If nnnn is zero, the virtual tape is eligible for scratch and will be scratched on the next run of the HVTUSCR utility.</p>

LOCID_---	<p>The local or off-site storage location name in which the virtual tape currently resides.</p> <p>If the name is *MAIN*, the true location name is blank. If the name is *RECRTD*, this is a temporary location name set by HVTURCV (in case the virtual tape has been recovered). This name will be reset when the e-Vaulted virtual tape returns.</p>	
(VD)	<p>The virtual tape vault status in the specified location. The VD value consists of two components:</p> <ul style="list-style-type: none"> • V - the status of the vaulting process • D - the direction of the vaulting process • H - 'to be remote-scratched conditionally' status. <p>The virtual tape is marked to be remote scratched conditionally (that is, only if the remote-scratch succeeds), but the process on the remote site did not end yet.</p> • J - 'to be remote-scratched unconditionally' status. <p>The virtual tape is marked to be remote-scratched unconditionally (that is, regardless of the remote-scratch success), but the process on the remote site did not end yet.</p>	
V	B	<p>'to be vaulted' status.</p> <p>The virtual tape is marked 'to be vaulted', but the e-Vault process has not yet started.</p>
	V	<p>'vaulted' status.</p> <p>The virtual tape resides in the remote off-site storage location whose name is listed under the LOCID column.</p>
	F	<p>'safe' status.</p> <p>The virtual tape resides in the current location (where the report was created) as an e-Vaulted virtual tape.</p>
	T	<p>'to be returned' status.</p> <p>The virtual tape is marked 'to be returned' but the return process did not start yet.</p>
	H	<p>'to be remote-scratched' status.</p> <p>The virtual tape is marked to be remotescratched but the process on the remote site did not end yet.</p>
	X	<p>'obtain extent size' status.</p> <p>The virtual tape is to be e-Vaulted to a type4 e-Vault location. The local server inquired for the remote off-site storage location allocation parameters (extent size, unit, unit count, etc.) and waits for the response.</p>
	Y	<p>'FTP-ed' status.</p> <p>The virtual tape dataset had been successfully FTP-ed to a remote type3 or type4 off-site storage location.</p>

	Z	'recovered after FTP' status. The virtual tape dataset had been successfully FTP-ed to a type3 or type4 offsite storage location, recovered on the remote site and waits for the local site to mark the virtual tape 'vaulted'.
D	Blank	The process described by the V column flows in a "local to remote" direction .
	2	The process described by the V column flows in a "remote to local" direction.
L	Blank	This value, in conjunction with the LOCID____ value, indicates that the virtual tape is replicated and that a local copy of it exists.
LB	SL	The virtual tape has a standard label.
	NL	The virtual tape does not have a standard label.
U	W	The virtual tape is currently being written by the system-id listed in the SID column.
	R	The virtual tape is currently being read by the system-id listed in the SID column. More systems may concurrently read the virtual tape, but only one system-id is listed.
	Blank	No system is currently accessing the virtual tape.
	Note: If a job that used the virtual tape was interrupted, as a result of the server failure or sudden IPL, the U field reflects the virtual tape status prior to the failure.	
R	*	The virtual tape is marked 'unusable' by the HVTURCV utility. This may have happened when HVTURCV failed to recover the virtual tape.
SID	<p>If U=R, this is one of the system-ids that are reading the virtual tape.</p> <p>If U=W, this is the system-id that is writing the virtual tape.</p> <p>If U=blank, this is the system-id that wrote the virtual tape last.</p> <p>If the virtual tape was recovered (JOBNAME=*RECRTD*), this field is blank.</p>	
ALC_M*	<p>Number of megabytes allocated to the virtual tape dataset.</p> <p>The following conditions apply to that field:</p> <ul style="list-style-type: none"> • If the field is blank, then: <ul style="list-style-type: none"> – Disk space statistics have not been set yet for this virtual tape. You may recover the virtual tape (using HVTURCV utility) in order to set the disk space statistics. – The virtual tape dataset is vaulted. – The virtual tape dataset has been manually deleted from the disk. • If the value is followed by an asterisk, the virtual tape dataset has been migrated to a real tape and does not actually reside on disk. 	
JOBNAME	The jobname that last opened the virtual tape for output. If *RECRTD* appears in this field, the VDB entry was recreated by the HVTURCV utility and the last user statistics are not available.	

JOBID	The jobid that last opened the virtual tape for output. If the virtual tape was recovered (JOBNAME=*RECRTD*), this field is blank.	
DATE	The date (YYYY.DDD) this virtual tape was last opened for output. If the virtual tape was recovered (JOBNAME=*RECRTD*), this field is blank.	
TIME	The time (HH:MM:SS) this virtual tape was last opened for output. If the virtual tape was recovered (JOBNAME=*RECRTD*), this field is blank.	
Virtual Tape	The associated virtual tape dataset name on disk.	
Dataset Name		

REP002 Report

The following information is printed by REP002 report:

```

19.40.51 HVT200I VTF MAINFRAME VER 3.0.0 UTILITY HVTUREP STARTED          *HVTUREP *
1GENERIC NAME                      DC#/SCR#/MNT# DC#/SCR#/MNT# DC#/SCR#/MNT#
-----
0M20.*.AAAAAAA.BBBBBBBB.CCCCCCCC.A    001_0003_0002 002_0003_0003 003_0001_0006
M20.EC7081                             013_0001_0000
M20.GDG.G????V??                       000_0000_0001 007_0000_0003 014_0000_0002
                                         042_0000_0003 076_0000_0001 078_0009_0002
M20.GDG1.G????V??                      000_0000_0003 003_0007_0001 007_0000_0005
                                         108_0033_0005
M20.GDG2.G????V??                      000_0001_0006 002_0003_0003 003_0001_0006
                                         014_0000_0001
M20.JAN????                             001_0003_0002 002_0001_0004 003_0003_0002
M20.JAN????.AAAAAAA.BBBBBBBB.CCCCCCCC 000_0000_0001
M20.KIKI.JAN??                          002_0001_0007 009_0001_0003 010_0001_0001
M20.KIKI.R????Q??.KUKU3                002_0001_0008
M20.KUKU.JAN??                          002_0001_0007 003_0007_0001 005_0003_0002
                                         010_0001_0001
M20.KUKU.JAN2019                        000_0000_0001
M20.KUKU1                               002_0001_0005 004_0006_0002
M20.KUKU3.JAN??                         000_0000_0001
M20.KUKU3.JAN??.AAAAAAA.BBBBBBBB.CCCCCCCC 000_0000_0001
M20.KUKU3.JAN??.AAAAAAA.BBBBBBBB.CCCCCCCC.A 000_0000_0001
M20.N2222022                           000_0000_0001
19.40.51 HVT201I UTILITY HVTUREP ENDED, RC=00          *HVTUREP *

```

The report lists the following fields:

GENERIC NAME	The dataset record generic name. If certain tape datasets could not be generated into generic names, this field contains their full dataset names.	
DC#/SCR#/MNT#	Up to five triplet values on a line, each one detailing scratch or mount events observations in one given day. If more than five triplet values should be printed, they are printed on a new line (without respecifying the generic name).	
	DC#	Number of days from the tape dataset creation till the listed event has occurred and observation recorded.
	SCR#	Number of scratch events on that day.
	MNT#	Number of mount events on that day.

Return Codes

The following return codes are returned by HVTUREP:

0	OK
8	Severe error, utility terminated.

HVTUSCR – Virtual Tapes Status Synchronization

As virtual tapes are used, their status changes from 'scratch' to 'active'. Periodically, the tape management system changes back this status to 'scratch', or changes the virtual tape required location from a local location to a remote off-site storage location (or vice versa). These changes should be reflected in the VDB, that is, the VDB should be synchronized with the tape management system database.

The utility will not change a virtual tape status if the tapepool the virtual tape belongs to or the VDB are in Read_Only mode.

The HVTUSCR utility can be used to synchronize the tape management system database and the VDB as follows:

- Update and optionally override the e-Vault indicators in the VDB.
- Scratch virtual tapes according to:
 - The local tape management system.
 - The MVS catalog.
 - Explicitly specified virtual tapes.

The scratch can be done:

- Immediately.
- Deferred by a user defined number of hours set for each tapepool in the corresponding TPOOL initialization statement.
- Forced, regardless of any defer conditions.

Once a virtual tape is scratched, its corresponding virtual tape dataset is physically deleted from the disk and the data is not available anymore (unless restored from backup).

- Reclaim virtual tapes that have been deferred-scratched but have not yet been scratched.
- If the e-Vault option is implemented, mark virtual tapes as eligible for:
 - e-Vault to a remote off-site storage location.
 - Return from a remote off-site storage location.
 - Remote-scratch (either immediate or defer).

The mark is done according to:

- The local tape management system.
- Explicitly specified virtual tapes.
- Update and optionally override the e-Vault indicators in the VDB.

If two or more VTE servers run in different CPUs sharing the same VDB and virtual tapes, the actual e-Vault or return process is performed by the VTE server running in the CPU under which the HVTUSCR utility marked the e-Vault or return request.

The HVTUSCR utility usually runs daily as part of the tape management system cleanup process.

Input

The input to HVTUSCR utility consists of either:

- One SYNC initialization statement (if TYPE is not VOLSER).
- At least one and as many as 20,000 SYNC initialization statements (if TYPE is VOLSER).

Note: If more than one SYNC initialization statements is present, HVTUSCR considers only the first encountered statement.

```
SYNC      TYPE={VOLSER,VOLSER=volser[,RECLAIM={YES|NO}]
           [,LOCID=locid] |
           CATALOG          |
```



```

TLMS
CA-1
CONTROL-T
ZARA
RMM
[ ,FORCE={force}]
[ ,TMSRESET={YES|NO}]
{VLTUPD|VFTP}

```

SYNC

The SYNC initialization statement requests that the scratch and e-Vault status of all the entries in the VDB will be synchronized with the tape management system database or MVS catalog, or that a specific virtual tape entry will be synchronized as per the user request.

TYPE={VOLSER|CATALOG|TLMS|CA-1|CONTROL-T|ZARA|RMM}

The TYPE parameter defines the method upon which the VDB synchronization will be performed.

VOLSER

When using this option, HVTUSCR handles separately each virtual tape specified in the VOLSER parameter. The virtual tape is handled regardless its status in the tape management system database or MVS catalog. Incorrect use of this option may cause discrepancies between the virtual tape status in the VDB and its status in the tape management system database (if used).

Note: If *LOCID=locid* is **not** specified, refer to HVTUSCR process when *TYPE=volser* is coded and *LOCID=locid* is not specified.

If *LOCID=locid* is specified, the virtual tape is not scratched. Refer to the parameter "LOCID=locid" (further down) for a description of HVTUSCR process for that option.

If more than one specific virtual tapes are to be handled, the SYNC initialization statement should be repeated.

A sample job to run HVTUSCR with *TYPE=VOLSER* can be found in member HVTUSCRV in the VTE SAMPLIB library.

CATALOG

This option is used by sites that do not use any tape management system and the dataset residing on virtual tapes are cataloged (either in the master or user catalog(s)).

Each virtual tape that all the datasets residing on it are not cataloged, is considered to be scratch.

During the MVS catalogs scan, the utility lists all relevant entries from the master and all connected user catalogs. User catalogs not connected to the master catalog are not scanned and the tape datasets cataloged in it are not listed. That means that virtual tapes that contain datasets cataloged in user catalogs accessed via JOBCAT or STEPCAT DD statements will be scratched. Accordingly, HVTUSCR with TYPE=CATALOG cannot be used when jobs catalog their datasets in user catalogs not connected to the master catalog.

If one or more user catalogs are not accessible, the utility terminates and no virtual tape is scratched. You may have connected user catalogs that are not accessible and do not really contain relevant entries. To allow the process of HVTUSCR, you can exclude these irrelevant user catalogs from the utility catalogs scan by specifying their names in the XUCAT001- XUCAT999 DD statements. If the disks where these user catalogs reside are not accessible, you need to specify any existing disk volume serial number and unit along with the user catalog names on the XUCAT001- XUCAT999 DD statements.

A sample job to run HVTUSCR with TYPE=CATALOG can be found in member HVTUSCRC in the VTE SAMPLIB library.

TLMS

When using this option, HVTUSCR utility scans a TLMS EARL report produced in an earlier step of the same job. That report contains the current VMF virtual tapes that are not scratch (active) along with their vault location.

All virtual tapes not listed in this report are considered to be scratch.

Refer to "HVTUSCR process when TYPE specifies a tape management system" in ["Special HVTUSCR Utility Processes"](#) on page 340 for details on HVTUSCR process when TYPE=TLMS.

A sample job to run HVTUSCR with TYPE=TLMS can be found in member HVTUSCRS in the VTE SAMPLIB library.



Attention: It is mandatory for the EARL report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run started.

The report generator used by previous VTE releases can be used with the current VTE release.

CA-1

When using this option, HVTUSCR utility scans a CA-1 EARL report produced in an earlier step of the same job. That report contains the current TMC virtual tapes that are not scratch (active) along with their vault location.

All virtual tapes not listed in this report are considered to be scratch.

Refer to "HVTUSCR process when TYPE specifies a tape management system" in ["Special HVTUSCR Utility Processes"](#) on page 340 for details on HVTUSCR process when TYPE=CA-1.

A sample job to run HVTUSCR with TYPE=CA-1 can be found in member HVTUSCR1 in the VTE SAMPLIB library.



Attention: It is mandatory for the EARL report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run started.

The report generator used by previous VTE releases can be used with the current VTE release.

CONTROL-T

When using this option, HVTUSCR utility scans a CONTROL-T CTRPT report produced in an earlier step of the same job. That report contains the current MDB virtual tapes that are not scratch (active) along with their vault location.

All virtual tapes not listed in this report are considered to be scratch.

Refer to "HVTUSCR process when TYPE specifies a tape management system" in ["Special HVTUSCR Utility Processes"](#) on page 340 for details on HVTUSCR process when TYPE=CONTROL-T.

A sample job to run HVTUSCR with TYPE=CONTROL-T can be found in member HVTUSCRT in the VTE SAMPLIB library.



Attention: It is mandatory for the CTRPT report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run started.

The report generator used by previous VTE releases can be used with the current VTE release.

RMM

When using this option, HVTUSCR utility scans an IBM DFSMSrmm EDGHSKP report produced in an earlier step of the same job. The report contains the current TCDB tapes from which the virtual tapes that are not scratch (active), along with their vault location, are filtered for processing.

All virtual tapes not listed in this report are considered to be scratch.

Refer to "HVTUSCR process when TYPE specifies a tape management system" in ["Special HVTUSCR Utility Processes"](#) on page 340 for details on HVTUSCR process when TYPE=RMM.

A sample job to run HVTUSCR with TYPE=RMM can be found in member HVTUSCRR in the VTE SAMPLIB library.



Attention: It is mandatory for the EDGHSKP report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run started.

The report generator used by previous VTE releases can be used with the current VTE release.

ZARA

When using this option, HVTUSCR utility scans a ZARA ZARAUTL report produced in an earlier step of the same job. The report contains the current ZARA database virtual tapes that are not scratch (active) along with their vault location.

All virtual tapes not listed in this report are considered to be scratch.

Refer to "HVTUSCR process when TYPE specifies a tape management system" in ["Special HVTUSCR Utility Processes"](#) on page 340 for details on HVTUSCR process when TYPE=ZARA.

A sample job running HVTUSCR with TYPE=ZARA can be found in member HVTUSCRZ in the VTE SAMPLIB library.



Attention: It is mandatory for the ZARAUTL report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run started.

The report generator used by previous VTE releases can be used with the current VTE release.

Value range:

N/A

Default:

None

VOLSER=volser

volser

This parameter is used only when TYPE=VOLSER and specifies the virtual tape volume serial number to be handled by the utility.

Value range:

One- to six-character alphanumeric string.

Default: None

RECLAIM={YES|NO}

This parameter is used only when TYPE=VOLSER and indicates whether the specific virtual tape should be reclaimed from its deferred-scratch status. For other TYPE values, the reclaim is handled automatically and this parameter is ignored.

YES	The virtual tape should be reclaimed from its deferred-scratch status. No action is taken if the virtual tape has already been scratched.
NO	The virtual tape should not be reclaimed from its deferred-scratch status.
Value range:	N/A
Default:	None

FORCE={YES|NO}

This parameter indicates whether virtual tapes eligible for scratch will be scratched regardless their deferred-scratch conditions.

YES	Virtual tape eligible for scratch will be scratched regardless their deferred-scratch status.
NO	Virtual tape eligible for scratch will be scratched considering their deferred-scratch status.
Value range:	N/A
Default:	None.

LOCID=*locid*

This parameter is used when TYPE=VOLSER and when TYPE= {VLTUPD|VFTP}. When the LOCID parameter is specified and TYPE=VOLSER, the virtual tape is not scratched. Rather, it is marked 'to be vaulted' or 'to be returned' (according to the *locid* value). When TYPE is not VOLSER, the e-Vault location name is extracted from the tape management system database and this parameter is ignored.

When used with TYPE={VLTUPD|VFTP}, the LOCID parameter specifies the name of the remote off-site storage location that contains the e-Vaulted virtual tapes whose e-Vault indicators are to be updated. That *locid* should be the same as the LOCID parameter of one of the VLTRMT initialization statements in the HVTVLtxx member.

<i>locid</i>	Specifies either the local vault location name or a remote off-site storage location name to which this virtual tape is to be moved. <ul style="list-style-type: none">• If <i>locid</i> is similar to the OWNLOCID parameter specified on the VLTOWN initialization statement and the virtual tape currently resides in a remote off-site storage location, the virtual tape is marked 'to be returned'.• If <i>locid</i> is similar to one of the LOCID parameters in the VLTRMT initialization statements and the virtual tape currently resides in the local location, the virtual tape is marked 'to be vaulted' to that location.
Value range:	One- to eight-character alphanumeric string or blanks surrounded by quotes.
Default:	None

TMSRESET={YES|NO}

This parameter indicates whether to reset the Open/Close/EOV recorded information from the local TMS once the respective virtual tape has been actually scratched. Once a virtual tape is scratched, the associated virtual tape dataset on disk is deleted and the virtual tape cannot be recovered anymore.

However, sites that backup the virtual tapes, may find it useful to keep the TMS record in case the virtual tape is restored and recovered after it has been inadvertently scratched.

The TMSRESET parameter is used only if the local TMS is DFSMSrmm. For any other TMS, the parameter is checked for valid syntax and is further ignored.

YES	The TMS data is to be reset once the virtual tape is actually scratched.
NO	The TMS data is not to be reset.
Value range:	N/A
Default:	YES

TYPE={VLTUPD|VFTP}LOCID=locid[,FORCE={YES|NO}]
TYPE={VLTUPD|VFTP}

This SYNC statement uses the same TYPE, LOCID, and FORCE parameters that are described above. The VLTUPD|VFTP value is described below.

Note: If more than one SYNC initialization statements is present, HVTUSCR considers only the first encountered statement.

This TYPE parameter indicates whether VDB entries containing locations or virtual tape dataset names that do not match the e-Vaulted virtual tape values will be overridden according to the FTP data. Mismatch of one of these values may indicate that a wrong off-site storage location is used, therefore this option should be used with caution.

The VLTUPD and VFTP parameters are the same in function. However, it is best to use the VLTUPD parameter. The VFTP parameter was used for an older version of this product and is provided for backward compatibility.

When using this option, and depending on the TYPE parameter value (3 or 5), HVTUSCR invokes the FTP program or the Cloud Tape Connector (CTC) to produce a report of dataset names whose high-level qualifier is identical to the high-level qualifier of the VTE virtual tape data sets.

That report contains all e-Vaulted virtual tape data sets currently found in the off-site storage location pointed to by the associated LOCID parameter value.

Keep the following considerations in mind:

- For 'known' virtual tapes, the VLTUPD parameter updates the appropriate VDB entry with location, virtual tape dataset name and e-Vault status. Please consider the associated FORCE parameter for additional options.
- For 'unknown' virtual tapes, the VLTUPD parameter ignores the entry.

A sample job to run HVTUSCR with TYPE=VLTUPD (or TYPE=VFTP) can be found in member HVTUSCRF in the VTE SAMPLIB library.

Value range:	N/A
Default:	None

TYPE=type[,VOLSER=volser][,LOCID=locid]
[,RECLAIM=reclaim]
[,TMSRESET=tmsreset][,FORCE=force]

This SYNC statement uses the same TYPE, VOLSER, LOCID, RECLAIM, and TMSRESET parameters that are described above. The FORCE parameter has been changed as described below.

Note: If more than one SYNC initialization statements is present, HVTUSCR considers only the first encountered statement.

FORCE=({YES|NO}[, {YES|NO}])

FORCE=({YES|NO}[, {YES|NO}])

Parameter's scope:

- For TYPE=VOLSER – The parameter's scope is the VOLSER specified in the associated statement.
- For all TYPE values except VOLSER – The parameter's scope is all VOLSER values included in the associated list.

The FORCE parameter indicates whether certain initial conditions will be overridden during the SYNC process. The parameter has two sub-parameters as follows:

1. The first sub-parameter is used in conjunction with either the deferred-scratch status or the e-Vault indicators of virtual tapes.
2. The second sub-parameter is used in conjunction with virtual tapes remote-scratch.

For the first sub-parameter, valid values are YES or NO, and are described below.

YES For all TYPE parameter values, except VLTUPD (or VFTP), virtual tapes eligible for scratch will be scratched regardless their deferred-scratch status.

Note: Virtual tapes that belong to tapepools that specify DEFERBY=9999 are never scratched regardless the FORCE first sub-parameter value.

For TYPE=VLTUPD (or VFTP), VDB entries that contain non-blank locations or virtual tape dataset names values that do not match the e-Vaulted virtual tapes values will be overridden according to the information on the remote off-site storage location.

Note: Mismatch of one of these values may indicate that a wrong off-site storage location is used therefore this option should be used with caution.

NO For all TYPE parameter values, except VLTUPD (or VFTP), virtual tapes eligible for scratch will be scratched considering their deferred-scratch status.

For TYPE=VLTUPD (or VFTP), if the VDB entry contains a location and/or virtual tape dataset name, they will not be overridden.

Value range: N/A

Default: NO

For the second sub-parameter, valid values are YES or NO, and are described below.

YES Remote-scratched virtual tapes will be returned to the scratch pool regardless of whether the virtual tape dataset was successfully deleted from the remote off-site storage location. This option should be used only when the remote off-site storage location that contains e-Vaulted virtual tape will never become operational again. If used with temporary non-operational off-site storage locations, orphan datasets may accumulate on it.

NO Remote-scratched virtual tapes will not be returned to the scratch pool as long as the virtual tape datasets are not successfully deleted from the remote off-site storage location.

Value range:	N/A
Default:	NO

For the second sub-parameter, valid values are YES or NO, and are described below.

Special HVTUSCR Utility Processes

HVTUSCR process when TYPE=VOLSER is coded and LOCID=locid is not specified

For each virtual tape specified in the VOLSER parameter (that does not contain the LOCID parameter), HVTUSCR scratches it by taking the following steps:

- If the virtual tape is in the local location and no deferred-scratch conditions apply to it, it is scratched.
- If the virtual tape is in the local location and deferred-scratch conditions apply to it, it is marked for deferred-scratch.
- If the virtual tape is in the local location and its deferred-scratch time expired, it is scratched.
- If the virtual tape is in a remote off-site storage location, it is either marked for deferred-scratch or for remote-scratch.

In addition, each virtual tape (even though not specified in the VOLSER parameter) whose deferred-scratch time expired is either scratched or marked for remote-scratch.

HVTUSCR process when TYPE specifies a tape management system

When TYPE specifies a tape management system, HVTUSCR synchronizes the VDB according to the tape management system database. In all these options, HVTUSCR scans a report produced by a respective tape management utility. The report of TLMS, CA-1, CONTROL-T, and ZARA contains only the active (not scratch) virtual tapes along with their vault location. The report of RMM contains all the tapes in the TCDB along with their status.

For each virtual tape found in the VDB, HVTUSCR takes the following steps:

- When the virtual tape is marked scratch in the tape management system report:
 - If no deferred-scratch conditions apply to the virtual tape or the deferred-scratch time expired, it is scratched or marked for 'remote-scratch'.
 - If deferred-scratch conditions apply to the virtual tape, it is marked for deferred-scratch.
- When the virtual tape is marked active in the tape management system report:
 - If the virtual tape is marked for deferred-scratch, it is reclaimed.
 - If the virtual tape residence is local but it is physically in a remote location, it is marked 'to be returned'.
 - If the virtual tape residence is remote but it is physically in the local location, it is marked 'to be vaulted'.



Attention: If the virtual tapes are controlled by an EDM and not defined to the tape management system, the respective TPOOL initialization statement must specify DEFERBY=9999. If this rule is not observed, virtual tapes may be inadvertently scratched (since they are not listed in the report as active virtual tapes).

General considerations when multiple options are specified on the SYNC initialization statement:

- If TYPE=VOLSER,VOLSER=volser,RECLAIM=YES is specified, the virtual tape is reclaimed and not scratched
- If TYPE=VOLSER,VOLSER=volser,RECLAIM=YES,LOCID=locid is specified, the virtual tape will be only reclaimed and the LOCID parameter is ignored. On a second run of the HVTUSCR utility, the LOCID parameter will be considered and the virtual tape will be marked 'to be vaulted' or 'to be returned'.

- If TYPE=VOLSER,VOLSER=*volser*,FORCE=YES,LOCID=*locid* is specified, the FORCE parameter is ignored and the virtual tape will not be scratched. Instead, it will be marked 'to be vaulted' or 'to be returned'.
- For all TYPE parameter values, if a virtual tape is to be reclaimed from a previous deferred-scratch condition and also is to be marked as 'to be vaulted' or 'to be returned', the virtual tape will be only reclaimed. On a second run of the HVTUSCR utility the virtual tape will be marked 'to be vaulted' or 'to be returned'.

Return Codes

The following return codes are returned by HVTUSCR:

0	OK
4	At least one virtual tape could not be handled properly. In a DFSMSrmm environment, the EDGXCI API failed at least once to update the DFSMSrmm control data set.
8	<ul style="list-style-type: none"> • Operator replied N to HVT335W message. • Severe error, utility terminated.
12	Utility abended or cancelled.

HVTUTAP – Adding/Deleting Virtual Tapes

HVTUTAP utility can be used to:

- Add virtual tapes to the VDB.

The added virtual tapes are marked as scratch (that is, eligible for mount on a scratch tape request). Since each new added virtual tape is automatically initiated, there is no need for any further run of the IEHINITT (or equivalent) utility.

- Delete one or more virtual tapes from the VDB, provided they are scratch and are not currently in use by a running job.
- Set and reset the 'Mount-On-Specific-Request-Only' attribute for existing virtual tapes.

The utility will not change a virtual tape status if the tapepool the virtual tape belongs to or the VDB are in Read_Only mode.

To keep from filling the journal file on mass updates of the VDB, the HVTUTAP utility does not journal the VDB updates (regardless of the journal enable/disable setting). Journaling can be explicitly enabled on each initialization statement.

A sample job to run HVTUTAP can be found in member HVTUTAP in the VTE SAMPLIB library.

Input

The input to HVTUTAP utility consists of at least one and as many as 20,000 TAP initialization statements.

```
TAP      FUNCTION={ADD|DEL|MNTSPEC|MNTALL},
          FIRST=volser1,
          LAST=volser2
          [,INCR={ALPHANUM|NUM}]
          [,JRN={ENABLE|DISABLE}]
```

TAP

The TAP initialization statement defines the function and parameters upon which:

- Virtual tapes are to be added or deleted from the VDB.
- The 'Mount-On-Specific-Request-Only' attribute is set or reset.

FUNCTION={ADD|DEL|MNTSPEC|MNTALL}

ADD	<p>Specifies that new virtual tapes are to be initiated and added to the VDB.</p> <p>When ADD-ing new virtual tapes, no virtual tape datasets are allocated on disk. This function just defines appropriate entries in the VDB.</p>
DEL	<p>Specifies that existing virtual tapes are to be deleted from the VDB (provided they are scratch and not in use).</p>
MNTSPEC	<p>Specifies that existing virtual tapes are to be assigned the 'Mount-On-Specific-Request-Only' attribute. The existing virtual tape can be scratch or active but not 'in-use'.</p> <p>Virtual tapes that bear this attribute will not be eligible for mount on a non-specific scratch request, even though they are scratch.</p> <p>The 'Mount-On-Specific-Request-Only' attribute can be set and reset for virtual tapes already found in the VDB. If you need to add new virtual tapes with this attribute, you must use two initialization statements, as follows:</p> <pre>TAP FUNCTION=ADD, FIRST=volser1, LAST=volser2 TAP FUNCTION=MNTSPEC, FIRST=volser1, LAST=volser2</pre>
MNTALL	<p>Specifies that existing virtual tapes are to be made eligible for all types of mount. The existing virtual tape can be scratch or active but not 'in-use'.</p>
Value range:	N/A.
Default:	None.

FIRST=volser1

<i>volser1</i>	<p>Specifies a volser identifying the first virtual tape volume serial number in the range of volsers to be processed by the utility.</p> <p>Value range: A 1 to 6 alphanumeric character string.</p> <p>Default: None.</p>
----------------	---

LAST=volser2

<i>volser2</i>	<p>Specifies a volser identifying the last virtual tape volume serial number in a range of volsers to be processed by the utility.</p> <p>Value range: A 1 to 6 alphanumeric character string.</p> <p>Default: None.</p>
----------------	--

Note: If only one volser is to be processed, *volser1* and *volser2* should be identical.

INCR={ALPHANUM|NUM}

Specifies the method by which the incrementation will be done from *volser1* to *volser2*.

ALPHANUM	Specifies that the volsers range (from <i>volser1</i> to <i>volser2</i>) increment will consist of alphanumeric characters.
NUM	Specifies that the volsers range (from <i>volser1</i> to <i>volser2</i>) increment will consist of numeric characters only.
	Value range: N/A.
	Default: NUM.

JRN={ENABLE|DISABLE}

Specifies whether the VDB journaling will be enabled or disabled for the processed virtual tapes in the range from *volser1* to *volser2*. This parameter affects the journaling process for HVTUTAP utility only.

ENABLE	Specifies that the VDB journaling will be enabled for the specified range (subject to the OPTIONS initialization statement setting).
DISABLE	Specifies that the VDB journaling will be disabled for the specified range (regardless of the OPTIONS initialization statement setting).
	Value range: N/A.
	Default: DISABLE.

Return Codes

The following return codes are returned by HVTUTAP:

0	OK
4	At least one to be added, deleted or attributed was not eligible for the requested operation.
8	Severe error, utility terminated.
12	Utility abended or cancelled.

HVTUV2V – Volume to Volume Copy

HVTUV2V utility can be used to copy a virtual tape onto a real tape or a real tape onto a virtual tape. The copy is performed on a volume-to-volume basis, that is, the whole input tape (either virtual or real) is copied onto the output tape. This includes all labels, tape marks and files. Following that copy, the output tape is totally overwritten (including the labels) and it becomes an identical image of the input tape.

An input tape cannot be split into multiple output tapes. If such a condition occurs, the copy fails.

More than one tape can be copied by the utility. If one of the copy processes fails, the utility resumes with the copy of the next tape.

The utility will not copy to a virtual tape if the tapepool the virtual tape belongs to or the VDB are in Read_Only mode.

The copy to the virtual tape or from the virtual tape is performed without actually mounting the virtual tape on a VTE virtual device. The access to the virtual tape datasets is done internally by HVTUV2V by simulating the server functionality, but without actually using it. In fact, the VTE server may be inactive while using HVTUV2V.

Using that copy method, two major advantages are accomplished:

- Possibility to use data found on virtual tapes even though the VTE server is down. By copying the virtual tape to a real tape, the data is immediately available to applications that need it.
- Possibility to copy real tapes to virtual tapes by retaining their original volume serial numbers. This enables sites to transparently move data between these two media.

Special Considerations

When using the HVTUV2V utility, some issues require special consideration. These issues are described below.

- If the utility is cancelled while copying a tape, the virtual tape (either input or output) remains in an 'in-use' status and the HVTURCV utility should be run in order to recover it.
- If as a result of a 'real to virtual' copy an 'unknown' virtual tape is added to the VDB, but for some reason the copy does not end successfully, the new created virtual tape is deleted. This is done in order to revert the virtual tape status (as it was before the copy started).
- Use of BLP processing should be permitted as follows:
 - For JES2: The JES2PARM's JOBCLASS statement should specify BLP=YES, for the class the utility will use.
Note that the default for this parameter as shipped by IBM is BLP=NO.
 - For JES3: The JES3PARM's CIPARM option list should enable the BLP processing.
Note that the default for this parameter as shipped by IBM is that the BLP processing is enabled.
- If the VDSN initialization statement specifies QUAL2=%TDSN or QUAL3=%TDSN, and the copy is from real to virtual, the respective virtual tape dataset name qualifier will be TEMPORAR.
- If the copy is from real to virtual, the virtual tape dataset will not be compressed (regardless the COMPRESS parameter in the appropriate TPOOL initialization statement).
- If the copy is from virtual to real, the output real tape must be initialized (by the IEHINITT utility or equivalent) prior to the copy.
- If new not-initiated tapes are to be used, specify RBLP=YES on the COPY initialization statement. After the copy completes, make sure to stick the right external label on the real tape.
- The utility does not change the catalog reference from the old input media to the new output media. That means that either:
 - The dataset(s) that were copied should be recataloged.
 - Jobs using the copied dataset(s) should explicitly specify the volume serial number and unit of the new media.

A sample job to run HVTUV2V can be found in member HVTUV2V in the VTE SAMPLIB library.

Input

The input to HVTUV2V utility consists of at least one and as many as 150,000 COPY initialization statements.

COPY	VOLSER=volser, FROM={R2V V2R}, RUNIT=unit [,RBLP={YES NO}] [,VREPLACE={YES NO}] [,REXPDT=expdt] [,RVLD={YES NO}]
------	--

COPY

The COPY initialization statement defines one tape to be copied.

VOLSER=volser

volser	Specifies the tape volume serial number that is to be copied. The real tape device for this volser will be dynamically allocated and the mount request will follow.
--------	---

Value range: One- to six-character alphanumeric string.
Default: N/A

FROM={R2V|V2R}

R2V Specifies that a real tape is copied onto a virtual tape. In that case, the real tape is opened for input only.

V2R Specifies that a virtual tape is copied onto a real tape. In that case, the real tape is opened for output.

Value range: N/A
Default: N/A

RUNIT=*unit*

unit Specifies the esoteric or generic name of a real tape device the real tape will be mounted on.

Value range: One- to eight-character alphanumeric string.
Default: N/A

RBLP={YES|NO}

YES Specifies that the label verification of the real tape will be bypassed. That means that:

- For input real tapes, the operator may mount a wrong tape that will be copied to a virtual tape.
- For output real tapes, there is no need to init them before the actual copy.

NO Specifies that the label verification of the real tape will be performed. If a wrong tape is mounted, the operator will be given a chance to re-mount the correct tape.

Value range: N/A
Default: NO

VREPLACE={YES|NO}

YES Specifies that if a copy is to be done onto an existing virtual tape, that virtual tape is to be overwritten.

NO Specifies that an existing virtual tape is not to be overwritten. If a copy is attempted on such an existing virtual tape, this copy is terminated.

Value range: N/A
Default: NO

REXPDT=*expdt*

expdt Specifies the EXPDT value to be associated with the real tape allocation. Note that if the output tape is real, this value will NOT replace the original EXPDT value associated with the input tape. The EXPDT value relevance is related only to tape management systems that might analyze this EXPDT value and take some actions according to it.

Value range: Any numeric value of the form yyddd where the yy is a two-digit year number (through 99) and ddd is a three-digit day number from 000 through 366.

Default: If REXPDT is not specified, no EXPDT value will be associated with the real tape allocation.

RVLD={YES|NO}

After copying a real SMS-managed tape onto a virtual tape and trying to mount the copied virtual tape (by explicitly specifying its volume serial number and virtual device), MVS overrides these parameters and mounts the real tape (according to the old tape volume entry specifications in the VOLCAT).

To overcome the problem, the tape volume entry should be deleted from the VOLCAT. The RVLD parameter provides that option.

YES	Specifies that the input real SMS-managed tape volume entry is to be deleted once its copy complete successfully. This option is ignored if the input tape is not real (that is, if FROM=V2R is specified).
NO	Specifies that the input real SMS-managed tape volume entry is not to be deleted. Value range: N/A Default: NO

Return Codes

The following return codes are returned by HVTUV2V:

0	All requested tapes have been copied successfully.
4	At least one tape copy failed.
8	Severe error, utility terminated.
12	Utility abended or cancelled.

HVTUVBK – VDB Backup

Specifies that the input real SMS-managed tape volume entry is not to be deleted. HVTUVBK utility can be used to mark the last journal file record that reflects the last update done to the VDB. No VDB updates (that require writing over this last journal file record) are allowed until the VDB is backed-up and the journal file last accessed record is reset.

This utility does not backup the VDB by itself. Instead, it is used in conjunction with an IDCAMS REPRO that backs-up the VDB.

A sample job to backup the VDB and mark the last usable journal file block can be found in member HVTUVBK in the SAMPLIB library.

The HVTUVBK job contains three steps:

- Before the actual backup starts, the utility marks the last block in the journal file indicating the last "secured" operation done to the VDB.
- IDCAMS REPRO to create a backup of the VDB.

The UNIT parameter should point to a real tape or disk. Under no circumstances should the VDB be backed up on a virtual tape (since deadlock may occur when VTE accesses it).

- The utility resets the journal so it can be overwritten till it reaches again the block that was marked on the first step.

The above method ensures that the interim updates to the VDB (done during its backup) are not considered secured and will not be overwritten until the next VDB backup

Input

The input to HVTUVBK utility consists of one BKP initialization statement.

```
BKP WHEN={START|END}
```

BKP

The BKP initialization statement defines the point in time HVTUVBK runs, either before or after the VDB backup.

WHEN={START|END}

START	Specifies that a VDB backup is about to start and the utility should temporarily mark the journal file block that secures the last update done to the VDB.
END	Specifies that a VDB backup successfully ended and the utility should permanently mark the journal file block that secures the last update done to the VDB.
Value range:	N/A.
Default:	None.

Return Codes

The following return codes are returned by HVTUVSI:

0	OK
8	Severe error, utility terminated.

HVTUVSI – VDB Initialization

HVTUVSI utility can be used to initialize a newly defined VDB file. An attempt to initialize an existing non-empty VDB file will fail. The utility will fail if the VDB is set to Read_Only mode.

A sample job to run HVTUVSI can be found in member HVTUVSI in the VTE SAMPLIB library.

Input

No input initialization statements are required by HVTUVSI utility.

The step that precedes the HVTUVSI utility in the HVTUVSI job defines the VDB file using IDCAMS utility. Special attention should be paid to its parameters:

RECORDS

The number of virtual tapes expected to be contained in the VDB.

Value range:	N/A.
Default:	200,000.

CISZ

The cluster's data component control interval size. The optimal value for a 3380 disk is 22528, and the optimal value for a 3390 disk is 18432.

Value range:	N/A.
Default:	18432.

RECORDSIZE

The size of one record representing one virtual tape entry in the VDB. The current record size is 256 and should not be changed.

Value range: N/A.
Default: 256.

Return Codes

The following return codes are returned by HVTUVSI:

0	OK
8	Severe error, utility terminated.

Chapter 33. Implementing VTE and Tape Management Systems

This chapter explains how VTE can be implemented in sites that run a tape management system. Details are given for the definitions of both VTE and the tape management system, and examples are set in order to illustrate these definitions.

Implementing in a CA CA-1 Environment

In order to implement VTE in a CA-1 environment, the virtual tapes should be defined to both VTE and CA-1.

In addition, if your site implements CA-1 tapepools for real tapes, and you wish to extend this support for virtual tapes, the tapepools should be defined to both VTE and CA-1.

After the definitions are set, CA-1 will control and manage the VTE virtual tapes like any other real tape.

This section explains how to:

- Add virtual tapes to the VTE VDB and the CA-1 TMC.
- Define tapepools to VTE and CA-1.
- Define tapepool rules to CA-1.

This example assumes the following tapepools and tapepool ranges:

- T_VTSWIZ – Tapepool name
- EC0100 – First virtual tape in the default tapepool
- EC0199 – Last virtual tape in the default tapepool
- EC1000 – First virtual tape in the T_VTSWIZ tapepool
- EC1099 – Last virtual tape in the T_VTSWIZ tapepool

According to the above:

- Volumes EC0100 to EC0199 belong to the default tapepool (named \$\$DPLT).
- Volumes EC1000 to EC1099 belong to the T_VTSWIZ tapepool.

It is also assumed that jobs whose jobname starts with M20WIZ get scratch tapes from T_VTSWIZ tapepool and all other jobs get scratch tapes from the default \$\$DPLT tapepool.

Add virtual tapes

Virtual tapes can be added to the virtual data base (VDB) in VTE and in the TMC for CA-1. The statements for each method are described below.

Add virtual tapes to the VTE VDB

Use HVTUTAP utility to add virtual tapes to the VDB.

For example, the following initialization statements are to be supplied to the utility:

```
TAP      FUNCTION=ADD,
          FIRST=EC0100,
          LAST=EC0199
TAP      FUNCTION=ADD,
```

```
FIRST=EC1000,  
LAST=EC1099
```

Add virtual tapes to the CA-1 TMC

If you have already predefined tapes in the TMC, you may use them as virtual tapes. If not, use TMSFORMT program (from CAI.SAMPJCL(JOBFORMT)) to add the virtual tapes to the TMC. The following statement is a sample statement supplied to TMSFORMT:

```
EXTEND TMC=(990100-990199,991000-991099),DSNB=50
```

Note: Volumes 99xxxx should be translated to ECxxxx by CA-1 exits TMSUX2E and TMSUX2U.

Define Tape pools

Tape pools can also be defined to VTE and to CA-1. Examples of initialization statements are provided below.

Define tape pools to VTE

VTE tape pools, if implemented, are defined in the HVTPOLxx member in the SAMPLIB library.

In the following example, the initialization statements define the VTE tape pools:

```
TPOOL      NAME=T_VTSWIZ,  
            DPOOL=D_VTSWIZ,  
            RANGE1=(EC1000,EC1099),  
            TAPECAP=(610,10),  
            CKPINTVL=0  
TPOOL      NAME=$$DPLT,  
            DPOOL=D_VTSWIZ,  
            RANGE1=(0,0),<--- Default tapepool!!!  
            TAPECAP=(610,04),  
            CKPINTVL=0
```

Define tape pools to CA-1

CA-1 tape pools are defined in member TMOSCRxx in CAI.PPOPTION library. For example:

```
SCRPOOL=T_VTSWIZ,RANGE=EC1000-EC1099
```

In the above example:

SCRPOOL

Name of the tapepool. It is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.

RANGE

RANGE

Note: The default tapepool need not be defined in TMOSCRxx member.

Define Tapepool Rules to CA-1

CA-1 tapepool rules, upon which scratch tapes are selected from different tape pools, are defined in member TMONSMxx in the CAI.PPOPTION library.

For example:

```
JOB=M20WIZ*,POOL=T_VTSWIZ
```

In the above example:

JOB

All scratch tapes for jobs whose jobname starts with M20WIZ will be requested from T_VTSWIZ tapepool. All other scratch tapes will be requested with no specific tapepool.

POOL The name of the tapepool. This name is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.

Note: With this being the only tapepool rule, all other jobs will be assigned no tapepool for their scratch mount requests. For these requests VTE will mount scratch tapes from its default tapepool range, that is, virtual tapes from EC0100 to EC0199 range.

Implementing in a CA-Dynam/TLMS Environment

In order to implement VTE in a CA-Dynam/TLMS environment, the virtual tapes should be defined to both VTE and CA-Dynam/TLMS.

In addition, if your site implements CA-Dynam/TLMS tapepools for real tapes, and you wish to extend this support for virtual tapes, the tapepools should be defined to both VTE and CA-Dynam/TLMS.

After the definitions are set, CA-Dynam/TLMS will control and manage the VTE virtual tapes like any other real tape.

This section explains how to:

- Add virtual tapes to the VTE VDB and CA-Dynam/TLMS VMF.
- Define tapepools to VTE and CA-Dynam/TLMS.
- Define tapepool rules to CA-Dynam/TLMS.

This example assumes the following tapepools and tapepools ranges:

- T_VTSWIZ – Tapepool name
- EC0100 – First virtual tape in the default tapepool
- EC0199 – Last virtual tape in the default tapepool
- EC1000 – First virtual tape in the T_VTSWIZ tapepool
- EC1099 – Last virtual tape in the T_VTSWIZ tapepool

According to the above:

- Volumes EC0100 to EC0199 belong to the default tapepool (named \$\$DPLT).
- Volumes EC1000 to EC1099 belong to the T_VTSWIZ tapepool.

It is also assumed that jobs whose jobname starts with M20WIZ get scratch tapes from T_VTSWIZ tapepool and all other jobs get scratch tapes from the default \$\$DPLT tapepool.

Add Virtual Tapes

Add virtual tapes to the VTE VDB

Use HVTUTAP utility to add the virtual tapes to the VDB.

For example, the following initialization statements are to be supplied to the utility:

```
TAP      FUNCTION=ADD,
          FIRST=EC0100,
          LAST=EC0199
TAP      FUNCTION=ADD,
          FIRST=EC1000,
          LAST=EC1099
```

Add virtual tapes to the CA-Dynam/TLMS VMF

If you have already predefined tapes in the VMF, you may use them as virtual tapes. If not:

- Edit member TLMSIDCK (from CAI.PPOPTION) and add the new range of volumes. The following statements are sample statements coded in TLMSIDCK member:

```
RANGE VSN(EC0100 EC0199)
RANGE VSN(EC1000 EC1099)
```

- Use TLMSVMFU program (from CAI.CTL55SLD(CATVMFRS)) to add the virtual tapes to the VMF.

Define Tape pools

Tape pools can be defined to VTE and to CA Dynam/TMLS. The statements for these methods are described below.

Define tape pools to VTE

VTE tape pools are defined in the HVTPOLxx member in the SAMPLIB library.

In the following example, the initialization statements define the VTE tape pools:

```
TPOOL      NAME=T_VTSWIZ,
            DPOOL=D_VTSWIZ,
            RANGE1=(EC1000,EC1099),
            TAPECAP=(610,10),
            CKPINTVL=0
TPOOL      NAME=$DPLT,
            DPOOL=D_VTSWIZ,
            RANGE1=(0,0),<--- Default tapepool!!!
            TAPECAP=(610,04),
            CKPINTVL=0
```

Define tape pools to CA-Dynam/TLMS

CA-Dynam/TLMS tape pools are defined in member CTOSCRxx in CAI.PPOPTION library.

```
SCRPOOL=T_VTSWIZ,RANGE=EC1000-EC1099
```

In the above example:

SCRPOOL	The name of the tapepool. This name is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.
RANGE	Range of virtual tapes in this tapepool. It is identical to the relevant range in the VTE tapepool.

Note: The default tapepool need not be defined in CTOSCRxx member.

Define Tapepool Rules to CA-Dynam/TLMS

CA-Dynam/TLMS tapepool rules, upon which scratch tapes are selected from different tape pools, are defined in member CTONSMxx in CAI.PPOPTION library.

For example:

```
JOB=M20WIZ*,POOL=T_VTSWIZ
```

In the above example:

JOB	All scratch tapes for jobs whose jobname starts with M20WIZ will be requested from T_VTSWIZ tapepool. All other scratch tapes will be requested with no specific tapepool.
POOL	Name of the tapepool. It is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.

Note: With this being the only tapepool rule, all other jobs will be assigned no tapepool for their scratch mount requests. For these requests, VTE will mount scratch tapes from its default tapepool range, that is, virtual tapes from EC0100 to EC0199 range.

Implementing in an IBM DFSMSrmm Environment

In order to implement VTE in a DFSMSrmm environment, the virtual tapes should be defined to both VTE and DFSMSrmm.

In addition, if your site implements DFSMSrmm tapepools for real tapes and you wish to extend this support for virtual tapes, the tapepools should be defined to both VTE and DFSMSrmm. After the definitions are set, DFSMSrmm will control and manage the VTE virtual tapes like any other real tape.

This section explains how to:

- Add virtual tapes to the VTE VDB and DFSMSrmm TCDB.
- Define tapepools to VTE and DFSMSrmm.
- Define tapepool rules to DFSMSrmm.

This example assumes the following tapepools and tapepools ranges:

- T_VTSWIZ – Tapepool name
- EC0100 – First virtual tape in the default tapepool
- EC0199 – Last virtual tape in the default tapepool
- EC1000 – First virtual tape in the T_VTSWIZ tapepool
- EC1099 – Last virtual tape in the T_VTSWIZ tapepool

According to the above:

- Volumes EC0100 to EC0199 belong to the default tapepool (named \$\$DPLT).
- Volumes EC1000 to EC1099 belong to the T_VTSWIZ tapepool.

It is also assumed that jobs whose jobname starts with M20WIZ get scratch tapes from T_VTSWIZ tapepool and all other jobs get scratch tapes from the default \$\$DPLT tapepool.

Add Virtual Tapes

Virtual tapes can be added to the VDB for VTE and the TCDB for DFSMSrmm. The statements for these methods are described below.

Add virtual tapes to the VTE VDB

Use HVTUTAP utility to add virtual tapes to the VDB.

For example, the following initialization statements are to be supplied to the utility:

```
TAP      FUNCTION=ADD,
          FIRST=EC0100,
          LAST=EC0199
TAP      FUNCTION=ADD,
          FIRST=EC1000,
          LAST=EC1099
```

Add virtual tape to the DFSMSrmm TCDB

If you have already predefined tapes in the TCDB, you may use them as virtual tapes. If not, use IKJEFT01 program to add the virtual tapes to the TCDB. The following statements are to be supplied to

IKJEFT01. Note that the first range definition (the default tapepool) does not contain any POOL parameter.

```
RMM
ADDVOLUME      EC0100 STATUS(SCRATCH) COUNT(100)    -
INITIALIZE(N)  -
MEDIANAME(EC0)          RELEASEACTION(SCRATCH) -
OWNER(M20)
ADDVOLUME      EC1000 STATUS(SCRATCH) COUNT(100)    -
INITIALIZE(N)  -
MEDIANAME(EC1) POOL(EC1*) RELEASEACTION(SCRATCH) -
OWNER(M20)
```

Define tapepools to VTE

Tapepools can be defined to VTE and to DFSMSrmm. The statements for these methods are described below.

Define tapepools to VTE

VTE tapepools are defined in the HVTPOLxx member in the SAMPLIB library.

In the following example, the initialization statements define the VTE tapepools:

```
TPOOL          NAME=T_VTSWIZ,
                DPOOL=D_VTSWIZ,
                RANGE1=(EC1000,EC1099),
                TAPECAP=(610,10),
                CKPINTVL=0
TPOOL          NAME=$$DPLT,
                DPOOL=D_VTSWIZ,
                RANGE1=(0,0),<--- Default tapepool!!!
                TAPECAP=(610,04),
                CKPINTVL=0
```

Define tapepools to DFSMSrmm

DFSMSrmm tapepools are defined in member EDGRMM00 in SYS1.PARMLIB library. For example:

```
/* ----- VTSWIZ tapepool ----- */
VLPPOOL DESCRIPTION('VTSWIZ POOL') -
EXPDTCHECK(N)          -
MEDIANAME(EC1)          -
NAME(T_VTSWIZ)          -
TYPE(R)                 -
SYSID(MM90)             -
PREFIX(EC1*)
/* ----- Default tapepool ----- */
VLPPOOL DESCRIPTION('$$DPLT') -
EXPDTCHECK(N)          -
MEDIANAME(EC0)          -
NAME($$DPLT)           -
TYPE(R)                 -
SYSID(MM90)             -
PREFIX(*)
```

In the above example:

- | | |
|---------------|---|
| NAME | The name of the tapepool. This name is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE. |
| PREFIX | Value that reflects a range of virtual tapes. All virtual tapes whose volume serial number starts with this prefix, belong to this tapepool. For example, if PREFIX(EC1*), the associated range in VTE is RANGE1=(EC1000,EC1099). |

Note:

- For the default tapepool, the prefix value is PREFIX(*)
- When defining the tapepools the NAME parameter should be explicitly defined so that the pool name will be displayed on the mount messages.

The RACK parameter's value on the MNTMSG statements should be RACK(999) (in order to get the poolname at the end of the message text).

The VLPOOL statement that defines the default tapepool (that is, the tapepool from which scratch tapes are selected for jobs not defined in the EDGUX100 exit), should contain a NAME value that is similar to the VTE default tapepool's NAME value.

Define tapepool rules to DFSMSrmm

DFSMSrmm tapepool rules, upon which scratch tapes are selected from different tapepools, are defined in exit EDGUX100. The source usually resides in the SYS1.SAMPLIB library and the load module resides in the SYS1.LINKLIB library.

Changes to EDGUX100 exit should be done after label POOLTAB. For example:

```
POOLTAB DS 0F START OF TABLE
          DC CL8'M20WIZ* ',CL44*'
          DC CL6'EC1*' POOL NAME PREF
          DC AL1(PL100_SET_ACLOFF) BYPASS ACL L
          DC CL8*' ',CL44*'
          DC CL6*' ' DEFAULT POOL
          DC AL1(PL100_SET_ACLOFF) BYPASS ACL L
```

In the above example, all scratch tapes for jobs whose jobname starts with M20WIZ will be requested from T_VTSWIZ tapepool.

Implementing in a BMC CONTROL-T Environment

In order to implement VTE in a CONTROL-T environment, the virtual tapes should be defined to both VTE and CONTROL-T. All other scratch tapes will be requested from the default tapepool (\$\$DPLT).

In addition, if your site implements CONTROL-T tapepools for real tapes and you wish to extend this support for virtual tapes, the tapepools should be defined to both VTE and CONTROL-T.

After the definitions are set, CONTROL-T will control and manage the VTE virtual tapes like any other real tape.

This section explains how to:

- Add virtual tapes to the VTE VDB and CONTROL-T MDB.
- Define tapepools to VTE and CONTROL-T.
- Define tapepool rules to CONTROL-T.

This example assumes the following tapepools and tapepools ranges:

- T_VTSWIZ – Tapepool name
- EC0100 – First virtual tape in the default tapepool
- EC0199 – Last virtual tape in the default tapepool
- EC1000 – First virtual tape in the T_VTSWIZ tapepool
- EC1099 – Last virtual tape in the T_VTSWIZ tapepool

According to the above:

- Volumes EC0100 to EC0199 belong to the default tapepool (named \$\$DPLT).
- Volumes EC1000 to EC1099 belong to the T_VTSWIZ tapepool.

It is also assumed that jobs whose jobname starts with M20WIZ get scratch tapes from T_VTSWIZ tapepool and all other jobs get scratch tapes from the default \$\$DPLT tapepool.

Add Virtual Tapes

Virtual tapes can be added to the VDB for VTE and the MDB for CONTROL-T. The statements for these methods are described below.

Add virtual tapes to the VTE VDB

Use HVTUTAP utility to add virtual tapes to the VDB.

For example, the following initialization statements are to be supplied to the utility:

```
TAP          FUNCTION=ADD,
              FIRST=EC0100,
              LAST=EC0199
TAP          FUNCTION=ADD,
              FIRST=EC1000,
              LAST=EC1099
```

Add virtual tapes to the CONTROL-T MDB

If you have already predefined tapes in the MDB, you may use them as virtual tapes. If not:

- Define the VTE media in CTTPARM by adding a new CTTMEDDF definition as follows:

```
CTTMEDDF NAME=VTE,DESC=VTE,
          CAPACITY=500,STKPCNT=80,UNITNAME=hvtes
```

In the above CTTMEDDF definition, *hvtes* is the VTE virtual devices esoteric name.

- Use ADDVOLS job from the CONTROL-T JCL library to add the virtual tapes to the MDB. The following statements are sample statements supplied to the ADDVOLS job:

```
TYPERUN MODE=ADD
RANGE FIRST=EC0100,LAST=EC0199,MEDIA=VTE,SCRATCH=Y
RANGE FIRST=EC1000,LAST=EC1099,MEDIA=VTE,SCRATCH=Y
```

In the above example, MEDIA is the media name as defined in CTTPARM under the VTE CTTMEDDF definition.

Define Tape pools

Tape pools can be defined to VTE and to CONTROL-T. The statements for these methods are described below.

Define tape pools to VTE

VTE tape pools are defined in the HVTPOLxx member in the SAMPLIB library.

In the following example, the initialization statements define the VTE tape pools:

```
TPOOL        NAME=T_VTSWIZ,
              DPOOL=D_VTSWIZ,
              RANGE1=(EC1000,EC1099),
              TAPECAP=(610,10),
              CKPINTVL=0
TPOOL        NAME=$$DPLT,
              DPOOL=D_VTSWIZ,
              RANGE1=(0,0),<--- Default tapepool!!!
              TAPECAP=(610,04),
              CKPINTVL=0
```

Define tape pools to CONTROL-T

CONTROL-T tape pools are defined in screen (TP.S). For example:

```
----- POOL T_VTSWIZ                      TABLE $$POOL      -----(TP.S)
COMMAND ==>                                SCROLL==>  CRSR
+-----+-----+-----+-----+-----+-----+-----+-----+
| POOL NAME   T_VTSWIZ                                OWNER DP2 |
```



```

| DESCRIPTION
|=====
| VOLUMES      FROM EC1000      TO EC1099
|                  FROM          TO
|===== >>>>>>>>>>>>>>>> END OF POOL DEFINITION PARAMETERS <<<<<<<<<<<<<<<< =====

```

In the above example:

- POOL NAME** Name of the tapepool. It is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.
- VOLUMES** Range of virtual tapes in this tapepool. It is identical to the relevant range in the VTE tapepool.

Note: The default tapepool volumes need not be defined in screen (TP.S).

Define tapepool rules to CONTROL-T

Tapepool rules can be defined for CONTROL-T. The procedure for doing that is described in this section.

CONTROL-T tapepool rules upon which scratch tapes are selected from different tapepools are defined in screen (TR). For example:

```

RULE: T_VTSWIZ LIB CTT.V514.RULES                                TABLE: RULE4
COMMAND ===>                                                    SCROLL===> CRSR
+-----+-----+-----+-----+-----+-----+-----+-----+
| RULE NAME | T_VTSWIZ | GROUP |          |          |          |          |          |          |
| OWNER     | M20      | SEQUENCE | PRIORITY | CONTINUE | SEARCH | Y | PROD (Prod/Test) |
|           |          |          |          |          |          |   | (Y/N)             |
| DESCRIPTION |          |          |          |          |          |   |                   |
| DOCMEM    | VTSWIZ   | DOCLIB  | CTT.V514.DOC |          |          |   |                   |
|=====+-----+-----+-----+-----+-----+-----+
| ON JOBNAME  = M20WIZ*                                     And/Or/Not A
| ON DATASET  = *                                           And/Or/Not
|=====+-----+-----+-----+-----+-----+
|                   O N   O P T I O N S
| ACCOUNT     - Job account                                PGM       - Program name
| DATASET     - Dataset name                                UCB       - Unit address range
| JOBNAME     - Processing jobname                          USERID    - User-id of the job
| MEDIA       - Type of media                               VOLSER    - Volume or volume range
| MGMTCLAS    - SMS Management Class
|=====+-----+-----+-----+-----+
| DO POOL     = T_VTSWIZ
| DO
|=====+-----+-----+-----+-----+

```

In the above example:

- JOBNAME** All scratch tapes for jobs whose jobname starts with M20WIZ will be requested from T_VTSWIZ tapepool. All other scratch tapes will be requested with no specific tapepool.
- POOL** The name of the tapepool. This name is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE

Note: With this being the only tapepool rule, all other jobs will be assigned no tapepool for their scratch mount requests. For these requests VTE will mount scratch tapes from its default tapepool range, that is, virtual tapes from EC0100 to EC0199 range.

Implementing in an ASG-Zara Environment

In order to implement VTE in an ASG-Zara environment, the virtual tapes should be defined to both VTE and ASG-Zara.

In addition, if your site implements ASG-Zara tapepools for real tapes, and you wish to extend this support for virtual tapes, the tapepools should be defined to both VTE and ASG-Zara.

After the definitions are set, ASG-Zara will control and manage the VTE virtual tapes.

This section explains how to:

- Add virtual tapes to the VTE VDB.
- Add virtual tapes to the ASG-Zara Database.
- Define tapepools to VTE.
- Define tapepools to ASG-Zara.
- Define tapepool rules to ASG-Zara.

This example assumes the following tapepools and tapepools ranges:

- T_VTSWIZ – Tapepool name
- EC0100 – First virtual tape in the default tapepool
- EC0199 – Last virtual tape in the default tapepool
- EC1000 – First virtual tape in the T_VTSWIZ tapepool
- EC1099 – Last virtual tape in the T_VTSWIZ tapepool

According to the above:

- Volumes EC0100 to EC0199 belong to the default tapepool (named \$\$DPLT).
- Volumes EC1000 to EC1099 belong to the T_VTSWIZ tapepool.

It is also assumed that jobs whose jobname starts with M20WIZ get scratch tapes from T_VTSWIZ tapepool and all other jobs get scratch tapes from the default \$\$DPLT tapepool.

Add Virtual Tapes

The method for adding virtual tapes depends on whether you are adding tapes to the VTE virtual database (VDB) or to the ASG-Zara database. Both methods are described below.

Add virtual tapes to the VTE VDB

Use HVTUTAP utility to add virtual tapes to the VDB.

For example, the following initialization statements are to be supplied to the utility:

```
TAP      FUNCTION=ADD,
          FIRST=EC0100,
          LAST=EC0199
TAP      FUNCTION=ADD,
          FIRST=EC1000,
          LAST=EC1099
```

Add virtual tapes to the ASG-Zara Database

If you have already predefined tapes in ASG-Zara Database, you may use them as virtual tapes. If not, use the ZARAUTL procedure to add the virtual tapes to the Database. The following statements are to be supplied to ZARAUTL procedure:

```
RANGE ADD VOLSER=(EC0100-EC0199)
RANGE ADD VOLSER=(EC1000-EC1099)
```

Define Tapepools

Tapepools can be defined to VTE and to ASG-Zara. The statements for these methods are described below.

Define tapepools to VTE

VTE tapepools are defined in the HVTPOLxx member in the SAMPLIB library.

In the following example, the initialization statements define the VTE tapepools:

```

TPOOL      NAME=T_VTSWIZ,
           DPOOL=D_VTSWIZ,
           RANGE1=(EC1000,EC1099),
           TAPECAP=(610,10),
           CKPINTVL=0
TPOOL      NAME=$$DPLT,
           DPOOL=D_VTSWIZ,
           RANGE1=(0,0),<--- Default tapepool!!!
           TAPECAP=(610,04),
           CKPINTVL=0

```

Define tapepools to ASG-Zara

ASG-Zara tapepools are defined in screen 2.3.1. For example:

```

+-----+
|ASG-Zara Scratch Pool Definition Facility|
|COMMAND ==>|
|
|Primary commands:EDIT
|Line commands: D -Delete In -Insert Rn -Repeat (n=1-
|Specify Scratch Pool logical names and description.
|Sel Pool      Start End      Descip
|Name          Range Range
|=====
|-- T_VTSWIZ    EC1000EC1099 RANGE 1
|
|***** BOTTOM OF DATA *****
+-----+

```

In the above example:

POOL NAME	The name of the tapepool. This name is identical to the name assigned to the relevant TPOOL's NAME parameter in VTE.
START/END RANGE	The range of virtual tapes in this tapepool. It is identical to the relevant range in the VTE tapepool.

Note: The default tapepool volumes do not need to be defined in the above screen (2.3.1).

Define tapepool rules to ASG-Zara

Tapepool rules can be defined for ASG-Zara. The procedure for doing that is described in this section.

ASG-Zara tapepool rules upon which scratch tapes are selected from different tapepools are defined in screen 2.3.2.

For example:

```

+-----+
|ASG-Zara          Scratch Pool Candidate Selection Facility|
|COMMAND ==>|
|
|Primary commands: EDIT TEST CHECK FIND NEXT
|Line commands: D -Delete In -Insert Rn -Repeat (n=1-9)
|Test DSN:
|
|Sel Dataset Name      GDG      Jobname      Pool
|Pattern              (Y/N)    Pattern      Name
|=====
|-- *                  N        M20WIZ*      T_VTSWIZ
|
|***** BOTTOM OF DATA *****
+-----+

```

In the above example, all scratch tapes for jobs whose jobname starts with M20WIZ will be requested from T_VTSWIZ tapepool. All other scratch tapes will be requested from the default tapepool (\$\$DPLT).

Note: With this being the only tapepool rule, all other jobs will be assigned no tapepool for their scratch mount requests. For these requests, VTE will mount scratch tapes from its default tapepool, that is, virtual tapes from EC0100 to EC0199 range.

Chapter 34. VTE Exits

This appendix describes the Virtual Tape Emulation exits.

EXIT001

This VTE exit, EXIT001, is invoked before the buffer is written to virtual tape and after a buffer is read from virtual tape. This exit may change data in the buffer. However, the data length should not be modified.

Module Name

HVTX001

Description

Tape/data buffer inspection

Coding Rules

Not applicable

Called By

VTE server

Residency

The independent module is loaded at VTE initialization to its address space.

Default

The initial VTE installation provides a dummy HVTX001 module.

Linkage Attributes

AMODE (31)

RMODE (24)

RENT

Environment

State: Supervisor

Key: 8

Input

- Address of calling function:
 - READ
 - WRITE
- Address of data buffer.
- Length of data buffer.
- Address of an 8 byte area to be passed between successive calls of the exit. This area may be used by the exit to pass information between its invocations.

Output

R15 - 0: OK

Sample

HVTX001 in the SAMPLIB library

Installation Member

USERMOD and HVTU001 in the SAMPLIB library.
job for

RECEIVE/APPLY

Installation Process

The installation steps follow:

- Edit member HVTX001 in the SAMPLIB library. This member is a sample exit. You may modify it as needed. When finished, save it and exit the member.
- If the changed exit does not contain references to external routines, skip the following step.
- Edit HVTU001 member in the SAMPLIB library
 - Change the ++JCLIN to contain appropriate INCLUDE statements for the invoked external routines.
 - If needed, add to it the JCL statements required for the correct external routines resolve.

Submit the job. The job should end with a return code of 0.

Activate

Bounce the VTE server.

EXIT002

This VTE exit, EXIT002, defines one control unit type and a series of attached devices to be supported by the VTE UIM. These devices will be used by the HCD process to define the VTE virtual devices.

One exit contains definitions for all VTE installations in a single system. If a new VTE environment is installed, the exit should be updated with the values of the new installation. This enables one VTE UIM module to support all VTE installations in the system.

Coding Rules

The exit is composed of a series of HVTX002 macro invocations that perform several functions depending on the invoked mode:

START Mode

Defines the control unit type to be used by the VTE devices. The START mode is invoked only once.

DEFINE Mode

Defines one attached device unit type along with its associated generic name, UCB device type, and preference value. One invocation of the macro in DEFINE mode defines one device unit.

If you run multiple VTE servers in a single MVS system, you need to define multiple device units. This ensures that each server will have its own VTE virtual devices. To define multiple device units, you must code multiple HVTX002 macros in DEFINE mode. Thereafter, use HCD to connect them to MVS, define appropriate esoteric names, etc.

As many as 40 device unit types can be defined by multiple invocations of the HVTX002 macro with DEFINE mode. The values specified for each invocation should be unique, that is, no two device unit types may have similar generic names, preference values or UCB device types.

END Mode

Concludes the definitions and generates the required assembler fields.

Module Name

HVTX002

Description

VTE virtual device definition parameters for the VTE UIM.

Macro Field Explanations

The macro can be invoked in one of the following three modes: START, DEFINE, END.

Each mode has its own parameters as described below:

START	The first occurrence of the macro should be in START mode. This mode should be invoked only once. The following parameters are associated with this mode:
-------	---

CUNAME A new control unit name to which all the VTE devices (3480, 3490, 3590, and 3592) will be attached.

Value range: 1 to 8-character string

Default: None

Note: As with the current VTE version, CUNAME is not needed anymore. It is left here for compatibility reasons.

DEFINE

In this mode you can define one attached device unit type along with its unique specific values. The following parameters are associated with this mode:

DEVUNIT A new control unit name to which all the VTE devices (3480, 3490, 3590, and 3592) will be attached.

Value range: 1 to 8-character string

Default: None

GENERIC A new generic name to be assigned to this VTE device type.

Value range: 1 to 8-character string

Default: None

TYPE 3480 This device unit simulates a 3480 device.
 3490 This device unit simulates a 3490 device.
 3590 This device unit simulates a 3590 or 3592 device.

Further selection between 3590 and 3592 is done by using the SIMULATE parameter of the VDEVICE initialization statement.

Value range: N/A

Default: None

UCBTBYT1 The first byte of the 4 bytes that make up the UCBTYP field of this VTE virtual device. The other 3 bytes will be set automatically as x'048080', x'048081' or x'048083' according to the TYPE value specified earlier.

When a dataset is cataloged, the UCBTYP of the device on which the dataset was mounted is kept in the catalog. This UCBTYP is then used when the dataset is accessed in order to allocate an appropriate device. UCBTBYT1 should **not** be changed after product implementation is started; otherwise MVS will fail in selecting devices eligible for virtual tape mount as requested by cataloged datasets.

Value range: Any two-digit hexadecimal from 00 to FF (except 78).

Default: None

	PREFVAL	The unique device unit preference value. IBM's HCD Device Validation Support manual gives a list of preference values used by IBM. Here you should code a value that is not used by IBM for its devices. Value range: Integer from 1 to 9999 Default: None
END		The last occurrence of the macro should be in END mode. This mode should be invoked only once.

Called By

VTE UIM module (CBDUCnnn)

Residency

As a non-executable CSECT of module CBDUCnnn (where nnn is the selected UIM number) residing in SYS1.NUCLEUS.

Default

The initial VTE installation provides a default HVTX002 module link-edited into the VTE UIM.

EXIT002 should be re-APPLY-ed only if the default supplied values are not suitable for your site.

Linkage Attributes

AMODE(31)

RMODE(ANY)

NON RENT

Environment

State: N/A

Key: N/A

Input

N/A

Output

N/A

Sample

HVTX002 in the SAMPLIB library.

Installation Member

USERMOD and HVTU002 in the SAMPLIB library.
job for RECEIVE/
APPLY

Copy UIM to HVTJCUIM in the SAMPLIB library.
SYS1.NUCLEUS

Installation Process

If this is not the first VTE installation in the system, perform the following installation steps using the SAMPLIB libraries from the first installation:

1. Edit member HVTX002 in the SAMPLIB library. This is an assembler source program containing a series of macro definitions that define several device units. Customize these macros as needed.
2. Enter member HVTU002 in the SAMPLIB library. It contains a job that will RECEIVE and APPLY USERMOD HVTU002.
3. Submit the job. The job should end with a return code of 0.

Activate

- Enter member HVTJCUIM in the SAMPLIB library. It contains a job that will copy the VTE UIM to SYS1.NUCLEUS. Check that the UIM modules being copied are indeed the VTE UIM modules and that no other modules are inadvertently replaced.

Submit the job. The job should end with a return code of 0.

- In order to use the devices added with EXIT002, enter the HCD panels and add new devices with the just introduced device types, then activate the new configuration. Further information is available in [“Step 1: Hardware configuration” on page 229](#).

EXIT004

This VTE exit, EXIT004, is invoked after a virtual tape mount process is completed, the virtual tape dataset is allocated, opened, and marked 'in-use'.

The mount process and this invoked exit are triggered by the first I/O sent to the VTE virtual device when a new virtual tape is mounted. This I/O is usually sent by the application's OPEN process.

The exit caller may invoke other services based on the virtual tape characteristics.

Note:

- At the time the application sends the first I/O to the virtual tape and the mount process is initiated, the application's OPEN process did not complete yet.
- The exit does not get control if an application opens a virtual tape and this open does not result in a mount request (for example, the exit does not get control for virtual tapes that are passed to later steps by means of VOL=(,RETAIN) or DISP=(,PASS)).

Module Name

HVTX004

Description

Mounted virtual tape characteristics inspection.

Called by

VTE server

Residency

An independent module is loaded at VTE initialization to its address space.

Default

The initial VTE installation provides a dummy HVTX004 module.

Linkage Attributes

AMODE(31)

RMODE(24)

RENT

Environment

State:
Supervisor

Key: 8

Input

- Address of the calling function.
- MOUNT
- Address of the mounted virtual tape volume serial number.
- Address of the virtual tape dataset name.

- Address of the tape dataset name.
- Address of the jobname that mounted the virtual tape.
- Address of the jobid that mounted the virtual tape.
- Address of the unitname the virtual tape dataset was allocated on.
- Address of the Management Class the virtual tape dataset was allocated on.
- Address of the Storage Class the virtual tape dataset was allocated on.
- Address of the Storage Group the virtual tape dataset was allocated on.

Output

R15 - 0: OK

Sample

HVTX004 in the SAMPLIB library

Installation Member

USERMOD and HVTU004 in the SAMPLIB library.
job for RECEIVE/
APPLY

Installation Process

The installation steps follow:

1. Edit member HVTX004 in the SAMPLIB library. This member is a sample exit. You may modify it as needed. When finished, save it and exit the member.
2. If the changed exit does not contain references to external routines, skip the following step.
3. Edit HVTU004 member in the SAMPLIB library
 - Change the ++JCLIN to contain appropriate INCLUDE statements for the invoked external routines.
 - If needed, add to it the JCL statements required for the correct external routines resolve.

Activate

Bounce the VTE server.

EXIT006

This VTE exit, EXIT006, is invoked as a result of a specific virtual tape mount request but before the relevant VTE component started to process the request.

The exit can either accept or reject the specific mounted virtual tape. When the mounted virtual tape is rejected, message HVT500W is issued on the running server or utility job log.

The mount process and this invoked exit are triggered by the first I/O sent to the virtual device when a specific virtual tape is mounted. This I/O is usually sent by the application's OPEN process.

The exit caller may invoke other services based on the virtual tape characteristics.

Note:

- At the time the application sends the first I/O to the virtual tape and the mount process is initiated, the application's OPEN process did not complete yet.
- The exit does not get control if an application opens a virtual tape and this open does not result in a mount request (for example, the exit does not get control for virtual tapes that are passed to later steps by means of VOL=(, RETAIN) or DISP=(, PASS).
- The exit does not get control for non-specific (scratch) mount requests.

Module Name

HVTX006

Description

Specific virtual tape mounts are either accepted or rejected.

Called by

- VTE server
- HVTUV2V utility
- Applications that use the VTE Access Method (VAM).

Residency

An independent module is loaded at the VTE server or utility initialization time to its address space.

Default

The initial VTE installation provides a sample HVTX006 module.

Linkage Attributes

AMODE(31)

RMODE(24)

RENT

Environment

State:

Supervisor

Key: 8

Input

- The address of a 1-byte area that indicates the state of the specific mounted virtual tape.

XL1'01' - (Known, Scratch)

A specific virtual tape has been requested. The specific virtual tape is a 'Known' virtual tape (that is, it is defined in the VDB) and is scratch.

XL1'02' - (Known, Active)

A specific virtual tape has been requested. The specific virtual tape is a 'Known' virtual tape (that is, it is defined in the VDB) and is active.

XL1'03' - (Unknown)

A specific virtual tape has been requested. The specific virtual tape is an 'Unknown' virtual tape (that is, it is not defined in the VDB).

XL1'04' - (Unknown, Oper)

A non-specific virtual tape has been requested. The tapepool was exhausted and the operator replied with a specific 'Unknown' virtual tape to message HVT317W.

- The address of an 8-byte area that indicates the environment from which the exit was invoked.

CL8'SERVER'

The exit is invoked by the server.

CL8'UV2V'

The exit is invoked by the HVTUV2V utility.

CL8'VAM'

The exit is invoked by the VTE Access Method (VAM) OPEN function (HVTVAM OPEN processing).

- The address of a 6-byte area that contains the specific virtual tape volser (volume serial number).
- The address of a 44-byte area that contains the first accessed virtual tape dataset name (the name coded or derived from the DD statement).
- The address of an 8-byte area that contains the name of the job (jobname) that requested the virtual tape mount.

- The address of an 8-byte area that contains the program name that requested the virtual tape mount.

Output

R15 - 0: OK

Sample

HVTX004 in the SAMPLIB library

Installation Member

USERMOD and HVTU004 in the SAMPLIB library.
job for RECEIVE/
APPLY

Installation Process

The installation steps follow:

1. Edit member HVTX004 in the SAMPLIB library. This member is a sample exit. You may modify it as needed. When finished, save it and exit the member.
2. If the changed exit does not contain references to external routines, skip the following step.
3. Edit HVTU004 member in the SAMPLIB library
 - Change the ++JCLIN to contain appropriate INCLUDE statements for the invoked external routines.
 - If needed, add to it the JCL statements required for the correct external routines resolve.

Activate

Bounce the VTE server.

Output

R15 - 0

Accept the mounted virtual tape.

R15 - 8

Reject the mounted virtual tape. When the mounted virtual tape is rejected, message HVT500W is issued on the running server or utility job log.

The result of rejecting a virtual tape depends on the environment that invoked the exit:

Server

This job is EQC-ed.

UV2V

The COPY of the specific virtual tape terminates and message HVT393I is issued. The utility then resumes with the next COPY initialization statement.

VAM

The OPEN function fails with a return code of 48. The application should then call the CLOSE function.

R15 - Any other value

Any value, except 8, will cause the virtual tape to be accepted.

Installation Member

USERMOD and HVTU006 in the SAMPLIB library.
job for RECEIVE/
APPLY

Installation Process

The installation steps follow:

1. Edit member HVTX006 in the SAMPLIB library. This member is a sample exit. You may modify it as needed. When finished, save it and exit the member.

2. If the changed exit does not contain references to external routines, skip the following step.
3. Edit HVTU006 member in the SAMPLIB library
 - Change the ++JCLIN to contain appropriate INCLUDE statements for the invoked external routines.
 - If needed, add to it the JCL statements required for the correct external routines resolve.
4. Submit the job. It should end with a return code of 0.

Activate

Bounce the VTE server.

Chapter 35. VTE Storage Usage

This appendix provides information on the storage requirements of Virtual Tape Emulation.

VTE Server Main Storage Usage

The following table gives a close approximation of the total main storage used by the VTE server as a function of its enabled features and amount of resources.

The calculation includes the storage explicitly GETMAIN-ed by the server, as well as storage implicitly used by the system on behalf of the server.

Table 23. Approximation of the total main storage used by the VTE server

Type of storage	Amount (bytes)
Common fixed area above the line (ESQA SP245)	6850+ A*512
Common area above the line (ECSA SP241)	640K
Common fixed area below the line (SQA SP245)	11340
Common area below the line (CSA SP241)	1388
LSQA/ELSQA (system usage)	11070K+ A*80K+ D*5K
Private area above the line	9278K+ A*20K+ B*584K+ C*1714K+ D*184K+ E*116K+ F*58K+ 37K (if VLT=ENABLE)
Private area below the line	31K+ B*4K+ C*17K+ D*7K

Legend

- A** Number of defined (generated) virtual devices
- B** Number of concurrent I/O subtasks (MAXTASK parameter in the PERFORM initialization statement)

- C** Number e-Vault subtasks (MAXTASK parameter in the VLTOWN initialization statement).
If VLT=DISABLE in the OPTIONS initialization statement, C=0.
- D** Number of active devices
- E** Number of DPOOL initialization statements.
- F** Number of VLTRMT initialization statements where TYPE=1.
If VLT=DISABLE in the OPTIONS initialization statement, then F=0.

Examples

Two examples are provided to show how many MBs the server will use in different scenarios.

Example 1

Assume that:

- VLT=DISABLE.
- The site generated 128 devices.
- The number of concurrent tasks is 128.
- 32 devices are active.

In this case, the server will use approximately 114MB.

Example 2

Assume that:

- The number of generated devices is 64.
- All other values are as defined in Example 1.

In this case, the server will use only 70MB.

Additional Main Storage Used by a Job Served by VTE

The amount of storage needed by a job can be a challenge to determine ahead of time. However, it is possible to estimate the storage required.

The following table gives a close approximation of the additional main storage used by a job served by VTE.

Type of storage	Amount (bytes)
Private area below the line	G*3K
Private area above the line	H*210K+
	I*2K+
	J*8K

Legend

- G** Dynamic allocations currently performed in the address space. Freed once the dynamic allocation completes.
- H** Dynamic allocations currently performed in the address space. Freed once the dynamic allocation completes.

- I** PAT alias volumes substitutions currently performed in the address space.
Freed once the dynamic allocation completes.
If PAT=DISABLE in the OPTIONS initialization statement, I=0.
- J** Number of active devices that contain virtual tapes compressed using compress type=3 (Lempel-Ziv).

Disk Storage Usage

The amount of tracks needed for disk storage will vary, depending on the type and size of the record, file, or volume that is being stored on disk.

Installation and Product Libraries

For information on VTE installation and the amount of storage needed for the product libraries, see the "DASD Storage Requirements" section of the *Program Directory for IBM System Storage VTE Mainframe [GI11-8936-00]*.

Operation Files

The storage requirements for virtual tapes for operational files will vary based on the type of record, file, and volume.

Virtual tape data sets

Each 'active' virtual tape volume is represented by a multi-volume disk dataset whose size is determined by the data contained in the virtual tape.

Scratch tapes are not represented by disk data sets.

VDB

The VDB contains a record for each 'active' or scratch virtual tape. Its size is 256 bytes. This file size should be calculated according to the number of virtual tapes.

For example, if the VDB contains 100,000 virtual tapes, its size would be 500 tracks.

Journal file

The journal file contains:

- One control record.
- Two records for each mounted virtual tape.
- One record for each virtual tape status change done any VTE utility.

A journal record size is 384 bytes and 16 records are blocked in one block.

The file size is calculated according to the above factors and the number of days that information should be kept in the journal file.

For example, if there are 600 mounts/day, 600 virtual tapes are handled by offline utilities and the VDB is backed-up every 2 days, the journal file size would be 45 tracks.

History file (optional)

The history file contains:

- One control record.
- One dataset record for each generic name.
- One record for each user pattern.

A dataset record uses 6 bytes for each observation and another 60 bytes for fixed data.

Assuming virtual tapes are used on a weekly basis, each dataset record uses 372 ($60 \times 52 \times 6$) bytes.

If 500000 different dataset records are to be handled, the history file size would be 3400 tracks.

History backup file (optional)

The history file is backed up by the history backup file. Its size is similar to the history file size.

Appendix A: List of Console Commands

There are several console commands available to help you perform activities in a timely manner. The table contains the list of most commonly used console commands.

Table 24. Console Commands	
Console Command	
Displays diagnostic information.	
<code>/F CUZCLOUD,DISPLAY DIAG</code>	
Updates the parameter values listed in the specified parameter member without having to restart Cloud Tape Connector.	
<code>/F CUZCLOUD,REFRESH MEMBER=CUZ#PARM (or other member name)</code>	
Initiates a restore request for a data set or initiates a stand-alone restore of the repository data set.	
<code>/F CUZCLOUD,RESTORE DSN=<i>dsname</i>[GEN=<i>nnn</i>] [NEWNAME=<i>new dsname</i>]</code>	
Restores a stand-alone restore of the repository data set from the cloud when the repository is not available.	
<code>/F CUZCLOUD,RESTORE CLD=<i>cllname</i> CLDLIST=<i>cldlst</i></code>	
Initiates the Cloud Tape Connector started task.	
<code>[START S] CUZCLOUD (Use either S CUZCLOUD or /F CUZCLOUD,START.)</code>	
Stops the Cloud Tape Connector started task.	
<code>[P STOP]CUZCLOUD (Use either P CUZCLOUD or /F CUZCLOUD,STOP.)</code>	
Cancels the started task.	
<code>CANCEL CUZCLOUD or C CUZCLOUD</code>	
Enables ZIIP processing.	
<code>/F CUZCLOUD,ZIIP ENABLE</code>	
Disables ZIIP processing.	
<code>/F CUZCLOUD,ZIIP DISABLE</code>	
Starts repository backup	
<code>/F CUZCLOUD,EXECUTE=BACKUP</code>	
Starts processing history data sets.	
<code>/F CUZCLOUD,EXECUTE=HISTORY</code>	

<i>Table 24. Console Commands (continued)</i>
Console Command
Restarts the staging process of orphaned files.
/F CUZCLOUD,EXECUTE=RESTART

Appendix B: Example of AT-TLS parameter setup

If you are enabling SSL encryption for any of the TCP/IP cloud traffic, IBM's Communication Server enables control through the Application Transparent Transport Layer Security (AT-TLS). This topic provides an example of the parameter setup for a TTLS rule that enables an SSL/TLS encryption policy.

The example below shows the code that would be used to set values for AT-TLS parameters.

```
TTLSRule                                     Default_CUZ
{
  LocalAddr                                 ALL
  RemoteAddr                               ALL
  LocalPortRangeRef                        portR1
  RemotePortRange                          443
  Jobname                                  CUZCLOUD <=== Can use wildcard, like CUZ*.
  Direction                                Both
  Priority                                  255
  TTLSGroupActionRef                       gAct1~CUZ
  TTLSEnvironmentActionRef                 eAct1~CUZ
  TTLSConnectionActionRef                  cAct1~CUZ
}
TTLSGroupAction                             gAct1~CUZ
{
  TTLSEnabled                             On
  Trace                                    255
}
TTLSEnvironmentAction                       eAct1~CUZ
{
  HandshakeRole                           Client
  EnvironmentUserInstance                  0
  TTLSKeyringParmsRef                     CUZ_keyring
  Trace                                    255
}
TTLSConnectionAction                       cAct1~CUZ
{
  HandshakeRole                           Client
  TTLSCipherParmsRef                      cipher~CUZ
  TTLSConnectionAdvancedParmsRef          cAdv1~CUZ
  CtraceClearText                         Off
  Trace                                    2
}
TTLSConnectionAdvancedParms                cAdv1~CUZ
{
  ResetCipherTimer                        0
  SecondaryMap                             Off
  TLSv1.2                                 On
  ApplicationControlled                    Off
}
TTLSKeyringParms                           CUZ_keyring
{
  Keyring                                 USERID/CLOUDrng
}
TTLSCipherParms                            cipher1~Default_Ciphers
{
  V3CipherSuites                          TLS_RSA_WITH_AES_256_CBC_SHA
  V3CipherSuites                          TLS_DHE_RSA_WITH_AES_256_CBC_SHA
  V3CipherSuites                          TLS_DH_RSA_WITH_AES_256_CBC_SHA
  V3CipherSuites                          TLS_DHE_DSS_WITH_AES_256_CBC_SHA
  V3CipherSuites                          TLS_DH_DSS_WITH_AES_256_CBC_SHA
  V3CipherSuites                          TLS_RSA_WITH_3DES_EDE_CBC_SHA
  V3CipherSuites                          TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
  V3CipherSuites                          TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
  V3CipherSuites                          TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
  V3CipherSuites                          TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
  V3CipherSuites                          TLS_RSA_WITH_AES_128_CBC_SHA
  V3CipherSuites                          TLS_DHE_RSA_WITH_AES_128_CBC_SHA
  V3CipherSuites                          TLS_DH_RSA_WITH_AES_128_CBC_SHA
  V3CipherSuites                          TLS_DHE_DSS_WITH_AES_128_CBC_SHA
  V3CipherSuites                          TLS_DH_DSS_WITH_AES_128_CBC_SHA
}
TTLSCipherParms                            cipher~CUZ
{
  V3CipherSuites                          TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
```

```

V3CipherSuites      TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
V3CipherSuites      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
V3CipherSuites      TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
V3CipherSuites      TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
V3CipherSuites      TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
V3CipherSuites      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
V3CipherSuites      TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites      TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites      TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites      TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites      TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites      TLS_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites      TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites      TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites      TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites      TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites      TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites      TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites      TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites      TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites      TLS_DH_DSS_WITH_AES_128_CBC_SHA
}
PortRange            portR1
{
  Port                1024-65535
}
PortRange            portR2
{
  Port                1024-65535
}

```

For information on setting options in a Parmlib member that is not for AT-TLS, refer to [“Cloud Connector Settings \(Parmlib Options\)”](#) on page 23.

Appendix C: Using the Cloud Connector batch API

Cloud Tape Connector supports the use of the Virtual Tape Emulation (VTE) to "vault" tapes, which saves virtual tape data to disk. Through an API module, CUZ#APIB, VTE can save data to and retrieve data from any cloud defined in Cloud Tape Connector.

Mapping the CUZ#APIB module

Module CUZ#APIT in the SCUZSAMP sample library includes examples of all the CUZ#APIB function calls. The CUZ#APIB module can be mapped using two methods:

CUZ#APIB FUNC=MAP

This map includes the data layout of the working storage in the CSECT of your module.

CUZ#APIB FUNC=DSECT or CUZ#APIB

This map will build the APIB Block as a DSECT, which is the default.

- CUZ#APIB - Passes the Call information to module CU#API
- CUZ#APIL - Maps the data sets residing on the cloud when the function, BUILD_CLOUD_LIST/LIST, is called.

This macro is also used to generate the following API calls:

Status

Returns the status of a specific cloud defined in Cloud Tape Connector.

Store

Copies an existing cataloged data set to the cloud.

Restore

Restores a data set from the cloud to z/OS.

Delete

Deletes a data set on the cloud.

List

Retrieves a list of data sets on a cloud.

Obtain

Obtains storage for this DSECT and initializes the contents of the APIB Control Block.

Free

Frees the APIB Control Block storage.

Programming Requirements

The table below lists the programming requirements for the CUZ#APIB module.

Table 25. CUZ#APIB programming requirements	
Requirement	Description or Valid Values
Minimum authorization	Problem/Supervisor state and any PSW key
Dispatchable unit mode	Task
Cross memory mode	PASN or HASN
AMODE	31-bit only
ASC mode	Primary

Table 25. CUZ#APIB programming requirements (continued)	
Requirement	Description or Valid Values
Authorization	APF authorization required.
Interrupt status	Enabled or disabled for I/O and external interrupts.
Locks	No locks held.
Control parameters	None
CUZ#APIB Control Block	Required for all function calls to Cloud Tape Connector.
Userid assigned for calling API functions	Required to define a valid OMVS segment for this userid.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module.

Table 26. Return codes for Module CUZ#APIB	
Return and Reason Codes	Description
Return Code - R15	APIB_RETURN_CODE
Reason Code - R0	APIB_REASON_CODE
Return Code - 0	Successful Call
Return Code - 8	Runtime Error
Return Code - 12	Internal Error - APIB not built properly

In addition to a return or reason code, you will also receive messages when CUZ#APIB is run. The APIB_RETURN_MESSAGE field will contain either an "error" message or a "success" message. All of the Write-to-Operator (WTO) messages that are generated by API calls are written to your address space and will appear as message CUZS###X in the CUZOUT DD.

Supported Functions

The CUZ#APIB module supports the following functions:

DSECT

Builds storage layout as a DSECT.

MAP

Builds storage layout as working storage/CSECT.

OBTAIN

Issues storage obtain and initializes APIB block.

FREE

Issues storage release of the APIB block.

GET_CLOUD_STATUS

Returns status of cloud connection.

COPY_TO_CLOUD or STORE

Copies an existing data set to the cloud.

RESTORE_FROM_CLOUD or RESTORE

Restores a data set from the cloud back to z/OS.

DELETE_FROM_CLOUD or DELTE

Deletes a data set on the cloud.

LIST

Obtains a list of data sets residing on a cloud.

Using the APIB Obtain function

Obtaining an APIB block can be accomplished by using the CUZ#APIB macro with the parameter, FUNC=OBTAIN. This method is recommended because this function will also initialize this block with information required for calling module CUZ#API.

Required Parameter

The only required parameter is APIB_REG, which is the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

Return Codes

The following information lists the return codes that are generated by the CUZ#APIB module when used with the FUNC=OBTAIN parameter:

Table 27. Return codes for Module CUZ#APIB with the OBTAIN parameter	
Return and Reason Codes	Description
Return Code - R15 - 0	Obtain Successful
Reason Code - R15 - 8	CTC Started Task is not active

Example 1

The example below shows the Obtain parameter with the APIB Block using R5 for the register.

```
CUZ#APIB FUNC=OBTAIN,APIB_REG=R5
```

Example 2

The example below shows that a register was not specified for the Obtain parameter. In this case, the APIB block will default to using R8 for the register.

```
CUZ#APIB FUNC=OBTAIN
```

Using the APIB Free function

Freeing the APIB obtained storage can be accomplished by using the CUZ#APIB macro with the parameter, FUNC=FREE.

Parameters

The Free function offers two parameters, APIB_REG and APIL_REG. These parameters are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

APIL_REG

This parameter specifies the User Register for the APIL block. This block will be freed if the List function is called. Possible registers are R2 – R13.

Example 1

The result of using the FUNC=FREE parameter is that the APIB_REG register is cleared. Refer to the examples below.

The example below shows the FREE parameter for the APIB Block using R5 as the register. This block will be freed if the List function is called using R7 as the register.

```
CUZ#APIB FUNC=FREE,APIB_REG=R5,APIL_REG=R7
```

Example 2

The example below shows that a register was not specified for the FREE parameter. In this case, the APIB block will default to using R8 for the register. This block will be freed in the List function is called using R7 as the register.

```
CUZ#APIB FUNC=FREE,APIL_REG=R7
```

Using the APIB Get Cloud Status function

Checking the status of a cloud connection can be accomplished by using the CUZ#APIB macro with the parameter, FUNC=GET_CLOUD_STATUS. It can also be specified as APIB_FUNCTION = G.

Parameters

The GET_CLOUD_STATUS function offers two parameters, APIB_REG and CLOUD_NAME. These parameters are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

CLOUD_NAME

This parameter specifies the Address or Register of the eight-character Cloud Name. This is a pre-defined cloud name that was specified through the **Parmlib Options** on the Main Menu.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module when used with the GET_CLOUD_STATUS function.

Table 28. Return codes for Module CUZ#APIB with GET_CLOUD_STATUS function	
Return and Reason Codes	Description
R15/APIB_RETURN_CODE = 0	Successful Connection
R15/APIB_RETURN_CODE = 8	Connection Error
R15/APIB_RETURN_CODE = 12	User Error
R0/APIB_REASON_CODE	Error Reason Code
APIB_RETURN_MESSAGE	Success or Error Message

Example 1

The example below shows the status of the cloud name, IBMCLLOUD. In this case, the APIB block is using Register 5 (R5).

```
IBMCLLOUD DC CL8'IBMCLLOUD'
          CUZ#APIB FUNC=GET_CLOUD_STATUS,      +
          APIB_REG=R5,                          +
          CLOUD_NAME=IBMCLLOUD

          LA    R2,IBMCLLOUD
          CUZ#APIB FUNC=GET_CLOUD_STATUS,      +
          APIB_REG=R5,                          +
          CLOUD_NAME=(R2)
```

Example 2

The example below shows the status of the cloud name, IBMCLLOUD. In this case, the cloud will default to using R8 for the register, assuming that register is pointing to the APIB block.

```
CUZ#APIB FUNC=GET_CLOUD_STATUS,      +
          CLOUD_NAME=IBMCLLOUD
CUZ#APIB FUNC=GET_CLOUD_STATUS,      +
          CLOUD_NAME=(R2)
```

Using the APIB Build Cloud List function

A list of data sets residing on the Cloud can be obtained by using the CUZ#APIB macro with the parameter, FUNC=BUILD_CLOUD_LIST or FUNC=LIST. An APIL (list) block is built with one Header record followed by the list of data sets in the cloud. It can also be specified as APIB_FUNCTION = L.

Required Parameters

The BUILD_CLOUD_LIST function offers the APIB_REG, APIL_REG, and CLOUD_NAME parameters. These parameters are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

APIL_REG

This parameter specifies the User Register for the APIL block.

CLOUD_NAME

This parameter specifies the address or register of the eight-character Cloud Name where the data will be saved. This is a pre-defined cloud name that was specified through the **Parmlib Options** on the Main Menu.

Optional Parameters

The BUILD_CLOUD_LIST function has an additional parameter, LIST_DATASET_FILTER, that is optional.

LIST_DATASET_FILTER

This parameter specifies the address or register of the data set name mask, which can be up to 44 characters in length. The LIST_DATASET_FILTER parameter will return only the data sets meeting the masking criteria. If this parameter is omitted, all of the data sets on the cloud will be returned.

The LIST_DATASET_FILTER supports masking like ISPF 3.4.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module when used with the BUILD_CLOUD_LIST function.

Table 29. Return codes for Module CUZ#APIB with BUILD_CLOUD_LIST function

Return and Reason Codes	Description
R15/APIB_RETURN_CODE = 0	Successful List
R15/APIB_RETURN_CODE = 8	Copy to Cloud Failed
R15/APIB_RETURN_CODE = 12	User Error
R0/APIB_REASON_CODE	Error Reason Code
APIB_RETURN_MESSAGE	Success or Error Message

Example 1

The example below shows how to obtain a list of data sets on cloud IBMCLLOUD. In this case, the APIB block is using Register 5 (R5). APIL is using R6 with filter criteria for a data set name (dsn) of MY.DSN*.

```

IBMCLLOUD DC CL08'IBMCLLOUD'
DSNFILTR DC CL44'MY.DSN*'
CUZ#APIB  FUNC=BUILD_CLOUD_LIST,      +
          APIB_REG=R5,                +
          APIL_REG=R6,                +
          LIST_DATASET_FILTER=DSNFILTR, +
          CLOUD_NAME=IBMCLLOUD

LA  R2,IBMCLLOUD
LA  R4,DSNFILTR
CUZ#APIB  FUNC=BUILD_CLOUD_LIST,      +
          APIB_REG=R5,                +
          APIL_REG=R6,                +
          LIST_DATASET_FILTER=(R4),   +
          CLOUD_NAME=(R2)

```

Example 2

The example below lists all of the data sets on cloud IBMCLLOUD.

```

IBMCLLOUD DC CL08'IBMCLLOUD'
CUZ#APIB  FUNC=BUILD_CLOUD_LIST,      +
          APIB_REG=R5,                +
          APIL_REG=R6,                +
          CLOUD_NAME=IBMCLLOUD

DSNFILTR DC CL44'*'
CUZ#APIB  FUNC=BUILD_CLOUD_LIST,      +
          APIB_REG=R5,                +
          APIL_REG=R6,                +
          LIST_DATASET_FILTER=DSNFILTR, +
          CLOUD_NAME=IBMCLLOUD

```

Using the APIB Copy to Cloud function

An existing cataloged data set can be copied to the cloud by using the CUZ#APIB macro with the parameter, FUNC=COPY_TO_CLOUD or FUNC=STORE. It can also be specified as APIB_FUNCTION = S.

Required Parameters

The COPY_TO_CLOUD function offers the APIB_REG, APIL_REG, and CLOUD_NAME parameters. These parameters are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

CLOUD_NAME

This parameter specifies the address or register of the eight-character Cloud Name where the data set will be copied to. This is a pre-defined cloud name that was specified through the **Parmlib Options** on the Main Menu.

COPY_DSN

This parameter specifies the address or register of the data set name, up to 44 characters in length, to be copied to the cloud. This data set must be cataloged and must either reside on TAPE or DISK.

RETENTION_PERIOD

This parameter specifies a number, up to four characters in length, for the amount of days the DSN will reside on the cloud. A retention period of 9999 indicates that this is a non-expiring cloud backup and will be kept indefinitely.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module when used with the COPY_TO_CLOUD function.

Table 30. Return codes for Module CUZ#APIB with COPY_TO_CLOUD function	
Return and Reason Codes	Description
R15/APIB_RETURN_CODE = 0	Successful Copy to Cloud
R15/APIB_RETURN_CODE = 8	Copy to Cloud Failed
R15/APIB_RETURN_CODE = 12	User Error
R0/APIB_REASON_CODE	Error Reason Code
APIB_RETURN_MESSAGE	Success or Error Message

Example 1

The example below shows how to copy a disk data set MY.DSN to cloud IBMCLLOUD. In this case, the APIB block is using Register 5 (R5) and the data is saved for five (5) days.

```
IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
RETPD     DC CL04'0005'
CUZ#APIB   FUNC=COPY_TO_CLOUD,
           APIB_REG=R5,
           CLOUD_NAME=IBMCLLOUD,
           RETENTION_PERIOD=RETPD,
           COPY_DSN=MYDSN

LA R2,IBMCLLOUD
LA R3,MYDSN
LA R4,RETPD
CUZ#APIB   FUNC=COPY_TO_CLOUD,
           APIB_REG=R5,
           CLOUD_NAME=(R2),
           RETENTION_PERIOD=(R4),
           COPY_DSN=(R3)
```

Example 2

The example below shows how to copy the disk data set MY.DSN to cloud IBMCLLOUD. The APIB block defaults to Register 8 (R8).

```
IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
CUZ#APIB   FUNC=COPY_TO_CLOUD,
           CLOUD_NAME=IBMCLLOUD,
           RETENTION_PERIOD=RETPD,
           COPY_DSN=MYDSN
```

```

LA    R2,IBMCLLOUD
LA    R3,MYDSN
CUZ#APIB  FUNC=COPY_TO_CLOUD,      +
        CLOUD_NAME=(R2),          +
        COPY_DSN=(R3)

```

Using the APIB Restore from Cloud function

Restoring a data set from a cloud can be accomplished by using the CUZ#APIB macro with the parameter, FUNC=RESTORE_FROM_CLOUD or FUNC=RESTORE. It can also be specified as APIB_FUNCTION = R.

Required Parameters

The RESTORE_FROM_CLOUD function offers several parameters, which are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

CLOUD_NAME

This parameter specifies the address or register of the eight-character Cloud Name containing the data set that will be restored from the cloud. This is a pre-defined cloud name that was specified through the **Parmlib Options** on the Main Menu.

RESTORE_DSN

This parameter specifies the address or register of the data set name, up to 44 characters in length, to be restored from the cloud. This data set must be uncatalogued and must reside on the cloud specified in the CLOUD_NAME parameter.

RESTORE_TO_TAPE_UNIT

This parameter specifies the address or register of an eight-character esoteric unit of a tape device. This parameter is ignored if RESTORE_TO_DDNAME is used.

RESTORE_TO_DASD_TAPE

This parameter indicates whether to restore to the DASD or Tape device. Valid values are D and T.

- **D** - Restore to DASD device. If the allocation on the DASD device fails, the restored DSN will be allocated on Tape instead. If the Restore DSN block size is greater than 32K, the DSN will be allocated on tape rather than DASD.
- **T** - Restore to Tape device.

These values are ignored if the RESTORE_TO_DDNAME parameter is used.

RESTORE_DELETE_OLD

This parameter indicates whether to delete an existing Restore to DSN. Valid values are Y and N, as described below:

- **Y** - Delete an existing Restore to DSN. This is the default.
- **N** - Do not delete an existing Restore to DSN.

These values are ignored if the RESTORE_TO_DDNAME parameter is used.

RETENTION_PERIOD

This parameter specifies a number, up to four characters in length, for the amount of days the restored DSN will reside on tape. A retention period of 9999 indicates that this is a non-expiring tape and will be kept indefinitely. Set the retention period to 0000 if restoring to DASD.

These values are ignored if the RESTORE_TO_DDNAME parameter is used.

RESTORE_WAIT_FOR_COMPLETION

This parameter indicates whether to wait until the Restore process is complete before returning control. Valid values are Y and N, as described below:

- **Y** - Wait for Restore to finish before returning control. This is the default.
A value of "Y" is required if the RESTORE_TO_DDNAME parameter is used.
- **N** - Do not wait for the Restore process to finish before returning control. Instead, return control immediately after the Restore process is scheduled.
A value of "N" is required for asynchronous processing.

Optional Parameters

The RESTORE_FROM_CLOUD function has a few additional parameters that are optional.

RESTORE_TO_DDNAME

This parameter specifies the address or register of the eight-character DD name that will be used for the Restore. The DD Name is the name of the user-allocated DD.

The RESTORE_TO_DDNAME parameter is mutually exclusive with the following parameters:

- RESTORE_DELETE_OLD
- RESTORE_TO_ALIAS
- RESTORE_TO_DASD_UNIT
- RESTORE_TO_DSN
- RESTORE_TO_STORCLASS
- RESTORE_TO_TAPE_UNIT
- RESTORE_VOLUME_COUNT
- RETENTION_PERIOD

RESTORE_REL_GEN

This parameter specifies the address or register containing a halfword with the relative generation of the backup. The value for the generation can either be a positive number or a negative value from 0 to 10.

If this parameter is omitted, the latest backup will be restored.

RESTORE_TO_DSN

This parameter specifies the address or register of the data set name, up to 44 characters in length. The data set from the cloud will be restored to this new DSN. If this parameter is omitted, the data set from the cloud will be restored to its original DSN or, if specified, to the data set name listed in the RESTORE_DSN parameter.

This parameter is mutually exclusive with the following parameters:

- RESTORE_TO_ALIAS
- RESTORE_TO_DDNAME

RESTORE_TO_ALIAS

This parameter specifies the address or register of the eight-character Alias Name. The data set from the cloud will be restored to this new Alias or high-level qualifier. If this parameter is omitted, the data set will be restored to its original DSN or, if specified, to the data set name listed in the RESTORE_DSN parameter.

This parameter is mutually exclusive with the following parameters:

- RESTORE_TO_DSN
- RESTORE_TO_DDNAME

If the RESTORE_TO_DSN and the RESTORE_TO_ALIAS parameters are omitted, the DSN will be restored to the original name.

RESTORE_TO_DASD_UNIT

This parameter specifies the address or register of the eight-character esoteric unit of a DASD device. The RESTORE_TO_DASD_UNIT parameter is ignored if the RESTORE_TO_DDNAME parameter is used.

RESTORE_TO_STORCLASS

This parameter specifies the address or register of an eight-character storage class. The data set will be restored with this new storage class. If the RESTORE_TO_STORCLASS parameter is omitted, the default z/OS SMS storage class will be used.

The RESTORE_TO_STORCLASS parameter will be ignored if the RESTORE_TO_DDNAME parameter is used.

RESTORE_VOLUME_COUNT

This parameter specifies the address or register of a three-character volume count for allocating the restore data set. If the RESTORE_VOLUME_COUNT parameter is omitted, the default z/OS SMS volume count will be used.

The RESTORE_VOLUME_COUNT parameter will be ignored if the RESTORE_TO_DDNAME parameter is used.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module when used with the RESTORE_FROM_CLOUD function.

Table 31. Return codes for Module CUZ#APIB with RESTORE_FROM_CLOUD function	
Return and Reason Codes	Description
R15/APIB_RETURN_CODE = 0	Successful Restore from Cloud
R15/APIB_RETURN_CODE = 8	Restore from Cloud Failed
R15/APIB_RETURN_CODE = 12	User Error
R0/APIB_REASON_CODE	Error Reason Code
APIB_RETURN_MESSAGE	Success or Error Message

Example 1

The example below restores data set MY.DSN to MY.RESTORED.DSN from cloud IBMCLD using Register 5 (R5) as the APIB block to Disk Unit SYSALLDA. The SMS storage class is SMSCLASS and the Retention Period is set to 0 days.

```
IBMCLD  DC CL08'IBMCLD'
MYDSN   DC CL44'MY.DSN'
TODSN   DC CL44'MY.RESTORED.DSN'
DISK    DC CL08'SYSALLDA'
TAPE    DC CL08'CART'
SMS     DC CL08'SMSCLASS'
RETPD   DC CL04'0000'
CUZ#APIB FUNC=RESTORE_FROM_CLOUD,      +
      APIB_REG=R5,                      +
      RESTORE_DSN=MYDSN,                +
      RESTORE_TO_DSN=TODSN,             +
      RESTORE_TO_DASD_UNIT=DISK,        +
      RESTORE_TO_TAPE_UNIT=TAPE,        +
      RESTORE_TO_STORCLASS=SMS,         +
      RESTORE_TO_DASD_TAPE=D,           +
      RESTORE_DELETE_OLD=Y,             +
      RETENTION_PERIOD=RETPD,           +
      RESTORE_WAIT_FOR_COMPLETION=Y,    +
      CLOUD_NAME=IBMCLD

LA      R2,IBMCLD
LA      R3,MYDSN
LA      R4,TODSN
LA      R6,DISK
LA      R7,SMS
LA      R8,RETPD
LA      R9,TAPE
CUZ#APIB FUNC=RESTORE_FROM_CLOUD,      +
      APIB_REG=R5,                      +
      RESTORE_DSN=(R3),                 +
```



```

RESTORE_TO_DSN=(R4),          +
RESTORE_TO_DASD_UNIT=(R6),    +
RESTORE_TO_TAPE_UNIT=(R9),    +
RESTORE_TO_STORCLASS=(R7),    +
RESTORE_TO_DASD_TAPE=D,       +
RESTORE_DELETE_OLD=Y,         +
RETENTION_PERIOD=(R8),        +
RESTORE_WAIT_FOR_COMPLETION=Y, +
CLOUD_NAME=(R2)

```

Example 2

The example below restores data set MY.DSN to RESTORED.DSN from cloud IBMCLLOUD using Register 5 (R5) as the APIB block to Disk Unit SYSALLDA. The SMS storage class is SMSCLASS and the Retention Period is set to 0 days.

```

IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
ALIAS     DC CL08'RESTORED'
DISK      DC CL08'SYSALLDA'
SMS       DC CL08'SMSCLASS'
RETPD     DC CL04'0000'
CUZ#APIB  FUNC=RESTORE_FROM_CLOUD,      +
          APIB_REG=R5,                  +
          RESTORE_DSN=MYDSN,            +
          RESTORE_TO_ALIAS=ALIAS,       +
          RESTORE_TO_DASD_UNIT=DISK,    +
          RESTORE_TO_TAPE_UNIT=TAPE,    +
          RESTORE_TO_STORCLASS=SMS,     +
          RESTORE_TO_DASD_TAPE=D,       +
          RESTORE_DELETE_OLD=Y,         +
          RETENTION_PERIOD=RETPD,       +
          RESTORE_WAIT_FOR_COMPLETION=Y, +
          CLOUD_NAME=IBMCLLOUD

```

Example 3

The example below restores data set MY.DSN to the original DSN from cloud IBMCLLOUD using Register 5 (R5) as the APIB block to Disk Unit SYSALLDA. The default SMS storage class is used and the Retention Period is set to 0 days.

```

IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
DISK      DC CL08'SYSALLDA'
RETPD     DC CL04'0000'
CUZ#APIB  FUNC=RESTORE_FROM_CLOUD,      +
          RESTORE_DSN=MYDSN,            +
          RESTORE_TO_DASD_UNIT=DISK,    +
          RESTORE_TO_TAPE_UNIT=TAPE,    +
          RESTORE_TO_STORCLASS=SMS,     +
          RESTORE_TO_DASD_TAPE=D,       +
          RETENTION_PERIOD=RETPD,       +
          CLOUD_NAME=IBMCLLOUD

```

Example 4

The example below restores data set MY.DSN to allocated DD RESTDD from cloud IBMCLLOUD using Register 8 (R8) as the APIB block. All other parameters are ignored since the data set is already allocated.

```

IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
RESTDD    DC CL08'RESTDD'
CUZ#APIB  FUNC=RESTORE_FROM_CLOUD,      +
          RESTORE_DSN=MYDSN,            +
          RESTORE_TO_DDNAME=RESTDD,     +
          CLOUD_NAME=IBMCLLOUD

```

Using the APIB Delete from Cloud function

A data set can be deleted from the cloud by using the CUZ#APIB macro with the parameter, FUNC=DELETE_FROM_CLOUD. It can also be specified as APIB_FUNCTION = D.

Parameters

The DELETE_FROM_CLOUD function offers two parameters, APIB_REG and CLOUD_NAME. These parameters are described below:

APIB_REG

This parameter specifies the User Register that is populated with the storage address of the APIB block. The R0, R1, R14, and R15 registers are reserved for use by the CUZ#APIB module.

If APIB_REG is not used, the default register is R8.

CLOUD_NAME

This parameter specifies the Address or Register of the eight-character Cloud Name. This is a pre-defined cloud name that was specified through the **Parmlib Options** on the Main Menu.

DELETE_DSN

This parameter specifies the address or register of the data set name, up to 44 characters in length, to be deleted from the cloud. This data set must reside on the cloud specified in the CLOUD_NAME parameter, but does not need to exist on z/OS.

Return Codes and Messages

The following information lists the return codes that are generated by the CUZ#APIB module when used with the DELETE_FROM_CLOUD function.

Table 32. Return codes for Module CUZ#APIB with DELETE_FROM_CLOUD function	
Return and Reason Codes	Description
R15/APIB_RETURN_CODE = 0	Successful Delete from Cloud
R15/APIB_RETURN_CODE = 8	Delete from Cloud Error
R15/APIB_RETURN_CODE = 12	User Error
R0/APIB_REASON_CODE	Error Reason Code
APIB_RETURN_MESSAGE	Success or Error Message

Example

The example below shows how to delete data set MY.DSN from cloud IBMCLLOUD using R5 as the APIB block.

```
IBMCLLOUD DC CL08'IBMCLLOUD'
MYDSN     DC CL44'MY.DSN'
CUZ#APIB  FUNC=DELETE_FROM_CLOUD,      +
          APIB_REG=R5,                  +
          CLOUD_NAME=IBMCLLOUD          +
          DELETE_DSN=MYDSN

LA  R2,IBMCLLOUD
LA  R3,MYDSN
CUZ#APIB  FUNC=DELETE_FROM_CLOUD,      +
          APIB_REG=R5,                  +
          CLOUD_NAME=(R2),             +
          DELETE_DSN=(R3)
```

Sample CUZ#APIT Member

The SCUZSAMP library contains several members with sample JCL. The CUZ#APIT member is included in the SCUZSAMP library and provides the sample JCL for using VTE with clouds defined in Cloud Tape Connector. This topic provides an example of the parameters available for assembler macro CUZ#APIB, which generates the DSECT and inline code to call any functions.

```
&LABEL    CUZ#APIB &FUNC=DSECT,                                     +
            &APIB_REG=R8,                                           +
            &APIL_REG=,                                             +
            &CLOUD_NAME=,                                           +
            &COPY_DSN=,                                             +
            &DELETE_DSN=,                                           +
            &RESTORE_DELETE_OLD=,                                    +
            &LIST_DATASET_FILTER=,                                   +
            &RESTORE_DSN=,                                           +
            &RESTORE_REL_GEN==H'0',                                  +
            &RESTORE_TO_DDNAME=,                                     +
            &RESTORE_TO_DSN=,                                        +
            &RESTORE_TO_DASD_TAPE=,                                  +
            &RESTORE_TO_DASD_UNIT=,                                  +
            &RESTORE_TO_TAPE_UNIT=,                                  +
            &RESTORE_TO_ALIAS=,                                     +
            &RESTORE_TO_STORCLASS=,                                  +
            &RESTORE_VOLUME_COUNT=,                                  +
            &RETENTION_PERIOD=,                                     +
            &RESTORE_WAIT_FOR_COMPLETION=Y                          +
            .*
            .*
            PUSH PRINT,NOPRINT
            PRINT OFF,NOPRINT
            LABEL    MACRO
            &***** $LADR &REG,&ADDR,&LENGTH=8
            .* $LAD*****
            .* addrR - This macro relieves the problem of determining whether the
            .* reloess passed to an 'LA' instruction is a register or a regular
            .* the catable value. It just checks to see if the first character of
            .* instADDR is an open parenthesis. If it is, a register-to-register
            .* instruction is generated instead of the LA Rn,ADDR.
            .*
            .*****
            .*
            GBLC &FECUSEY
            AIF ('&ADDR'(1,1) EQ '*').INDIRECT
            AIF ('&ADDR'(1,1) EQ '(').ISREG
            AIF ('&ADDR'(1,1) NE ' ').NOTLIT
            AIF (('&LENGTH'(1,1) EQ 'w') OR ('&LENGTH'(1,1) EQ 'w')).L_WORD
            AIF ('&FECUSEY' NE 'Y').PAST_Y1
            LAY    &REG,=CL&LENGTH.&ADDR
            AGO    .EX_1
            .PAST_Y1 ANOP
            LA      &REG,=CL&LENGTH.&ADDR
            .EX_1 ANOP
            AGO    .EXIT
            .*
            .L_WORD ANOP
            &ACNT    SETA K'&ADDR                                     literal word aligned
            &ACNT    SETA &ACNT-2                                     number of characters
            &ADD2     SETC '&ADDR'(2,&ACNT)                          less quotes
            AIF ('&FECUSEY' NE 'Y').PAST_Y2                          strip off quotes
            &LABEL    LAY    &REG,=A(&ADD2)
            AGO    .EX_2
            .PAST_Y2 ANOP
            &LABEL    LA      &REG,=A(&ADD2)
            .EX_2 ANOP
            AGO    .EXIT
            .*
            .NOTLIT ANOP
            AIF ('&ADDR'(1,2) EQ '=A').ADCON
            AIF ('&ADDR'(1,2) EQ '=a').ADCON
            AIF ('&FECUSEY' NE 'Y').PAST_Y3
            &LABEL    LAY    &REG,&ADDR
            AGO    .EX_3
            .PAST_Y3 ANOP
            &LABEL    LA      &REG,&ADDR
            .EX_3 ANOP
```

```

      AGO      .EXIT
.*
      .ADCON   ANOP
&LABEL      L      &REG,&ADDR
      AGO      .EXIT
.*
      .ISREG   ANOP
&LABEL      LR      &REG,&ADDR
      AGO      .EXIT
.*
      .INDIRECT ANOP
&ACNT       SETA  K'&ADDR
&ACNT       SETA  &ACNT-1
&ADD2       SETC  '&ADDR'(2,&ACNT)
&LABEL      AIF  ('&ADD2'(1,1) EQ '().ISREG_I
      L      &REG,&ADD2
      AGO      .EXIT
.*
      .ISREG_I ANOP
&ACNT       SETA  &ACNT-2
&ADD2       SETC  '&ADD2'(2,&ACNT)
&LABEL      L      &REG,0(,&ADD2)
      .EXIT   ANOP
.*
      MEND
      POP PRINT,NOPRINT
.*
&CUZ        GBLC  &CUZ
      SETC  'CUZ'
      LCLA  &COUNT
&COUNT     GBLC  &COUNTER
&COUNT     SETA  &COUNTER
&COUNT     SETA  &COUNT+1
&COUNTER     SETC  '&COUNT'
.*
      AIF ('&FUNC' EQ 'DSECT').MAPIT
      AIF ('&FUNC' EQ 'MAP').MAPIT
      AIF ('&FUNC' EQ 'OBTAIN').OBTAIN
      AIF ('&FUNC' EQ 'GET_CLOUD_STATUS').GETSTAT
      AIF ('&FUNC' EQ 'COPY_TO_CLOUD').COPYCLD
      AIF ('&FUNC' EQ 'STORE').COPYCLD
      AIF ('&FUNC' EQ 'RESTORE_FROM_CLOUD').RESTCLD
      AIF ('&FUNC' EQ 'RESTORE').RESTCLD
      AIF ('&FUNC' EQ 'FREE').FREE
      AIF ('&FUNC' EQ 'DELETE').DELETE
      AIF ('&FUNC' EQ 'DELETE_FROM_CLOUD').DELETE
      AIF ('&FUNC' EQ 'LIST').LIST
      AIF ('&FUNC' EQ 'BUILD_CLOUD_LIST').LIST
      MNOTE 12,'Invalid Function - Look at Macro doc for list valid Func'
      MEXIT
      .MAPIT   ANOP
      * * * * *
      * CUZ#APIB Mapping. Control Block used for All API calls to Cloud *
      * Tape Connector. *
      * * * * *
      .
      AIF ('&FUNC' NE 'MAP').DSECTD
      .DSECT   ANOP
&PIB_BLK_S   DS      0F
      AGO      .DSECTC
      .DSECTD ANOP
&PIB_BLK_S   DS      DSECT
      .DSECTC ANOP
&PIB_EYE_CATCHER DC    CL08'CUZ#APIB' Eye Catcher
&PIB_VERSION     DC    CL06'V1R1M1'
&PIB_FUNCTION     DS    CL01 Function of Call
&PIB_FUNCTION_GET_STATUS EQU  C'G' Return Cloud Connection Status
&PIB_FUNCTION_STORE EQU    C'S' Store a Dataset on Cloud
&PIB_FUNCTION_RESTORE EQU   C'R' Restore a Dataset from Cloud
&PIB_FUNCTION_DELETE EQU    C'D' Delete a Dataset on Cloud
&PIB_FUNCTION_LIST EQU     C'L' Obtain List of Datasets on Cloud
&PIB_PSW_KEY      DS      X Callers PSW Key
&PIB_BLK_LENGTH   DS      F Length of this Block
&PIB_RETURN_CODE  DS      F Return Code
&PIB_REASON_CODE  DS      F Reason Code
&PIB_CUZ#APIL_ADDR DS      F Addr of Cloud List DSECT CUZ#APIL
&PIB_CUZ#API_MOD_ADDR DS     F Addr of Module CUZ#API
&PIB_RESTORE_REL_GEN DS      H Relative Generation to Restore
&PIB_CHAR_AREA    DS      0S Starting Character Area
&PIB_CLOUD_NAME    DS     CL08 Cloud Connector Name
&PIB_CLOUD_TYPE    DS     CL01 Returned Cloud_TYPE

```

```

APIB_CLOUD_TYPE_HCP      EQU C'H'  Cloud Type Hitachi
APIB_CLOUD_TYPE_FTP      EQU C'F'  Cloud Type FTP
APIB_CLOUD_TYPE_S3       EQU C'S'  Cloud Type S3
APIB_CLOUD_TYPE_CLS      EQU C'C'  Cloud Type S3-Compatible
APIB_CLOUD_TYPE_SFT      EQU C'L'  Cloud Type Softlayer
APIB_CLOUD_TYPE_NOT_FND  EQU C'N'  Cloud Server was not Found
APIB_CLOUD_STAT          DS CL01    Returned Cloud_TYPE
APIB_CLOUD_STAT_VALID    EQU C'V'  Valid Connection to Cloud Server
APIB_CLOUD_STAT_NO_CON   EQU C'N'  No Valid Connection to Cloud Server
APIB_RETURN_MESSAGE      DS CL132   Error Message Returned
APIB_CLOUD_COPY_DSN      DS CL44    Dataset Name being copied to cloud
APIB_RESTORE_TO_DATASET  DS CL44    Dataset Name to restore to
APIB_RESTORE_TO_ALIAS    DS CL08    New Alias for restore DSN - Rename
APIB_RESTORE_TO_DDNAME   DS CL08    Restore to Previously Allocated DD
APIB_RESTORE_DASD_UNIT   DS CL08    DASD Unit to restore DSN
APIB_RESTORE_TAPE_UNIT   DS CL08    Tape Unit to restore DSN
APIB_RESTORE_TO_STORCLAS DS CL08    SMS Storage Class for restore DSN
APIB_RESTORE_VOLUME_CNT  DS CL03    Volume Count - Char 3 (001-256)
APIB_RESTORE_DASD_TAPE   DS CL01    D or T - Restore to Tape or DASD?
APIB_DELETE_OLD_RST_DSN  DS CL01    Y or N - Delete Old Restore DSN?
APIB_RESTORE_WAIT_COMP   DS CL01    Y or N - Wait for Restore Complete
APIB_RETENTION_PERIOD    DS CL04    Retention Period for Store/Restore
APIB_LIST_DATASET_FILTER EQU APIB_RESTORE_TO_DATASET
APIB_CHAR_AREA_LEN       EQU *-APIB_CHAR_AREA
APIB_CLOUD_CTLG_RESTORE  DS CL01    Reserved Storage Area
APIB_RESERVED           DS CL79    Reserved Storage Area

APIB_BLK_LEN            EQU *-APIB_BLK_S
APIB_BLK                EQU APIB_BLK_S,*-APIB_BLK_S
APIB_SUBPOOL            EQU 94

```

```

* * * * *
* CUZ#APIL Mapping.  Control Block returned on FUNC=BUILD_CLOUD_LIST*
*
* -APIB_CUZ#APIL_ADDR contains Address of this Block after call
*
* -First Record is the header record with an eye catcher, length
*   of storage block obtained and the nbr of datasets returned.*
*
* -APIL_DATASET_COUNT contains nbr of Datasets to follow Header
*
* -After First Record, the list of datasets follow.
*
* USING APIB_BLK,R5
* USING APIL_BLK,R6
* CUZ#APIB FUNC=BUILD_CLOUD_LIST,      +
*     APIB_REG=R5,                      +
*     APIL_REG=R6,                      +
*     LIST_DATASET_FILTER=DSNFILTR,    +
*     CLOUD_NAME=IBMCLOUD
* L      R4,APIL_DATASET_COUNT      Loop Counter
* LA     R6,APIL_NEXT_DSN          Skip Over Header Record
*DSN_LOOP DS 0H
* CLC    APIL_BACKUP_DATASET,MYDSN Is this my dsn?
* BNE    NEXT_DSN
*NEXT_DSN DS 0H
* BCT    R4,DSN_LOOP
*
* * * * *

```

```

APIB_BLK_S              DSECT
APIB_BACKUP_DATASET     DS CL44    z/OS Dataset Backed up to Cloud
APIB_CLOUD_DATASET      DS CL44    Name of Dataset on Cloud
APIB_BACKUP_GEN_ID       DS CL08    Absolute Generation or STCK Time
APIB_BACKUP_RELATIVE_GEN DS PL02    Relative Generation 0, -1...
APIB_BACKUP_REPO_STATUS  DS CL01    Is this DSN in CTC Repository ?
APIB_BACKUP_MATCHED      EQU C'M'  Backup on Cloud and in CTC Repo
APIB_BACKUP_ON_CLOUD_ONLY EQU C'R'  DSN found on Cloud not CTC Repo
APIB_BACKUP_ON_REPO_ONLY EQU C'Z'  DSN found on CTC Repo not on Cloud
APIB_NEXT_DSN           DS 0S
APIB_HEADER_RECORD      DS 0S      First Record Description
APIB_EYE_CATCHER        DC CL08    'CUZ#APIL' Eye Catcher
APIB_BLOCK_LENGTH       DS F       Length of this Block
APIB_DATASET_COUNT      DS F       Nbr of Datasets in this block
APIB_PSW_KEY            DS X       Storage Acquired in PSW Key
APIB_NEXT_DSN           ORG ,

APIB_BLK_LEN            EQU *-APIB_BLK_S
APIB_BLK                EQU APIB_BLK_S,*-APIB_BLK_S
APIB_SUBPOOL            EQU 94
MEXIT

```

```

.GETSTAT ANOP
      AIF ('&APIB_REG' EQ '').APIBERR
      AIF ('&CLOUD_NAME' EQ '').CLOUDERR
      PUSH USING
APIB  USING APIB_BLK,&APIB_REG
      MVI APIB.APIB_FUNCTION,APIB_FUNCTION_GET_STATUS
      LA R0,APIB.APIB_CHAR_AREA
      LHI R1,APIB_CHAR_AREA_LEN
      SGR R14,R14
      L R15,=X'40000000'
      MVCL R0,14
      $LADR R1,&CLOUD_NAME
      MVC APIB.APIB_CLOUD_NAME,0(R1)
      IC R15,APIB.APIB_PSW_KEY
*      IC R15,=X'80'
      SPKA 0(R15)
      LGF R15,APIB.APIB_CUZ#API_MOD_ADDR
      LR R1,&APIB_REG
      BASR R14,R15
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_RETURN_CODE
      LGF R0,APIB.APIB_REASON_CODE
      DROP APIB
      POP USING
      MEXIT

.*
.COPYCLD ANOP
      AIF ('&APIB_REG' EQ '').APIBERR
      AIF ('&CLOUD_NAME' EQ '').CLOUDERR
      AIF ('&COPY_DSN' EQ '').CPDSNERR
      AIF ('&RETENTION_PERIOD' EQ '').RETNPERR
      PUSH USING
APIB  USING APIB_BLK,&APIB_REG
      MVI APIB.APIB_FUNCTION,APIB_FUNCTION_STORE
      LA R0,APIB.APIB_CHAR_AREA
      LHI R1,APIB_CHAR_AREA_LEN
      SGR R14,R14
      LGF R15,=X'40000000'
      MVCL R0,14
      $LADR R1,&CLOUD_NAME
      MVC APIB.APIB_CLOUD_NAME,0(R1)
      $LADR R1,&COPY_DSN
      MVC APIB.APIB_CLOUD_COPY_DSN,0(R1)
      $LADR R1,&RETENTION_PERIOD
      MVC APIB.APIB_RETENTION_PERIOD,0(R1)
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_CUZ#API_MOD_ADDR
      LR R1,&APIB_REG
      BASR R14,R15
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_RETURN_CODE
      LGF R0,APIB.APIB_REASON_CODE
      DROP APIB
      POP USING
      MEXIT

.*
.DELETE ANOP
      AIF ('&APIB_REG' EQ '').APIBERR
      AIF ('&CLOUD_NAME' EQ '').CLOUDERR
      AIF ('&DELETE_DSN' EQ '').DLDSNERR
      PUSH USING
APIB  USING APIB_BLK,&APIB_REG
      MVI APIB.APIB_FUNCTION,APIB_FUNCTION_DELETE
      LA R0,APIB.APIB_CHAR_AREA
      LHI R1,APIB_CHAR_AREA_LEN
      SGR R14,R14
      LGF R15,=X'40000000'
      MVCL R0,14
      $LADR R1,&CLOUD_NAME
      MVC APIB.APIB_CLOUD_NAME,0(R1)
      $LADR R1,&DELETE_DSN
      MVC APIB.APIB_CLOUD_COPY_DSN,0(R1)
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_CUZ#API_MOD_ADDR
      LR R1,&APIB_REG
      BASR R14,R15
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)

```

```

LGF R15,APIB.APIB_RETURN_CODE
LGF R0,APIB.APIB_REASON_CODE
DROP APIB
POP USING
MEXIT

.*
.LIST ANOP
      AIF ('&APIB_REG' EQ '').APIBERR
      AIF ('&APIL_REG' EQ '').APILERR
      AIF ('&CLOUD_NAME' EQ '').CLOUDERR
      PUSH USING
APIB  USING APIB_BLK,&APIB_REG
      MVI APIB.APIB_FUNCTION,APIB_FUNCTION_LIST
      LA R0,APIB.APIB_CHAR_AREA
      LHI R1,APIB_CHAR_AREA_LEN
      SGR R14,R14
      LGF R15,=X'40000000'
      MVCL R0,14
      $LADR R1,&CLOUD_NAME
      MVC APIB.APIB_CLOUD_NAME,0(R1)
      AIF ('&LIST_DATASET_FILTER' EQ '').FLTRALL
      $LADR R1,&LIST_DATASET_FILTER
      MVC APIB.APIB_LIST_DATASET_FILTER,0(R1)
      AGO .FLTRDON
.FLTRALL ANOP
.MVI APIB.APIB_LIST_DATASET_FILTER,C'*'
.FLTRDON ANOP
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_CUZ#API_MOD_ADDR
      LR R1,&APIB_REG
      BASR R14,R15
      IC R15,APIB.APIB_PSW_KEY
      SPKA 0(R15)
      LGF R15,APIB.APIB_RETURN_CODE
      LGF R0,APIB.APIB_REASON_CODE
      LGF &APIL_REG,APIB.APIB_CUZ#APIL_ADDR
      DROP APIB
      POP USING
      MEXIT

.*
.RESTCLD ANOP
.* Validate required parms are present and have valid values
      AIF ('&APIB_REG' EQ '').APIBERR
.* AIF ('&CLOUD_NAME' EQ '').CLOUDERR
      AIF ('&RESTORE_WAIT_FOR_COMPLETION' NE 'Y' AND '&RESTORE_WAIT_+
FOR_COMPLETION' NE 'N').WAITERR
      AIF ('&RESTORE_DSN' EQ '').RSDSNERR
      AIF ('&RESTORE_WAIT_FOR_COMPLETION' EQ 'N' AND '&RESTORE_TO_DD+
NAME' NE '').WAITDDR
.* If Restore to DDNAME is requested, remainder of parms are
.* not used since the DD is being allocated by the user.
      AIF ('&RESTORE_TO_DDNAME' GT '').RSTDDDED
      AIF ('&RESTORE_TO_DASD_UNIT' EQ '').RDASDERR
      AIF ('&RESTORE_TO_TAPE_UNIT' EQ '').RTAPEERR
      AIF ('&RETENTION_PERIOD' EQ '').RETNPERR
      AIF ('&RESTORE_TO_DASD_TAPE' NE 'D' AND '&RESTORE_TO_DASD_TAPE+
' NE 'T').DSDTPERR
      AIF ('&RESTORE_DELETE_OLD' NE 'Y' AND '&RESTORE_DELETE_OLD' NE+
'N').DLOLDERR
      AIF ('&RESTORE_TO_DSN' NE '' AND '&RESTORE_TO_ALIAS' NE '').AL+
IASERR
      AIF ('&RESTORE_TO_DSN' NE '' AND '&RESTORE_TO_DDNAME' NE '').D+
DNERR1
      AIF ('&RESTORE_TO_ALIAS' NE '' AND '&RESTORE_TO_DDNAME' NE '')+
.DDNERR2
      AGO .GENREST
.* If Restore to DDNAME is requested, send warnings that all
.* parms below will be ignored.
.RSTDDDED ANOP
      AIF ('&RESTORE_TO_DSN' EQ '').NWRDASD
      MNOTE 4,'RESTORE_TO_DSN is ignored with RESTORE_TO_DDNAME'
.NWRDASD ANOP
      AIF ('&RESTORE_TO_ALIAS' EQ '').NWRALIS
      MNOTE 4,'RESTORE_TO_ALIAS is ignored with RESTORE_TO_DDNAME'
.NWRALIS ANOP
      AIF ('&RESTORE_TO_DASD_TAPE' EQ '').NWRDSTP
      MNOTE 4,'RESTORE_TO_DASD_TAPE is ignored with RESTORE_TO_DDNAME'
.NWRDSTP ANOP
      AIF ('&RESTORE_TO_DASD_UNIT' EQ '').NWDASD
      MNOTE 4,'RESTORE_TO_DASD_UNIT is ignored with RESTORE_TO_DDNAME'
.NWDASD ANOP

```

```

        AIF ('&RESTORE_TO_TAPE_UNIT' EQ '').NWTAPE
MNOTE 4,'RESTORE_TO_TAPE_UNIT is ignored with RESTORE_TO_DDNAME'
.NWTAPE ANOP
        AIF ('&RETENTION_PERIOD' EQ '').NWRETPD
MNOTE 4,'RETENTION_PERIOD is ignored with RESTORE_TO_DDNAME'
.NWRETPD ANOP
        AIF ('&RESTORE_TO_STORCLASS' EQ '').NWSTORC
MNOTE 4,'RESTORE_TO_STORCLASS is ignored with RESTORE_TO_DDNAME'
.NWSTORC ANOP
        AIF ('&RESTORE_DELETE_OLD' EQ '').GENREST
MNOTE 4,'RESTORE_DELETE_OLD is ignored with RESTORE_TO_DDNAME'
.GENREST ANOP
        PUSH USING
APIB    USING APIB_BLK,&APIB_REG
        MVI APIB.APIB_FUNCTION,APIB_FUNCTION_RESTORE
        LA R0,APIB.APIB_CHAR_AREA
        LHI R1,APIB_CHAR_AREA_LEN
        SGR R14,R14
        L R15,=X'40000000'
        MVCL R0,14
        AIF ('&CLOUD_NAME' EQ '').NOCLDNM
        $LADR R1,&CLOUD_NAME
        MVC APIB.APIB_CLOUD_NAME,0(R1)
.NOCLDNM ANOP
        $LADR R1,&RESTORE_DSN
        MVC APIB.APIB_CLOUD_COPY_DSN,0(R1)
        $LADR R1,&RESTORE_REL_GEN
        MVC APIB.APIB_RESTORE_REL_GEN,0(R1)
        AIF ('&RESTORE_TO_DDNAME' EQ '').NODDNAM
        $LADR R1,&RESTORE_TO_DDNAME
        MVC APIB.APIB_RESTORE_TO_DDNAME,0(R1)
        MVI APIB.APIB_RESTORE_WAIT_COMP,C'N'
        AGO .CALLRST
.NODDNAM ANOP
        AIF ('&RESTORE_TO_DSN' EQ '').NORDSN
        $LADR R1,&RESTORE_TO_DSN
        MVC APIB.APIB_RESTORE_TO_DATASET,0(R1)
.NORDSN ANOP
        AIF ('&RESTORE_TO_ALIAS' EQ '').NOALIAS
        $LADR R1,&RESTORE_TO_ALIAS
        MVC APIB.APIB_RESTORE_TO_ALIAS,0(R1)
.NOALIAS ANOP
        $LADR R1,&RESTORE_TO_DASD_UNIT
        MVC APIB.APIB_RESTORE_DASD_UNIT,0(R1)
        $LADR R1,&RESTORE_TO_TAPE_UNIT
        MVC APIB.APIB_RESTORE_TAPE_UNIT,0(R1)
        AIF ('&RESTORE_TO_STORCLASS' EQ '').NOSMS
        $LADR R1,&RESTORE_TO_STORCLASS
        MVC APIB.APIB_RESTORE_TO_STORCLAS,0(R1)
.NOSMS ANOP
        $LADR R1,&RETENTION_PERIOD
        MVC APIB.APIB_RETENTION_PERIOD,0(R1)
        MVI APIB.APIB_RESTORE_DASD_TAPE,C'&RESTORE_TO_DASD_TAPE'
        MVI APIB.DELETE_OLD_RST_DSN,C'&RESTORE_DELETE_OLD'
        MVI APIB.APIB_RESTORE_WAIT_COMP,C'&RESTORE_WAIT_FOR_COMPLETI+
        ON'
.CALLRST ANOP
        IC R15,APIB.APIB_PSW_KEY
        SPKA 0(R15)
        L R15,APIB.CUZ#API_MOD_ADDR
        LR R1,&APIB_REG
        BASR R14,R15
        IC R15,APIB.APIB_PSW_KEY
        SPKA 0(R15)
        L R15,APIB.APIB_RETURN_CODE
        L R0,APIB.APIB_REASON_CODE
        DROP APIB
        POP USING
        MEXIT
.*
.OBTAIN ANOP
        AIF ('&APIB_REG' EQ '').APIBERR
        AIF ('&APIB_REG' EQ 'R0').APIBR0ER
        AIF ('&APIB_REG' EQ '(R0)').APIBR0ER
        AIF ('&APIB_REG' EQ 'R1').APIBR1ER
        AIF ('&APIB_REG' EQ '(R1)').APIBR1ER
        AIF ('&APIB_REG' EQ 'R2').APIBR2ER
        AIF ('&APIB_REG' EQ '(R2)').APIBR2ER
        AIF ('&APIB_REG' EQ 'R14').APIBR14ER
        AIF ('&APIB_REG' EQ '(R14)').APIBR14ER
        AIF ('&APIB_REG' EQ 'R15').APIBR15ER
        AIF ('&APIB_REG' EQ '(R15)').APIBR15ER

```



```

APIB    PUSH    USING
        USING  APIB_BLK,&APIB_REG

        STORAGE OBTAIN,LENGTH=APIB_BLK_LEN,          +
            SP=APIB_SUBPOOL,      User Defined Subpool  +
            CALLRKY=YES,          +
            LOC=ANY,              Default Location      +
            COND=YES

        LTR    R15,R15          Storage Obtain Successful?
        BNZ    CUZ_OBTAIN_ERROR&COUNTER
        LR     &APIB_REG,R1      Save Storage Blk in callers reg
        LA     R0,APIB.APIB_CHAR_AREA
        LHI    R1,APIB_CHAR_AREA_LEN
        SGR    R14,R14
        L      R15,=X'40000000'
        MVCL   R0,14
        MVC    APIB.APIB_EYE_CATCHER,=CL08'CUZ#APIB' Eye Catcher
        MVC    APIB.APIB_VERSION,=CL06'V1R1M1'
        XR     R2,R2            Clear R2
        IPK     Obtain PSW Key
        STC    R2,APIB.APIB_PSW_KEY Save Callers PSW Key
        MVC    APIB.APIB_BLK_LENGTH,=A(APIB_BLK_LEN)
        XC     APIB.APIB_RETURN_CODE,APIB.APIB_RETURN_CODE
        XC     APIB.APIB_REASON_CODE,APIB.APIB_REASON_CODE
        LOAD   EP=&CUZ.#API,ERR=CUZ_LOAD_ERROR&COUNTER
        ST     R0,APIB.APIB_CUZ#API_MOD_ADDR
        SGR    R15,R15
        B      CUZ_OBTAIN_EXIT&COUNTER
CUZ_OBTAIN_ERROR&COUNTER DS 0H
        WTO    '&CUZ.8401E-&SYSECT-Error Obtaining Storage for APIB Loca+
            1 Control Block'
        LGHI   R15,8
        B      CUZ_OBTAIN_EXIT&COUNTER
CUZ_LOAD_ERROR&COUNTER DS 0H
        WTO    '&CUZ.8402E-&SYSECT-Load Failed for &CUZ.#API Module-Proc+
            ess Aborting'
        LGHI   R15,8
CUZ_OBTAIN_EXIT&COUNTER DS 0H
        DROP   APIB
        POP     USING
        MEXIT

.FREE    ANOP
        AIF    ('&APIB_REG' EQ '').APIBERR
        AIF    ('&APIL_REG' EQ '').APILERR
        AIF    ('&APIB_REG' EQ '').APIBERR
        AIF    ('&APIB_REG' EQ 'R0').APIBR0ER
        AIF    ('&APIB_REG' EQ '(R0)').APIBR0ER
        AIF    ('&APIB_REG' EQ 'R1').APIBR1ER
        AIF    ('&APIB_REG' EQ '(R1)').APIBR1ER
        AIF    ('&APIB_REG' EQ 'R2').APIBR2ER
        AIF    ('&APIB_REG' EQ '(R2)').APIBR2ER
        AIF    ('&APIB_REG' EQ 'R14').APIBR14ER
        AIF    ('&APIB_REG' EQ '(R14)').APIBR14ER
        AIF    ('&APIB_REG' EQ 'R15').APIBR15ER
        AIF    ('&APIB_REG' EQ '(R15)').APIBR15ER
        AIF    ('&APIL_REG' EQ 'R0').APIBR0ER
        AIF    ('&APIL_REG' EQ '(R0)').APIBR0ER
        AIF    ('&APIL_REG' EQ 'R1').APIBR1ER
        AIF    ('&APIL_REG' EQ '(R1)').APIBR1ER
        AIF    ('&APIL_REG' EQ 'R2').APIBR2ER
        AIF    ('&APIL_IF ('&APIL_RE').APIBR2ER
        AIF    ('&APIL_IF ('&APIL_RE).APIBR14ER
        AIF    ('&APIL_IF ('&APIL_RE').APIBR14ER
        AIF    ('&APIL_IF ('&APIL_RE).APIBR15ER
        AIF    ('&APIL_IF ('&APIL_RE').APIBR15ER
        PUSH   USING USH USING
APIB    USING  APIB_BSING APIB_BLK
APIL    USING  APIL_BSING APIL_BLK
*       Free the APIL Block if it was Obtained
        LGF    &APIL_REG,APIB.APIB_CUZ#APIL_ADDR
        LTR    &APIL_REG,&APIL_REG
        BZ     CUZ_NO_APIL_BLK&COUNTER
        L      R15,APIL.APIL_BLOCK_LENGTH
        IC     R2,APIL.APIL_PSW_KEY

        STORAGE RELEASE,LENGTH=(R15),          +
            ADDR=(&APIL_REG),      Addr of APIL Block  +
            SP=APIL_SUBPOOL,      Default Subpool      +
            CALLRKY=YES

```

```

XR      &APIB_REG,&APIB_REG    Clear Register

*      Free the APIB Block
CUZ_NO_APIB_BLK&COUNTER DS 0H
XR      R2,R2                  Clear R2
IC      R2,APIB.APIB_PSW_KEY

        STORAGE RELEASE,LENGTH=APIB_BLK_LEN,
        ADDR=(&APIB_REG),      Addr of APIB Block          +
        SP=APIB_SUBPOOL,       Default Subpool             +
        KEY=(R2)               Obtain in Users PSW Key

XR      &APIB_REG,&APIB_REG    Clear R2
DELETE EP=&CUZ.#API
DROP   APIB,APIB
POP    USING
MEXIT

.APIBERR ANOP
MNOTE 12,'APIB_REG is Required for addressability'
MEXIT

.APILERR ANOP
MNOTE 12,'APIL_REG is Required for addressability'
MEXIT

.CLOUDERR ANOP
MNOTE 12,'CLOUD_NAME is Required for this function'
MEXIT

.CPDSNERR ANOP
MNOTE 12,'COPY_DSN is Required for this function'
MEXIT

.DLOLDERR ANOP
MNOTE 12,'RESTORE_DELETE_OLD must be a Y or N'
MEXIT

.DLDSNERR ANOP
MNOTE 12,'DELETE_DSN is Required for this function'
MEXIT

.DSDTPERR ANOP
MNOTE 12,'RESTORE_TO_DASD_TAPE must be a D or T'
MEXIT

.ALIASERR ANOP
MNOTE 12,'RESTORE_TO_DSN and RESTORE_TO_ALIAS are mutually exclusive'
MEXIT

.DDNERR1 ANOP
MNOTE 12,'RESTORE_TO_DSN and RESTORE_TO_DDNAME are mutually exclusive'
MEXIT

.DDNERR2 ANOP
MNOTE 12,'RESTORE_TO_ALIAS and RESTORE_TO_DDNAME are mutually exclusive'
MEXIT

.RDASDERR ANOP
MNOTE 12,'RESTORE_TO_DASD_UNIT is Required for this function'
MEXIT

.RTAPEERR ANOP
MNOTE 12,'RESTORE_TO_DASD_UNIT is Required for this function'
MEXIT

.RETNPERR ANOP
MNOTE 12,'RETENTION_PERIOD is Required for this function'
MEXIT

.WAITDDR ANOP
MNOTE 12,'WAIT_FOR_COMPLETION = N with RESTORE_TO_DDNAME option not allowed'
MEXIT

.APIBR0ER ANOP
MNOTE 12,'R0 can not be used with this function. R0-R2, R14-R15 in use'
MEXIT

.APIBR1ER ANOP
MNOTE 12,'R1 can not be used with this function. R0-R2, R14-R15 in use'
MEXIT

.APIBR2ER ANOP
MNOTE 12,'R2 can not be used with this function. R0-R2, R14-R15 in use'
MEXIT

.APIBR14ER ANOP
MNOTE 12,'R14 can not be used with this function. R0-R2, R14-R15 in use'
MEXIT

.APIBR15ER ANOP
MNOTE 12,'R15 can not be used with this function. R0-R2, R14-R15 in use'
MEXIT

```

MEND

***** Bottom of Data *****

Appendix D: Cloud Tape Connector Messages

All messages generated by Cloud Tape Connector are described in this chapter. A description of each message and information on resolving the problem are provided.

ISPF Messages

This section lists all of the messages that may display while using the menus, options, and fields that are part of this product's interface (ISPF screens). The ISPF messages are listed below.

p

CUZ001E	Multiple executions of the ISPF interface in one TSO session is not allowed.
----------------	---

Explanation:

You are attempting to use multiple ISPF interfaces in one TSO session, which is not allowed.

User response:

Open multiple TSO sessions to use more than one ISPF interface.

CUZ007E	Invalid value entered - Please enter a valid value from the list displayed.
----------------	--

Explanation:

A value was entered that is not allowed on the current panel.

User response:

Change the value to one that is allowed, based on the list displayed.

CUZ008E	Invalid Line Command - Enter "D" to Delete Backup, "R" to Restore Dataset or "X" to Exclude the backup from the display list
----------------	---

Explanation:

The only valid line commands are D, R, and X. The Exclude (X) line command is helpful when used with the RESTORE primary command, which restores the 0 generation of every data set displayed on this list. You can use DSN masking to narrow the list and then use the Exclude (X) line command to further exclude other data sets matching the mask before typing the RESTORE command.

User response:

Change the value to D, R, or X, based on the descriptions listed in the message and on the panel. Be sure you are typing the line command in the "CMD" column next to the line you want to process.

CUZ009E	Option not available at this time. Cloud Connector Started Task is
----------------	---

currently Inactive. Start the Cloud Connector Started Task.

Explanation:

The option you are attempting to use is not available because the Started Task is not active.

User response:

Use the Start command, such as the SDSF command, /S CUZCLOUD, to start the Cloud Tape Connector Started Task. The status of the Started Task is displayed on the Main Menu. When you see "Active" as the status, you can select any option you want.

CUZ010E	Enter a valid Dataset Name containing Cloud Connector Parmlib options.
----------------	---

Explanation:

Type the name of the data set that holds the parameter member. The default data set is the one with the low level of SCUZPARM. The initial contents of the CUZ#PARM member will be copied from the sample provided in SCUZPARM. However, you may have specified a different data set name when you configured this product.

User response:

Type the default data set name, which is SCUZPARM. Otherwise, try to determine what data set name was specified by your administrator when the product was configured.

CUZ011E	Enter a PDS Member name where Cloud Connector Parms will be saved.
----------------	---

Explanation:

A PDS member name is required to store the parameter values for this product.

User response:

Type up to eight characters for a member name from a partitioned data set where you want to save the product parameters. The default member name is CUZ#PARM.

**CUZ012E Parmlib Dataset not cataloged.
Enter a valid Parmlib DSN.**

Explanation:

The Parmlib Dataset name that you provided is not cataloged.

User response:

Type a different data set name. The default Parmlib Dataset name is SCUZPARM.

**CUZ013E Parmlib Dataset cannot be
allocated.**

Explanation:

You are attempting to allocate a Parmlib Dataset, but the data set cannot be allocated.

User response:

Verify that the data set exists and is available for use.

**CUZ014E Parmlib Dataset cannot be
opened.**

Explanation:

You are attempting to open a Parmlib Dataset, but the data set cannot be opened.

User response:

Verify that the data set name is spelled correctly. Also verify that the data set exists and is available for use.

**CUZ015E Member does not exist in Parmlib -
Press enter to create member or
change the member name
entered.**

Explanation:

You have typed a member name that does not exist in the Parmlib data set.

User response:

If you want to create a new member using the name you typed, press Enter. If not, type a different member name.

**CUZ016E Enter a "Y" to Abend on Errors or
"N" Not to Abend on Errors.**

Explanation:

Indicate how you want to handle errors that may occur. You can choose to "abend" (stop) processing or allow the program to continue processing even if an error occurs. The default is "N".

User response:

Type "N" to continue processing, even if an error occurs (default). Type "Y" to force the program to abend when an error occurs.

**CUZ017E User abend code must be a
numeric value between 01 and 99.**

Explanation:

The value you entered was either non-numeric or is not between 01 and 99.

User response:

Type a number between 01 and 99 for the user abend code.

**CUZ018E Debug Mode must be an "A" for
debug All jobs, "J" to debug by
Jobname mask (wildcarding) or
"N" to turn off Debug Mode.**

Explanation

You must specify a value for the Debug Mode option. The valid values are:

- A - Debug all jobs
- J - Debug only the jobs that match this jobname or mask. You can use an asterisk (*) in the mask.
- N - No, do not use debugging. Turn off Debug Mode.

User response:

Type one of the values listed in the message (A, J, or N) to indicate which Debug Mode to use.

**CUZ019E When Debug Mode is set to "J"
(Jobs), a Jobname mask is
required. Enter a specific Jobname
or Jobname Mask (MYJOB*).**

Explanation:

This field requires a one- to eight-character job name.

User response:

You must type a specific job name or a mask for the job name. For example, to debug all job names that begin with "MYJOB" you would type an asterisk at the end (MYJOB*).

**CUZ020E Enter a "Y" to have Messages
written to console or "N" to
suppress Message writing to
console. Warning: Setting to "Y"
may cause flooding if Debug All is
turned on.**

Explanation:

Indicate whether or not you want messages to be displayed on the console. The default is "N", which suppresses the writing of messages to the console. If you choose "Y" and Debug All is turned on, a large amount of messages will be sent to the console.

User response:

While it is recommended that you use the default value of "N", you can change it to "Y" if you prefer.

**CUZ021E Max Cloud Backup Generations
must be a numeric value between
1 and 10. This is the nbr of
generations that will be kept on
the cloud server.**

Explanation:

When the same data is saved multiple times on the cloud, each version is referred to as a "generation". For example, if you save MYDATA for the first time on 11/1/2016, that is the first generation. If you save MYDATA on 11/2/2016, that is the second generation. Both generations of MYDATA are stored on the cloud and subsequent generations will be as well, up to the number you specify in the Max Cloud Backup Generations field. When the maximum number of generations is reached, the new version will be saved and the oldest generation will be deleted from the cloud and removed from the z/OS repository. The maximum number of generations is 10.

User response:

Type a value between 1 and 10 for the number of generations that you want to be saved on the cloud server.

CUZ022E The Cell Pool Size must be numeric value between 25600000 and 99999999. This is the Memory Pools required for copying data to the cloud. The recommended value is 5524288.

Explanation:

You must determine how large each memory cell pool must be to contain the data that will be stored in the cloud. The size must be between 25600000 and 99999999, and the recommended value is 5524288.

User response:

Type a number between 25600000 and 99999999 to specify a size for the cell pools.

CUZ023E The Primary and Secondary Nbr of Cells must be a numeric value between 10 and 9999. This is the nbr of Memory objects that can be obtained. These Memory Cells are not part of your Region size. The recommended value is 250.

Explanation:

You must determine how many primary memory cells and secondary memory cells you will need for the data that will be stored in the cloud. You can specify a number between 10 and 9999. The recommended value is 250.

User response:

Type a number between 10 and 9999 to specify how many Primary and Secondary cells will be needed.

CUZ024E The Memory Cell Extents must be a numeric value between 02 and 9999. 250 is the recommended value.

Explanation:

You must determine how many memory cell extents you want to allow if the maximum number of primary and secondary cells is exceeded. The recommended value is 250.

User response:

Type a number between 02 and 9999 for the number of Memory Cell Extents that are needed.

CUZ025E Please enter a "Y" to Stage backups to DASD or "N" to bypass staging and write directly to the Cloud.

Explanation:

Specify "Y" if you want to "stage" the data by saving it to a temporary file before sending it to the Cloud. If the connection to the cloud is lost, the staged backup could be used to continue the backup once the connection is restored. The recommended value is "Y" for this field.

Specify "N" if you want to write a backup directly to the Cloud. While this may seem to be the obvious choice, the process could be interrupted or disconnected if there are network issues. If a disconnection occurs, no data is saved to the cloud, which means you have two options:

- Restart the job.
- Add the data set to the Include/Exclude list to be backed up on a regularly scheduled basis, such as every hour.

Also copying directly to the cloud can be somewhat slow, based on the speed of the network connection. Therefore, it is recommended that you specify "Y" to stage the data before copying it to the cloud.

User response:

Type a "Y" or an "N" as you prefer.

CUZ026E The Stage Alias must be a valid z/OS Cataloged Alias.

Explanation:

The name of the Staging Dataset Alias is not valid because it is not cataloged. Either add the alias to a catalog or use a different alias name that is already cataloged. The default name is CUZSTAGE.

User response:

Specify a one- to eight-character name for the Staging Dataset Alias or use the default name, CUZSTAGE.

CUZ027E The Staging Allocation must be in "T" (Tracks) or "C" (Cylinders).

Explanation:

The only valid values for the Staging Dataset Allocation field on the Parmlib Staging Options screen are "C" for Cylinders or "T" for Tracks.

User response:

Type either "T" or "C" in the Staging Dataset Allocation field to specify the type of storage you want to use.

CUZ028E The Staging Primary/Secondary Allocations must be a numeric value between 1 and 99999999. Make sure this is large enough to handle all backups.

Explanation:

Staging is recommended to ensure no loss of data in case of connectivity issues with the cloud during the backup process. Consider how many primary and secondary space allocations you will need to stage backups before they are copied to the cloud. The numeric values you specify in the Primary Space Allocation field and the Secondary Space Allocation fields must be between 1 and 99999999.

The allocations must be large enough to handle any and all data sets being backed up. All unused space is freed at the end of the job and the staging data set is deleted after the data has been successfully written to the cloud.

User response:

Type a value between 1 and 99999999 to indicate how many Staging Primary/Secondary Allocations should be assigned.

CUZ029E The Retry count is the nbr of times we will attempt to write stage data to the Cloud when Cloud write errors occur. This must be a numeric value between 1 and 9999.

Explanation:

When you have staged data to DASD and attempt to copy that data to the Cloud, ideally the data will be copied without any issues. However, if the network connection to the Cloud is lost during the copy process, Cloud Tape Connector will continue to try to copy the data to the Cloud based on the number of "retry attempts" you specify in the Error Retry Count field.

User response:

On the Parmlib Staging Options screen, type a number between 1 and 9999 in the Error Retry Count field.

CUZ030E The Retry Interval is the nbr of seconds to wait between Cloud write retries. This must be a valid numeric between 1 and 999 seconds.

Explanation:

One of the fields on the Parmlib Staging Options screen is the "Error Retry Interval Secs" field which

specifies how many seconds to wait before attempting to copy data to the Cloud again. When you have staged data to DASD and attempt to copy that data to the Cloud, ideally the data will be copied without any issues. However, if the network connection to the Cloud is lost during the copy process, Cloud Tape Connector will continue to try to copy the data to the Cloud based on the number of "retry attempts" you specify in the Error Retry Count field. The amount of time between retry attempts is determined by the value you specify in the "Error Retry Interval Secs" field.

User response:

Type a number between 1 and 999 to specify how many seconds to wait before attempting to write to the Cloud again.

CUZ031E Invalid Line Command. Enter a "C" to Create a new Cloud Filter, "D" to Delete the Cloud Filter or "E" to Edit the Cloud Filter.

Explanation:

On the Cloud Filter Display screen, you can create a new cloud filter (C), edit an existing cloud filter (E), or delete a cloud filter (D). If you specify "C" in the Cmd field, you are asked to provide the following information about the new cloud filter:

- filter type
- name of the cloud
- retention period
- filter criteria

If you specify "E" or "D" in the Cmd field, another window appears where you can make changes to an existing filter or confirm the deletion of a filter.

User response:

Type either a C, D, or E in the Cmd line, depending on what you want to do with a cloud filter.

CUZ032E Invalid Filter Criteria - Please enter an "S" to filter by SMS Storage class, "D" to filter by Dataset, "E" to filter by Esoteric Unit, or "R" to backup Cloud Connector Repository.

Explanation:

The value you entered as the Filter Type is not valid. The Repository filter specifies where to save a backup of the entire Cloud Tape Connector repository. No criteria is allowed for this filter type and only one Repository filter can be defined. A backup of the repository is performed every "n" minutes, based on the value you specified in the "Auto Bkup Repository Min" field on the General Options screen (under Product Settings).

User response:

Type either "S" (SMS Storage Class), "D" (Dataset), "E" (Esoteric Unit), or "R" (Repository) to specify a Filter Type for this cloud filter.

CUZ033E The Cloud Name entered has not been defined yet or is invalid. This parmlib member can not be saved until the cloud server is defined.

Explanation:

The Cloud Name you specified is either spelled incorrectly or it has not yet been "defined" to this product. Choose the "Settings" option on the Main Menu, and then select the "Cloud Servers" option to create a new cloud server definition.

User response:

Type a different Cloud Name or add a new cloud definition, as needed.

CUZ034E The Retention Period is invalid. Please enter the nbr of days to keep these backups on the cloud server.

Explanation:

When creating or editing filter criteria for a cloud backup, you must specify a value for the Retention Period field to indicate the number of days that you want to retain this data on the cloud. Valid numbers for the Retention Period are 0000 - 9999. A value of zero is helpful for testing so you can run the expire job and retest without having to wait days for the data sets to expire.

User response:

Type a number between 0000 and 9999 in the Retention Period field.

CUZ035E The Filter Criteria is invalid. Please enter valid filter options for capturing data to the cloud. Wildcarding is available here. Ex: Storage class filter - MYCLASS*.

Explanation:

You can filter data by SMS storage class, data set name, or an esoteric unit. You can also use an asterisk (*) as a wildcard for other characters. In the example, the filter MYCLASS* will select all items with a storage class that begins with "MYCLASS".

User response:

Specify a different filter.

CUZ036E Invalid Value - Please enter a "Y" to delete this filter criteria or "N" to bypass deletion.

Explanation:

On the Cloud Filter Display screen, you can choose to create, edit, or delete filter criteria for cloud backups.

When you type "D" to delete an existing filter, a window displays information about that filter. You must specify a value to either confirm that you want to delete this filter (Y) or return to the previous screen without deleting the filter (N).

User response:

Type "Y" to confirm you want to delete this filter or "N" to keep this filter (not delete it).

CUZ037E A Parmlib Member is required. Please enter a valid Parmlib member name.

Explanation:

Parmlib Member names must be one to eight characters long. The default Parmlib Member name is CUZ#PARM, which is in the sample PARMLIB data set, SCUZPARM.

User response:

Type a name up to eight characters long in the Parmlib Member field. Use CUZ#PARM if you do not have a different Parmlib Member created for this purpose.

CUZ038E Allocation Error - The ISPF DD is already allocated and cannot be deallocated - Process not completed.

Explanation:

The ISPF DD allocation failed. The DD is already allocated and cannot be unallocated for this TSO session. The process did not complete successfully.

User response:

No response required.

CUZ039E Allocation Error - An error was encountered allocating the ISPWRK1 or ISPWRK2 DD - Process not completed.

Explanation:

The ISPWRK1 or ISPWRK DD allocation failed. The process did not complete successfully.

User response:

Verify TSO session parameters are set correctly for your site prior to allocation of these DD statements.

CUZ040E An error occurred opening the file tailoring ISPWRK1 or ISPWRK2 DD.

Explanation:

An error occurred when attempting to open the ISPF work files: ISPWRK1 or ISPWRK2.

User response:

Retry the operation.

CUZ041I The column sizes have been reduced for initial display purposes. Enter command CSIZE

to set the column widths you desire. Enter command CORDER to arrange the columns in the order you desire.

Explanation:

You can modify the column sizes and the column order to fit your purposes, or you can leave the display as it appears.

User response:

No action is required, however you can resize or re-order the columns to suit your display preferences.

CUZ042E	No Cloud Servers have been defined. Saving a parmlib member without any cloud definitions is not allowed. Enter option "3" to define a cloud server.
----------------	---

Explanation:

There must be a filter criteria defined to be able to save a PARMLIB member.

User response:

Select Option 1 (Settings) on the Main Menu and then Option 3 (Cloud Servers). Use the "C" (Create) line command to add a cloud server definition. This is the cloud that will hold the backup. You can add multiple cloud definitions.

CUZ043E	No Filter Criteria has been established. Saving a parmlib member without any filter definitions is not allowed. Enter option "4" to define a filter definition.
----------------	--

Explanation:

To save a PARMLIB member, define the filter criteria.

User response:

At least one filter criteria must be defined before the started task can start. Without any cloud servers defined or filter criteria defined, the product has nothing to do. To save a PARMLIB member, define the filter criteria. Select Option 1 (Settings) on the Main Menu, then Option 4 (Backup Filter Criteria). Use the "C" (Create) line command to create filter criteria to select the data sets you want to backup onto the cloud.

CUZ044E	Cloud Name has already been defined. A Cloud Name used in a Cloud definition must be unique. Change the Cloud Name or delete the cloud with the same name.
----------------	---

Explanation:

You are creating a new cloud server definition, but the name you typed in the "Cloud Name" field is already used by a different cloud server definition. Each Cloud Name must be unique.

User response:

Type a different Cloud Name for this new cloud definition or delete the existing cloud definition that uses the same name.

CUZ045E	Max Backup History Tasks must be a numeric value between 1 and 99. This is the nbr of datasets already created that will be backed up to the cloud at the same time. Set this to the max nbr of tape drives that can be allocated at the same time.
----------------	--

Explanation:

The number you specify in the "Max Backup History Tasks" field indicates the maximum number of tape drives that can be allocated at the same time to write existing data sets to the cloud. This parameter will prevent Cloud Tape Connector from absorbing all of the tape units in your system.

User response:

Type a number between 1 and 99 for the maximum number of tape drives that can be allocated at the same time to write existing data sets to the cloud.

CUZ046E	The Restore Alias must be a valid z/OS Cataloged Alias.
----------------	--

Explanation:

When restoring a data set that is currently saved to the cloud, you can either restore it to the existing data set name or you can use a different data set name, which is an "alias" name. In this case, the alias that you entered cannot be used because the alias is not cataloged.

User response:

Either add the alias to the catalog or specify a different alias that is already cataloged.

CUZ047E	Max Restore Tasks must be a numeric value between 1 and 99. This will regulate the nbr of tape drives that can be occupied at the same time.
----------------	---

Explanation:

When restoring data sets from the cloud, setting the "Max Restore Tasks" parameter will allow you to control the number of tape drives that Cloud Connector can use at the same time. Each drive will be used to restore one data set. The default value is 5.

User response:

Type a number between 1 and 99 for the maximum number of restore tasks that can occur simultaneously to restore data sets from the Cloud(s). Do not type a number that is higher than the actual number of tape drives you have at your site.

CUZ048E **When Restore to DASD is set to Yes, a valid DASD device is required to restore. Please enter a valid DASD Device type.**

Explanation:

The "restore" process copies data from the cloud to DASD or tape. If you have a value of "Y" in the "Restore to DASD" field, you must specify a one- to eight-character device name in the "Restore DASD Unit" field. The default name is SYSALLDA. You can use this default or specify a DASD unit device that is appropriate for your site.

If the block size is greater than 32K, the restored data must be sent to tape rather than DASD. For this reason, you must also specify a value in the "Restore TAPE Unit" field. The default name is TAPE. You can use this default or specify a value that is appropriate for your site.

User response:

Specify a DASD device in the "Restore DASD Unit" field or use the default, which is SYSALLDA. Also provide a value for the "Restore TAPE Unit" field or use the default, which is TAPE.

CUZ049E **A Tape Device type is required for restoring datasets from the Cloud. Please enter a valid Tape Device type.**

Explanation:

When restoring data sets from the cloud, if a data set has a block size that is greater than 32K, the data set will be restored to tape. Also, if you specify an "N" in the "Restore to DASD" field, the data set will be restored to tape.

User response:

Specify a one- to eight-character value for the "Restore TAPE Unit" field or use the default, which is TAPE.

CUZ050E **The Retention Period is invalid. Please enter the nbr of days to keep the restored datasets on tape.**

Explanation:

The Retention Period indicates how long to keep the data on tape. The number of days can range from 0 - 9999. A value of zero is helpful for testing so you can run the expiration job and retest without having to wait days for the data set to expire.

User response:

Type a number between 0 and 9999 for the number of days to keep the restored data on tape.

CUZ051E **Invalid value. Please enter a "Y" to restore datasets to DASD or "N" to restore datasets on tape.**

Explanation:

The only valid values for this field are Y or N. Data sets from the cloud can be restored to a DASD unit as long as the block size (blksize) is less than 32K. If it is larger than 32K, the data set must be restored to tape.

User response:

Type either "Y" to restore data sets to DASD or "N" for tape. If you type a "Y" in the "Restore to DASD" field, you must also provide values for the "Restore DASD Unit" and "Restore TAPE Unit" fields. You must always specify a value for the "Restore TAPE Unit" field.

CUZ052E **Invalid value. Please enter a numeric value between 1 and 50. Nbr of IO Buffers default value is 10.**

Explanation:

When staging data on DASD before copying the data to the cloud, it is important to have several Input/Output Buffers so the job writing the data does not have to wait for server I/O. The default number of buffers is 10, but you can specify up to 50.

User response:

Type a value from 1 - 50 in the "Nbr of IO Buffers" field for the number of I/O buffers for staging data to DASD. The default is 10.

CUZ053E **Invalid Line Command - Enter "D" to Delete the DSN line, "I" to Insert a new DSN line or "R" to Repeat this line.**

Explanation

When you choose Option 4, Backup History Datasets, on the Main Menu, you can create a list of data sets to include or exclude. The data sets on this list are copied to the cloud in the background processing of the Cloud Connector started task. The only valid line commands available on the Include/Exclude History screen are the following:

- I - Insert a blank line to create a new data set filter.
- D - Delete a blank line or an existing data set filter.
- R - Repeat this line.

User response:

Type either "I" to insert a new filter, "D" to delete a filter, or "R" to repeat a line.

CUZ054E **Invalid Include/Exclude indicator. Please enter an "I" to Include these Datasets or "E" to Exclude these Datasets.**

Explanation

The only valid values for the Inc/Exc column are the following:

- I - Include these data sets when performing a backup.
- E - Exclude these data sets from a backup.

User response:

Change the value you typed in the Inc/Exc column to an "I" or an "E".

CUZ055E	Invalid Dataset Masking. Please enter valid Dataset Like criteria to include/exclude in history processing.
----------------	--

Explanation:

The data set mask you entered as criteria on the History Include/Exclude screen is not valid.

User response

Type up to 44 characters for the data set name, data set mask, or data set filter you want to include or exclude during backups of history data sets. Use an asterisk (*) to indicate "any characters" are accepted. For example, if you typed MYUSERID.MYOLDLIB.* as the mask, all of the data sets owned by MYUSERID in the MYOLDLIB library would be selected for processing.

If an "I" is assigned to this data set mask, the data sets will be included in processing. If an "E" is specified, the data sets will be excluded from backup history processing.

CUZ056E	Invalid Entry. Please enter a "Y" to restore this dataset, "N" to cancel restoring this dataset or "A" to restore all the datasets selected.
----------------	---

Explanation:

On the Restore Dataset Confirmation window, you are asked whether you want to restore the data set that is listed in the window. If you typed an "R" on two or more lines on the Cloud Dataset Display screen, a confirmation window displays for each data set to verify that you want to restore it. If you selected a large number of data sets and are certain that you want to restore all of them, type "A" in the Restore Dataset field to restore them all.

User response:

Specify "Y" (yes), "N" (no), or "A" (all) in the Restore Dataset field.

CUZ057E	Invalid Restore Alias. The Restore Alias must be a valid z/O Cataloged Alias.
----------------	--

Explanation:

You can restore the data set to the same name, a different name (alias), or to the default alias name (CUZRESTR).

User response:

If you want to restore the data set to a different name, type up to eight characters for the alias. The default alias is CUZRESTR. If you prefer to use the original data set name, rather than an alias, leave this field blank.

CUZ058E	Invalid Entry. Please enter a "Y" to delete this cloud backup, "N" to cancel deleting this cloud backup or "A" to delete all the cloud backups selected.
----------------	---

Explanation

On the Delete Cloud Backup Confirmation window, you are asked whether you want to delete the backup data set that is listed in the window.

- If you only selected this one backup data set to be deleted, type either "Y" (yes) or "N" (no) in the Delete Cloud Backup field.
- If you typed a "D" on two or more lines on the Cloud Dataset Display screen, a confirmation window displays for each data set to verify that you want to delete it.
- If you selected a large number of data sets and are certain that you want to delete all of them, type "A" in the Delete Cloud Backup field to delete them all.

User response:

Type a "Y" (yes), an "N" (no), or an "A" (all) in the Delete Cloud Backup field. The default value is N.

CUZ059E	Auto Bkup Repository minutes must be a numeric value between 05 and 9999. This will create a backup of the repository to the cloud every <i>n</i> minutes. It will also rebuild the Include / Exclude history list plus restart any failed cloud staging writes.
----------------	---

Explanation:

A backup of the repository can be saved to the cloud on a regular basis. When this happens, the Include/Exclude history list is also rebuilt and any staging jobs that failed when attempting to write data to the cloud are also restarted.

User response:

Type a number between 05 and 9999 for the number of minutes to wait between automatic backups to the cloud repository. The default is 60.

CUZ060E	Cloud Name Required. A Cloud Name is required for all datasets in the include list. Please enter a
----------------	---

valid pre-defined cloud name listed in the Parmlib options.

Explanation:

When creating a new data set filter on the Include/Exclude History screen, you must specify a Cloud Name so Cloud Tape Connector knows where to copy the data. This Cloud Name must have been previously defined to Cloud Tape Connector. To create a cloud server definition, select the "Settings" option on the Main Menu, then choose the "Cloud Servers" option on the Parmlib Options Main Menu.

User response:

Type up to eight characters in the Cloud Name field to specify a name associated with this new filter.

CUZ061E Invalid Primary command. Only ISPF primary commands are supported on this screen.

Explanation:

You are attempting to perform an operation that requires the use of an ISPF primary command.

User response:

Type an ISPF primary command, not a line command.

CUZ062E Repository Filter already created. Only 1 Repository Filter may be created per Cloud Connector subsystem.

Explanation:

You are attempting to create a cloud filter using "R" (repository) as the Filter Type, when another repository filter already exists. The repository can be backed up to only one cloud, so only one repository filter is allowed.

User response:

Specify a different type of filter. Other filter types include Storage Class (S), Esoteric Unit (E), and Dataset (D).

CUZ063W No Repository Filter was found. Please define a Repository Filter to enable backing up Cloud Connector repository to a cloud server. If you choose not to define a Repository Filter, auto repository backups will be disabled.

Explanation:

In this scenario, automatic backups to a repository on a cloud cannot be performed because no Repository Filter has been defined. No data sets are being written to the repository.

User response:

Define a Repository Filter to enable the repository to be automatically backed up to the cloud.

CUZ064E Storage acquisition error obtaining another memory cell. There is not enough storage to display all datasets. Narrow down your dataset like criteria and re-enter.

Explanation:

The "Dataset Like" field on the Enter Dataset Criteria screen contains a value that results in too many data sets being selected for display.

User response:

If you used an asterisk (*) in this field, try to be more specific with a data set name. To increase the number or size of memory cells, refer to ["General Options"](#) on page 24.

CUZ065E Error building a storage cell pool. Check the region size of your TSO session and try again.

Explanation:

The cell pool requires more space than is currently available in the region.

User response:

Increase the region size of your TSO session and try again.

CUZ066E Invalid value. Please enter a "1" to define an S3-compatible cloud server, "2" for FTP, "3" for HCP, "4" for SFT, or "5" for S3 or "6" for Proxy server.

Explanation

The value you entered in the "Create Cloud Type" field is not supported. Valid cloud types include:

1. CLS - S3-Compatible
2. FTP - File Transfer Protocol
3. HCP - Hitachi Control Platform
4. SFT - SoftLayer (IBM)
5. S3 - Simple Storage Service (Amazon)
6. PRX - Proxy Server

User response:

Specify one of the Cloud Types listed above.

CUZ067E Invalid Dataset Name. The dataset name entered does not meet z/OS dataset naming standards. Cursor is pointing to location of dataset error location.

Explanation:

When saving your settings to a parameter member, you must provide the name of a data set where the member will be saved. The data set name can be up to 44 characters in length. The default data set name is

SCUZPARM, which is the sample data set that is included with this product.

User response:

Type up to 44 characters for the name of the data set that will contain the parameter member.

CUZ068E **Invalid Member Name. The member name entered does not meet z/OS dataset naming standards.**

Explanation:

The Parmlib Member name you specified contains invalid characters. The SCUZPARM data set included with this product contains an example of a parameter member, CUZ#PARM.

User response:

Type a one- to eight-character name for the parameter member. The default is CUZ#PARM.

CUZ069I **Member *parameter member* was successfully saved.**

Explanation:

The listed Parmlib Member has been saved.

User response:

.

CUZ070I **Member &PARMLMBR was not saved.**

Explanation:

The Parmlib Member listed in the message was not saved.

User response:

This is an informational message. No action is required.

CUZ071E **Invalid Value - Please enter a "Y" to delete this cloud server definition or "N" to bypass deletion**

Explanation:

At this point, you must choose to go forward with deleting the cloud server definition (Y) or cancel it (N).

User response:

The only valid entries for this field are "Y" (yes, delete this cloud server definition) or "N" (no, cancel the deletion and keep this cloud server definition).

CUZ072E **The Restore Alias can not be the same as the Staging Alias. Please enter an Alias different than the Staging Alias**

Explanation:

The Restore Alias and the Staging Alias must be different Alias names.

User response:

If you want to restore the data set to a different name, type up to eight characters for the Restore Alias. The default alias is CUZRESTR. If you prefer to use the original data set name, instead of an alias, leave this field blank.

CUZ073E **The filter criteria can not be more than 8 characters for storage class and esoteric unit filters**

Explanation:

The name of an esoteric unit filter or a storage class filter can be up to eight characters. An asterisk (*) can be used as a wildcard.

User response:

Specify filter criteria that is a maximum of eight characters for a storage class or an esoteric unit.

CUZ074E **The filter dataset mask does not meet z/OS dataset naming standards. Please enter a valid dataset mask**

Explanation:

You typed characters in the data set mask that are not acceptable.

User response:

Change the data set mask to ensure it meets naming standards.

CUZ075E **The restore dataset name does not meet z/OS dataset naming standards. Please enter a valid restore dataset name**

Explanation:

You typed characters in the data set name that are not acceptable.

User response:

Change the data set name to ensure it meets naming standards.

CUZ076E **Restoring to Disk is not supported when the block size exceeds 32768. This Dataset must be restored to tape**

Explanation:

If the block size is greater than 32K, the restored data will be sent to tape rather than DASD.

User response:

Specify a value in the "Restore TAPE Unit" field. The default name is TAPE. You can use this default or specify a tape value that is appropriate for your site.

CUZ077E **Invalid Restore Unit. Unit must be T for Tape or D for Disk**

Explanation:

You have typed an invalid character in the Restore Unit Type field on the Restore Dataset Confirmation screen.

You can choose to restore data to Tape (T) or to Disk (D).

User response:

Type either a "T" (tape) or a "D" (disk) in the Restore Unit Type field.

CUZ078E **Restore Alias and Restore Dataset are mutually exclusive. Please enter an Alias or Dataset, but not both.**

Explanation:

When restoring data, you can either specify up to 44 characters for a Restore Dataset or up to 8 characters for a Restore Alias, but not both.

User response:

Specify either a Restore Dataset or a Restore Alias.

CUZ079E **Restoring dataset is a GDG without a GDG base. Option not allowed.**

Explanation:

The data set being restored is part of a generation data group (GDG), but there is no base for the GDG. In this case, restoration of the data set is not allowed.

User response:

Define a GDG base for this data set alias or remove/change the last node so it does not meet GDG naming standards.

CUZ080E **Invalid DASD Unit parameter. Please enter a valid DASD Esoteric unit parameter.**

Explanation:

The DASD unit your specified is not defined as a valid esoteric DASD unit at your installation.

User response:

Enter a DASD esoteric unit parameter allowed by your installation.

CUZ081E **Invalid Tape Unit parameter. Please enter a valid Tape Esoteric unit parameter**

Explanation:

The TAPE unit your specified is not defined as a valid esoteric TAPE unit at your installation.

User response:

Enter a TAPE esoteric unit parameter allowed by your installation.

CUZ082E **Invalid value. Please enter a "Y" to delete the existing backup or "N" to bypass deleting the existing backup.**

Explanation

You are being asked to confirm that you want to delete an existing backup. The only valid entries in this field are:

- Y - Yes, delete the existing backup.
- N - No, cancel the deletion process and keep the existing backup.

User response:

Specify either "Y" or "N" as you prefer.

CUZ083E **Restore will fail because the restore dataset is cataloged and delete old backups is set to No. Set delete old backups to Yes or manually delete the old backup.**

Explanation:

If a new restore data set is already cataloged, the restore process will fail. The old backup data set must be deleted before the restore process is started.

User response:

Either specify "Yes" in the "Delete Old Backups" field or manually delete the old backup so that the cataloged data set can be restored.

CUZ084E **The same dataset is being restored more than once. Please select only one dataset with the same name to be restored at one time.**

Explanation:

Two data sets with the same name have been selected to be restored. However, only one of the data sets can be restored.

User response:

Change your selections so that only one data set is restored.

CUZ085E **Option not allowed. You can not reply "All" when entering a restore to dataset name. Multiple restores to the same dataset name will fail.**

Explanation:

The "All" value cannot be used in this case because the same data set name cannot be restored multiple times.

User response:

Type a "Y" to restore the data set and a new screen will appear for the other selected data sets.

CUZ086E **Invalid Value. S3-Compatible requires the Key ID to be a 1 to 128 byte alphanumeric string. Please enter a valid Key ID.**

Explanation:

The Key ID is the server ID that is required for signing into this cloud. The Key ID must be 1 - 128 characters in length.

User response:

Type a Key ID that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ087E	Invalid Value. S3-Compatible requires the Secret Access Key to be a 1 to 128 byte alphanumeric string. Please enter a valid Secret Access Key.
----------------	---

Explanation:

The Secret Access Key is similar to a password and is associated with the Key ID that allows access to this cloud. The Secret Access Key must be 1 - 128 characters in length.

User response:

Type a Secret Access Key that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ088E	Invalid Value. S3-Compatible requires the IP Address to be a 1 to 128 byte alphanumeric string. Please enter a valid IP Address.
----------------	---

Explanation:

The Internet Protocol Address is used to connect to the cloud. The IP Address must be 1 - 128 characters in length.

User response:

Type an IP Address that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ089E	Invalid Value. S3-Compatible requires the Bucket Name to be a 3 to 63 byte alphanumeric string. Please enter a valid Bucket Name.
----------------	--

Explanation:

The Bucket Name is the name or location of the repository on the cloud where the data will be saved. The Bucket Name must be 3 - 63 characters in length.

User response:

Type a Bucket Name that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ090E	Invalid Value. FTP requires the User ID to be a 1 to 64 byte alphanumeric string. Please enter a valid User ID.
----------------	--

Explanation:

The User ID is the server ID that is required for signing into this cloud. The User ID must be 1 - 64 characters in length.

User response:

Type a User ID that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ091E	Invalid Value. FTP requires the Password to be a 1 to 64 byte alphanumeric string. Please enter a valid Password.
----------------	--

Explanation:

This password is associated with the User ID that allows access to this cloud. The password must be 1 - 64 characters in length.

User response:

Type a password that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ092E	Invalid Value. FTP requires the IP Address to be a 1 to 128 byte alphanumeric string. Please enter a valid IP Address.
----------------	---

Explanation:

The Internet Protocol Address is used to connect to the cloud. The IP Address must be 1 - 128 characters in length.

User response:

Type an IP Address that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ093E	Invalid Value. SoftLayer requires the User ID to be a 1 to 64 byte alphanumeric string. Please enter a valid User ID.
----------------	--

Explanation:

The User ID is the server ID that is required for signing into this cloud. The User ID must be 1 - 64 characters in length.

User response:

Type a User ID that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ094E	Invalid Value. SoftLayer requires the Password to be a 1 to 64 byte alphanumeric string. Please enter a valid Password.
----------------	--

Explanation:

This password is associated with the User ID that allows access to this cloud. The password must be 1 - 64 characters in length.

User response:

Type a password that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ095E **Invalid Value. SoftLayer requires the IP Address to be a 1 to 128 byte alphanumeric string. Please enter a valid IP Address.**

Explanation:

The Internet Protocol Address is used to connect to the cloud. The IP Address must be 1 - 128 characters in length.

User response:

Type an IP Address that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ096E **Invalid Value. S3 requires the Key ID to be a 1 to 128 byte alphanumeric string. Please enter a valid Key ID.**

Explanation:

The Key ID is the server ID that is required for signing into this cloud. The Key ID must be 1 - 128 characters in length.

User response:

Type a Key ID that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ097E **Invalid Value. S3 requires the Secret Access Key to be a 1 to 128 byte alphanumeric string. Please enter a valid Secret Access Key.**

Explanation:

The Secret Access Key is similar to a password and is associated with the Key ID that allows access to this cloud. The Secret Access Key must be 1 - 128 characters in length.

User response:

Type a Secret Access Key that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ098E **Invalid Value. S3 requires the IP Address to be a 1 to 128 byte alphanumeric string. Please enter a valid IP Address.**

Explanation:

The Internet Protocol Address is used to connect to the cloud. The IP Address must be 1 - 128 characters in length.

User response:

Type an IP Address that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ099E **Invalid Value. S3 requires the Bucket Name to be a 3 to 63 byte alphanumeric string. Please enter a valid Bucket Name.**

Explanation:

The Bucket Name is the name or location of the repository on the cloud where the data will be saved. The Bucket Name must be 3 - 63 characters in length.

User response:

Type a Bucket Name that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ100E **Invalid Value. HCP requires the User ID to be a 1 to 64 byte alphanumeric string. Please enter a valid User ID.**

Explanation:

The User ID is the server ID that is required for signing into this cloud. The User ID must be 1 - 64 characters in length.

User response:

Type a User ID that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ101E **Invalid Value. HCP requires the Password to be a 1 to 64 byte alphanumeric string. Please enter a valid Password.**

Explanation:

This password is associated with the User ID that allows access to this cloud. The password must be 1 - 64 characters in length.

User response:

Type a password that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ102E **Invalid Value. HCP requires the IP Address to be a 1 to 128 byte alphanumeric string. Please enter a valid IP Address.**

Explanation:

The Internet Protocol Address is used to connect to the cloud. The IP Address must be 1 - 128 characters in length.

User response:

Type an IP Address that is the proper length and that uses either alphabetic or numeric characters, or a combination of both.

CUZ103E Authorization Failure - You are not authorized to view this dataset

Explanation:

You are attempting to access a data set that you do not have the authority to use.

User response:

Either choose a different data set or talk to your System Administrator about granting you the proper authority.

CUZ104E Authorization Failure - You are not authorized to write to this dataset. Saving this member is not allowed in this dataset.

Explanation:

You have attempted to save a member to a data set, but you do not have the authority to write to this data set.

User response:

Either try saving this member to a different data set or contact your System Administrator about granting the authority you need to write to this data set.

CUZ105E You cannot restore to a relative GDG "(+1)", it must be an absolute GDG or no GDG. Remove the relative GDG from the end of the dataset.

Explanation:

When specifying a data set name, use an absolute generation data group (the 0 generation), not a relative GDG. For example, if you are restoring a data set, `USERID.DISK.CUZ.GDG0`, you could restore it to `RESTR.DISK.CUZ.GDG0` (notice that "USERID" is changed to "RESTR"), but not to a relative data set, such as `RESTR.DISK.CUZ(+1)`.

User response:

Either specify a data set name with an absolute GDG or one that does not have a GDG. Another option is to remove the relative GDG from the end of the data set name.

CUZ106E Invalid Value. SoftLayer requires the Container to be a 1 to 128 byte alphanumeric string. Please enter a valid Container.

Explanation:

This type of server definition requires a "Container" as one of the values in the definition.

User response:

Type up to 128 characters to identify the Container that will hold the saved data.

CUZ107I Restore complete - <message>

Explanation:

The Restore Request you waited for is now complete. It may have completed with errors. <Message> is the message returned by the Restore process.

User response:

If the message indicates that an error occurred, correct the error and re-submit the request.

CUZ108E Authorization Failure - You must have alter authority on DSN Parmlib Data Set Name in order to restore it.

Explanation:

To restore a data set from the cloud back to z/OS, the TSO User ID must have alter RACF authority on the original data set. You received this message because your User ID does not have alter authority on the original data set.

User response:

Either take steps to have alter RACF authority assigned to your User ID or use a TSO User ID that does have that type of authority.

CUZ109E Invalid Volume Count - The Restore Volume Count must be a numeric value between 1 and 256.

Explanation:

The **Restore Volume Count** parameter is used when allocating a **Restore to Dataset** value, and will inform z/OS of the number of volumes this data set can span.

User response:

Make sure the volume count is large enough to handle any data sets being restored.

CUZ110E Invalid Volume Count - The Staging Volume Count must be a numeric value between 1 and 256.

Explanation:

The **Staging Volume Count** parameter is used when allocating a **Staging Dataset** value, and will inform z/OS of the number of volumes this data set can span.

User response:

Make sure the volume count is large enough to handle any data sets being backed up.

CUZ111E Invalid Value - Enter a "Y" to Catalog the backup to the cloud or "N" to keep backup cataloged as is

Explanation

This parameter, **Catalog to Cloud**, determines whether will re-catalog the data set on the cloud.

- If set to "Y", will delete the disk data set or uncatalog the tape data set and re-catalog it with *volume serial* = CLOUD. The data set will be restored back to z/OS when allocated.

- If set to "N", a copy of the data set will reside on the cloud and the data set will also reside on z/OS.

User response:

Type either "Y" or "N" as described above.

CUZ112E Option not Allowed - Recataloging a repository backup filter is not supported.

Explanation:

This **Catalog to Cloud** option is not allowed on a Repository type filter. The repository backup data set is not cataloged and only resides on the cloud.

User response:

Clear any value you specified for the **Catalog to Cloud** option for this repository.

CUZ113I Dataset *data set* was successfully deleted and uncataloged from the cloud

Explanation:

Dataset *data set* was cataloged with *volume serial* = CLOUD. The data set was deleted from the cloud and was also uncataloged from z/OS.

User response:

CUZ114E An error occurred uncataloging dataset *data set name* from the cloud.

Explanation:

Data set *data set name* was cataloged with *volume serial* = CLOUD. An attempt was made to uncatalog the data set from z/OS, however, it could not be changed. The data set was deleted from the cloud, but is still cataloged to z/OS.

User response:

Try to determine why the data set could not be uncataloged from z/OS.

CUZ115E Enter a "Y" to Share the Repository across LPARS or "N" for non-sharing. Sharing requires VSAM RLS.

Explanation

This option determines if the repository is going to be shared across multiple LPAR's in the same sysplex. Valid options are:

- Y – VSAM RLS (record-level sharing) is enabled for the integrity of the repository.
- N – VSAM RLS is disabled, but only one CUZCLOUD STC (started task) can be started with the same repository.

User response:

Specify either "Y" or "N". If you are uncertain, choose "N".

CUZ116E Invalid Value - The Roll in GDG Base must be a "Y" to roll in the GDS, "N" to leave it in Defer (Not Rolled In), or "C" for Conditionally Roll in the GDS. Press F1 for a full explanation of these values.

Explanation

The **Roll into GDG Base** field allows you to restore active or deferred generation data group files, and also to force the restoration of a GDS data set, if the situation requires it. Indicate whether to modify the generation data group (GDG) index. Valid values are:

- **Y** - For SMS-managed DASD and Tape data sets, the generation data set (GDS) will be rolled into the active generation data group (GDG) index, regardless of its current state. The dataset (GDS) can be referenced by a relative generation number.
- **N** - For SMS-managed DASD data sets, the GDS will be restored in Deferred state, regardless of its current state. For SMS-managed Tape data sets, the GDS will be restored, but the data set will be renamed from G####V## to G####X##.
- **C** - This is the default. The restore process varies, depending on the present state of the GDS.

For more information, you can press F1 for a help screen or see [“Considerations for restoring GDG's” on page 74](#) for a detailed explanation of these values.

User response:

Type "Y", "N", or "C" in the **Roll into GDG Base** field. If you do not know which value to choose, use the default value of "C".

CUZ117W Encryption Failure - An error was encountered generating an encryption key. Refer to the z/OS Integrated Cryptographic Service Facility (ICSF) Application Programmer's Guide for a list of Return and Reason codes. RC=return code RS=reason code

Explanation:

The return code and reason code listed in the message may help you determine the source of the encryption key error. Refer to Appendix A in the *IBM z/OS Cryptographic Services: Integrated Cryptographic Service Facility (ICSF) Application Programmer's Guide* (SC14-7508) for a list of Return and Reason codes.

User response:

CUZ118E Invalid Value - Enter a "Y" to Encrypt the Cloud Definitions or

**"N" to save the Cloud Definitions
in displayable format**

Explanation:

When saving or updating a Parmlib member, you can choose to encrypt the cloud definitions. This prevents the actual information from being displayed, printed, or sent as write-to-operation (WTO) messages at startup.

User response:

Specify "Y" or "N".

CUZ119E	Encryption Services are not available. Encryption is disabled.
----------------	---

Explanation:

The encryption process is not available at this time, so the cloud encryption feature has been disabled.

User response:

Wait a few minutes and try again.

CUZ120E	Encryption Failure - An error was encountered encrypting the parmlib values. Refer to the z/OS Integrated Cryptographic Service Facility (ICSF) Application Programmer's Guide for a list of Return and Reason codes. RC=return code RS=reason code
----------------	--

Explanation:

The return code and reason code listed in the message may help you determine the source of the encryption error. Refer to Appendix A in the *IBM z/OS Cryptographic Services: Integrated Cryptographic Service Facility (ICSF) Application Programmer's Guide* (SC14-7508) for a list of Return and Reason codes.

User response:

CUZ121W	Display Limit Reached - The number of lines to display is larger than the screen can hold. Please refine your display list using the filter above (Jobname Like, Dataset Like or Cloud Like)
----------------	---

Explanation:

There are too many items to display on the screen. You will need to specify filter criteria to reduce the number of lines displayed.

User response:

To shorten the list, specify criteria based on job name, data set name, or cloud name.

CUZ122E	Invalid Value. TCP/IP protocol requires port number to be between 1 to 65535. Please enter a valid port number.
----------------	--

Explanation

Identifies the port by which Cloud Tape Connector attempts to connect to the cloud server. Valid values must be between 1 - 65534. The default is 80.

User response:

Specify a different number.

CUZ123E	Enter a "Y" to enable compression, "N" to disable it.
----------------	--

Explanation:

The "Compress" field indicates whether to use zEnterprise Data Compression (zEDC) to compress the data when transferring to the cloud. The default is N (no).

User response:

Specify either Y or N. If the field is blank, the default is N.

CUZ124E	Connection Retry Attempts must be a numeric value between 0 and 99. This is the number of times the product will attempt to reconnect if the socket connection is dropped.
----------------	---

Explanation:

The "Connection Retry Attempts" field indicates the number of times to attempt to reconnect with the cloud if the socket connection is dropped. Valid values are 0-99. The default is 3.

User response:

Type a number from 0 - 99.

CUZ125E	Connection Retry Time must be a numeric value between 0 and 999. The amount of seconds to wait between connection attempts.
----------------	--

Explanation:

The "Connection Retry Time" field indicates the number of seconds to wait between attempts to connect to the cloud. Valid values are 0 - 999. The default is 30 seconds.

User response:

Type a different number in the field.

CUZ126E	Data Transfer Retry must be a numeric value between 0 and 99. This is a number of times the product will have to retry unsuccessful I/O operation.
----------------	---

Explanation:

The "Data Transfer Retry" field indicates the number of times to retry a data transfer (I/O operation) if it is unsuccessful the first time. Keep in mind that a data transfer can be significantly more expensive than other

network communications, because the Part Size can be 100MB and above. The default is 0.

User response:

Specify a value between 0 - 99. If you are uncertain, use the default value of 0 or another low number.

CUZ127E **Enter a "Y" to enable Virtual Host URL Style, "N" to disable it.**

Explanation:

Indicates whether the connection to an S3-Compatible (CLS) or Amazon Simple Storage Service (S3) server should have the bucket name in front of the CLS or S3 host name. For example, if this is enabled for an S3 server, the product will connect to `bucketname.s3.amazonaws.com`, instead of the standard connection to `s3.amazonaws.com`. This is useful in some installations where advanced network routing is in place.

User response:

Type either "Y" for yes or "N" for no. The default is N.

CUZ128E **Enter a "Y" to enable Server-side Encryption, "N" to disable it.**

Explanation

Specifies whether server-side encryption will be supported.

- If a "Y" (yes) is specified, Cloud Connector will add the proper header to CLS traffic to allow it to be in compliance with a bucket policy that requires server-side encryption.
- If an "N" (no) is specified or if the field is left blank, no additional header will be used. The default is N.

User response:

Type a "Y" or an "N" (or leave blank) to indicate whether to enable server-side encryption.

CUZ129E **Number of Threads is a numeric value between 0 and 20. This is a number of additional threads used for writing data to the cloud.**

Explanation:

The "Number of Threads" field specifies that multiple concurrent threads will be used when writing data to S3 or CLS clouds. Using multiple threads can help large backups complete more quickly. With each thread that is specified, there is an increase in the amount of memory that is needed. The default is 0.

User response:

Specify a value between 0 - 20. If you specify a value of 0, no multi-threading will be performed.

CUZ130E **Part size must be a numeric value between 5 and 100. This is a number the maximum amount of**

data, in MB, to be transferred in a transaction.

Explanation:

Indicates the maximum amount of data (part size), in MB, that will be transferred to the cloud during a transaction. Specify a number between 5 – 100. The default is 5MB. The bigger the number, the better the performance will be, but at a memory cost. Specifying 100MB will gain the most in terms of performance, but does allocate a large memory area to hold the data.

User response:

Type a number between 5 - 100. The default is 5.

CUZ131E **Invalid Value. TCP/IP protocol requires port number to be between 1 to 65535 for FTP connection. Please enter a valid port number.**

Explanation:

Identifies the port by which attempts to connect to the cloud server. Valid values are 1 - 65535. The default is 80.

User response:

Type a number from 1 - 65535. If you are uncertain what to enter, use 80, which is the default.

CUZ132E **Invalid Value. Enter a "Y" to refresh the Parm's in the Cloud Connector Started Task. Enter an "N" to bypass the STC refresh**

Explanation:

If you have changed any of the parameters and would like to have them take effect immediately, specify a "Y" in this field to refresh the started task so that it picks up on the changes you've made. If you do not want to refresh the started task immediately, specify "N" and the parameters will take effect the next time that the started task is active.

User response:

Type a "Y" or an "N" (or leave blank) to indicate whether to refresh the parameters in the parmlib member immediately.

CUZ133E **Invalid Value. Enter a "Y" to start History Processing in the Cloud Connector Started Task. Enter an "N" to wait until the next History Interval to start History Processing.**

Explanation

If you want to process the history files immediately, specify a "Y". If you want to wait until the next scheduled interval, specify "N".

To process the history files, the started task must be active and no other LPAR can be performing a backup of the history files. For example, if you are on LPAR A2Z01, but LPAR B2Y02 is currently backing up the history files, you will not be able start the history processing on A2Z01 until B2Y02 is done.

User response:

Specify a "Y" or an "N".

CUZ135E **Write SMF Record Type is invalid. Must be a number between 128 and 1151.**

Explanation:

This value will cause Cloud Tape Connector to write SMF records for every action that will be copied to the cloud, deleted from the cloud, and restored from the cloud. Enter "0" to turn off SMF processing.

User response:

Correct the number used as it is not in the acceptable range.

CUZ136E **Time-out must be a numeric value between 0 and 999. This number is the maximum time in seconds that socket is allowed to be in control after read call.**

Explanation

The "Time-out of socket" parameter changes the maximum amount of time (in seconds) that the sockets are allowed to be in control during read (BPX1RED) calls. Valid values are 0 - 999. Zero means that no time-out restrictions are applied. The default is 60 seconds.

User response:

Specify a value between 0 - 999. If you specify a value of 0, no time-outs will be applied for read calls.

CUZ138E **Invalid Value. Proxy Server requires a Server Address be a 1 to 128 byte alphanumeric string. Please enter a valid Server Address**

Explanation:

The server address can range from 1 - 128 characters (letters and numbers). You may have used a symbol in the server address that is not allowed.

User response:

Verify the server address and try again.

CUZ139E **Option Not Allowed. Cloud Connector allows only 1 Proxy Server definition and a Proxy Server has already been defined.**

Explanation:

Only one Proxy Server can be defined and you are attempting to add another one.

User response:

Either use the proxy server that is already defined or revise the definition to point to a different proxy server.

CUZ140E **Invalid Value. Enter a "Y" to use a Proxy Server or "N" to bypass Proxy Server support.**

Explanation:

You entered a value that is not "Y" or "N". Only those values can be specified for this field.

User response:

Change the value in the "Proxy Server" field to Y or N. The default is N. If you haven't defined a proxy server yet, you must do so before you can specify a "Y" in this field. See ["Specifying a Proxy Server cloud"](#) on page 45 for information on defining a proxy server for use with Cloud Tape Connector.

CUZ141E **Invalid Value. A Proxy Server has not been defined yet. Create a Proxy with line command "C" and select option 6. Define a Proxy Server prior to setting this value to "Y".**

Explanation:

While the "Proxy Server" field does take the values Y and N, the "Y" is only valid if a proxy server has already been defined.

User response:

Follow the steps in the message to define a proxy server first, then you can change the value in the "Proxy Server" field to Y.

CUZ142E **Option not allowed. You are attempting to delete a Proxy Server when Cloud *cloudname* has option Use Proxy Server set to "Yes". Turn this option off before deleting the Proxy Server.**

Explanation:

You cannot delete a proxy server when there is a "Y" in the "Use Proxy Server" field for this cloud.

User response:

You must change the value in the "Use Proxy Server" field to "N" to turn off this feature/proxy. If you want to change the definition of the proxy server to point to a different server, you can do that by specifying the "E" (edit) line command in the CMD column on the Cloud Server Display screen and then choosing the Proxy Server.

CUZ143E	Option not allowed. This Dataset must connect to a cloud definition other than a PROXY server.	A VSAM READ error occurred while attempting to access the data set specified for the DB2 Control Data Set. The return code is provided for diagnostic purposes.
Explanation:	You are attempting to connect to the Proxy Server, but this is not the type of cloud you need for this dataset.	
User response:	Select a different type of cloud definition. Choose the one that matches the type of cloud available at your site.	
CUZ144E	Invalid Line Command - Enter a "C" to Create a new Cloud Server, "D" to Delete the Cloud Server or "E" to Edit the Cloud Server	
Explanation:	The only line commands that are accepted on this screen are C, D, and E. You typed a different letter so it was considered invalid.	
User response:	Type either C, D, or E as a line command.	
CUZ902E	A DB2 subsystem ID has to be entered for processing.	
Explanation:	This process requires a DB2 subsystem ID.	
User response:	Type a one- to four-character DB2 subsystem ID to enable processing.	
CUZ903E	The default GDG base dataset name could not be located.	
Explanation:	The default name for the Generation Data Group (GDG) base data set could not be found.	
User response:	Check the spelling of the base data set name for the default generation data group (GDG) or create a new data set name.	
CUZ904E	The specified dataset could not be opened for I/O.	
Explanation:	A VSAM open error occurred while attempting to open the Control Data Set.	
User response:	Verify that the data set is accessible. Also verify the data set name.	
CUZ905E	An unexpected return code from VSAM was encountered while doing a read of the control file. RC=return code	
Explanation:		
		A VSAM READ error occurred while attempting to access the data set specified for the DB2 Control Data Set. The return code is provided for diagnostic purposes.
		User response: No response required
CUZ906I	The control file record for DB2 subsystem <i>sysname</i> has been successfully updated.	
Explanation:	This is an informational message only.	
User response:	No action is required.	
CUZ907E	An unexpected return code from VSAM was encountered while doing an update operation of the control file. RC1=return code	
Explanation:	A VSAM update error occurred while attempting to update the data set specified for the DB2 Control Data Set. The return code is provided for diagnostic purposes.	
User response:	No response is required.	
CUZ908I	The control file record for DB2 subsystem <i>sysname</i> has been successfully added.	
Explanation:	This is an informational message only.	
User response:	No action is required.	
CUZ909E	Invalid value. Valid options are 1 and 2.	
Explanation:	The character or number you typed is not acceptable. The only values that are acceptable are 1 and 2.	
User response:	Type 1 or 2 to select the option you want.	
CUZ910E	An unexpected return code from VSAM was encountered while doing an add operation to the control file. RC1=return code	
Explanation:	A VSAM error occurred while attempting to perform an add operation to the Control Data Set.	
User response:	Review the return code to determine the type of error that has occurred.	
CUZ911E	No Find string specified.	

Explanation:

The FIND command requires specification of a target string.

User response:

Type one or more characters that you want to locate using the FIND command.

CUZ912E Find string not found

Explanation:

The requested find string was not found. If this is the string you were searching for, then it does not exist.

User response:

Adjust the number of characters in the string you originally typed and try the search again or type a completely different string.

CUZ913I The control file record has been successfully updated.

Explanation:

This message indicates that the update process was completed.

User response:

No response is required.

CUZ914E An unknown column was specified using the SORT command.

Explanation:

When using the SORT command, you must specify a column name that you want to use as the basis for sorting the table. The column you specified with the SORT command is not known.

User response:

Verify that you correctly typed the name of the column or select another column.

CUZ915E SORT is not supported for the specified column.

Explanation:

The column you attempted to SORT is not supported as a column on which to sort.

User response:

Refer to the sort columns listed on the Define Sort Columns panel for a list of valid columns on which the sort can be based and redefine the sort.

CUZ916E Sort column not entered. Column name or number must be specified.

Explanation:

A column was not specified with the SORT. A column name or number must be specified for the SORT command.

User response:

If the column name is used, ensure that all spaces in the name are replaced with an underscore.

CUZ917E Put an ending quote at the end of the string.

Explanation:

You must place a quotation mark at the end of the string.

User response:

Type a quotation mark at the end of the string.

CUZ918I CHARS string not found. Press PF5 to continue from top.

Explanation:

The indicated character string was not found.

User response:

To continue searching for the character string from the top of the dialog, press PF5.

CUZ919I CHARS string not found. Press PF5 to continue from bottom.

Explanation:

The indicated character string was not found.

User response:

To continue searching for the character string from the bottom of the dialog, press PF5.

CUZ920E File tailoring open returned a file tailoring already in progress condition

Explanation:

An attempt to perform file tailoring for utility customization failed. There was a file tailoring session already in progress. File tailoring sessions cannot be performed concurrently.

User response:

CUZ921E File tailoring open returned the output file already in use condition -- ENQ failed

Explanation:

An attempt to open the Control Data Set failed with an ENQ error. The data set is already open for output.

User response:

Verify that you are the only user attempting to access this file.

CUZ922E File tailoring open returned the skeletal file or output file not allocated condition

Explanation:

An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:

Verify that all required files are allocated prior to performing file tailoring.

CUZ923E	File tailoring open returned a severe error condition
Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on open.	
User response: Verify that all required files are allocated and accessible prior to performing file tailoring.	
CUZ924E	File tailoring open returned an unknown code -- severe error
Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on open.	
User response: Verify that all required files are allocated and accessible prior to performing file tailoring.	
CUZ925E	File tailoring close returned a file not open condition -- severe error
Explanation: An attempt to perform file tailoring failed because a File-Not-Open condition was encountered on close.	
User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.	
CUZ926E	File tailoring close returned an output file in use condition
Explanation: An attempt to perform file tailoring failed because an Output-File-In-Use condition was encountered on close.	
User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.	
CUZ927E	File tailoring close returned a skeletal file or output file not allocated condition
Explanation: An attempt to close file tailoring failed because either a tailoring skeleton file or output file was not allocated.	
User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.	
CUZ928E	File tailoring close returned a severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CUZ929E	File tailoring close returned an unknown code -- severe error
Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on close.	
User response: Verify that all required files are allocated and accessible prior to performing file tailoring.	
CUZ930E	File tailoring close returned an output member exists in the output library and NOREPL was specified
Explanation: An attempt to perform file tailoring failed because the close process could not replace the pre-existing tailored member in the output file.	
User response: Change the output member name to a new name or ensure that the output library allows for member replacement.	
CUZ931E	File tailoring include returned a skeleton does not exist condition
Explanation: An attempt to perform file tailoring failed because the tailoring process could not locate a required tailoring skeleton.	
User response: Assure that all required files are allocated to perform file tailoring.	
CUZ932E	File tailoring include returned a skeleton in use -- ENQ failed condition
Explanation: An attempt to access a tailoring skeleton failed with an ENQ error (member-in-use).	
User response: Verify that all required tailoring files are allocated and that there are no other tailoring sessions running concurrently.	
CUZ933E	File tailoring include returned a data truncation or skeleton library or output file not allocated condition

Explanation:

An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:

Verify that all required files are allocated prior to performing file tailoring.

CUZ934E File tailoring include returned a severe error condition

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CUZ935E File tailoring include returned an unknown condition -- severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CUZ936E Allocation error - The ISPFDD DD is already allocated and cannot be deallocated - Process not completed

Explanation:

The ISPFDD allocation failed. The DD is already allocated and cannot be deallocated for this TSO session. The process did not complete successfully.

User response:

CUZ937E Allocation Error - An error was encountered allocating the ISPWRK1 or ISPWRK2 DD - Process not completed

Explanation:

The ISPWRK1 or ISPWRK2 DD allocation failed. The process did not complete successfully.

User response:

Verify TSO session parameters are set correctly for your site prior to allocation of these DD statements.

CUZ938E Field Required - The data set entered is a partitioned data set and the member name is required

Explanation:

A required field was not specified. The data set entered is a PDS (partitioned data set) and a member in this PDS must be referenced

User response:

Enter a valid member name for PDS access.

CUZ939E The only valid values are "T" for tracks and "C" for cylinders

Explanation:

You specified an invalid value. The only valid values are "T" for tracks and "C" for cylinders

User response:

Specify a valid value.

CUZ940E The specified data set could not be found in the MVS catalog.

Explanation:

The specified data set could not be found in the MVS catalog.

User response:

Ensure that the data set name is correct.

CUZ941I The RFIND key works only after a FIND character string is entered.

Explanation:

A repeat FIND (RFIND) was issued before a FIND command was issued. You must issue FIND before RFIND will work.

User response:

Issue FIND prior to attempting to issue RFIND.

CUZ942E Invalid Sort number. Enter a valid digit.

Explanation:

An invalid character was entered in the Srt column. Valid characters are the digits 1, 2, 3,... up to 9, or the number of sortable columns, whichever is less.

User response:

Specify a valid sort number.

CUZ943E Same Sort number entered twice

System action:

The same sort number was entered for more than one column. The screen is positioned to the second instance. Sort sequence numbers must be unique.

User response:

Specify a valid sort number.

CUZ944E Sort sequence skips a number.

Explanation:

The selected sorting sequence skips a number. This is not allowed. The screen is positioned to a selection whose number is lacking an immediate predecessor. The sort sequence is completely rebuilt from the Cmd (and Dir) information. Any previously existing sort sequence is entirely replaced. It is not added to or extended by the new entries.

User response:

Please specify a valid sort sequence that does not skip a number.

CUZ945E Invalid Dir entered. Must be A or D (ascending/descending).

Explanation:

The selected sorting direction is invalid. Only A (ascending) or D (descending) can be specified. A blank indicates ascending (default).

User response:

Specify a valid sorting direction.

CUZ946E Dir not valid without Ord.

Explanation:

A sorting direction was selected for a column that was not selected to be sorted. Sorting direction is only a valid choice for selected columns.

User response:

Select a sorting direction and order.

CUZ947E Max Sort Columns exceeded. Sorting first 10 columns.

Explanation:

More columns were selected for sorting than are supported. Nine columns can be selected. Under certain circumstances the limit is less than nine, due to internal constraints. For example, sorting a date field can be implemented by three sorts of partial column fields. In that case, the column would count as three toward the maximum of nine, not one.

User response:

Specify the appropriate allowable maximum number of sort columns.

CUZ948E Fix Columns cannot exceed screen size.

Explanation:

More columns were selected to be fixed than will fit on the screen.

User response:

Remove the (F) selection character from one or more columns.

CUZ950E Invalid selection character. "F" and "U" are valid.

Explanation:

An invalid Cmd character was entered. Valid characters are F (fix) and U (unfix). Fix causes the column to move to the fixed area on the left side of the screen. Fixed columns do not scroll horizontally when LEFT or RIGHT scrolling commands are issued. Unfix moves the column out of the fixed area, and allows it to scroll horizontally when LEFT and RIGHT scroll commands are issued.

User response:

Either remove the invalid character or enter a valid one.

CUZ951E Invalid entry. Must be numeric.

Explanation:

An invalid Cmd value was entered. Cmd values must be numeric. If the column is fixed, the number must be in the fixed range. If the column is not fixed, the number must be in the unfixed range.

User response:

Either remove the invalid number or enter a valid one.

CUZ952E Invalid entry for fixed column.

Explanation:

An invalid Cmd value was entered for a fixed column. Valid selections for fixed column are up to the number of fixed columns.

User response:

Either remove the invalid number or enter a valid one.

CUZ953E Invalid entry for unfixed column.

Explanation:

An invalid Cmd value was entered for an unfixed column. The number must be less than the number of columns, and greater than the number of fixed columns.

User response:

Either remove the invalid number or enter a valid one.

CUZ954E Invalid value entered for column size: non-numeric data.

Explanation:

An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields.

User response:

Either remove the invalid number or enter a valid one.

CUZ955E Invalid value entered for column size: out of range.

Explanation:

An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields. MIN is the smallest acceptable value. MAX is the largest acceptable value.

User response:

Either remove the invalid number or enter a valid one.

CUZ956E Total fixed column sizes cannot exceed screen size.

Explanation:

The Cmd values entered would result in the sum of the fixed column sizes to exceed the screen size. This is not allowed. The fixed columns are those with an or in the Fix column. Fixed columns are always displayed, and so must fit on the screen.

User response:

Either change the fixed column sizes so that the total is less than the screen size or cancel to return to the previous panel.

CUZ957E New configuration makes this column size invalid.

Explanation:

The requested column sizes make at least one unfixed column unable to be displayed. The cursor is positioned on the value where the problem was detected. The unfixed area on the screen would be too small to show the column where the cursor is placed.

User response

Do one of the following:

- Make the column where the cursor is smaller so that it can fit in the available unfixed area.
- Set it to its maximum size (width).
- Make the fixed area smaller.
- Cancel to return to the previous panel.

CUZ958E Column does not fit in unfixed area in new configuration.

Explanation:

The requested column sizes would make the unfixed column where the cursor is positioned undisplayable. The unfixed area on the screen would be too small to show this column.

User response:

Shrink the fixed area by either unfixing columns or making fixed columns smaller. The column where the cursor is cannot be partially displayed (min-max) so its size cannot be changed.

CUZ959E New configuration makes this column size invalid.

Explanation:

Fixing the requested columns would shrink the available area for unfixed columns unacceptably. One or more unfixed columns would not fit in the remaining unfixed area of the screen. The cursor is placed on a row that represents one such column. Therefore, the requested configuration is not allowed.

User response:

To change column sizes, cancel out of the CFIX function and invoke the CSIZE function. Either cancel to exit CFIX with no change or blank out one or more FIX selections until an allowable fixed size is reached.

CUZ960E Invalid fixed selections. Would not leave enough space for this column.

Explanation:

Fixing the columns requested would make at least one unfixed column undisplayable. The cursor is positioned on the row that represents one such unfixed column, whose minimum displayable size would not fit in the available screen area.

User response

Shrink the requested fixed area by either:

- Requesting fewer fixed columns.
- Unfixing one or more fixed columns.
- Cancel out of CFIX and invoke CSIZE in order to shrink one or more fixed columns enough so that all unfixed columns have the space they require.

CUZ962E Duplicate Cmd values entered.

Explanation:

Duplicate Cmd numbers were entered. The cursor points to the second instance of a Cmd value.

User response:

Either change this value, clear it, or exit the CORDER function.

CUZ963E Cursor not on data element.

Explanation:

CEXPAND was issued and the cursor was not located on a valid (expandable) area. CEXPAND requires the cursor to be positioned on a data element (non-heading area) in the dynamic area of the display. Or CEXPAND can be issued specifying the row and column of the data element to expand.

User response:

Ensure the cursor is located on a valid (expandable) area prior to issuing the CEXPAND command.

CUZ964E Invalid scroll amount for CRIGHT. Must be numeric.

Explanation:

Invalid (non-numeric) parameter to CRIGHT specified. CRIGHT accepts one numeric parameter: the number of columns to scroll right. If no parameter is entered a value of 1 is assumed.

User response:

Specify a numeric parameter to the CRIGHT command.

CUZ965E Invalid scroll amount for CLEFT. Must be numeric.

Explanation:

Invalid (non-numeric) parameter to CLEFT specified. CLEFT accepts one numeric parameter: the number of columns to scroll left. If no parameter is entered, a value of 1 is assumed.

User response:

Specify a numeric parameter to the CLEFT command.

CUZ966E Invalid parameter to ICRIGHT; must be numeric.

Explanation:

A parameter to ICRIGHT is not numeric. ICRIGHT (inner column scroll right) accepts either zero, one, or two numeric parameters. ICRIGHT can be abbreviated as ICR.

User response:

Specify a valid, numeric parameter for ICRIGHT.

CUZ967E Parameter to ICRIGHT too long. Invalid.

Explanation:

A parameter to ICRIGHT is too long. ICRIGHT does not process more than eight digits in a parameter, which is more than double any reasonable value.

User response:

Specify a valid parameter for ICRIGHT.

CUZ968E Parameter to ICRIGHT is zero. Invalid.

Explanation:

A parameter to ICRIGHT has the value zero. This is not supported.

User response:

Specify non-zero parameters to ICRIGHT.

CUZ969E ICRIGHT: unspecified column.

Explanation:

ICRIGHT was invoked with no parameters and the cursor is not positioned in the dynamic panel area.

User response:

Either put the cursor in the column that should be scrolled or specify the column by number. Column numbers can refer to visible columns (in the current display window) only. Number starts at 1, on the left side.

CUZ971E ICRIGHT: Column number specified is too big.

Explanation:

A column number parameter to ICRIGHT must be between 1 and the number of columns currently on the display screen.

User response:

To refer to a column by number you must first position the display window so that the desired column is visible.

CUZ972E Invalid parameter to ICLEFT; must be numeric.

Explanation:

A parameter to ICLEFT is not numeric. ICLEFT (inner column scroll left) accepts either zero, one, or two

numeric parameters. ICLEFT can be abbreviated as ICL.

User response:

Specify a valid parameter for ICLEFT.

CUZ973E Parameter to ICLEFT too long. Invalid.

Explanation:

A parameter to ICLEFT is too long. ICLEFT does not process more than eight digits in a parameter which is more than double reasonable value.

User response:

Specify a parameter less than or equal to eight digits for ICLEFT.

CUZ974E Parameter to ICLEFT is zero. Invalid.

Explanation:

A parameter to ICLEFT has the value zero. This is not supported.

User response:

Specify a non-zero number for ICLEFT.

CUZ975E ICLEFT: unspecified column.

Explanation:

ICLEFT was invoked with no parameters and the cursor is not positioned in the dynamic panel area.

User response:

Either put the cursor in the column that should be scrolled or specify the column by number. Column numbers can refer to visible columns (in the current display window) only. Numbering starts at 1 on the left side.

CUZ976E Column selected not sortable. Sort selection list presented.

Explanation:

You cannot perform a SORT on the column you selected. Valid sort columns are displayed in the sort selection list.

User response:

Sort on one of the valid columns displayed in the selection list.

CUZ977E ICLEFT: Column number specified is too big.

Explanation:

A column number parameter to ICLEFT must be between 1 and the number of columns currently on the display screen.

User response:

To refer to a column by number, you must first position the display window so that the desired column is visible.

CUZ978E Invalid column number specified for SORT (not numeric).

Explanation:

Invalid column number parameter to CSORT specified (non-numeric).

User response:

Specify a column number parameter to CSORT that is between 1 and the number of columns currently on the display screen. This can be followed by a direction value A or D (ascending/descending).

CUZ979E Invalid column number specified. Too many digits.

Explanation:

Invalid parameter to CSORT specified. More than eight digits were specified. Parsing stops at eight digits.

User response:

Specify a column number parameter between 1 and the number of columns currently on the display screen. This can be followed by a direction value A or D (ascending/descending).

CUZ980E Invalid column number specified: zero.

Explanation:

Invalid parameter to CSORT was specified (zero).

User response:

Specify a column number parameter to CSORT that is between 1 and the number of columns currently on the display screen. This can be followed by a direction value A or D (ascending/descending).

CUZ981E Invalid column number specified: out of range.

Explanation:

Invalid parameter to CSORT was specified.

User response:

Specify a column number parameter to CSORT that is between 1 and the number of columns currently on the display screen. This can be followed by a direction value A or D (ascending/descending).

CUZ982E Invalid view. View adjusted.

Explanation:

The current view was adjusted but not deleted. The saved view did not match the report requirements. This could be caused by the report changing or the view file getting corrupted.

User response:

The adjusted view will be used. You can issue CSET to modify the view.

CUZ983E Invalid view. View deleted.

Explanation:

Invalid data was found in a view for this report. The view was deleted and contents ignored. This could be caused by the report changing or the view file getting corrupted.

User response:

You can issue CSET to create a view that will match current report.

CUZ984E Unexpected return code from TBSTATS: rc

Explanation:

An unexpected failure issuing TBSTATS was received.

User response:

Refer to *ISPF Services Guide* (SC34-4819-03) for (hex) return code descriptions. Also, review the ISPTLIB and ISPTABL allocations. For information about ISPTLIB and ISPBABL see the ISPF manuals.

CUZ985E View Library not allocated.

Explanation:

A view input library has not been allocated. In order for a user to save and use report customizations that are created via the CSET command, ISPTABL and ISPTLIB must be allocated.

User response:

Refer to *ISPF Services Guide* (SC34-4819) for information on ISPTLIB and ISPTABL.

CUZ986E TBCREATE failed. RC=rc

Explanation:

TBCREATE was issued to create a view. It failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ987E TBOPEN failed. RC=rc

Explanation:

TBOPEN was issued to open a view. It failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ988E TBGET failed. RC=rc

Explanation:

A TBGET produced a return code (as indicated in the message).

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to

ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ989E TBMOD failed. RC=rc

Explanation:

A TBMOD produced an error and return code (as indicated in the message).

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ990E TBCLOSE failed. RC=rc

Explanation:

TBCLOSE failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ991E TBDELETE failed. RC=rc

Explanation:

TBDELETE failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ992E Invalid selection.

Explanation:

A command that is not supported on this panel was selected.

User response:

Issue a valid command for the panel.

CUZ993E Permanent view not supported

Explanation:

Something is preventing views from being saved. The permanent view flag cannot be set to Y. The most likely cause of this is that either ISPTLIB or ISPTABL (or both) have not been properly allocated.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to *ISPF Services Guide* (SC34-4819).

CUZ994E Invalid row number.

Explanation:

CEXPAND was issued with an invalid parameter of zero. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a valid parameter count for use with CEXPAND.

CUZ995E Invalid column number.

Explanation:

CEXPAND was issued with an invalid parameter of zero. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a valid parameter count for use with CEXPAND.

CUZ996E Invalid digits

Explanation:

CEXPAND was issued with an invalid parameter of zero. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a valid parameter count for use with CEXPAND.

CUZ997E Too many digits.

Explanation:

CEXPAND was issued with an invalid parameter of zero. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a valid parameter count for use with CEXPAND.

CUZ998E Zero parameter invalid.

Explanation:

CEXPAND was issued with an invalid parameter of zero. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a non-zero parameter.

CUZ999E Invalid parameter count: must be either two or zero parms.

Explanation:

CEXPAND was issued with an invalid number of parameters. CEXPAND can be issued with no parameters and the cursor on a data field, or with two parameters. The two parameters are the row number, followed by the column number of the data element to be expanded. The row number is counted down from the top, starting with the first scrollable row (heading not counted) The column number is counted from left to right, starting with the left column in the current display window.

User response:

Specify a valid parameter count for use with CEXPAND.

Started Task Messages for Cloud Tape Connector

This section lists all of the messages that may display if an error occurs while initiating the started task for Cloud Tape Connector to launch this product. The started task messages are listed below.

CUZ0001E CUZ Started Task is not APF Authorized

Explanation:

The started task for this product has not been APF authorized so it is not allowed to continue.

User response:

Add CUZ to the list of products that are APF authorized.

CUZ0002E Invalid Parameter Specified - MBR=

Explanation:

The member name (MBR) specified in the parameter is not valid.

User response:

Correct the member name and retry. Also verify that this parameter uses a member name.

CUZ0003E Cloud Connector Initialization Failed

Explanation:

This message informs you that there was a problem with the initialization process when starting .

User response:

Additional messages will be issued to assist in locating and correcting the error. Correct the error and submit the job again.

CUZ0004I CUZ Started Task Terminated

Explanation:

The started task will terminate after an APF Authorization test failure.

User response:

Look for additional messages for more information.

CUZ0005E STEPLIB DDNAME REQUIRED

Explanation:

The Cloud Connector started task requires a STEPLIB.

User response:

Add the product load library to the STEPLIB. Ensure that the STEPLIB is APF authorized.

CUZ0006E UNABLE TO OBTAIN SUBPOOL 241 STORAGE

Explanation:

The Cloud Connector started task will permanently load some modules in CSA subpool 241. There might be a CSA shortage.

User response:

Check CSA usage for shortages.

CUZ0007I Cloud Connector for z/OS Termination in Progress

Explanation:

Cloud Connector is shutting down.

User response:

No action is required.

CUZ0008E STARTED TASK INIT FAILURE: MISSING MODULE *module name*

Explanation:

The Cloud Connector started task was attempting to load the module listed in the message, but it was missing.

User response:

.

CUZ0009I Cloud Connector for z/OS Initialization Started.

Explanation:**User response:**

CUZ0010E **Cloud Connector for z/OS Already Active**

Explanation:

The product is already running so there is no need to start it again.

User response:

CUZ0012I **Cloud Connector for z/OS Termination is complete**

Explanation:**User response:**

CUZ0013I **INSTALL ***** EP: *******

Explanation:

This message is for your information only and is part of a series of messages listing modules that are being installed and the address entry point (EP).

User response:

CUZ0014I **INSTALL ***** EP: *******

Explanation:

This message is for your information only and is part of a series of messages listing modules that are being installed and the address entry point (EP).

User response:

CUZ0015I **INSTALL DLPA ***** LP:
***** EP: ***** L:

Explanation:

This message is for your information only and is part of a series of messages listing modules that are being installed and the address entry point (EP).

User response:

CUZ0016E **INSTALL DLPA ***** LP:
***** EP: ***** L:
***** Dynamic LPA ADD
failed, RC(**), RS(*****)**

Explanation:

This message is issued when a call to the dynamic LPA service routine fails. The return and reason code are included in the message text.

User response:

CUZ0017E **Need Update Auth on RACF Facility Class CSVDYLPA.ADD.****

Explanation:

The product cannot ADD to CSVDYLPA because it does not have UPDATE authority in RACF.

User response:

Change the RACF authority to UPDATE to allow adding modules with CSVDYLPA.ADD.**.

CUZ0018W **module name - No Filter Criteria was found in Parmlib**

Explanation:

This message is issued by module *module name*. The Parmlib member offers several different types of parameters for setting general options, staging options, defining cloud servers, and backup and restore options. None of these options have been set in the Parmlib member, which means that is not able to perform any functions.

User response:

Use Option 1 on the Cloud Connector Main Menu to specify settings for the Parmlib options.

CUZ0019W **module name - SVC Hooks are disabled**

Explanation:

This message is issued by module *module name*. It indicates that the hooks for the supervisor call (SVC) are not available or have been removed from use.

User response:

CUZ0020I **module name - Terminating all tasks**

Explanation:

This message is issued by module *module name*. It displays when is shutting down all of the tasks that are running. This message is accompanied by message CUZ0021I, which lists the tasks that is detaching.

User response:

CUZ0021I **module name - Detaching module**

Explanation:

This message is issued by module *module name*. It lists a *module* that is being detached so that all tasks can be terminated.

User response:

CUZ0022I **module name - Closing CUZ.OUT DD**

Explanation:

This message is issued by module *module name*.

User response:

CUZ0023I **module name - Termination Complete**

Explanation:

This message is issued by module *module name*. It informs you of the status of this process.

User response:

CUZ0024E ***module name* - Error Obtaining ASID Vector Table**

Explanation:

This message is issued by module *module name*. An address space identifier (ASID) vector table could not be accessed, causing an error.

User response:

CUZ0025E ***module name* - Open failed for CUZOUT**

Explanation:

This message is issued by module *module name*. The CUZOUT data set could not be opened.

User response:

CUZ0026I ***module name* - zIIP Processing Disabled**

Explanation:

This message is issued by module *module name*. The IBM z Systems Integrated Information Processor (zIIP) provides an option to help free-up general computing capacity in your processor. If your processor is zIIP-enabled, can take advantage of this feature. You can enable or disable zIIP processing. This message indicates that the zIIP processing has been disabled.

User response:

If you would like more information on how to enable and disable zIIP processing, refer to [“Managing zIIP processing”](#) on page 80.

CUZ0027I ***module name* - zIIP Processing Enabled**

Explanation:

This message is issued by module *module name*. The IBM z Systems Integrated Information Processor (zIIP) provides an option to help free-up general computing capacity in your processor. If your processor is zIIP-enabled, can take advantage of this feature. You can enable or disable zIIP processing. This message indicates that the zIIP processing has been enabled.

User response:

If you would like more information on how to enable and disable zIIP processing, refer to [“Managing zIIP processing”](#) on page 80.

CUZ0028E ***module name* - Parmlib errors prohibit cloud processing**

Explanation:

This message is issued by module *module name*. It indicates that there is a problem with settings that are specified in the Parmlib and the problem is preventing cloud processing.

User response:

You can modify Parmlib settings and verify that cloud servers have been defined for use by . Refer to [“Cloud Connector Settings \(Parmlib Options\)”](#) on page 23 for more information.

CUZ0029W ***module name* - DCB Nodes are active at shutdown waiting 15 Seconds**

Explanation:

This message is issued by module *module name*. It indicates that more time is needed before can shut down.

User response:

CUZ0030I ***module name* - Waiting for Stage to Cloud to Finish**

Explanation:

This message is issued by module *module name*. It informs you of the status of this process. The backup was staged to DASD and is now being copied to the cloud.

User response:

CUZ0031I ***module name* - Waiting for Repository Mnt to Finish**

Explanation:

This message is issued by module *module name*. It indicates that a tape for this repository is being mounted.

User response:

CUZ0032I ***module name* - Waiting for Message Print to Finish**

Explanation:

This message is issued by module *module name*. It indicates that a message or message file is being printed. Processing will continue when printing is done.

User response:

CUZ0086E ***mmmmmmmm* Delete failed: DCBS OBTAIN failed, U(*userid*)**

Explanation:

This message is issued by module *mmmmmmmm* to indicate it was unable obtain the DCBS control block. This control block is needed to instruct the cloud storage location to delete the backup data set that is being deleted from the repository. User *userid* initiated the request.

User response:

Retry the request. If the problem persists, contact IBM Technical Support.

CUZ0087E ***mmmmmmmm* Delete failed: Cloud node not found, U(*userid*)**

Explanation:

This message is issued by module *mmmmmmmm* to indicate it was unable to delete the backup data set because the cloud name listed as the backup's location cannot be found. User *userid* initiated the request.

User response:

Confirm the contents of your cloud definitions. Ensure your cloud definitions contain the cloud name listed on the backup data set repository record. Retry the request. Contact IBM Technical Support if the problem persists.

CUZ0088E **operation Failure, FDBK(*feedback*) , U(*userid*), type KEY(*key*)**

Explanation:

This message is issued to indicate that a VSAM error occurred. The *operation* will be PNT, GET, PUT, ERS, or UNK. The *feedback* is the RPLDFDBK field from the RPL. The *userid* indicates which user is making the request. The *type* indicates whether this is a profile or discrete entry. The *key* is the key of the record being processed when Cloud Tape Connector encountered the error.

User response:

Retry the request. Contact IBM Technical Support if the problem persists.

CUZ0090I **Entry deleted, U=*userid*, T=*type*, D=*dataset***

Explanation:

User *userid* successfully deleted backup data set *dataset* from the repository and the cloud. The *type* will be either DATA or PROF. A DATA entry defines a specific data set backup. A PROF entry is a profile of a group of specific data set discrete entries. The PROF entry is created by the Cloud Connect address space when the first backup is create and deleted with the last backup is deleted from the repository. Cloud Tape Connector

User response:

No action is required.

CUZ0092I **Restore successful, U=*userid*, D=*dataset*, CF=*cloudfile***

Explanation:

The Restore request did not encounter any errors. The request was initiated by User *userid* for Data Set *dataset* that resides in cloud file *cloudfile*.

User response:

No action is required.

CUZ0093E **Restore failed, RC=(*nn*), U=*userid*, DSN=*dataset***

Explanation:

This message indicates that the restore request initiated by User *userid* for backup data set *dataset* failed with return code (RC) *nn*.

User response:

Retry the request. An attempt to restore to DASD could encounter space problems.

CUZ0101I ***module name* - Stage to copy process completed successfully**

Explanation:

This message is issued by module *module name*. It indicates that the stage to copy processed successfully.

User response:

CUZ0102I ***module name* - Cloud Processing Bypassed**

Explanation:

This message is issued by module *module name*. It indicates that Cloud Connector bypassed the stage to copy process. This is typically the result of an attempt to process a data set with an unsupported data set organization (DSORG).

User response:

CUZ0103E ***module name* - Error Allocating Staging File**

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to allocate a file for staging data, which is copying a backup to DASD before moving it to the cloud.

User response:

Review the Staging Options in the Parmlib member to see if any changes need to be made, especially the size of the staging space. On the Main Menu, choose Option 1 (Parmlib Options), then Option 2, Staging Options.

CUZ0104E ***module name* - Error Allocating Staging File for Deletion**

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to delete a staging file after the backup file was copied to the cloud.

User response:

Review the Staging Options in the Parmlib member to see if any changes need to be made. On the Main Menu, choose Option 1 (Parmlib Options), then Option 2, Staging Options.

CUZ0105E ***module name* - Error Freeing Staging File**

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to free a file that was used to copy data to DASD, referred to as staging. After the staged data is moved to the cloud, the staging file can be freed for future use.

User response:

Review the Staging Options in the Parmlib member to see if any changes need to be made. On the Main Menu, choose Option 1 (Parmlib Options), then Option 2, Staging Options.

**CUZ0106E *module name* - Cloud Processor
Retry in progress - Setting Timer**

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to copy a staging file (backup) from DASD to the cloud. One or more attempts will be made to copy to the cloud, based on the value you specified in the "Error Retry Count" field. The amount of seconds between retry attempts is specified in the "Error Retry Interval Secs" field on the Staging Options parameter screen.

User response:

Review the Staging Options in the Parmlib member to see if any changes need to be made. On the Main Menu, choose Option 1 (Parmlib Options), then Option 2, Staging Options.

**CUZ0201E *module name* - ASID Table not
good**

Explanation:

This message is issued by module *module name*. It indicates that the ASID vector table ID was incorrect. This is an internal error indicative of a corrupted ASID vector table.

User response:

**CUZ0202E *module name* - DCBS Head Ptr is
zeroes**

Explanation:

The message is issued by module *module name*. It indicates that the head pointer in the DCBS contained all zeroes. This is an internal error indicative of a processing error.

User response:

**CUZ0203E *module name* - STC has been
cancelled-Staging Aborting**

Explanation:

The message is issued by module *module name*. It indicates that the started task was cancelled, so the process of staging data to DASD has also been cancelled.

User response:

**CUZ0204E *module name* - FORMAT-1 CCWs in
use**

Explanation:

The message is issued by module *module name*. It indicates that Format 1 CCWs are in use.

User response:

**CUZ0205E *module name* - EXCPVR CCWs in
use**

Explanation:

The message is issued by module *module name*. It indicates that EXCPVR CCWs are in use.

User response:

**CUZ0206E *module name* - non-zero RC from
CUZ#DATA**

Explanation:

The message is issued by module *module name*. It indicates that the return code generated from the CUZ#DATA module was greater than zero, which indicates an error.

User response:

**CUZ0207E *module name* - non-zero CLEANUP
RC**

Explanation:

The message is issued by module *module name*. The CUZ#DATA CLEANUP function completed with a return code greater than zero.

User response:

**CUZ0208E *module name* - Command Chain bit
is off**

Explanation:

This message is issued by module *module name*. It indicates that the CCW Command Chain bit is off.

User response:

**CUZ0209E *module name* - Channel Processor
Close in Progress**

Explanation:

This message is issued by module *module name*. It indicates that Channel Processor Close is in progress.

User response:

CUZ0301E Error retrieving WTDE

Explanation:

TBD

User response:

**CUZ0302E CUZ.CLOUD DD open failed, RC
(**)**

Explanation:

The cloud could not be accessed. A return code is provided to help determine the problem.

User response:

CUZ0303I **CUZ.#WTDR work-to-do task ready**

Explanation:

User response:

CUZ0304I **CUZ.#WTDR work-to-do task terminating**

Explanation:

User response:

CUZ0305E ***** failure, FDBK(*****),
****KEY(key info)**

Explanation:

TBD

User response:

CUZ0401E **module name - COMPARE
PROGRAM TERMINATED WITH
ERROR**

Explanation:

This message is issued by module *module name*. The CUZJCOMP utility program compares two data sets to see if they are the same. However, the job ended with an error instead of producing a report.

User response:

The CUZJCOMP job in the SCUZSAMP library must be modified for your environment prior to use. Review the instructions at the top of the CUZJCOMP job to make changes to the job before you submit it.

CUZ0402E **module name - TWO DATASETS
COMPARED WERE NOT EQUAL**

Explanation:

This message is issued by module *module name*. The CUZJCOMP job in the SCUZSAMP library is used to compare two data sets, such as a data set on tape and the same data set stored on the cloud. This comparison showed that the two data sets are not the same. When the CUZJCOMP job is run, it produces a "Dataset Compare Report" that describes the results of the comparison.

User response:

Review the "Dataset Compare Report" and the accompanying "Dataset Analysis Report" to determine which block contains the data that is different. See the "Tape Compare Utility" on page 105 for more information.

CUZ0403E **module name - MISMATCHED
BLOCK COUNT ERROR, TMC
COUNT=**

Explanation:

This messages is issued by module *module name*. The CUZJCOMP job in the SCUZSAMP library is used to compare two data sets, such as a data set on tape and the same data set stored on the cloud. This comparison showed that the two data sets do not have the same number of blocks. When the CUZJCOMP job is run, it produces a "Dataset Compare Report" that describes the results of the comparison.

User response:

Review the "Dataset Compare Report" and the accompanying "Dataset Analysis Report" to see if there is any additional information that will help you resolve the error. See the "Tape Compare Utility" on page 105 for more information.

CUZ0404E **module name - ABEND S237-04
SUPPRESSED BY DCB ABEND
EXIT**

User response:

This message was issued by module *module name*.

User response:

CUZ0405E **module name - ABEND S413-5C
SUPPRESSED BY DCB ABEND
EXIT**

Explanation:

This message is issued by module *module name*.

User response:

CUZ0406E **module name - ABEND S637-B4
SUPPRESSED BY DCB ABEND
EXIT**

Explanation:

This message is issued by module *module name*.

User response:

CUZ0407E **module name - ABEND S637-B8
SUPPRESSED BY DCB ABEND
EXIT**

Explanation:

This message is issued by module *module name*.

User response:

CUZ0408E **module name - VOLSER OF TAPE
WAS CHANGED DURING OPEN**

Explanation:

This message is issued by module *module name*.

User response:

CUZ0409I **module name - TAPE ANALYZE
USED SCOPE=VOLUME. IOS000I
MESSAGES OK**

Explanation:

This message was issued by module *module name*. It indicates the status of this process.

User response:

CUZ0410E ***module name* - OPEN OF DATASET FAILED**

Explanation:

This message is issued by module *module name*. When the CUZJCOMP job in the SCUZSAMP library tried to compare two data sets, one of the data sets could not be opened. Check the CUZJCOMP job to see if the data set names and directory paths were specified correctly.

User response:

If generated, review the "Dataset Compare Report" and the accompanying "Dataset Analysis Report" to see if there is any additional information that will help you resolve the error. See the ["Tape Compare Utility"](#) on page 105 for more information.

CUZ0411E ***module name* - DSN1 DATASET IS NOT ON A TAPE DEVICE**

Explanation:

This message is issued by module *module name*. When the CUZJCOMP job, located in the SCUZSAMP library, tried to compare two data sets, the first data set listed in the job could not be found on a tape drive. Check the CUZJCOMP job to see if the data set name and directory path were specified correctly.

User response:

If generated, review the "Dataset Compare Report" and the accompanying "Dataset Analysis Report" to see if there is any additional information that will help you resolve the error. See the ["Tape Compare Utility"](#) on page 105 for more information.

CUZ0412E ***module name* - DSN2 DATASET IS NOT ON A TAPE DEVICE**

Explanation:

This message is issued by module *module name*. When the CUZJCOMP job, located in the SCUZSAMP library, tried to compare two data sets, the second data set listed in the job could not be found on a tape drive. Check the CUZJCOMP job to see if the data set name and directory path were specified correctly.

User response:

If generated, review the "Dataset Compare Report" and the accompanying "Dataset Analysis Report" to see if there is any additional information that will help you resolve the error. See the ["Tape Compare Utility"](#) on page 105 for more information.

CUZ0501E **Parmlib processing had errors**

Explanation:

The sample library, SCUZSAMP, has a default parameter library (Parmlib) member, CUZ#PARM. To

make changes to the default settings in CUZ#PARM, choose Option 1, Cloud Connector Settings, on the Main Menu.

User response:

After modifying the CUZ#PARM member, try the job again.

CUZ0502E **INVALID OPERAND FOR COMMAND**

Explanation:

Several commands are provided to manage tasks and processing options. The operand specified for this command is either the wrong operand or has an incorrect value specified for the operand.

User response:

Refer to [Chapter 7, "Operational Considerations,"](#) on page 71 for a list of commands and operands.

CUZ0503E **Restore processing had errors.**

Explanation:

An attempt to copy a data set from the cloud to a tape device had processing errors. Several commands are provided to manage tasks and processing options, including the Restore process.

User response:

Refer to [Chapter 7, "Operational Considerations,"](#) on page 71 for a list of commands and operands, and for additional information on how to restore a data set.

CUZ0504E **INVALID COMMAND**

Explanation:

Several commands are provided to manage tasks and processing options. A command was not specified correctly or was not valid for this process.

User response:

Refer to [Chapter 7, "Operational Considerations,"](#) on page 71 for a list of commands and operands for various processes.

CUZ0505E **VERIFY COMMAND FAILED: INVALID SYNTAX**

Explanation:

The VERIFY command is used as part of the compare process to ensure that the data on the cloud matches the data on tape. In this case, the syntax for the VERIFY command was incorrect, perhaps listing the wrong data set to compare.

User response:

Review the VERIFY command in the job to ensure that the data set names are typed correctly.

CUZ0506E **RESET COMMAND FAILED: INVALID SYNTAX**

Explanation:

The RESET command is used as part of the compare process, which ensures that the data on the cloud matches the data on tape. In this case, the syntax for the RESET command was incorrect.

User response:

Review the RESET command in the job.

CUZ0507I RESET ALL COMMAND COMPLETE

Explanation:

User response:

CUZ0508I RESET SVC COMMAND COMPLETE

Explanation:

User response:

**CUZ0509I SVC DEBUG ACTIVE FOR
 JOBNAME *jobname***

Explanation:

This is a status message for the *jobname* listed in the message.

User response:

CUZ0510I zIIP Processing is now Enabled

Explanation:

The IBM z Systems Integrated Information Processor (zIIP) provides an option to help free-up general computing capacity in your processor. If your processor is zIIP-enabled, can take advantage of this feature. zIIP processing will be set based on the presence of the NOZIIP DD in the CUZCLOUD started task JCL procedure. If the DD is not present or is commented out, zIIP processing will be active by default.

User response:

For more information on how to enable or disable zIIP processing, refer to [“Managing zIIP processing”](#) on page 80. If you do not want to change the current setting, no action is required.

CUZ0511I zIIP Processing is now Disabled

Explanation:

The IBM z Systems Integrated Information Processor (zIIP) provides an option to help free-up general computing capacity in your processor. If your processor is zIIP-enabled, can take advantage of this feature. zIIP processing will be set based on the presence of the NOZIIP DD in the CUZCLOUD started task JCL procedure. If the DD is not present or is commented out, zIIP processing will be active by default.

User response:

For more information on how to enable or disable zIIP processing, refer to [“Managing zIIP processing”](#) on page 80. If you do not want to change the current setting, no action is required.

CUZ0512I SVC DEBUG DISABLED

Explanation:

User response:

CUZ0513E Invalid Data Set Name value

Explanation:

The data set name is not the correct length or contains characters that are not allowed.

User response:

Change the data set name to meet requirements.

CUZ0514E Invalid Cloud name value

Explanation:

A cloud name can be from 1 - 8 characters in length. Clouds are defined for use through the Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu, and then with Option 3, Cloud Servers.

User response:

Refer to [“Cloud Server Options”](#) on page 27 for more information.

CUZ0515E Invalid Generation value

Explanation:

When specifying which data set to restore, you can specify the most recent backup (generation 0) or a previously saved backup of the data set, referred to as a generation. You set the maximum number of generations you want to save to a cloud backup by specifying a value from 1 - 10 on the Parmlib General Options screen. When performing Restore processing, you can specify which generation you want to restore from the cloud, with 0 referring to the current version, 1 is the previous version, and so on. However, if a generation value is specified, but a matching prior backup version does not exist, you will receive an error.

User response:

Refer to [“General Options”](#) on page 24 and to [“Restore commands”](#) on page 84 for more information.

CUZ0516E Invalid Cloud list value

Explanation:

When restoring a repository, a Cloud List specifies a "fixed" file name for the backup copy of the repository. The file name currently specified is not acceptable.

User response:

Refer to [“Restoring a Repository”](#) on page 89 for more information on how to construct a Cloud List file name.

**CUZ0517E Invalid Cloud restore
 subparameter**

Explanation:

When restoring a cloud data set or a complete repository, you can use different commands and

parameters for that process. One of the parameters used was not acceptable.

User response:

Refer to [“Restoring a data set”](#) on page 72 and to [“Restoring a Repository”](#) on page 89 for more information on Restore commands and parameters.

CUZ0518I CUZ. DIAGNOSTIC DISPLAY:

Explanation:

This message is combined with other messages to provide diagnostic data.

User response:

CUZ0519I SDA ADDRESS *address*

Explanation:

User response:

CUZ0520E INVALID COMMAND SYNTAX

Explanation:

One of the commands in the job was typed incorrectly.

User response:

Review the commands in [Chapter 7, “Operational Considerations,”](#) on page 71 to determine the correct syntax.

CUZ0601E *module name* Abend Detected

Explanation:

This message is issued by module *module name*, and indicates that an abend occurred while using *module name*.

User response:

CUZ0701I *message text*

Explanation:

The text varies for this message, but is informational in nature and does not require a response.

User response:

**CUZ0702E *module name* - Cloud Connector
STC Not Active - Processing
bypassed**

Explanation:

This message is issued by module *module name*. It indicates that the product's started task is not running so processing is bypassed.

User response:

Start the started task for Cloud Connector.

**CUZ0703E *module name* - Non numeric
retention period set to 7**

Explanation:

This message is issued by module *module name*. The value specified for the Retention Period was not a number so the value was set to 7. Valid numbers for

the Retention Period are 0000 - 9999. A value of zero is helpful for testing as it enables you to run the expiration job and retest without having to wait days for the data sets to expire.

User response:

Type a number between 0000 and 9999 in the Retention Period field.

**CUZ0801E *module name* - STC is not active.
Cloud copy bypassed**

Explanation:

This message is issued by module *module name*. The started task for Cloud Connector has not been started yet so a copy process cannot be performed.

User response:

Start the started task and try the copy job again.

CUZ0803E *module name* - Parmlib DD Missing

Explanation:

This message is issued by module *module name*. An attempt to call the Parmlib member, CUZ#PARM, failed because the DD statement specifying the Parmlib was missing from the started task. The CUZ#INEX job, which backs up history data sets, was not run due to the missing Parmlib DD.

User response:

Correct the Parmlib DD statement in CUZCLOUD and try again.

**CUZ0804E *module name* - History Mbr
Missing-Processing Bypassed**

Explanation:

This message is issued by module *module name*. The History Member, normally CUZ#INEX, is needed for backing up history data sets. However, this member cannot be located so the job that backs up history data sets to the cloud could not be run.

User response:

Specify a History Member. Refer to [“General Options”](#) on page 24 and to [Chapter 8, “Backing up existing data sets,”](#) on page 97 for more information.

**CUZ0805E *module name* - CUZ#INEX DD
Missing-Initialization Failed**

Explanation:

This message is issued by module *module name*. It indicates that the DD statement for CUZ#INEX is missing so the initialization of the job failed.

User response:

Refer to [“General Options”](#) on page 24 and to [Chapter 8, “Backing up existing data sets,”](#) on page 97 for more information about the Include/Exclude History data set and member.

**CUZ0806E *module name* - Error Opening
CUZ#INEX DD**

Explanation:

This message is issued by module *module name*. It indicates that an error occurred while attempting to open the DD for CUZ#INEX, which is normally used for backing up history data sets.

User response:

Refer to “General Options” on page 24 and to Chapter 8, “Backing up existing data sets,” on page 97 for more information about the Include/Exclude History data set and member.

CUZ0807E *module name* - Member not found in CUZ#INEX DD

Explanation:

This message is issued by module *module name*. It indicates that a member name was not listed in the CUZ#INEX DD statement.

User response:

Refer to “General Options” on page 24 and to Chapter 8, “Backing up existing data sets,” on page 97 for more information about the Include/Exclude History data set and member.

CUZ0808E *module name* - Error allocating Parmlib Member

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to allocate CUZ#INEX.

User response:

Refer to “General Options” on page 24 and to Chapter 8, “Backing up existing data sets,” on page 97 for more information about the Include/Exclude History data set and member.

CUZ0809E *module name* - Error Obtaining Storage for DCBS Node

Explanation:

This message is issued by module *module name* to indicate that it was unable to obtain storage for a DCBS control block. The DCBS is a multifaceted block of storage used in many areas of . Processing for *module name* does not continue and the service requesting the DCBS fails.

User response:

If the problem persists, contact IBM Technical Support for assistance.

CUZ0810E *module name* - Error Acquiring Cloud API Storage

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to access storage on a cloud for history backup data sets.

User response:

If the problem persists, contact IBM Technical Support for assistance.

CUZ0901E *module name* - Parmlib DD Missing

Explanation:

This message is issued by module *module name*. It indicates that a DD statement is missing for the CUZ#PARM parameter library.

User response:

CUZ0902E *module name* - CUZ#PARM DD Missing - Initialization Failed

Explanation:

This message is issued by module *module name*. It indicates that a DD statement is missing for the CUZ#PARM parameter library, which caused the initialization to fail.

User response:

CUZ0903E *module name* - Error Opening CUZ#PARM DD

Explanation:

This message is issued by module *module name*. It indicates that while attempting to open the DD statement for CUZ#PARM, an error occurred. The DD statement may need to be revised.

User response:

Refer to “General Options” on page 24 for more information about this Parmlib member.

CUZ0904E *module name* - Member not found in CUZ#PARM DD

Explanation:

This message is issued by module *module name*. It indicates that a member name must be specified in the DD statement for CUZ#PARM. The DD statement may need to be revised.

User response:

Refer to “General Options” on page 24 for more information about the Parmlib member.

CUZ0905E *module name* - Error allocating Parmlib Member

Explanation:

This message is issued by module *module name*. It indicates that a problem occurred when attempting to allocate CUZ#PARM, which is the Parmlib member.

User response:

Refer to “General Options” on page 24 for more information about the Parmlib member.

CUZ1001E Cross memory initialization failure: Missing SDA

Explanation:

The message is issued by module CUZ#PCRI. It indicates that Cloud Connector failed to initialize its Cross Memory facility because it was unable to find the SDA control block. This is indicative of an internal error.

User response:
Retry initialization.

CUZ1101E ***module name* execution failure:
Missing or invalid SDA**

Explanation:
This message is issued by module *module name*. It indicates that Cross Memory module *module name* could not locate the SDA control block. This is indicative of an internal failure.

User response:
Restart the Cloud Connector address space.

CUZ1201E ***module name* - ASID Table not
good**

Explanation:
This message is issued by module *module name*. It indicates that the acronym for the ASID table control block was invalid. This is indicative of an internal error.

User response:

CUZ1202E ***module name* - DCBS Head Ptr is
zeroes**

Explanation:
This message is issued by module *module name*. It indicates that the DCBS Head pointer is zero. This is indicative on an internal error.

User response:

CUZ1203E ***module name* - STC has been
cancelled - Staging Aborting**

Explanation:
This message is issued by module *module name*. It indicates that the started task was cancelled so the process of staging data to DASD has also been cancelled.

User response:
Restart the started task if you did not intend to cancel it.

CUZ1204E ***module name* - FORMAT-1 CCWs in
use**

Explanation:
The message is issued by module *module name*. It indicates that detected the use of FORMAT-1 CCWs and recorded this in the DCBS internal control blocks.

User response:

CUZ1205E ***module name* - EXCPVR CCWs in
use**

Explanation:

The message is issued by module *module name*. It indicates that detected EXCPVR CCW usage and recorded this in the DCBS internal control blocks.

User response:

CUZ1206E ***module name* - non-zero CLEANUP
RC**

Explanation:

This message is issued by module *module name*. A return code greater than zero was returned from the CLEANUP process, indicating an error.

User response:

CUZ1207W ***module name* - Command Chain bit
is off**

Explanation:

This message is issued by module *module name*. It indicates that the Command Chain bit is off but the CCW is not the last one. This is an internal error possibly indicative of a premature end of the channel program.

User response:

CUZ1208I ***module name* - Channel Processor
Close in Progress**

Explanation:

This message is issued by module *module name*. It indicates that channel processor close is in progress for a data set processed by .

User response:

CUZ1209I ***module name* - DCBS node not
found**

Explanation:

Explanation: The message is issued by module *module name*. It indicates that there was no DCBS control block structure suggesting that the data set be processed by .

User response:

CUZ1501W ***module name* - Invalid Auto
Backup Minutes Parm-Setting to
60**

Explanation:

This message is issued by module *module name*. It indicates that the Auto Backup Repository Minutes field on the "Parmlib General Options" screen allows you to specify a value between 05 - 9999. This message indicates that a value outside that range was entered in the field and is not valid. The setting was changed to 60 minutes.

User response:

For more information, see [“General Options” on page 24.](#)

**CUZ1502E *module name* - Error Opening
SYSIN DD**

Explanation:

This message is issued by module *module name*. It indicates that while attempting to open the DD for SYSIN, an error occurred.

User response:

**CUZ1503I *module name* - Shutdown in
Progress**

Explanation:

The message is issued by module *module name*. It indicates that IBM Cloud Connector is shutting down.

User response:

**CUZ1504E *module name* - Invalid Return code
calling History Processor**

Explanation:

This message is issued by module *module name*. It indicates that when processing a backup history data set, a return code was generated that is not allowed for the CUZ#REPM module, possibly as a result from a call to the CUZ#INEX module.

User response:

Look for warning and error messages produced by module CUZ#INEX, which is called by CUZ#REPM during history processing. You may have included data sets with DSORG types that are not supported in the "include" list, such as VSAM or PDS file types. The DSORG for a copy to the cloud must be a sequential file.

**CUZ1505E *module name* - Error Obtaining
Staging DCB Buffer Storage**

Explanation:

This message is issued by module *module name*. It indicates that an error occurred while attempting to locate storage for the Data Control Block (DCB) buffer that is needed for the staging process, which copies data to DASD before sending the data to a cloud.

User response:

CUZ1601I Displaying record, *=*record info***

Explanation:**User response:**

CUZ1602I We are in the write_to_dsn routine

Explanation:**User response:**

CUZ1603I We are in the write_to_rep routine

Explanation:**User response:**

**CUZ1901E *module name* - Cloud Connector
Errors - Job Abending**

Explanation:

This message is issued by module *module name*. It is accompanied by message CUZ1902E, which has additional instructions on how to determine what error has caused the job to abend.

User response:

**CUZ1902E *module name* - Look in CUZOUT DD
in STC for Abend Reason**

Explanation:

This message is issued by module *module name*. It indicates that the explanation for why the job abended is listed in the CUZOUT DD in the started task.

User response:

**CUZ1903I *module name* - Open Detected for
DDNAME *ddname***

Explanation:

This message is issued by module *module name*. It provides the name of a DD that has been opened.

User response:

**CUZ2001I *module name* - Close Detected for
DDNAME**

Explanation:

This message is issued by module *module name*. It indicates that detected a Close for the DDNAME it is processing.

User response:

**CUZ2002I *module name* - Close Hook
Successfully Disabled**

Explanation:

The message is issued by module *module name*. It indicates that the Close intercept is disabled and will not perform any processing.

User response:

**CUZ2003E *module name* - Invoking z/OS
Repository had errors**

Explanation:

This message is issued by module *module name*. It indicates that when was attempting to call a z/OS Repository, multiple errors occurred.

User response:

Verify the Repository name and the directory path for the z/OS Repository. On the Main Menu, choose Option 1 (Parmlib), then choose Option 3 (Cloud Servers), to edit the Repository information currently specified for an existing cloud server definition.

CUZ2004I DCB Cloud Node Found

Explanation:

This message indicates that during Close processing, found a DCBS node for the data set, and therefore will process the data set.

User response:

CUZ2005I *module name* - Closing the DCB is complete RC=0

Explanation:

This message is issued by module *module name*. It indicates that no errors were encountered while closing the data control block, which is why the return code is 0.

User response:

CUZ2006E *module name* - Closing the DCB has completed with errors

Explanation:

This message is issued by module *module name*. It indicates that when attempted to close a data control block (DCB), multiple errors occurred.

User response:

Try to determine the reason for the errors.

CUZ2007I *module name* - Cloud Processing Ended with Success

Explanation:

The message is issued by module *module name*. It indicates that processing completed successfully. The selected data set has been copied to the cloud or staged for later processing.

User response:

CUZ2008I *module name* - Cloud Processing Ended with errors

Explanation:

This message is issued by module *module name*. It indicates that when attempted to save data to or retrieve data from a cloud, multiple errors occurred.

User response:

If you want to verify the Cloud name and the directory path, choose Option 1 (Parmlib) on the Main Menu, then choose option 3, Cloud Servers, to edit the cloud server definition or to create a new definition.

CUZ2009E *module name* - Cloud Process has Abended

Explanation:

This message is issued by module *module name*. It indicates that when attempted to save data to or retrieve data from a cloud, the process did not complete, resulting in an abend.

User response:

Verify the Cloud name and the directory path. On the Main Menu, choose Option 1 (Parmlib), then choose

Option 3 (Cloud Servers) to edit the cloud server definition or to create a new definition.

CUZ2010E *module name* - Cloud Process Ended via ECB Post

Explanation:

This message is issued by module *module name*. It indicates that when attempted to save data to or retrieve data from a cloud, the process ended due to an Event Control Block (ECB) post.

User response:

Verify the Cloud name and the directory path. On the Main Menu, choose Option 1 (Parmlib), then choose Option 3 (Cloud Servers) to edit the cloud server definition or to create a new definition.

CUZ2011I *module name* - Waiting for Cloud ECB to Post-30 secs

Explanation:

This message is issued by module *module name*. It indicates that is waiting for Cloud activity to complete.

User response:

CUZ2012E *module name* - Post Error Routing Executing

Explanation:

This message is issued by module *module name*. It indicates that a POST attempt encountered an error and is executing its Error Return Routine (ERRET). This is possibly the result of an internal error.

User response:

CUZ2013E *module name* - Cloud Connector Errors-Job Abending

Explanation:

This message is issued by module *module name*. It indicates that when attempted to save data to or retrieve data from a cloud, the job ended before the process was complete.

User response:

Verify the Cloud name and the directory path. On the Main Menu, choose Option 1 (Parmlib), then choose Option 3 (Cloud Servers) to edit the cloud server definition or to create a new definition.

CUZ2014E *module name* - Look in CUZOUT DD in STC for A+

Explanation:

This message is issued by module *module name*. Review the CUZOUT DD statement in the started task for more information.

User response:

CUZ2101E CUZSDA REQUEST FAILURE: UNKNOWN

Explanation:

This message indicates that the service responsible for managing the SDA control block received an unknown request. This is indicative of an internal error.

User response:

CUZ2102E CUZSDA CREATE FAILURE: APF AUTHORIZATION

Explanation:

This message indicates that the load modules do not have the necessary APF authorization. This is indicative of an incomplete installation and configuration. Review the configuration instructions and ensure the load module is APF authorized.

User response:

CUZ2103E CUZSDA CREATE FAILURE: STORAGE OBTAIN

Explanation:

TBD

User response:

CUZ2104E CUZSDA CREATE FAILURE: IEANTCR RC(***)**

Explanation:

TBD

User response:

CUZ2105E CUZSDA DELETE FAILURE: APF AUTHORIZATION

Explanation:

TBD

User response:

CUZ2106E CUZSDA DELETE FAILURE: IEANTDL RC(***)**

Explanation:

TBD

User response:

CUZ2201E CUZ.SVC22 - Cloud Connector Errors - Job Abending

Explanation:

This message is accompanied by message CUZ2202E, which has additional instructions on how to determine what error has caused the job to abend.

User response:

CUZ2202E *module name* - Look in CUZOUT DD in STC for Abend Reason

Explanation:

This message is issued by module *module name*. The explanation for why the job abended is listed in the CUZOUT DD in the started task.

User response:

CUZ2203I *module name* - Open Detected for DDNAME *ddname*

Explanation:

This message is issued by module *module name*. It provides the name of a DD that has been opened.

User response:

CUZ2401I *module name* - Shutdown in Progress

Explanation:

This message is issued by module *module name*. It indicates that the address space is shutting down.

User response:

CUZ2402I *module name* - Waiting for Initialization to complete

Explanation:

This message is issued by module *module name*. It indicates that this task is waiting for the address space initialization to complete.

User response:

CUZ2403E *module name* - Invalid Cloud End Status / ECB Post encountered

Explanation:

This message is issued by module *module name*. It indicates an internal processing error.

User response:

CUZ2404E *module name* - Error Releasing Storage for DCBS Definition

Explanation:

This message is issued by module *module name*. It indicates that encountered an error while releasing storage for a DCBS control block.

User response:

CUZ2405I *module name* - Shutdown in Progress

Explanation:

This message is issued by module *module name*. It indicates that the address space is being shutdown.

User response:

CUZ2501E SVC INSTALLATION ERROR: NOT APF AUTHORIZED

Explanation:

The product needs to be APF authorized before it can be installed.

User response:

CUZ2502E *module name* SVC PROCESSING: INVALID REQUEST

Explanation:

This message is issued by module *module name*. It indicates that the SVC intercept initialization encountered an internal error.

User response:

CUZ2503E SVC Installation Error: No Steplib

Explanation:

This message indicates that detected an internal error during SVC intercept installation. The address space does not contain the appropriate STEPLIB and is unable to find the necessary load modules.

User response:

CUZ2504E SVC Installation Error: Load

Explanation:

This message indicates that detected an internal error during SVC intercept installation.

User response:

CUZ2505E SVC INSTALLATION ERROR: SVC ROUTINE CSA AREA SIZE

Explanation:

This message indicates that detected an error while installing its SVC intercepts.

User response:

CUZ2506I SVC INSTALLATION: REPLACE DETECTED FOR *****

Explanation:**User response:**

CUZ2507E SVC INSTALLATION ERROR: CUZSDA VALIDATION

Explanation:

This message indicates that encountered an error while attempting to install its SVC intercepts.

User response:

CUZ2508E SVC INSTALLATION ERROR: LOAD CUZSVCHK

Explanation:

This message indicates that encountered an error while installing its SVC intercepts.

User response:

CUZ2509E SVC INSTALLATION ERROR: NO STEPLIB

Explanation:

This message indicates that encountered an error while installing its SVC intercepts.

User response:

CUZ2510E SVC INSTALLATION ERROR: ALCSVCHK SP241 STORAGE OBTAIN.

Explanation:

This message indicates that encountered an error while installing its SVC intercepts.

User response:

CUZ2511E SVC INSTALLATION ERROR: SVCUPDTE RC(return code)

Explanation:

This message indicates that encountered an error while installing its SVC intercepts. SVCUPDTE failed with RC value *return code*.

User response:

CUZ3001E *module name* - Invalid WTO Message Parm Passed

Explanation:

This message is issued by module *module name*. It indicates that a parameter that was passed in a write-to-operator message is not valid.

User response:

Correct the parameter name or value and try again.

CUZ3002E *module name* - No DCB or LCB passed.

Explanation:

This message is issued by module *module name*. It indicates that the message request did not include a DCB or LCB control block. This is indicative of an internal error.

User response:

CUZ3003E *module name* - Message Number:

Explanation:

This message is issued by module *module name*. It provides the message number of the message that failed to display as a result of the error reported in message CUZ3002E. It is displayed along with message CUZ3002E.

User response:

CUZ3101E *module name* - DCB Not Open- Messages print stopped

Explanation:

This message is issued by module *module name*. It indicates that when attempting to print messages, a Data Control Block could not be opened, which caused printing to stop.

User response:

Try to determine what prevented the DCB from being open.

CUZ3102I *module name* - STC Shutdown in progress

Explanation:

This message is issued by module *module name*. It indicates that a shutdown of the address space is in progress.

User response:

CUZ3201E **BLOCK READ EXCEEDS 65535 BYTES AND NOT LBI**

Explanation:

detected a block size that exceeded the maximum value for processing that is not using the Large Block Interface (LBI).

User response:

CUZ3202E **BLOCK READ EXCEEDS 256K BYTES USING LBI**

Explanation:

detected a block size that exceeded the maximum value for processing that is not using the Large Block Interface (LBI).

User response:

CUZ3203E **INVALID VALUE ON SCOPE PARAMETER**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3204E **INVALID VALUE ON ACTION PARAMETER**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3205E **INVALID VALUE ON MODE PARAMETER**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3206E **ACTION CONTROL CARD NOT SUPPLIED**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3207E **SCOPE CONTROL CARD NOT SUPPLIED**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3208E **MODE CONTROL CARD NOT SUPPLIED**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3209E **INVALID CONTROL CARD VERB**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3210E **INVALID VALUE ON PRINT PARAMETER**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3211E **INVALID VALUE ON PRINT-FORMAT PARM**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3212E **INVALID VALUE ON PRINT-BEGIN PARM**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3213E **INVALID VALUE ON PRINT-END PARAMETER**

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3214E **PRINT END NUMBER < PRINT BEGIN NUM**

Explanation:

Change the value for PRINT END to a number that is greater than the PRINT BEGIN number.

User response:

CUZ3215E PARM MUST BEGIN BEFORE COL. 60

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3216E SUB-PARM MUST BEGIN BEFORE COL. 66

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3217E PRINT WITH SCOPE=VOLUME REQUIRES BLP

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3218I RECORD PRINTING IS SUPPRESSED

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3219E COMPARE NOT VALID WITH SCOPE=VOLUME

Explanation:

This message describes a control card error. Correct the statement and try again.

User response:

CUZ3301E *module name* - Error Obtaining Working Storage

Explanation:

This message is issued by module *module name*. It indicates that while attempting to "stage" (copy) the data to DASD before placing it on a cloud, there was a problem in obtaining working storage.

User response:

Try increasing the amount of space that is allocated by adjusting the values on the Parmlib Staging Options screen. See ["Staging Options" on page 26](#) for more information.

CUZ3302E *module name* - Error Obtaining Work Buffer Area

Explanation:

This message is issued by module *module name*. It indicates that while attempting to "stage" (copy) the

data to DASD before placing it on a cloud, there was a problem in obtaining a work buffer area.

User response:

Try increasing the amount of space that is allocated by adjusting the values on the Parmlib Staging Options screen. See ["Staging Options" on page 26](#) for more information.

CUZ3303E *module name* - Error Obtaining DCB Buffer Area

Explanation:

This message is issued by module *module name*. It indicates that while attempting to "stage" (copy) the data to DASD before placing it on a cloud, there was a problem in obtaining a DCB buffer area.

User response:

Try increasing the amount of space that is allocated by adjusting the values on the Parmlib Staging Options screen. See ["Staging Options" on page 26](#) for more information.

CUZ3304E *module name* - Invalid Function Code in DCBS Node

Explanation:

This message is issued by module *module name*. It indicates that detected an error during DCBS processing. It is indicative of an internal error.

User response:

Try increasing the amount of space that is allocated by adjusting the values on the Parmlib Staging Options screen. See ["Staging Options" on page 26](#) for more information.

CUZ3305I *module name* - Cleanup in Progress-Waiting for Post

Explanation:

This message is issued by module *module name*. It indicates that stage processing is waiting for a termination request.

User response:

CUZ3306I *module name* - Hit End of Linklist-Waiting for Post

Explanation:

This message is issued by module *module name*. It indicates that processing is complete and waiting for a Post.

User response:

CUZ3307I *module name* - Entered PLO Restart Logic

Explanation:

This message is issued by module *module name*. It indicates PLO Restart Logic is processing.

User response:

CUZ3308I *module name* - Channel
Appendage has ended

Explanation:

This message is issued by module *module name*. It indicates that Channel Appendage has ended.

User response:

CUZ3309I *module name* - Waiting 2 Seconds
for Appendage Post

Explanation:

This message is issued by module *module name*. It indicates that is waiting for its Appendage to Post.

User response:

CUZ3310E *module name* - Error Allocating
Staging File for Deletion

Explanation:

This message is issued by module *module name*. It indicates that an error occurred when the staging process attempted to delete the staging file after copying the data to the cloud.

User response:

CUZ8001E *module name* - Error Obtaining
Storage for CELL Pool Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error obtaining storage for a cell pool.

User response:

CUZ8002I *module name* - Reusing a Cell

Explanation:

This message is issued by module *module name*. It indicates that cell pool processing is reusing an available, unused cell pool.

User response:

CUZ8003I *module name* - Obtaining a New
Cell

Explanation:

This message is issued by module *module name*. It indicates that detected an error obtaining a new cell pool.

User response:

CUZ8004E *module name* - Cell Pool Get Failed

Explanation:

This message is issued by module *module name*. It indicates that detected an error while attempting to get a cell pool.

User response:

CUZ8005E *module name* - Error Obtaining
Storage for CELL Pool Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a Cell Pool Definition.

User response:

CUZ8006E *module name* - Error Releasing
Storage for CELL Pool Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a Cell Pool Definition.

User response:

CUZ8101E *module name* - Error Obtaining
Storage for DCBS Local Control
Block

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a DCBS local control block.

User response:

CUZ8102E *module name* - Error Obtaining
Storage for DCBS Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a DCBS Definition.

User response:

CUZ8103E *module name* - Error Releasing
Storage for DCBS Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a DCBS Definition.

User response:

CUZ8104E *module name* - Error Releasing
Storage for DCBS Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a DCBS Definition.

User response:

CUZ8201E *module name* - Error Obtaining
Storage for Cloud Definition

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a Cloud Definition.

User response:

CUZ8202E ***module name* - Error Obtaining Storage for Cloud Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a Cloud Definition.

User response:

CUZ8203E ***module name* - Error Releasing Storage for CLOUD Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a Cloud Definition.

User response:

CUZ8204E ***module name* - Error Releasing Storage for CLOUD Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a Cloud Definition.

User response:

CUZ8301E ***module name* - Error Obtaining Storage for FILTER Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a Filter Definition.

User response:

CUZ8302E ***module name* - Error Obtaining Storage for FILTER Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while obtaining storage for a Filter Definition.

User response:

CUZ8303E ***module name* - Error Releasing Storage for Filter Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a Filter Definition.

User response:

CUZ8304E ***module name* - Error Releasing Storage for Filter Definition**

Explanation:

This message is issued by module *module name*. It indicates that detected an error while releasing storage for a Filter Definition.

User response:

CUZ9999E ***module_name* - Recovery Cleanup Invoked**

Explanation:

The module listed at the beginning of the message has invoked the recovery cleanup process.

User response:

If you need more information, review the messages issued by this module.

FEC Scroll Messages

This section lists all of the messages that may display while attempting to use scroll commands with this product. The FEC Scroll messages are listed below

CUZA900E **Invalid Column Function value.
Valid values: 1, 2, 3, 4**

Explanation:

An invalid character was entered in the Column Function field.

User response:

Specify a valid character (1, 2, 3, or 4).

CUZA901E **Invalid Permanent View value.
Valid values: Y, N**

Explanation:

An invalid value was entered in the Permanent View field.

User response:

Correct the value or cancel. Valid values are Y and N.

CUZA902E **Invalid Reset View value. Valid values are Y, N**

Explanation:

An invalid character was entered in the Reset View field. Valid characters are Y and N.

User response

Specify a valid value or cancel. Valid values are:

- Y - Resets all customizations.
- N - Customizations are not reset.

CUZA903E **Invalid Stop Sorting value. Valid values: Y, N**

Explanation

The specified stop sorting value is not valid. Valid values are:

- Y - Indicates that sorting will be stopped.
- N - Indicates that sorting will continue.

User response:

Specify a valid value or cancel.

CUZA904E Invalid command in FORM display**Explanation:**

The command you issued when viewing the FORM display was not valid.

User response:

Valid commands for FORM display include NROW and PROW.

CUZA905E FORM command not supported from CSETUP function**Explanation:**

The FORM command was issued from a CSETUP function. FORM is not supported while in a CSETUP function (CSETUP functions include CFIX, CORDER, CSIZE and CS).

User response:**CUZA906E Invalid parameter for NROW. Must be numeric.****Explanation:**

The parameter you specified for NROW (next row) was not numeric and is therefore invalid.

User response:

Specify a numeric value corresponding to the number of rows to advance. The default value for NROW is 1.

CUZA907E Invalid parameter for PROW. Must be numeric.**Explanation:**

The parameter you specified for PROW (previous row) was not numeric and is therefore invalid.

User response:

Specify a numeric value corresponding to the number of rows to scroll back. The default value for PROW is 1.

CUZA908E Invalid parameter for NROW. Too many digits.**Explanation:**

An invalid parameter for the NROW (next row) keyword was specified. More than eight digits were specified. Parsing stops at eight digits.

User response:

A parameter of NROW must be between 1 and the number of rows in the current report display. If no parameter is specified, 1 is assumed.

CUZA909E Invalid parameter for PROW. Too many digits.**Explanation:**

Invalid parameter to PROW (previous row) specified. More than eight digits were specified. Parsing stops at eight digits.

User response:

A parameter of PROW must be between 1 and the number of rows in the current report display. If no parameter is specified, 1 is assumed.

CUZA910E CSETUP command not supported from FORM function**Explanation:**

CSETUP functions are not supported while in the FORM display. CSETUP functions include CFIX, CORDER, CSIZE, CSORT, and CSETUP (CSET).

User response:

Exit the current FORM function before issuing a CSETUP function.

CUZA911E Invalid ICR command. Use RIGHT command.**Explanation:**

ICR is only valid with columns that are not their maximum size. You can see the column's current and maximum sizes by issuing CSIZE.

User response:

RIGHT and LEFT commands can be used to see all parts of this column.

CUZA912E Invalid ICL command. Use LEFT command.**Explanation:**

ICL is only allowed with columns that are not their maximum size. You can see the column's current and maximum sizes by issuing CSIZE.

User response:

RIGHT and LEFT commands can be used to see all parts of this column.

CUZA913E Format mix data element not updated.**Explanation:**

Format MIX data cannot be updated when only part of the data is displayed.

User response:**CUZA914E FORM command not supported from FORM function****Explanation:**

FORM was issued from within a FORM display. This is not supported.

User response:

CUZA915E **FORM PF keys set; NROW = *nrow*
PROW = *prow***

Explanation

The NROW (next row) and PROW (previous row) commands are used to move the FORM display window to another row. The UP, DOWN, LEFT, and RIGHT commands move the FORM display window within the current row.

Row, as mentioned above, refers to the row from the original report display, not any reformatted FORM display row.

By default, NROW advances the FORM display to the next row. If NROW *n* is issued, the FORM display will advance *n* rows.

Similarly, PROW moves the FORM display window to the immediately prior row PROW *n* moves the current FORM display window to the *n*th prior row.

User response:

CUZA916E **Invalid CNUM parm. Valid parms are ON, OFF, or blank.**

Explanation:

CNUM was issued with an invalid parameter. Issuing CNUM with no parameter acts as an ON/OFF toggle. ON and OFF are the only parameters accepted. ON turns the CNUM display on. OFF turns the CNUM display off.

User response:

Use a valid CNUM parameter (ON, OFF, or blank).

CUZA917E **Report width for print too large.**

Explanation:

The report width exceeds the maximum print width.

User response:

The maximum report width that is currently supported is 32,760.

CUZA918E ***string* not found. Press PF5 to continue from top.**

Explanation:

The indicated character string was not found.

User response:

To continue searching for the character string from the top of the dialog, press PF5.

CUZA920I **Chars *chars* found *n* times**

Explanation:

Indicates the number of times the specified character was found.

User response:

CUZA921I **Chars *chars* found *n* times**

Explanation:

Indicates the number of times the specified character was found.

User response:

CUZA922I **Search for CHARS *chars* was successful.**

Explanation:

Indicates the search for the indicated characters produced matches.

User response:

CUZA923E **Check for misspelled keywords or embedded blanks in search string.**

Explanation:

Indicates there may be invalid keywords or blanks embedded within the search string.

User response:

Verify and correct the search string to remove embedded blanks or to correct keywords.

CUZA924E ***string1* and *string2* cannot both be specified for FIND command.**

Explanation:

You specified two strings for the FIND command.

User response:

You must specify one FIND string at a time.

CUZA925E **Put quotes (" ") around the string of characters to be displayed.**

Explanation:

The string of characters is not enclosed in quotes.

User response:

Place the string of characters in side quotes.

CUZA926E **Maximum parameter length is 80**

Explanation:

The parameter you specified is too long.

User response:

Place the string of characters in side quotes.

CUZA927E **Invalid COLS parm. Valid parms are ON, OFF, or blank**

Explanation:

COLS was issued with an invalid parameter. Issuing COLS with no parameters acts as an ON/OFF toggle. ON and OFF are the only parameters accepted.

User response:

Enter COLS ON to turn on the COLS display or COLS OFF to turn off the COLS display.

CUZA930I **No columns eligible for resizing**

Explanation:

You cannot re-size any columns.

User response:**CUZA931I** **No columns eligible for sorting****Explanation:**

You cannot sort any columns.

User response:**CUZA932E** **TBMOD failed. RC=return code****Explanation:**

An unexpected return code occurred during TBMOD.

User response

Suggested diagnostics:

- See TBMOD in the *z/OS ISPF Services Guide*.
- Review ISPTLIB allocation.
- Review security-controlled access to ISPTLIB data sets.

CUZA933E **Invalid column name: missing quote****Explanation:**

SORT or CSORT was issued with a parameter that had an initial quotation character, but not a second closing quotation character.

User response:

Either clear the command line and select the desired sort column(s) from the displayed selection list or correct the command on the command line.

CUZA934E **More than 9 columns specified****Explanation:**

SORT or CSORT was issued with too many columns specified as sort columns. A maximum of 9 sort columns can be specified.

User response:

Either clear the command line and select the desired sort column(s) from the displayed selection list or correct the command on the command line.

CUZA935E **Invalid column name****Explanation:**

SORT or CSORT was issued with a column parameter that does not match any column name. A list of the correct column names is seen in the SORT selection panel.

User response:

Either clear the command line and select the desired sort column(s) from the displayed selection list or correct the command on the command line.

CUZA936E **Invalid row selection character****Explanation:**

An invalid selection character was entered in the SSID selection list. The only valid selection character is S.

Alternatively, place the cursor on the desired line and press ENTER (without a line selection character).

User response:

Clear the invalid character.

CUZA937E **Only one row selection allowed****Explanation:**

More than one SSID was selected from the SSID selection list. A maximum of one SSID can be selected.

User response:

Clear all, or all but one row selection character.

CUZA938E **Invalid command****Explanation:**

An invalid command was entered on the SSID selection list panel.

User response:

Clear the command.

CUZA939E **Read of control file failed****Explanation:**

Reading the control data set failed.

User response:

Check the product setup (accessed from the main menu) to view the control data set currently in use. Verify that the data set name is correct.

CUZA943E **Invalid command****Explanation:**

An invalid command was issued. It is not supported on the current panel.

User response:

Check the command for typographical error. Clear or correct the command.

CUZA944I **Empty History****Explanation:**

This is an informational message. The history database is empty. If commands were previously entered, then either HCLEAR was issued or the size of the history database was set to 0. If ISPTABL and ISPTLIB are not allocated, history is not remembered across sessions, and each new session has an empty history database.

User response:

No action is required. To verify allocation of ISPTLIB and ISPTABL, ISRDDN and ISPLIBD can be useful; refer to the ISPF manuals for information on ISRDDN and ISPLIBD.

CUZA945E **Invalid history size limit****Explanation:**

An invalid character was found in the History Size Limit field. Only numeric values from 0-999 are valid.

User response:

Enter a valid value in the History Size Limit field.

CUZA948E TBOPEN failed. RC=return code**Explanation:**

TBOPEN for the history table failed. return code is the return code from the TBOPEN service.

User response:

Check ISPTLIB allocation. Verify the data sets in ISPTLIB. Verify it is a valid PDS. See ISPF manuals for ISPTLIB requirements.

CUZA951E History cleared**Explanation:**

History was cleared either by issuing the HCLEAR command or by setting the History Size Limit to 0.

User response:**CUZA952E Unable to list data sharing members. Display failed****Explanation:**

Command failed attempting to get a list of data sharing members. The reason code and return code are listed in the message.

User response:

Look up the reason code and return code in the DB2 Messages and Codes manual for your version of DB2.

CUZA954E Invalid command**Explanation:**

An invalid command was issued from the data sharing members list/selection panel.

User response:

Clear the command.

CUZA955E No member selected**Explanation:**

You exited the data sharing member selection panel without selecting a data sharing member.

User response:**CUZA956E Invalid row selection character****Explanation**

An invalid selection character was entered in the History output display. A command listed in the History display can be selected for execution either by selecting it with an "S" selection character, or by placing the cursor anywhere on a line within the command and pressing Enter.

When selecting by cursor placement, the cursor can be on the line selection input line, which also has a command number, or on a line with some command text.

User response:

Clear the invalid character.

CUZA957E Only one row selection allowed**Explanation:**

More than one command was selected from the History display. Only one History command can be selected.

User response:

Clear all, or all but one row selection character.

CUZA958E Invalid row selection character**Explanation:**

An invalid selection character was entered in the displayed list of data sharing members. A data sharing member in this display can be selected by selecting it with an "S" selection character, or by placing the cursor anywhere on the desired row and pressing Enter.

User response:

Clear the invalid character.

CUZA959E Only one row selection allowed**Explanation:**

More than one member was selected from the list of displayed data sharing members.

User response:

Clear all, or all but one row selection character.

CUZA960E Cannot list commands without SSID**Explanation:**

A command was issued to select a command syntax diagram, but no SSID has been selected. Syntax diagrams cannot be displayed until an SSID has been selected.

User response:

Select an SSID. You can generate a list of SSIDs by clearing the SSID field, or entering a ? (question mark).

CUZA964E Invalid parameter**Explanation:**

An invalid parameter was used with a command.

User response:

Clear the parameter.

CUZA966E Sort column specified more than once. Selection panel invoked.**Explanation:**

You cannot specify the same sort column twice. The selection panel displays the columns you can choose to sort.

User response:

Specify a different column for the sort process.

CUZA966E	Sort column specified more than once. Selection panel invoked.
-----------------	---

Explanation:

You cannot specify the same sort column twice. The selection panel displays the columns you can choose to sort.

User response:

Specify a different column for the sort process.

CUZA969E	A name of a valid partitioned data set and member name are required.
-----------------	---

Explanation:

This field requires that you specify the name of a partitioned data set (PDS) and the name of a member in the PDS.

User response:

Specify the name of a PDS and a member name.

CUZA970E	A problem was encountered in allocating the files necessary for ISPF file tailoring. Please try again.
-----------------	---

Explanation:

An error occurred when dynamically allocating the ISPF work files ISPFIL, ISPWK1, or ISPWK2.

User response:

Retry the operation. Contact IBM Software Support if the problem persists.

CUZA971I	Display MEPL Job <i>jobname</i> successfully submitted
-----------------	---

Explanation:

This is an informational message. The job listed in the message was submitted for processing

User response:

CUZA972E	Command is not supported on this screen. Please enter a valid command or clear the primary command line.
-----------------	---

Explanation:

An invalid command was entered in the Option line.

User response:

Correct the command or clear the Option line.

CUZA973E	Allocation error. An error was encountered allocating the ISPFIL DD. Attempting to continue.
-----------------	---

Explanation:

While attempting to allocate the ISPFIL DD, and error occurred. is trying to continue processing.

User response:

Check to see if the ISPFIL DD is already allocated.

CUZA974E	Display MEPL internal error - at least one DSN required
-----------------	--

Explanation:

TBA

User response:

Specify a data set name.

CUZA975E	Display MEPL internal error - Invalid eyecatcher length
-----------------	--

Explanation:

TBA

User response:

Specify a different value for the length.

Product Messages

This section lists all of the messages that may display while using the product. The messages are listed below.

CUZS001I	<i>prodname</i> Starting. Version <i>v.r.mm</i>
-----------------	--

Explanation:

The address space is initiating and has starting reading its parameter library (Parmlib). The *v.r.mm* specifies the Version, Release, and Modification level of the executing product.

User response:

No action is required.

CUZS002I	<i>prodname</i> complete. RC=<i>return code</i>
-----------------	--

Explanation:

This message informs you that the product, *prodname*, has completed the start up process and also lists the return code.

User response:

No action is required.

CUZS003I	Parmlib Cards:
-----------------	-----------------------

Explanation:

This message indicates the start of the control cards read from the Parmlib. An image of each control card read is displayed in message CUZS004I.

User response:

No action is required.

CUZS004I	<i>message</i>
-----------------	-----------------------

Explanation:

This message is the second of two messages that display parameter library (parmlib) information.

User response:

No action is required.

CUZS005I *module name - Jobname job name*
DCB data control block - Cloud
Connector filtering has started

Explanation:

This message is issued by module *module name*. It indicates that filtering process has initiated to determine if Jobname *job name* DCB *data control block* should be processed. If filtering is successful, data written to DCB *data control block* will also be copied to the cloud storage destination named in the matching filter criteria.

User response:

No action is required.

CUZS006I *module name - Jobname job name*
DCB data control block - DCB
Rejected because DD was not
opened for output.

Explanation:

This message is issued by module *module name*. It indicates that Jobname *job name* DCB *data control block* will not be processed by . Although the filter criteria matched, the data set is being opened for “input”. only intercepts “output” data sets.

User response:

No action is required.

CUZS007I *module name - Jobname job name*
DCB data control block - DCB
Rejected because Dataset is on
Disk

Explanation:

This message is issued by module *module name*. It indicates that job *job name* DCB *data control block* will not be processed by . Although filter criteria matched, the data set is written to DASD. only intercepts data sets written to Tape.

User response:

No action is required.

CUZS008I *module name - Jobname job name*
DCB data control block - DCB
Rejected because no Dataset
Name was assigned to DD

Explanation:

This message is issued by module *module name*. It indicates that job *job name* DCB *data control block* will not be processed by . Although filter criteria matched, the DD does not specify a data set.

User response:

No action is required.

CUZS009I *module name - Jobname job name*
DCB data control block - Dataset
retrieved from DCB is data set
name

Explanation:

This message is issued by module *module name* to report the backup data set *data set name* referenced by DCB *data control block*. This data set will be copied to a cloud storage location as designated by the filter criteria that caused its selection.

User response:

No action is required.

CUZS010I *module name - Jobname job name*
DCB data control block - DCB
Rejected because no Esoteric Unit
was assigned to DD

Explanation:

This message is issued by module *module name* to indicate that job *job name* DCB *data control block* will not be processed. Although filter criteria matched, the DD does not specify a valid esoteric unit.

User response:

If the data set is one that should be copied to the cloud, verify the catalog information describing the data set. Otherwise, no action is necessary.

CUZS011I *module name - Jobname job name*
DCB data control block - Esoteric
Unit retrieved from DCB is esoteric
unit

Explanation:

This message is issued by module *module name* to report the esoteric unit name *esoteric unit* referenced by DCB *data control block*. This information describes the original device type used for the backup data set, which is recorded in the repository record for optional use in restore processing

User response:

No action is required.

CUZS012I *module name - Jobname job name*
DCB data control block - SMS
Storage Class retrieved from DCB
is storage class

Explanation:

This message is issued by module *module name* to report the SMS Storage Class *storage class* referenced by DCB *data control block*. This information describes the original storage class used for the backup data set, which is recorded in the repository record for optional use in restore processing.

User response:

No action is required.

CUZS013I ***module name - Jobname job name***
DCB data control block - DCB
Rejected because it did not match
any Filter Criteria in Parmlib

Explanation:

This message is issued by module *module name* to indicate that job *job name* DCB data control block will not be processed. There was no matching filter criterion.

User response:

CUZS014I ***module name - Jobname job name***
DCB data control block - DCB Has
been selected for Cloud
Processing based on Storage Class

Explanation:

This message is issued by module *module name* to indicate that the Storage Class referenced by job *job name* DCB data control block matches filter criteria contain in the Parmlib. The backup data set will be copied to the cloud storage location designated on the matching filter criteria.

User response:

CUZS015I ***module name - Jobname job name***
DCB data control block - DCB Has
been selected for Cloud
Processing based on Dataset
Name

Explanation:

This message is issued by module *module name* to indicate that the Dataset Name referenced by job *job name* DCB data control block matches filter criteria contain in the Parmlib. The backup data set will be copied to the cloud storage location designated on the matching filter criteria.

User response:

CUZS016I ***module name - Jobname job name***
DCB data control block - DCB Has
been selected for Cloud
Processing based on Esoteric Unit

Explanation:

This message is issued by module *module name* to indicate that the Esoteric Unit name referenced by job *job name* DCB data control block matches filter criteria contain in the Parmlib. The backup data set will be copied to the cloud storage location designated on the matching filter criteria.

User response:

CUZS017I ***module name - Cloud Name cloud***
name was successfully connected
to

Explanation:

This message is issued by module *module name*. It indicates that the address space contacted and established a connection to Cloud Name *cloud name*. This cloud storage location can now be used as a target for backup copy operations.

User response:

CUZS018E ***module name - Cloud Name cloud***
name connection error - error

Explanation:

This message is issued by module *module name*. The product was unable to connect to cloud *cloud name*. The connection failure returns error *error*.

User response:

Confirm that the cloud definition information in the parameter member is correct. If the problem persists, contact IBM Technical Support.

CUZS019I ***module name - Jobname job name***
DCB data control block - DCB Has
been successfully opened for
Cloud Processing

Explanation:

IBM Cloud Connector has successfully connected to job *job name* and will copy data written to DCB data control block to the cloud named in the filter criteria that selected this DCB.

User response:

CUZS020E ***module name - Jobname job name***
DCB data control block - Cloud
Name cloud name in Filter Criteria
was not defined in Parmlib

Explanation:

This message is issued by module *module name*. was unable to connect to job *job name* DCB data control block because Cloud *cloud name* was not defined.

User response:

Confirm that the cloud definition information in the parameter member is correct. Make sure that the cloud definitions include all the cloud names used any filter criteria. If the problem persists, contact IBM Technical Support.

CUZS021I ***module name - Jobname job name***
DCB data control block - DCB will
be written to Cloud Name cloud
name

Explanation:

This message is issued by module *module name*. will copy all data written to DD data control block of job *job name* to Cloud name *cloud name*.

User response:

CUZS022E ***module name - Jobname job name***
DCB data control block - Cloud

**Name *cloud name* has encountered
Connection issues - Process
Bypassed**

Explanation:

This message is issued by module *module name*. encountered an error while attempting to connect or write to Cloud *cloud name*. Data written to DCB *data control block* of job *job name* will not be written to the designated cloud destination.

User response:

Check your communications network for errors or contact your Network administrator, providing them with the connection information regarding Cloud *cloud name*. If the problem persists, contact IBM Technical support.

**CUZS023E *module name* - Cloud Name
location has not been defined -
History Processing Bypassed**

Explanation:

This message is issued by module *module name* to indicate that it was unable to find the designated cloud storage location *location*. The historical backups will not be copied to cloud storage.

User response:

Confirm that the cloud definition information in the parameter member is correct. Make sure that the cloud definitions include all cloud names used any filter criteria. If the problem persists, contact IBM Technical Support.

**CUZS024W *module name* - Jobname *job name*
DCB *data control block* DCB has
BLKSIZE>32K. May cause Aux
Storage Consumption.**

Explanation:

This message is issued by module *module name* to indicate that DCB *data control block* use of a BLKSIZE greater than 32K will force to utilize additional storage for staging processing. If you are using staging, data destined for cloud storage is first copied to temporary DASD locations and later moved to cloud storage destinations. Although a block size of greater than 32K is allowed for tape, MVS does not allow it for DASD. Because of this, *data control block* requires additional processing and storage utilization to create the temporary DASD staging data sets.

User response:

To avoid this, turn off the Stage-to-DASD option. You may however need to weigh performance in using Stage-to-DASD versus not using it.

**CUZS025E *module name* - Jobname *job name*
DCB *data control block* - Staging
Data Failure. Cloud copy
bypassed.**

Explanation:

This message is issued by module *module name* to indicate that it encountered an I/O failure while attempting to write to the staging data set. The data set allocated to DCB *data control block* will not be copied to its designated cloud storage location. This error will be accompanied by z/OS MVS messages related to I/O failures, but the most common cause of this is lack of DASD space availability.

User response:

You may want to review and adjust the "Staging Options" that are set in the Parmlib Options, which is Option 1 on the Main Menu. Check to ensure that the values specified for the primary and secondary space allocations are large enough to stage the cloud data.

**CUZS026I *module name* - Jobname *job name*
DCB *data control block* Cloud
Connector Processor has
Successfully Started**

Explanation:

This message is issued by module *module name*. The product has started successfully.

User response:

**CUZS027E *module name* - Jobname *job name*
DCB *data control block* All Memory
Cells Exhausted. Processing
stopped.**

Explanation:

This message is issued by module *module name*. The amount of memory cells that were specified in the Parmlib are not large enough to continue cloud processing. Increase the number of primary cells to be greater than 30.

User response:

Update the amount of memory cells and the cell pool size in the Parmlib. Select **Option 1 (Settings)** on the Main Menu, then select **Option 1 (General Options)** on the Parmlib Options screen. Change the value in the **Memory Primary Cells** field to a number greater than 30.

**CUZS028I *module name* - Jobname *job name*
DCB *data control block* - Channel
End Appendage has Started**

Explanation:

This message is issued by module *module name*. This message is for your information only.

User response:

**CUZS029I *module name* - Jobname *job name*
- Waiting for job to complete
before completing STC Shutdown**

Explanation:

This message is issued by module *module name* to inform you that the Cloud Tape Connector started task will be shutdown after the job *job name* completes.

User response:

CUZS030E ***module name - Cloud Connector Initialization task has abended - Abend code: code***

Explanation:

This message is issued by module *module name*. There is a problem with the initialization task. An abend code is listed in the message to help determine what type of problem occurred.

User response:

Use the abend code listed in the message to determine what caused the abend and correct the problem.

CUZS031E ***module name - Cloud Connector Initialization error -error text***

Explanation:

This message is issued by module *module name*. An error occurred while attempting to initialize . The *error text* provides additional information on this error.

User response:

Refer to the additional information in the message to determine how to correct the error.

CUZS032E ***module name - No valid Cloud Connections found - All processing disabled***

Explanation:

This message is issued by module *module name* and indicates that could not connect to any cloud servers so all processing was stopped. A cloud definition has not been created for any cloud servers.

User response:

Create one or more cloud definitions so that can locate a cloud and connect to it. Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose the Cloud Servers option and use the "C" line command to create a new cloud definition.

CUZS033D ***module name - Jobname job name DCB data control block - Waiting 5 seconds for Channel End to post***

Explanation:

This diagnostic message informs you that the program is waiting five seconds for the Channel End to post.

User response:

No action is required.

CUZS034I ***module name - Jobname job name DCB data control block - Cloud***

Write Function has completed successfully

Explanation:

This message is issued by module *module name*. This message informs you that data was written to the cloud successfully.

User response:

CUZS035E ***module name - Jobname job name DCB data control block - Cloud Write Function has Abended - Abend Code: abend code***

Explanation:

This message is issued by module *module name*. An error occurred while attempting to write data to the cloud. An abend code is provided in this message.

User response:

Refer to the abend code to determine why the write process generated an error.

CUZS036I ***module name - Jobname job name DCB data control block - Waiting for Cloud Write process to complete***

Explanation:

This message is issued by module *module name*. Data is being written to the cloud and the program is waiting for this process to finish.

User response:

CUZS037E ***module name - Jobname job name DCB data control block - Cloud Finish Bypassed - Process Abnormal End***

Explanation:

This message is issued by module *module name*. The job *job name* was attempting to finish a cloud process, but the process ended abnormally.

User response:

CUZS038E ***module name - Jobname job name DCB data control block - Cloud Process Bypassed due to previous errors***

Explanation:

The cloud process that is normally performed by the job listed in the message has been bypassed due to errors that occurred previously.

User response:

CUZS039I ***module name - Jobname job name DCB data control block - Cloud Staging Process has Started***

Explanation:

This message informs you that the cloud staging process has started.

User response:

CUZS040E *module name - Jobname job name*
DCB data control block - Cloud
Staging Process has Abended -
Abend Code: *abend code*

Explanation:

This message is issued by module *module name*. An error occurred while attempting to stage data that is being sent to the cloud. An abend code is provided in this message.

User response:

Refer to the abend code to determine why the staging process generated an error.

CUZS041E *module name - Jobname job name*
DCB data control block - Channel
Program Terminated - Stage/
Cloud not running

Explanation:

This message is issued by module *module name*. An error occurred while attempting to stage data that is being sent to the cloud.

User response:

CUZS042E *module name - Jobname job name*
DCB data control block - Error
Obtaining Cell Pool - Check cell
pool sizes in Parmlib

Explanation:

This message is issued by module *module name*. There is an issue with the size of the cell pool, which is causing a problem with this job. The cell pool size can be adjusted on the Parmlib General Options screen.

User response:

Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose General Options and type a larger value in the "Memory Cell Pool Size" field.

CUZS043E *module name - Jobname job name*
DCB data control block - text

Explanation:

This message is issued by module *module name*. There is an issue with this job, which is explained in the text listed at the end of the message.

User response:

Try to correct the error, if possible.

CUZS044E *module name - Jobname job name*
DCB data control block - Failed to
open, bypassing Cloud Processing

Explanation:

This message is issued by module *module name*. The job *job name* did not open so cloud processing did not occur.

User response:

Try to correct the error, if possible.

CUZS045E *module name - Dataset data set*
name - No Cloud Server found,
bypassing Cloud Processing

Explanation:

This message is issued by module *module name*. The data set *data set name* did not contain a cloud server definition so cloud processing did not occur.

User response:

Create a cloud definition so that can locate the cloud and connect to it. Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose the Cloud Servers option and use the "C" line command to create a new cloud definition.

CUZS046E *module name - Repository backup*
bypassed. No Repository Filter
criteria found

Explanation:

This message is issued by module *module name*. The Repository filter specifies where to save a backup of the entire repository. No criteria is allowed for this filter type and only one Repository filter can be defined. A backup of the repository is performed every "n" minutes, based on the value you specified in the "Auto Bkup Repository Min" field on the General Options screen (under Parmlib Settings). If you specify a value in the "Auto Bkup Repository Min" field, but do not create a Repository filter to specify where the repository backup should be saved, this error message displays.

User response:

Create a Repository Backup filter. Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose the Backup Filter Criteria option and use the "C" line command to create a new Repository Backup filter.

CUZS047I *module name - Dataset data set*
name - Has been selected for
Cloud Processing - Dataset Name

Explanation:

This message is issued by module *module name* and informs you that the data set name listed in the message has been selected for cloud processing.

User response:

CUZS048E *module name - Cloud Name cloud*
name Not found for backing up
Cloud Connector Repository

Explanation:

This message is issued by module *module name*. The cloud *cloud name* was supposed to be used for a backup of the repository, but the cloud name was not found. This may indicate that the wrong cloud name was used or that cloud has not been defined for use by .

User response:

Verify the spelling of the cloud name and that a Repository Filter has been created. Only one repository is allowed. If there is no cloud definition for this cloud name, you must make one. Create a cloud definition so that can locate the cloud and connect to it. Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose the Cloud Servers option and use the "C" line command to create a new cloud definition.

CUZS049E ***module name - Dataset data set name not found - Cloud Processing bypassed***

Explanation:

This message is issued by module *module name*. The data set *data set name* is supposed to be used in cloud processing, but the data set could not be found.

User response:

Verify the data set name. Also use Option 3, Cloud Datasets, on the Main Menu to enter selection criteria to determine which data sets you want to process.

CUZS050I ***module name - Dataset data set name - Has been excluded from Cloud Processing - Exc List***

Explanation:

This message is issued by module *module name*. The data set *data set name* listed in the message has been placed on the "Exclude" list for cloud processing.

User response:

CUZS051I ***module name - Dataset data set name - Has already been backed up to the cloud, processing bypassed***

Explanation:

This message is issued by module *module name* and informs you that the data set *data set name* has already been copied to the cloud. No additional processing of this data set is required.

User response:

CUZS052E ***module name - Jobname job name DCB data control block - Error writing data to repository***

Explanation:

This message is issued by module *module name*. An error occurred while job *job name* tried to write data to the repository.

User response:

Check to see if there is a connectivity issue with that cloud. If the connection went down during the repository backup, that could cause an error. If that is the case, submit the job again.

CUZS053E ***module name - Jobname job name DCB data control block - Write to Cloud retry count exceeded***

Explanation:

This message is issued by module *module name*. The job *job name* has attempted to write to the cloud multiple times and has now exceeded the amount of retry attempts allowed.

User response:

Attempt to determine why the job cannot write to the cloud. Perhaps there is a connectivity issue. If you want to increase the number of times the job should attempt to write to the cloud before generating this error message, you can adjust that value. Choose Option 1, Cloud Connector Settings (Parmlib Options), on the Main Menu. On the Parmlib Options Menu, choose the Staging Options and change the number in the "Error Retry Count" field. The maximum value is 9999.

CUZS054I ***module name - Jobname job name DCB data control block - Staging to Cloud Write Restart Started***

Explanation:

This message is issued by module *module name*. The job *job name* has restarted the process of writing staged data to the cloud.

User response:

CUZS055E ***module name - Jobname job name DCB data control block - STC has been Cancelled. Cloud Process Terminating***

Explanation:

This message is issued by module *module name*. The job *job name* has cancelled the started task and any cloud processing is now ending.

User response:

CUZS056I ***module name - Cloud Connector Repository being backed up to Cloud cloud name***

Explanation:

This message is issued by module *module name* to indicate that the VSAM Repository has been written to Cloud *cloud name*. This is part of the process used to create a copy of the VSAM repository at a cloud

storage destination designated by Repository Backup Filter criteria. Refer to “Repository Restore” for more information.

User response:

Record the cloud name for future reference.

CUZS057E ***module name - Error invoking IDCAMS on Repository backup to Cloud location***

Explanation:

This message is issued by module *module name*. An error occurred when calling IDCAMS while performing a backup of the Cloud Connector Repository to the cloud storage *location*.

User response:

CUZS058I ***module name - Dataset data set name was successfully deleted from Cloud***

Explanation:

This message is issued by module *module name* and indicates that the data set *data set name* was deleted from the cloud.

User response:

CUZS059D ***module name - Jobname job name DCB data control block - Open Attempt on DCB in Progress***

Explanation:

This is a diagnostic message. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change **DEBUG** to OFF on the **Parmlib General Options** screen (Option 1.1 from Main Menu).

CUZS060D ***module name - Jobname job name DCB data control block - Macro Format format***

Explanation:

This is a diagnostic message. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change DEBUG to OFF on the Parmlib General Options screen.

CUZS061D ***module name - Jobname job name DCB data control block - Macro Format Qualified for Cloud Process***

Explanation:

This is a diagnostic message regarding the macro format of the Data Control Block (DCB). In this case, the macro format does qualify for cloud processing. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change **DEBUG** to OFF on the **Parmlib General Options** screen (Option 1.1 from Main Menu).

CUZS062D ***module name - Jobname job name DCB data control block - Macro Format Not Qualified for Cloud Process***

Explanation:

This is a diagnostic message regarding the macro format of the Data Control Block (DCB). only processes DCB's with a macro format of "puts" or "writes". The macro format provided in this diagnostic message is the internal hexadecimal format. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change **DEBUG** to OFF on the **Parmlib General Options** screen (Option 1.1 from the Main Menu).

CUZS063E ***module name - Jobname job name DCB data control block - Move Data to Cell Pool Failed***

Explanation:

This message is issued by module *module name*. The job *job name* uses the specified DCB. An attempt to move data to a cell pool failed.

User response:

CUZS064I ***module name - Jobname job name DCB data control block - Attaching module name Module***

Explanation:

This message is issued by module *module name*. The job *job name* uses the specified DCB *data control block* and is attaching the module *module name*.

User response:

CUZS065I *module name* - Jobname *job name*
DCB data control block - Dataset
opened with DISP=MOD Cannot be
Processed

Explanation:

This message is issued by module *module name*. The job name listed in the message uses the specified DCB. The data set that was opened cannot be processed because of the DISP=MOD command.

User response:

CUZS066I *module name* - Cloud Write tasks
still active - Waiting 10 seconds

Explanation:

This message is issued by module *module name*. The tasks that write to the cloud are still working. The job will wait 10 seconds.

User response:

CUZS067I *module name* - Max History Tasks
Reached - Waiting 30 Seconds for
retry

Explanation:

This message is issued by module *module name*. The maximum number of history tasks has already been reached. The job will wait 30 seconds before trying again.

User response:

However, you could increase the maximum number of history tasks by changing the value specified in the "Max Backup History Tasks" field on the Parmlib General Options screen.

CUZS068I *module name* - Enq failure on
Volume - Waiting for other tasks to
finish

Explanation:

This message is issued by module *module name*. This message informs you that the program must wait for other tasks to finish before attempting to queue another task for this volume.

User response:

CUZS069I *module name* - History Dataset
data set is scheduled to be written
to Cloud location

Explanation:

This message is issued by module *module name* and informs you that the existing history data set *data set name* is currently scheduled to be written to Cloud location.

User response:

CUZS070E *module name* - Cross memory
initialization failure: Missing SDA

Explanation:

This message is issued by module *module name* when it is unable to locate the SDA or the SDA is corrupted. The SDA is the principle storage control block for

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS071E *module name* - Execution failure:
Missing SDA

Explanation:

This message is issued by module *module name* when it is unable to locate the SDA or the SDA is corrupted. The SDA is the principle storage control block for

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS072E *module name* - Cross memory
initialization failure: Invalid
function

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it encountered an invalid function in the parameter list.

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS073E *module name* - Cross memory
initialization failure: Linkage
Index Reserve failure

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it encountered an error while attempting to reserve a linkage index.

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS074E *module name* - Cross memory
initialization failure: Index Set
failure

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it encountered an error while attempting to set the authorization index.

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS075E ***module name* - Cross memory initialization failure: Entry Table Create failure**

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it encountered an error while attempting to create an entry table.

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS076E ***module name* - Cross memory initialization failure: Entry Table Connect failure**

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it encountered an error while attempting to connect its entry table.

User response:

Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS077I ***module name* - Cross memory environment initialized**

Explanation:

This message is issued by module *module name* to indicate that has successfully constructed its Program Call (PC) environment.

User response:

CUZS078E ***module name* - Cross memory initialization failure; PC routine *routine name* failed to load**

Explanation:

This message is issued by module *module name* when it is unable to establish its PC environment. In this case, it was unable to load the necessary load modules.

User response:

Confirm that the address space has access to its product load modules. Restart the address space. If the problem persists, contact IBM Technical Support for assistance.

CUZS079I ***module name* - Cross memory environment cleanup completed**

Explanation:

This message is issued by module *module name* to indicate it has completed removal of the cross-memory environment.

User response:

CUZS080E ***module name* Execution failure: DCBS storage obtain failed**

Explanation:

This message is issued by module *module name* to indicate it was unable to obtain storage for a DCBS control block. The DCBS is a multifaceted block of storage used in many areas of IBM Cloud Tape Connector. Processing for module *module name* does not continue and the service requesting the DCBS fails.

User response:

If the problem persists, contact IBM Technical Support for assistance.

CUZS081E ***module name* Execution failure: Invalid RSTR block**

Explanation:

This message is issued by module *module name* to indicate that a restore requestor provided an invalid Restore Request block (RSTR). This is an internal error.

User response:

Contact IBM Technical Support.

CUZS082I ***module name* Initialized**

Explanation:

This message is issued by module *module name* to indicate that it has successfully completed initialization processing.

User response:

CUZS083E ***module name* Execution failure: Invalid function**

Explanation:

This message is issued by module *module name* to indicate that the module, a PC routine, was called with an invalid function or invalid parameter list. Execution terminates. This is an internal error.

User response:

Contact IBM Technical Support.

CUZS084I ***module name* Scheduling restore, DSN=*data set name***

Explanation:

This message is issued by module *module name* to indicate it has scheduled a restore for data set *data set name* at the user's request.

User response:

CUZS085I ***module name* Completing restore request, DSN=*data set name***

Explanation:

This message is issued by module *module name* to indicate it has completed restore processing for data set *data set name*.

User response:

CUZS086E ***module name* Delete failed: DCBS
OBTAIN failed, U (*userid*)**

Explanation:

This message is issued by module *module name* to indicate it was unable to obtain the DCBS control block. This control block is needed to instruct the cloud storage location to delete the backup data set that is being deleted from the repository. User *userid* initiated the request.

User response:

Retry the request. If the problem persists, contact IBM Technical Support.

CUZS087E ***module name* Delete failed: Cloud
node not found, U (*userid*)**

Explanation:

This message is issued by module *module name* to indicate it was unable to delete the backup data set because the cloud name listed as the backup's location cannot be found. User *userid* initiated the request.

User response:

Confirm the contents of your cloud definitions. Ensure your cloud definitions contain the cloud name listed on the backup data set repository record. Retry the request. Contact IBM Technical Support if the problem persists.

CUZS088E ***operation* Failure, FDBK(*feedback*),
U (*userid*), *type* KEY(*key*)**

Explanation:

This message is issued to indicate that a VSAM error occurred. The operation will be PNT, GET, PUT, ERS, or UNK. The *feedback* is the RPLDFDBK field from the RPL. The *userid* is that of the user making the request. The *type* indicates whether this is a profile or discrete entry. The *key* is the key of the record being processed when encountered the error.

User response:

Retry the request. If the problem persists, contact IBM Technical Support.

CUZS089I ***module name* Restore requested,
GEN=*ggg*, DSN=*data set name***

Explanation:

This message is issued by module *module name* to indicate that a Restore operator command has been used to initiate a restore request. The data set to restore is *data set name* and the generation requested is *ggg*. The generation specification indicates how far back version selection should go.

User response:

CUZS090I **Entry deleted, U=*userid*, T=*type*,
D=*dataset***

Explanation:

User *userid* successfully deleted backup data set *dataset* from the repository and the cloud. The *type* will be either DATA or PROF. A DATA entry defines a specific data set backup. A PROF entry is a profile of a group of specific data set discrete entries. The PROF entry is created by the address space when the first backup is created, and deleted when the last backup is deleted from the repository.

User response:

CUZS091E ***module name* Request failed,
Cloud does not contain the
requested data set**

Explanation:

This message is issued by module *module name* to indicate that the specified backup data set cannot be restored. The cloud storage location does not contain it.

User response:

Retry the request. Confirm your request. Make sure you specified the correct name.

CUZS092I **Restore successful, U=*userid*,
D=*dataset*, CF=*cloudfile***

Explanation:

The Restore request did not encounter any errors. The request was initiated by User *userid* for Data Set *dataset* that resides in Cloud file *cloudfile*.

User response:

CUZS093E **Restore failed, RC (*nn*), U=*userid*,
DSN=*data set***

Explanation:

This message indicates that the restore request initiated by User *userid* for backup data set *data set* failed with return code (RC) *nn*.

User response:

Retry the request. An attempt to restore to DASD could encounter space problems.

CUZS094E ***module name* Restore failed:
Dynamic allocation failed for
backup**

Explanation:

This message is issued by module *module name* to indicate that the restore request was unable to allocate the output data set. If restoring to DASD, this could be the result of insufficient available space. You could also have an issue with the device unit name; it might be incorrectly specified in the restore request. Review the system log for allocation messages related to this request.

User response:

Correct any errors that caused the failure and retry the request.

CUZS095E ***module name* Restore failed: Open failed for backup**

Explanation:

This message is issued by module *module name* to indicate that the restore request was unable to open the output data set. Review the system log for any allocation errors.

User response:

Correct any errors and retry the request.

CUZS096E ***module name* Restore failed: DCBS OBTAIN failed**

Explanation:

This message is issued by module *module name* to indicate that the restore request was unable to obtain a DCBS control block. This control block is required to request and obtain the backup data set from the cloud storage location.

User response:

Increase the address space region size. If problem persists, contact IBM Technical Support.

CUZS097E ***module name* Restore failed: Cloud node not found**

Explanation:

This message is issued by module *module name* to indicate that the restore request was unable to find the cloud node named in the backup data set repository record.

User response:

Confirm the contents of your cloud definitions. Ensure your cloud definitions contain the cloud name listed on the backup data set repository record. Retry the request. If the problem persists, contact IBM Technical Support.

CUZS098E ***module name* Restore failed, CUZ1CLDI failed, *data set name***

Explanation:

This message is issued by module *module name* to indicate that the interface responsible communicating with cloud storage failed. In this case a Restore Request or an attempt to retrieve data from cloud storage failed. The message text describes the specific error.

User response:

Correct the error described in the message text if possible and retry. If the problem persists, contact IBM Technical Support.

CUZS099E ***module name* Restore ALL failed**

Explanation:

An error occurred while attempting to restore multiple data sets, which caused the restore process to fail.

User response:

Try choosing fewer data sets to restore, instead of all of them.

CUZS101I ***module name* Placing Repository restore request on queue**

Explanation:

This message is issued by module *module name* to indicate that a request to restore the Cloud Connector VSAM Repository has been placed on the request queue. Refer to [“Restoring a Repository” on page 89](#) for more details.

User response:

CUZS102I ***module name* Cloud Connector Repository List staged to *data set name***

Explanation:

This message is issued by module *module name* to indicate that the “List” has been written to data set *data set name*. Once this “staged” data set is copied to the designated cloud location, it is deleted. This is part of the process used to create a copy of the VSAM repository at a cloud storage destination designated by Repository Backup Filter criteria. This “List” is a fixed name file that resides on cloud storage only and contains a list of repository records describing the VSAM Repository. Refer to [“Restoring a Repository” on page 89](#) for more information.

User response:

Record the list name *data set name* for future reference.

CUZS103W ***module name* - Output DSN Allocation failed on DASD - Allocating on Tape**

Explanation:

This message is issued by module *module name*. It is a warning message that an attempt was made to allocate a restore data set on disk, but the allocation failed. An attempt will now be made to allocate a restore data set on tape.

User response:

CUZS104I **Entry deleted due to new generation, D=*data set***

Explanation:

This message indicates that the backup data set *data set* was deleted from the repository and cloud due to generation processing. A newly created backup caused the number of available backup data sets to exceed the maximum number of generations. Thus, the Cloud Tape Connector address space deletes the oldest version of the backup data set.

User response:

No action is required. However, you can change the Max Cloud Backup Gens to increase or reduce the number of backup generations.

**CUZS105I ALESERV failure prevented
Restore response**

Explanation:

The address space was unable to report Restore results back to the cross-memory requester. In this instance, the requester will not receive a return code or a message regarding the status of the request. Messages regarding the results of the Restore request are still reported in the started task address space and on the System log. This could result from the Requester's address space being terminated before the Restore request was complete.

User response:

Review the started task address space or System log for more information. If the problem persists, contact IBM Technical Support.

**CUZS106E *module name* - Error retrieving
WTDE**

Explanation:

The message is issued by module *module name*. It indicates an internal error occurred that prevented Repository operations.

User response:

**CUZS107I *module name* - Repository task
ready**

Explanation:

This message is issued by module *module name*. It indicates that the task responsible for managing the VSAM Repository is ready for processing.

User response:

**CUZS108I *module name* - Repository task
terminating**

Explanation:

This message is issued by module *module name*. It indicates that the task managing access to the VSAM repository is terminating. This is typically due to a shutdown request, but it can also occur as a result of a catastrophic failure.

User response:

No action is required if the message displayed due to a shutdown request. However, if this is not the result of a shutdown request, review the system log for error messages describing a possible failure. Contact IBM Technical Support for assistance, if needed.

**CUZS109E *module name* - CUZCLOUD DD
open failed, RC(rc)**

Explanation:

The message is issued by module *module name* to indicate an error occurred during initialization. was unable to OPEN the VSAM Repository and failed with return code *rc*. This can occur as a result of an incomplete installation or damage to the VSAM Repository data set.

User response:

Confirm the successful completion of all installation and configuration steps. Confirm that the VSAM Repository allocated to the CUZCLOUD DD statement exists and is accessible. Review the system log for additional error messages related to this failure, including security and IOS failures.

**CUZS111I *module name* - Repository Cross
System Sharing active**

Explanation:

This message is issued by module *module name*. It indicates that the VSAM data set assigned to the CUZCLOUD DD statement in the address space (i.e. The Repository) was opened with DISP=SHR. With that, the data set will be shared with other address spaces on other z/OS MVS images. This is contingent upon all the address spaces using DISP=SHR.

User response:

CUZS112I *module name* - Restore task ready

Explanation:

This message is issued by module *module name*. It informs you that the task responsible for Restore requests is ready for processing.

User response:

**CUZS113E *module name* - Error retrieving
RTDR**

Explanation:

The message is issued by module *module name*. It indicates an internal error occurred while attempting to process a Restore request.

User response:

**CUZS114I *module name* - Restore scheduler
task terminating**

Explanation:

The message is issued by module *module name*. It indicates that the task responsible for scheduling Restore processing is terminating. This is usually because of a shutdown request.

User response:

No action is required if the message displayed due to a shutdown request. However, if this is not the result of a shutdown request, review the system log for error messages describing a possible failure. Contact IBM Technical Support for assistance, if needed.

CUZS115I ***module name* - Termination
waiting on restore tasks.**

Explanation:

This message is issued by module *module name*. A request to terminate the address space has been delayed because there are active Restore tasks. Once those Restore tasks complete, the address will terminate.

User response:

CUZS116I ***module name* - Attaching restore
for DSN=*data set name***

Explanation:

This message is issued by module *module name*. It indicates that the task responsible for managing Restore processing has initiated a specific restore request. In this case, the request to restore *data set name* executes as a sub-task of the address space.

User response:

CUZS117E ***module name* - Error retrieving
RTCR**

Explanation:

The message is issued by module *module name*. It indicates an internal error occurred while attempting to process a request to free the storage used for a particular Restore request.

User response:

CUZS118I ***module name* - Restore task
cleanup ready**

Explanation:

This message is issued by module *module name*. It informs you that the task responsible for freeing storage used by Restore tasks is ready to process requests.

User response:

CUZS119I ***module name* - Restore task
cleanup terminating**

Explanation:

This message is issued by module *module name*. It informs you that the task that frees storage acquired by the Restore task is terminating. Usually, this is a result of a shutdown request. However, it could be the result of a failure.

User response:

No action is required if this message is due to a shutdown request. However, if this is not the result of a shutdown request, review the log for error conditions. Contact IBM Technical Support if needed.

CUZS120I ***module name* - Restore task
cleanup complete for DSN=*data
set name***

Explanation:

This message is issued by module *module name*. It indicates that any storage obtained for Restore processing for *data set name* has been freed.

User response:

CUZS121I ***module name* - Jobname *job name*
DCB data control block - Cell Pool -
I/O Flagged for Suspension**

Explanation:

This message is issued by module *module name*. This message indicates that all of the memory cells are full and I/O is being suspended until either staging or the cloud writer process catches up. When the other task catches up, the I/O process will be resumed.

User response:

CUZS122I ***module name* - Jobname *job name*
DCB data control block - I/O Has
Been Temporarily Suspended**

Explanation:

This message is issued by module *module name*. This is an informational message that all memory cells are full and IO has been suspended. You will receive another message, CUZS0123I, when staging or the cloud writer catches up and IO is resumed.

User response:

CUZS123I ***module name* - Jobname *job name*
DCB data control block - I/O Has
Been Resumed**

Explanation:

This message is issued by module *module name*. Previously the I/O process had been suspended due to the memory cells being full during the staging or cloud writing process. However, there is no longer an issue and the I/O process has now been resumed.

User response:

CUZS124I ***module name* - Staging Dataset
data set name successfully
allocated for cloud copy**

Explanation:

This message is issued by module *module name*. It indicates that the staging data set *data set name* was successfully allocated as an interim location. The data set being copied to the cloud will reside in this staging data set until the copy-to-cloud operation completes successfully, at which time will delete the staging data set.

User response:

CUZS125I ***module name* - Disk Dataset *data
set name* successfully allocated for
cloud copy**

Explanation:

This message is issued by module *module name*. It indicates that successfully allocated an existing data set, *data set name*, in preparation for copying this data set to the Cloud. However, will not delete this data set once copy-to-cloud operations complete.

User response:

CUZS126W *module name* - Dataset *data set name* contains a DSORG of value which is not supported for cloud processing

Explanation:

This message is issued by module *module name*. The only data set organization values allowed for cloud processing are PS (Physical Sequential), DA (Direct Access), and U (Undefined). However the DSORG value associated with this data set is not allowed for cloud processing.

User response:

CUZS127W *module name* - Dataset *data set name* is not a sequential file - Cloud processing bypassed

Explanation:

This message is issued by module *module name*. The data sets you add to the include history list must have the data set organization of PS (Physical Sequential) or U (Undefined). Partitioned data sets (PDS and PDSE) and VSAM files are not supported.

User response:

To copy a VSAM file or partitioned data set to the cloud, you will first need to run a utility program, such as IDCAMS, to create a sequential data set and then have that output data set sent to the cloud, either by filter criteria or through the use of the History Include/Exclude data set list.

CUZS128E *module name* - Jobname *job name* Error allocating Staging file - Cloud processing bypassed

Explanation:

This message is issued by module *module name*. One option for capturing data while it is being written to tape is Staging data to DASD. This option allows you to quickly save the data to a DASD data set called a staging file. After the batch job has completed, the contents of the staging file are written to the cloud by the Started Task. Unfortunately, an error occurred while the job listed in the message attempted to allocate a staging file, causing to bypass the staging process completely.

User response:

Modify the staging values on the Parmlib Staging Options screen and try again.

CUZS129E *module name* - Jobname *job name* Error allocating Staging file - Process aborting

Explanation:

This message is issued by module *module name*. One option for capturing data while it is being written to tape is Staging data to DASD. This options allows you to quickly save the data to a DASD data set called a staging file. After the batch job has completed, the contents of the staging file are written to the cloud by the Started Task. Unfortunately, an error occurred while the job listed in the message attempted to allocate a staging file, and the file cannot be opened.

User response:

Modify the staging values on the Parmlib Staging Options screen and try again.

CUZS130W *module name* - Jobname *job name* Unable to Page-Fix DECB Buffer Storage areas

Explanation:

This message is issued my module *module name* to indicate that it was unable to page fix the storage areas used to optimize I/O performed to staging processing. The “Number of IO Buffers” Staging option (Parmlib Options) determines the size of a circular buffer queue used when reading the staging data sets. These storage areas require a Page-Fix specification.

User response:

Reduce the Number of IO Buffers in the Staging Options. If problem persists, contact IBM Technical Support.

CUZS131I *module name* - Dataset *data set name* is restarting Stage to Cloud *data set name* copy

Explanation:

This message is issued by module *module name* to indicate that it is restarting the staging process for data set *data set name* that previously failed.

User response:

CUZS132E *module name* - Error Allocating Cloud Connector Repository DSN *data set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error attempting to allocate staging data set *data set name*. As a result, it discontinues the process. Review the system log for possible causes. This is part of the process used to create a copy of the Cloud Connector VSAM Repository at a cloud storage destination designated by Repository Backup Filter criteria. Refer to “Restoring a Repository” on page 89 for more information.

User response:

Correct errors that caused the failure. If the problem persists, contact IBM Technical Support.

CUZS133E *module name - Error Allocating
Cloud Connector List Staging DSN
data set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error attempting to allocate “List” staging data set *data set name*. As a result, it discontinues the process. Review the system log for possible causes. This is part of the process used to create a copy of the Cloud Connector VSAM Repository at a cloud storage destination designated by Repository Backup Filter criteria. This “List” is a fixed name file that resides on cloud storage only and contains a list of repository records describing the VSAM Repository. Refer to [“Restoring a Repository”](#) on page 89 for more information.

User response:

Correct errors that caused the failure. If the problem persists, contact IBM Technical Support.

CUZS134E *module name - Error Opening
Cloud Connector List Staging DSN
data set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error attempting to open “List” staging data set *data set name*. As a result, it discontinues the process. Review the system log for possible causes. This is part of the process used to create a copy of the Cloud Connector VSAM Repository at a cloud storage destination designated by Repository Backup Filter criteria. This “List” is a fixed name file that resides on cloud storage only and contains a list of repository records describing the VSAM Repository. Refer to [“Restoring a Repository”](#) on page 89 for more information.

User response:

Correct errors that caused the failure. If the problem persists, contact IBM Technical Support.

CUZS135E *module name - Error Allocating
Cloud Connector Staging DSN
dataset name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error attempting to allocate staging data set *data set name*. As a result, it discontinues the process. Review the system log for possible causes. This is part of the process used to create a copy of the Cloud Tape Connector VSAM Repository at a cloud storage destination designated by the Repository Backup Filter criteria.

User response:

Correct errors that caused the failure. If the problem persists, contact IBM Technical Support.

CUZS136E *module name - Error Opening
Cloud Connector Staging DSN data
set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error attempting to open staging data set *data set name*. As a result, it discontinues the process. Review the system log for possible causes. This is part of the process used to create a copy of the Cloud Connector VSAM Repository at a cloud storage destination designated by Repository Backup Filter criteria.

User response:

Correct errors that caused the failure. If the problem persists, contact IBM Technical Support.

CUZS137E *module name - Error Freeing
Staging DSN data set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error trying to free staging data set *data set name*. As a result, it discontinues processing for this data set. However, as de-allocation is the last function performed in copying a backup data set to a cloud destination, the data set has been copied.

User response:

If the problem persists, contact IBM Technical Support.

CUZS138E *module name - Error Allocating
Staging DSN data set name*

Explanation:

This message is issued by module *module name* to indicate that it experienced an error trying to allocate staging data set *data set name*. As a result, it discontinues processing for this data set. Review the system log to determine a possible cause for this failure. The most common cause of allocation failures is insufficient DASD space.

User response:

Resolve the issue that caused the failure and retry the request. If the problem persists, contact IBM Technical Support.

CUZS139E *module name - Error on Catalog
Superlocate for DSN data set name*

Explanation:

This message is issued by module *module name* to indicate it failed during an attempt to gather information from the catalog for data set *data set name*. As a result, it discontinues processing for this

data set. Review the system log to determine a possible cause for this failure.

User response:

Retry the request. If the problem persists, contact IBM Technical Support.

CUZS140E ***module name* - Error Opening
Staging DSN *data set name***

Explanation:

This message is issued by module *module name* to indicate that it encountered an error attempting to open staging data set *data set name*. As a result, it discontinues processing for this data set. Review the system log to determine the possible cause for the Open error.

User response:

Correct the cause of the Open error if possible. If the problem persists, contact IBM Technical Support.

CUZS141E ***module name* - End of File
Encountered reading Staging DSN
*data set name***

Explanation:

This message is issued by module *module name* to indicate that it encountered an unexpected end-of-file (EOF) indication while reading the staging data set *data set name*. As a result, it will discontinue processing this data set.

User response:

Retry the request. If the problem persists, contact IBM Technical Support.

CUZS142E ***module name* - Staging file *file*
name found with no CUZ#REPO
record**

Explanation:

This message is issued by module *module name* to indicate that it found a data set named *data set name* that appears to be a staging file, but it does not have proper control information. A CUZ#REPO record is included as the first block of any backup data set copied to the cloud. This data set does not have such a record. This can happen if a non-staging data set is inadvertently allocated under the High Level Qualifier (HLQ) defined for staging data sets.

User response:

If this is truly not a staging data set, change the HLQ so that it is not treated as such. If the problem persists, contact IBM Technical Support.

CUZS143I ***module name* - Dataset *data set*
name is scheduled to be written to
Cloud *cloud name* from Staging
File**

Explanation:

This message is issued by module *module name* to indicate that a currently staged data set *data set name* is scheduled to be written to the cloud storage location *cloud name* designated by its matching filter criteria.

User response:

No action is required.

CUZS144I ***module name* - Dataset *data set*
name successfully written to
Cloud *location***

Explanation:

This message is issued by module *module name* to indicate that it has successfully written the backup data set *data set name* to the cloud storage location named *location*.

User response:

CUZS145D ***module name* - Jobname *job name*
DCB *data control block* - Cell Pool
Cleanup in progress. Wait for Post**

Explanation:

This is a diagnostic message indicating that is cleaning up the allocated cell pools and making them available for reuse. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change **DEBUG** to OFF on the **Parmlib General Options** screen (Option 1.1 on the Main Menu).

CUZS146D ***module name* - Jobname *job name*
DCB *data control block* - Cell Pool
Contention. Wait for Post**

Explanation:

This is a diagnostic message indicating that the is experiencing cell pool contention, which could potentially cause the job to slow down. The module writing the data to the cell pool cannot be used by the "staging" or "cloud write" function until the cell is full. This type of message only displays when DEBUG is turned ON. Only use DEBUG when expressly told to do so by IBM Technical Support. The information in this message will help them to determine the cause of the problem you are experiencing.

User response:

No action is required, unless specifically requested by IBM Technical Support. Change **DEBUG** to OFF on the **Parmlib General Options** screen (Option 1.1. on the Main Menu).

CUZS147I *module name - Dataset data set name successfully Staged to DASD*

Explanation:

This message is issued by module *module name*. It informs you that the process of writing the data set *data set name* to DASD for staging has finished successfully. Now the stage-to-cloud copy process is running under the started task.

User response:

CUZS148I *module name - Dataset data set name - Error determining current backup status, processing bypassed*

Explanation:

This message is issued by module *module name* and informs you that the data set *data set name* will not be copied to the cloud. encountered an unexpected error while attempting to determine if a record of this data set already exists in cloud storage.

User response:

Review the system log for other error messages describing potential problems with the VSAM repository. Retry the request. Contact IBM Technical Support if the problem persists.

CUZS149I *module name - Retry count exceeded for Dataset data set name variable*

Explanation:

This message is issued by module *module name* and indicates that the maximum number of attempts to copy data to data set *data set name* has been reached.

User response:

However, if you would like to increase the retry count, see “[Staging Options](#)” on page 26.

CUZS150I *module name - Restarting Cloud Copy for Dataset data set name variable*

Explanation:

This message is issued by module *module name* and indicates that the another attempt is being made to copy the data set *data set name* to the cloud

User response:

CUZS151I *API program name-Func function code -RC return code*

Explanation:

The API program listed in the message performed a function, such as restore (R) data. The return code indicates whether the function was processed successfully. A return code of 0 indicates a successful completion, 8 indicates a run time error, and 12 indicates an internal error.

User response:

CUZS152E *API program name-Func function code -Error:-number-Connecting to cloud cloud name*

Explanation:

An error occurred when the program listed in the message tried to connect to the specified cloud.

User response:

Check the cloud name to ensure the name is valid. If it is valid, try to determine the cause of the connectivity issue with this particular cloud.

CUZS153E *API program name-Func function code -Error:-number-Building list from cloud cloud name*

Explanation:

An error occurred when the API program tried to create a list of the data sets that are currently located on the specified cloud.

User response:

Check the cloud name to ensure the name is valid. If it is valid, there may be no data sets on that specific cloud or there may be an issue with the program listed in the message.

CUZS155E *API program name-Func function code -Invalid Function Code Passed in APIB Block. Process Aborted*

Explanation:

An error occurred when a function code was passed to the specified program that is not valid for use in this program. The program stopped and ended all processing.

User response:

Change the function code to a valid one for this program.

CUZS156E *API program name-Func function code -Cloud Name cloud name was not found in Cloud Connector Started Task*

Explanation:

The specified cloud is not listed in the started task for Cloud Connector. The default name for the started task is CUZCLOUD.

User response:

Either change the cloud name to one that is in the started task or add the cloud name to the started task. Also the CUZCLOUD started task must be active in order for data sets to be written to or restored from the cloud.

CUZS157E ***API program name-Func function code -Catl Loc Failure on DSN data set name.***

Explanation:

The specified data set could not be located in the catalog, which resulted in the program being unable to perform the specified function.

User response:

Change the data set name to one that is known to be in the catalog or choose a different catalog. Contact IBM Support for assistance, if needed.

CUZS158E ***API program name-Func function code -Copy to DSN not supplied in APIB Block. Process Aborted.***

Explanation:

The program could not copy the data set as expected, so the program stopped the process.

User response:

Contact IBM Support for assistance, if needed.

CUZS159E ***API program name-Func function code -Restore to Tape/Disk not equal to "T" or "D". Process Aborted***

Explanation:

The program could not restore the tape or disk because a character other than "T" or "D" was specified.

User response:

Try running the program again using "T" for tape or "D" for disk.

CUZS160E ***API program name-Func function code -Delete old Restore DSN not equal to "Y" or "N". Process Aborted'***

Explanation:

The program could not delete the old data set that was used for restore processes because a character other than "Y" or "N" was specified.

User response:

Try running the process again using "T" for tape or "D" for disk.

CUZS161E ***API program name-Func function code -Restore Alias / DSN are mutually exclusive. Process Aborted.***

Explanation:

You cannot specify both an alias and a data set name for this restore process. The restore process was not performed.

User response:

Specify an alias or a data set name, but not both, and try the restore process again.

CUZS162E ***API program name-Func function code -Restore Alias / DDNAME are mutually exclusive. Process Aborted.***

Explanation:

You cannot specify both an alias and a DD name for this restore process. The restore process was not performed.

User response:

Specify an alias or a DD name, but not both, and try the restore process again.

CUZS163E ***API program name-Func function code -Restore to DSN / DDNAME are mutually exclusive. Process Aborted.***

Explanation:

You cannot specify both a data set name and a DD name for this restore process. The restore process was not performed.

User response:

Specify a data set name or a DD name, but not both, and try the restore process again.

CUZS164E ***API program name-Func function code -Restore keyword is a Required Keyword for this Restore. Process Aborted.***

Explanation:

The restore process was not performed because the keyword listed in the above message is missing from the RESTORE command.

User response:

Ensure you include the missing keyword and try the restore again.

CUZS165E ***API program name-Func function code -DASD Unit unit name is not a valid DASD Device. Process Aborted.***

Explanation:

The function was not performed because the DASD device listed in the above message is not valid. The default is SYSALLDA.

User response:

Check the name of the DASD unit or enter a different DASD unit and try the restore again.

CUZS166E ***API program name-Func function code -Tape Unit unit name is not a valid Tape Device. Process Aborted.***

Explanation:

The function was not performed because the Tape device listed in the above message is not valid. The default is CART.

User response:

Check the name of the Tape unit or enter a different Tape unit and try the process again.

CUZS167E *API program name-Func function code -Restore Wait for Completion NE "Y" or "N. Process Aborted.*

Explanation:

The restore function was not performed because it expected a "Y" or "N" to indicate whether to wait for the restore to complete. An invalid character was specified.

User response:

Change the value to "Y" or "N" and try the restore process again.

CUZS168E *API program name-Func function code - Restore to DSN data set name Cataloged with Delete Old = "N". Process Aborted.*

Explanation:

The restore function was not performed because a value of "N" indicated that the restore process should not delete the old data set. A value of "Y" is required to delete the old data set.

User response:

Change the value to "Y" and try the restore process again.

CUZS169I *API program name-Func function code -DSN data set name Scheduled for Restore from Cloud cloud name.*

Explanation:

This message informs you that you have successfully scheduled the specified data set to be restored from the specified cloud.

User response:

No action is required.

CUZS170I *API program name-Func function code -DSN data set name Successfully copied to cloud cloud name.*

Explanation:

This message informs you that you have successfully copied the specified data set to the cloud.

User response:

No action is required.

CUZS171I *API program name-Func function code -DSN data set name*

Successfully Restored from cloud cloud name.

Explanation:

This message informs you that you have successfully restored this data set from the specified cloud.

User response:

No action is required.

CUZS172E *API program name-Func function code -DSN data set name Restore Failed from cloud cloud name.*

Explanation:

An attempt to restore this data set from the specified cloud was not successful.

User response:

Verify the data set name and the cloud name and try the process again. There also may be connectivity issues with the cloud at this time.

CUZS173E *API program name-Func function code -DDNAME Specified is not Allocated - Process Aborted.*

Explanation:

The DD name is not allocated so the function cannot be performed.

User response:

Either allocate the DD name or choose a different DD name that is already allocated, and try again.

CUZS174E *API program name-Func function code -DSN data set name Specified was not found in repository. Process Aborted.*

Explanation:

The data set name cannot be found in the repository. There is only one repository so that indicates that the data set name is either invalid or was not included as part of the repository.

User response:

Verify the spelling of the data set name or specify a different data set name from this repository.

CUZS175E *API program name-Func function code -DSN data set name-GEN generation number was not found in repository. Process Aborted.*

Explanation:

The specific generation (backup) of the data set could not be located in the repository.

User response:

Specify a different generation number that is a part of this repository.

CUZS176I *API program name-Func function code -DSN data set name was*

**successfully deleted from cloud
cloud name**

Explanation:

This message informs you this data set was deleted from the specified cloud.

User response:

No action is required.

CUZS177E **API program name-Func function
code -DSN data set name was not
found on cloud cloud name.**

Explanation:

The specified cloud does not contain this data set.

User response:

Verify that you have the correct data set name or try looking for the data set on a different cloud.

CUZS178E **API program name-Func function
code -An Error was encountered
deleting DSN data set name from
cloud cloud name.**

Explanation:

An error occurred while attempting to delete this data set from the specified cloud.

User response:

Verify that you have the correct data set name. Also check to ensure that there are no connectivity issues with this cloud.

CUZS179E **API program name-Func function
code -Error creating cloud list list.**

Explanation:

An error occurred while attempting to create a list of defined clouds.

User response:

Contact IBM Support for assistance, if needed.

CUZS180I **API program name-Func function
code -Cloud List Filter filter criteria
Returned data set name Dataset
(s)**

Explanation:

The data sets that match the filter criteria are listed in this message.

User response:

No action is required.

CUZS181E **module or API program name -
Func function code - Not
Authorized to Restore Dataset
data set name**

Explanation:

You do not have the type of RACF authority that allows you to allocate the restore data set.

User response:

Contact your System Administrator to be given the appropriate RACF authority.

CUZS182D **module name - Jobname job name
DCB data control block -
Processing Close**

Explanation:

This is a diagnostic message indicating that has control when closing a file.

User response:

CUZS183E **module name - Detected VSI
length incompatibility**

Explanation:

This message is issued by module *module name*. It indicates that the address space detected an anomaly in the control blocks necessary to facilitate Repository sharing. The length of the VSAM Shared Information (VSI) control block has changed since it was last recorded by Repository processing. This anomaly could lead to structural errors in the Repository if it goes unresolved.

User response:

Shutdown the address space and restart it. Contact IBM Technical Support if the problem persists.

CUZS184D **module name - Jobname job name
- Hooking Catalog Locate on DSN
data set name**

Explanation:

This is a diagnostic message indicating that intercepted a "catalog locate" on the DSN listed in the message.

User response:

CUZS185D **module name - Jobname job name
- Locate Failed on DSN data set
name**

Explanation:

This is a diagnostic message indicating that a "catalog locate" failed on the DSN listed in the message.

User response:

CUZS186D **module name - Jobname job name
Dataset data set name Not on the
Cloud**

Explanation:

This is a diagnostic message indicating that the data set listed in the message was not cataloged to the cloud (volser = cloud).

User response:

CUZS187I **module name - Jobname job name
Cloud Dataset data set name
Scheduled for Restore**

Explanation:

This message provides the name of a data set that was cataloged to the cloud and is now being restored back to z/OS.

User response:

CUZS188E *module name - Jobname job name*
Cloud Dataset data set name CTC
Started Task is not active -
Restore Cancelled

Explanation:

The data set listed in the messages was cataloged to the cloud, but the started task for was not started (not active) so the restore back to z/OS has been cancelled.

User response:

Specify the command to start . See Chapter 3, “Configuring Cloud Tape Connector,” on page 7 for more information on the started task.

CUZS189E *module name - Jobname job name*
Cloud Dataset data set name
Uncatalog Failed - Restore from
Cloud Cancelled

Explanation:

To restore a dataset to z/OS when it is cataloged to the cloud, must first uncatalog the data set from the cloud before attempting to restore it back to z/OS. In this case, the uncatalog process failed, so the restore was cancelled.

User response:

See “Catalog Support” on page 90 for more information.

CUZS190I *module name - Jobname job name*
Cloud Dataset data set name -
Restore from Cloud Successful

Explanation:

A data set that was cataloged to the cloud has been successfully restored back to z/OS.

User response:

CUZS191E *module name - Jobname job name*
Cloud Dataset data set name -
Restore from Cloud Failed

Explanation:

The dataset listed in the message is cataloged to the cloud. An attempt was made to restore this data set back to z/OS, but the restore failed.

User response:

See “Catalog Support” on page 90 for more information.

CUZS192I *module name - Jobname job name*
Cloud Dataset data set name -
Recataloged with VOLSER= CLOUD

Explanation:

This message provides the name of a data set that was uncataloged from z/OS and recataloged to the cloud.

User response:

See “Catalog Support” on page 90 for more information on cataloging data sets to the cloud.

CUZS193E *module name - Jobname job name*
Cloud Dataset data set name -
Recataloging to Cloud Failed

Explanation:

The data set listed in the message needed to be recataloged to the cloud, but the recatalog attempt failed.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS194D *module name - Jobname job name*
- Catalog Locate Invoked on DSN
data set name

Explanation:

This is a diagnostic message indicating that the catalog hook has been invoked on the DSN listed in the message.

User response:

See “Catalog Support” on page 90 for more information on cataloging data sets to the cloud.

CUZS195E *module name - Repository Cross*
System Sharing failed to initialize

Explanation:

This message is issued by module *module name*. It indicates that attempts to establish the Repository sharing environment failed. Additional messages indicate possible causes.

User response:

Restart the address space. Contact IBM Technical Support if the problem persists. Change the started task CUZCLOUD DD to DISP=OLD to prevent sharing the Repository until the problem is resolved.

CUZS196E *module name - Detected incorrect*
VSI environment

Explanation:

This message is issued by module *module name*. It indicates that the address space detected an anomaly in the control blocks necessary to facility Repository sharing. The address space could not verify the VSAM Shared Information (VSI) control block. This anomaly could lead to structural errors in the Repository if it goes unresolved.

User response:

Shutdown the address space and restart it. Contact IBM Technical Support if the problem persists.

CUZS197E ***module name - Error Freeing Disk
DSN data set name for
Recataloging to Cloud***

Explanation:

When recataloging to the cloud, if the data set is on disk, it must be allocated and deleted. The deleting of the original DSN failed.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS198E ***module name - Error Uncataloging
Tape DSN data set name for
Recataloging to Cloud***

Explanation:

An attempt was made to re-catalog this tape data set to the cloud. However, the uncataloging of the original DSN from tape has failed.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS199E ***module name - Error Re-Cataloging
DSN data set name for
Recataloging to Cloud***

Explanation:

An error was encountered when attempting to catalog a dataset to the cloud.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS200I ***module name - DSN data set name
has been successfully Re-
Cataloged to Cloud***

Explanation:

This message indicates that recataloging to the cloud was successful.

User response:

CUZS201E ***module name - Authorization
Failure on DSN data set name Re-
Cataloging to Cloud***

Explanation:

The user is not authorized to recatalog the specified dataset to the cloud.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS202E ***module name - Error Uncataloging
Disk DSN data set name for
Recataloging to Cloud***

Explanation:

An error occurred when attempting to allocate the specified disk data to be deleted and re-cataloged to the cloud.

User response:

See “Catalog Support” on page 90 for more information on re-cataloging data sets to the cloud.

CUZS203I ***Restore successful, U=userid, New
name data.set.name***

Explanation:

This message is issued when a user, listed as *userid*, requests the Restore be done with a new name, which is *data.set.name*, and that restore is successful.

User response:

CUZS212I ***module name - Func function DSN
data set name was successfully
Uncataloged from the Cloud***

Explanation:

When the last data set generation has been deleted from the cloud (either by module CUZ#EXPR or through ISPF, option 3, Delete line command), the data set will be uncataloged from the cloud. This message indicates that the specified data set was uncataloged successfully.

User response:

CUZS213E ***module name Func function name
DSN data set name Error
Uncataloging from the Cloud***

Explanation:

An error was encountered when attempting to uncatalog the specified data set from the cloud.

User response:

See “Catalog Support” on page 90 for more information.

CUZS214E ***module name Func function name
Catalog to Cloud not equal to "Y"
or "N". Process aborted***

Explanation:

The "Catalog to Cloud" field on the Cloud Filter Edit screen and the CATALOG_TO_CLOUD option in the sample "history include list" member, CUZJINCL, only accept "Y" or "N" as valid values. A different value was specified so the process has aborted.

User response:

Change the value to either "Y" or "N". See “Backup Filter Criteria” on page 48 for more information on this "Catalog to Cloud" field.

CUZS215I ***module name Func function name
Cloud cloud name Type cloud type
has a successful connection.***

Explanation:

This API message indicates the status of the cloud connection. In this instance, the named cloud and cloud type (HCP, FTP, etc.) has a successful connection.

User response:

**CUZS216I Repository Backup Bypassed-
Backups taken on another LPAR**

Explanation:

With the repository sharing feature, only one LPAR will do the repository backups if the same data set is used across LPARs. This message indicates that the repository backups are being taken on another LPAR and bypassed on this LPAR.

User response:

**CUZS217E *module name* Func *function name*
Retention Period is required for
this function. Process aborted.**

Explanation:

A retention period is required for a Restore function.

User response:

Specify a value between 1 - 9999 in the "Retention Period" field. See ["Backup Filter Criteria"](#) on page 48 for more information on this field.

**CUZS218E *module name* Func *function name*
Dataset *data set name* was not
found in the repository. Process
aborted.**

Explanation:

The Repository contains information about all of the data sets copied to clouds defined in . The data set *data set name* generation (x) being restored or deleted via the API function was not found in the repository so the function was not performed.

User response:

See Chapter 11, ["Cloud Tape Connector Repository,"](#) on page 115 for more detailed information.

**CUZS229E <variable> - DSORG of dataset
name dataset is not PS**

Explanation:

The specified dataset is not organized correctly. A staging data set must be organized as a physically sequential (PS) data set.

User response:

Change the organization of the specified dataset to PS.

**CUZS230E *module name*-Staging Dataset
Alias *alias dataset name* is invalid.
Cloud Connector terminated.**

Explanation

The *module name* is the program module that generated this message. The staging alias you are attempting to use is not valid. The staging alias MUST be a unique qualifier to be used only by . DO NOT use an alias being used by other users or products (TSOID).

Note: The stage alias is required even if staging has been turned off. This alias is also used for creating backups of the repository and copy them to the cloud. ended due to the bad dataset alias.

User response:

Correct the name of the staging alias in the Parmlib and restart the started task (STK) for . For more information on staging and the use of a dataset alias, refer to ["Method 2: Staging data to DASD"](#) on page 7. Refer to ["Staging Options"](#) on page 26 for more information on specifying a staging dataset alias in the Parmlib using ISPF menus.

**CUZS231E *module name*-DSN *data set name*
is not a staging dataset. The
dataset is retained. It's not
deleted.**

Explanation

The *module name* is the program module that generated this message. A staging file, *data set name*, was found without the correct naming standards, indicating that the staging dataset alias is invalid. The staging alias MUST be a unique qualifier to be used only by . DO NOT use an alias being used by other users or products (TSOID).

Note: The stage alias is required even if staging has been turned off. This alias is also used for creating backups of the repository and copying them to the cloud.

User response:

Specify a unique staging alias in the "Staging Dataset Alias" on the Parmlib Staging Options screen. For more information, refer to ["Staging Options"](#) on page 26.

**CUZS239I *module name* - z/OS DSN *dataset*
name copied to Cloud *cloud*
*dataset name***

Explanation

This message indicates that a dataset was copied successfully to a cloud. It displays the z/OS dataset name with the corresponding cloud dataset name.

System action:

The dataset was copied to the cloud.

User response:

No further action is required.

CUZS240W	<i>module_name dataset_name</i> Allocation failed on DASD, successful on tape	CUZS247I	<i>module_name - Dataset</i> <i>dataset_name - Is already in copy</i> to cloud queue
Explanation <p>This is a Warning message which informs you that the data set was restored on tape, but not on disk. The <i>module_name</i> is the module from which the message was issued, while <i>dataset_name</i> is the DSN of the dataset for which the restore was issued.</p>		Explanation <p>This message indicates that the dataset listed in the message is already in the copy-to-cloud queue and will not be added again to the queue by another history request.</p>	
User response: <p>Review whether any updates need to be done.</p>		User response: <p>No action is required.</p>	
CUZS243E	<i>module_name - Error Writing SMF</i> <i>Rectype record_type -</i> <i>RC(return_code) RS(reason_code)</i>	CUZS248E	<i>module_name-Share Options on</i> <i>Repository need to be (2,3). Run</i> <i>samplib mbr CUZJALTR</i>
Explanation: <p>In the module shown as <i>module_name</i>, an error occurred while attempting to write an SMF record. The <i>record_type</i>, <i>return_code</i>, and <i>reason_code</i> are provided to help you assess the problem.</p>		Explanation: <p>You need to run the alter repository job. The new SAMPLIB member, CUZJALTR, will change the share options from (4,3) to (2,3). When you see this message, the CUZCLOUD STC (started task) will not start.</p>	
User response: <p>Use the <i>return_code</i> and <i>reason_code</i> to try to determine why the error occurred. Also verify that there is enough space to hold the SMF records that are being written.</p>		User response: <p>Run the alter repository job, which is CUZJALTR in the SAMPLIB.</p>	
CUZS246I	<i>module_name - History Processing</i> <i>skipped until next interval. Other</i> <i>History Tasks Active.</i>	CUZS249E	<i>module_name-Check Parmlib</i> <i>Option Share Repository across</i> <i>LPARs needs to be "Y" for sharing</i>
Explanation: <p>This message informs you that history datasets are currently being copied to the cloud, and during this time another interval was triggered. When history tasks are already active, this work must complete before any new history include/exclude statements will be processed. History processing will run again only if the history queue is empty or it is submitted as a batch job. This message will appear if the history job has been started within the STC (started task) or the next interval has been triggered.</p>		Explanation: <p>This message is displayed when there is an open error on the repository. If you are sharing the same repository dataset on multiple LPARs, the STC (started task) will start on the first LPAR and will fail on all the other LPARs as long as the STC is still up on the first one. This message indicates you need to change the new parmlib option, Share Repository Across LPARs, to a "Y". If this option is not set to "Y", sharing is not allowed and z/OS will not allow the file to be opened on a second LPAR.</p>	
User response: <p>No action is required.</p>		User response: <p>On the Parmlib General Options screen (option 1.1 from the Main Menu), enter "Y" in the "Share Repository across LPARs" field.</p>	

Appendix E: List of VTE Messages

This appendix lists the messages generated by Virtual Tape Emulation (VTE), which is sometimes referred to as "VTFM" in the messages. A description of each message and suggestions on how you can resolve the problem are provided, where possible.

Many of the messages include a return code. For a description of these return codes, refer to the *IBM z/OS MVS Programming: Assembler Services Reference* (SA23-1372). You can find this book and many other publications on the IBM Knowledge Center:

<https://www.ibm.com/support/knowledgecenter/>

All of the messages have a severity code as the last character of the message ID. Severity codes are described below.

Table 33. Severity codes	
Code	Description
I	Information only. No operator action is required.
W	Warning message. The involved component continues its normal work. Results might not be as expected. The operator may have to take some action. Read the "User Response" text to determine the appropriate course of action.
E	Error. The involved component continues its normal work, however it stops handling the unit of work that was in progress at the time the error occurred. The operator may have to take some action. Some errors might be correctable. Read the "User Response" text to determine the appropriate course of action.
S	Severe message. The involved component may stop its normal work. The operator must take some action to overcome the problem. Read the "User Response" text to determine the appropriate course of action.

HVT001I VTF Mainframe VER *version* SERVER *serverid* INITIALIZATION

Explanation

This is an informative message issued by the VTE server during initialization. The following variables are in the message:

version

Indicates the VTE product version.

serverid

Specifies the VTE server ID, as specified in the IDENT initialization statement.

System action:

None

User response:

None

HVT003S ***SYNTAX ERROR ON ABOVE '*stat*' STATEMENT, RC=*rc* (*text*)

Explanation

During VTE server initialization, while refreshing of one of the refreshable members or a utility run, VTE detected a syntax error in the initialization statement (as listed in message HVT100I). The variables and return codes are explained below:

stat

The initialization statement name for which one of its parameters encountered the syntax error. If the *stat* value is the name of a specified parameter, one of this parameter's inner positional sub-parameters encountered the syntax error.

rc

The return code generated by this error. Possible values are:

12	Invalid initialization statement name. The initialization statement could not be classified as a recognized VTE initialization statement.
16	One or more mandatory keywords were not supplied.

	<ul style="list-style-type: none"> The text field lists up to two missing keywords. Missing inner positional sub-parameters are listed as the associated main keyword parameter followed by a sequence number.
20	Initialization statement contains redundant data, unrecognized or too many parameters.
28	Initialization statement is longer than 512 characters.

text

Short explanation of the error.

System action

The system action varies depending on when the error occurred:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response:

Correct the erroneous initialization statement, then restart the VTE server, retry the refresh or rerun the utility.

HVT004S *VALIDITY ERROR ON ABOVE STATEMENT, KWD '*kwd*'**

Explanation

During VTE server initialization, refreshing of one of the refreshable members or a utility run, VTE detected an unacceptable value in one or more keywords in an initialization statement (as listed in message HVT100I).

The only variable is *kwd*, which specifies the keyword on which the validation error was detected.

If the keyword's value contains multiple inner sub-parameters, and one of these sub-parameters has an unacceptable value, the erroneous subparameter is displayed as *parm#%*, where:

parm

Indicates the parameter name suffixed by #.

%

Indicates the inner sub-parameter sequence number.

System action

The system action varies depending on when the error occurred:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response:

Correct the erroneous initialization statement, then restart the VTE server, retry the refresh or rerun the utility.

HVT005S

INITIALIZATION PARAMETERS ERROR, RC=*rc*

Explanation

During VTE server initialization or a utility run, VTE encountered one or more errors. More detailed explanations about these errors precede this message.

The only variable in this message is *rc*, which refers to the return code generated by this error. Possible values are:

8	Initialization statement syntax error, validation error, essential initialization statement missing or duplicate initialization statement.
12	HVTSIN file could not be opened.
16	The maximum number of a multiple handled initialization statement was exceeded.

System action

The system action varies depending on when the error occurred:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.

User response:

Correct the condition that caused this message, then restart the VTE server or rerun the utility.

HVT006S

UCB LOOKUP ROUTINE FAILED, RC=*rc*

Explanation

During VTE server initialization, VTE detected that it could not obtain information about the virtual devices it handles or the virtual devices were not defined correctly.

The only variable in this message is *rc*, which refers to the return code generated by this error. Possible values are:

08	The maximum number of 256 virtual tape devices was exceeded.
24	The EDTINFO system service failed to supply the units defined under the VTE esoteric name (the ESOTERIC parameter in the VDEVICE initialization statement).
28	The UCBSCAN system service failed to supply information about the units defined under the VTE esoteric name (the ESOTERIC parameter in the VDEVICE initialization statement).
32	The EDTINFO system service failed to supply the device type associated with the VTE esoteric name (the ESOTERIC parameter in the VDEVICE initialization statement).
36	The device numbers defined to the VTE virtual tape devices were not contiguous.

System action:

The initialization phase fails and the VTE server shuts down.

User response

The recommended operator action depends on the return code:

08	Change the VTE esoteric name (defined in HCD) to contain up to 256 virtual devices.
24	Verify that the ESOTERIC parameter value defined on the VDEVICE initialization statement defines a valid value defined in HCD. Message HVT300S gives the EDTINFO system service return code. Try to locate the problem using that value. If the problem cannot be located, collect and forward the appropriate documentation to customer support for technical assistance.
28	Message HVT356S gives the UCBSCAN system service return code. Try to locate the problem using that value. If the problem cannot be located, collect and forward the appropriate documentation to customer support for technical assistance.

32	Message HVT300S gives the EDTINFO system service return code. Try to locate the problem using that value. If the problem cannot be located, collect and forward the appropriate documentation to customer support for technical assistance.
36	Make new HCD definitions for the VTE esoteric name, with a contiguous range of device numbers. When finished, ACTIVATE the new I/O configuration and restart the VTE server.

HVT007I

**serverid SERVER OBTAINED
X'amount' BYTES AT X'address'
(SP245)**

Explanation

This is an informative message issued by the VTE server during initialization. This message indicates that VTE successfully obtained its work area in ESQA (SP245). The following variables are in the message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

amount

A hexadecimal number indicating the number of bytes allocated in ESQA (SP245) to be used as a work area.

address

A hexadecimal number indicating the address in ESQA where the VTE server work area resides.

System action:

None

User response:

None

HVT008I

**serverid SERVER INTERFACE
ESTABLISHED AT X'address'**

Explanation

Informative message issued by the VTE server during initialization. This message indicates that VTE successfully established its I/O interface to MVS. The following variables are in the message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

address

A hexadecimal number indicating the address in fixed SQA where the VTE server I/O interface module resides.

System action:

None

User response:

None

**HVT009I INITIALIZATION OF *serverid*
SERVER SUCCESSFULLY
COMPLETED**

Explanation:

This is an informative message issued by the VTE server during initialization, which indicates that the VTE server successfully completed its initialization. The only variable in the message is the *serverid*, which is the VTE server ID, as specified in the IDENT initialization statement.

System action:

None

User response:

None

**HVT010E *oper* FAILED FOR DD=*ddname*
DSN=*dsn***

Explanation

The VTE server or a utility attempted to manipulate the DDname listed in the message, but due to an error this attempt failed. The most common cause of this failure is a missing DDname. The following variables are in the message:

oper

The attempted operation. Possible values are:

- **OPEN** – OPEN attempt
- **OPENJ** – OPEN TYPE=J attempt
- **CLOSE** – CLOSE attempt

ddname

The DDname for which the operation attempt failed.

dsn

The dsn name associated with the above DDname.

System action

The system action varies depending on when the error occurred:

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs when accessing a virtual tape dataset, the job using the virtual tape is EQC-ed.
- If the error occurs in a utility run, that utility terminates.

User response

If the error occurs when accessing a virtual tape dataset, the job using the virtual tape is EQC-ed.

OPEN or OPENJ

Verify why the DDname listed in the message could not be opened, fix the problem and, if needed, restart the VTE server.

CLOSE

Collect and forward the appropriate documentation to customer support for technical assistance.

**HVT011E STORAGE OBTAIN FAILED FOR
CONTROL BLOCK '*blk*'**

Explanation:

During regular work, the VTE server failed to GETMAIN storage for the listed control block. As a result, vital internal information could not be stored. The only variable in this message is *blk*, which is the control block name for which the GETMAIN failed. The name itself is not listed here and is for internal problem determination purposes.

System action:

The job using the virtual tape at the time of error is EQC-ed.

User response

The problem may occur due to unexpected storage accumulation. However, the VTE server shutdown clears this problem., so the VTE server bounce may relieve it.

Collect and forward the appropriate documentation to customer support for technical assistance.

**HVT013W *serverid* SERVER SHUTTING
DOWN**

Explanation

This is an informative message issued by the VTE server on shutdown (either normal or abnormal). The only variable in the message is the *serverid*, which is the VTE server ID, as specified in the IDENT initialization statement.

System action:

The VTE server shuts down.

User response:

None

**HVT015I *serverid* SERVER *type* (S) SCAN
COMPLETED AFTER POOL(S)
REFRESH**

Explanation

This is an informative message issued by the VTE server after pool(s) refresh, either on server startup or by explicit operator command. This message indicates that the subtask for monitoring disk free space or the subtask for monitoring scratch tapes successfully completed the diskpool(s) or tapepool(s) scan. The following variables are in the message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

type

The newly scanned pools are either DISKPOOLS or TAPEPOOLS.

System action

The system action depends on the value *type*:

DISKPOOL

VTE server issues a `F server, D=DSP` command to display the status of the diskpool(s).

TAPEPOOL

VTE server issues a `F server, D=TSP` command to display the status of the tapepool(s).

User response:

None

**HVT016S O/S RELEASE *rel* IS LOWER THAN
REQUIRED. MINIMUM
SUPPORTED RELEASE IS OS/390**

Explanation:

During VTE server initialization, VTE detected that the operating system release is lower than required. The minimum MVS release required by VTE is OS/390. The only variable for this message is *rel*, which is the operating system release as detected by the VTE server.

System action:

VTE initialization fails and the VTE server shuts down.

User response:

Report the problem to the system programmer. An MVS operating system upgrade may be necessary.

HVT019I *option* IS NOW *status*

Explanation

This is an informative message issued by the VTE server after the operator enabled or disabled one of the VTE options. It is issued as a result of one of the following operator commands:

- `F server, optionON`
- `F server, optionOFF`

The following variables are in the message:

option

Identifies the option being enabled/disabled.

status

Indicates if the named option is enabled or disabled.

For an explanation of VTE options, refer to [Chapter 31, “VTE Operation,”](#) on page 297.

System action:

The option is enabled or disabled, as required.

User response:

None

HVT020I TRACE (*type*) IS NOW *status*

Explanation

This is an informative message issued by the VTE server after the operator turned on or off one of the VTE internal traces. It is issued as a result of one of the following operator commands:

- `F server, typeON`
- `F server, typeOFF`

The following variables are in the message:

type

Identifies the type of trace being activated/deactivated.

status

Indicates if the trace is turned ON or OFF.

For an explanation of VTE internal traces, refer to [“Trace VTE Operation”](#) on page 311

System action:

None

User response:

None

HVT021E ENTERED *text* NOT RECOGNIZED

Explanation

One of the following occurred:

- A VTE operator command was not recognized by the VTE server.
- A valid VTE operator command containing an invalid parameter was issued.

The only variable in this message is *text*. Possible values are:

COMMAND

An invalid VTE operator command was issued.

REFRESH

An invalid request to refresh one of the refreshable members was issued.

CPU% VAL

An invalid value for the CPU% parameter was entered.

DISPLAY

An invalid value for the DISPLAY parameter was entered.

KILL

An invalid value for the KILL parameter was entered.

System action:

VTE ignores the invalid operator command.

User response

- If an invalid command was entered, reissue a valid VTE operator command containing valid parameters.
- If an invalid request to refresh one of the refreshable members was issued, reissue the request and make sure the member name suffix to be refreshed contains exactly two characters.
- If an invalid value for the CPU% parameter was entered, reissue the request and make sure the value is a number from 0 to 99.
- If an invalid value for the DISPLAY parameter was entered, reissue the command and make sure the value is:
 - A (or ACTIVE)
 - DSP (or DSPACE)
- If an invalid value for the KILL parameter was entered, reissue the command and make sure its value is CUU or TCB.

HVT022I **action ACCEPTED FOR**
VOLUME=volume (aliasvol),
CUU=cuu (poolname)

Explanation

This is an informative message appearing after the VTE server received an operator command referring to one of its virtual devices. The following variables are in this message:

action

MOUNTTAP indicates that VTE accepted a virtual tape mount on a given VTE virtual device.

volume

The mounted virtual tape volume serial number. For scratch volumes, this field appears as PRIVAT or SCRTCH.

aliasvol

If the mounted virtual tape is accessed through a PAT alias volume, this is its volume serial number. Otherwise, the volume is identical to the volume value.

cuu

The device number of the virtual device on which the mount was accepted.

poolname

The tapepool name requested by the TMS (or the default tapepool name if the TMS did not assign one).

System action:

None

User response:

None

HVT023I **VOLUME=volume (aliasvol)**
UNLOADED, CUU=cuu C=cpu_cp/
cpu_ziip/cpu_zicp, I=blki
(cin#_inm), O=blko (cout#_outm/
uout#_uoutm/perc%)

Explanation

This is an informative message issued by the VTE server when it unloads a virtual tape. The following variables are in this message:

volume

The volume serial number of the unloaded virtual tape.

aliasvol

If the unloaded virtual tape was accessed through a PAT alias volume, this is its volume serial number. Otherwise, the value is identical to the volume value.

cuu

The device number of the virtual device from which the volume was unloaded.

cpu_cp

CPU CP time (in seconds) used by the server's I/O subtasks to serve the virtual tape I/Os. This value represents the CPU time on CP processor(s) for workloads that are not zIIP-eligible.

This value does not include:

- SRB time (normally used by the server I/Os to disk).
- Other servers with tasks that are not directly related (i.e. disk space monitor, MCS console, etc.).

The total CPU time used by the server to serve the virtual tape I/Os is the sum of the following values:

- *cpu_cp*
- *cpu_ziip*
- *cpu_zicp*

For an uncompressed virtual tape, 5 to 10 percent should be added for the SRB time and overhead

related to other subtasks. For a compressed virtual tape, the above percent decreases with the compression ratio, that is, when the compression ratio is higher, the relative SRB overhead is lower.

cpu_ziip

CPU zIIP time (in seconds) used by the server's I/O subtasks to serve the virtual tape I/Os. This value represents the CPU time on zIIP processor(s) for a workload that is zIIP-eligible and was dispatched on zIIP processor(s).

The total CPU time used by the server to serve the virtual tape I/Os is the sum of the following values:

- SRB time (normally used by the server I/Os to disk).
- Other servers with tasks that are not directly related (i.e. disk space monitor, MCS console, etc.).

The total CPU time used by the server to serve the virtual tape I/Os is the sum of the following values:

- *cpu_cp*
- *cpu_ziip*
- *cpu_zicp*

cpu_zicp

CPU zIIP on CP time (in seconds) used by the server's I/O subtasks to serve the virtual tape I/Os. This value represents the CPU time for a workload that is zIIP-eligible, but as a result of various factors, it was dispatched on CP processor(s).

Several factors that may cause zIIP-eligible workload to be dispatched on CP processor(s) are:

- The defined percent of zIIP-eligible work to be actually offloaded to zIIP processor(s) is less than 100. This value is based on the ZIIP% parameter on the PERFORM initialization statement, or the parameter value changed through the server operator commands.
- The zIIP processors are fully utilized.
- There are no online zIIP processor(s) but PROJECTCPU=YES is specified.

The total CPU time used by the server to serve the virtual tape I/Os is the sum of the following values:

- *cpu_cp*
- *cpu_ziip*
- *cpu_zicp*

blki

Number of blocks that have been read (or 0). The "read" block count also includes any counts for tape marks.

cin#

Number of bytes that have been read (or 0), subject to the adjacent *inm* multiplicity factor. This number reflects the compressed (if applicable) value and is rounded up or down to the nearest integer. The "read" bytes count also includes any counts for tape marks.

inm

The *cin#* multiplicity factor. Valid values are:

- K - *cin#* is expressed in units of Kilo_bytes (1024).
- M - *cin#* is expressed in units of Mega_bytes (1048576).

blko

Number of blocks that have been written (or 0). The "written" block count also includes any counts for tape marks.

cout#

Number of bytes that have been written (or 0), subject to the adjacent *outm* multiplicity factor. This number reflects the compressed (if applicable) value and is rounded up or down to the nearest integer. The "written" bytes count also includes any counts for tape marks.

outm

The *cout#* multiplicity factor. Valid values are:

- K - *cout#* is expressed in units of Kilo_bytes (1024).
- M - *cout#* is expressed in units of Mega_bytes (1048576).

uout#

Number of bytes that have been written (or 0), subject to the adjacent *uoutm* multiplicity factor. This number reflects the *uncompressed* value and is rounded up or down to the nearest integer. The "written" bytes count also includes any counts for tape marks.

uoutm

The *uout#* multiplicity factor. Valid values are:

- K - *uout#* is expressed in units of Kilo_bytes (1024).
- M - *uout#* is expressed in units of Mega_bytes (1048576).

perc

Written data compression percent. This number is calculated only for the data written in the current job and does not include previous data that may exist on the virtual tape.

A value of 0 means 'no compression', while a value of 99 is the maximum possible compression.

To get the compression ratio, the following formula should be used:

```
ratio = 100 / (100-perc)
```

For example, a compression percent of 65% is equivalent to a compression ratio of 1:2.85.

System action:

None

User response:

None

HVT024E **MCS FUNCTION=*func* FAILED, RTNCODE=X'*rc*', RSNCODE=X'*rs*'**

Explanation

During VTE server initialization or normal message processing, the VTE extended MCS console detected an error. The following variables are in this message:

func

The type of function. Possible values are:

- ACTIVATE – The MCSOPER ACTIVATE system service failed to activate the VTE extended MCS console environment.
- GETMSG – The MCSOPMSG GETMSG system service failed to retrieve a console message.

rc

The hexadecimal return code returned by MCSOPER or MCSOPMSG system service.

rs

The hexadecimal reason code returned by MCSOPER or MCSOPMSG system service.

System action

- If the error occurs during the VTE extended MCS console activation phase, the VTE server shuts down.
- If the error occurs during retrieving of a console message, the VTE server bypasses the processing of the current message and continues to the next message.

User response

This error may be the result of an environmental condition (for example, a problem in the site's console environment). Try to locate the problem using the return and reason codes.

If the problem cannot be located, collect and forward the appropriate documentation to customer support for technical assistance.

HVT025I ***serverid* SERVER MCS CONSOLE TERMINATING**

Explanation:

This is an informative message which indicates that the VTE extended MCS console is terminating as a result of a VTE server shutdown. The only variable in this message is *serverid*, which is the VTE server ID, as specified in the IDENT initialization statement.

System action:

None

User response:

None

HVT026E **BAD MCS STATEMENT VALIDATION STRING *string*, RC=*rc***

Explanation

During processing of a console message, the internal validation routine detected that one of the VALIDATn parameters of the MCS initialization statement is not supported.

string

The value specified in the VALIDATn parameter of the MCS initialization statement.

rc

The return code generated by this error. Possible values are:

8	Validation parameter not recognized.
12	Validation string could not be classified.

System action:

Validation of the console message is bypassed. VTE may incorrectly interpret the console message.

User response

Enter the VTE HVTMCSxx initialization statements member and fix the invalid VALIDATn parameter values. Then bounce the VTE server.

If the problem cannot be located, collect and forward the appropriate documentation to customer support for technical assistance.

HVT027S *****ESSENTIAL INITIALIZATION STATEMENT MISSING - '*stmt*'**

Explanation:

During VTE server initialization, refreshing of one of the refreshable members or a utility run, VTE detected that an essential initialization statement was missing. The only variable in this message is *stmt*, which is the missing initialization statement name.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the

remaining initialization statements, then the VTE server shuts down.

- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response:

Add the missing initialization statement, then restart the VTE server, retry the refresh or rerun the utility.

HVT028S *DUPLICATE *stmt***

Explanation

During VTE server initialization, refreshing of one of the refreshable members or a utility run, VTE detected one of the following:

- An initialization statement was specified more than once.
- A tapepool name was specified more than once on two or more TPOOL initialization statements.
- A diskpool name was specified more than once on two or more DPOOL initialization statements.
- A vault locid was specified more than once on two or more VLTRMT or VLTOWN initialization statements.
- A RULE filtlist or PRULE filtlist name was specified more than once on two or more RFILT or PFILT utilization statements.

This message follows message HVT100I listing the duplicate initialization statement.

The only variable in this message is *stmt*, which can be any of the following values:

STATEMENT

A regular initialization statement has been specified more than once.

TPOOL

A TPOOL initialization statement NAME's value has been specified on more than one TPOOL initialization statement.

DPOOL

A DPOOL initialization statement NAME's value has been specified on more than one DPOOL initialization statement.

VLTRMT

A VLTRMT initialization statement LOCID's value has been specified on more than one VLTRMT or VLTOWN initialization statement.

RFILT

An RFILT or PFILT initialization statement NAME's value has been specified on more than one RFILT or PFILT initialization statement, respectively.

DPTRN

A DPTRN initialization statement NAME's value has been specified on more than one DPTRN initialization statement.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response

Delete the duplicate initialization statement, then restart the VTE server, retry the refresh or rerun the utility.

HVT029I HIGH CPU% UTILIZATION SET TO *nn%*

Explanation:

This is an informative message issued by the VTE server after the operator changed the high CPU% utilization to be used by the server. the only variable in this message is *nn*, which represents the new high CPU% utilization value used by the server.

System action:

If the current CPU% utilization is higher than the new value, the server will adjust it to be below that value. This adjustment will take approximately 15 seconds.

User response:

None

HVT030I *type* VAULT OPTION SET TO *nnnn*

Explanation

This is an informative message issued by the VTE server during initialization or after the operator changed one of the e-Vault components operation mode.

The following variables are listed in this message:

type

The e-Vault component whose operation mode is set at server initialization or changed. Valid values are:

- VLTRET – Failing e-Vault processes are retried.
- VLTSUS – The e-Vault processing activity is suspended or resumed.

nnnn

Specifies the value entered by the operator for the operator command.

System action:

The e-Vault component operation mode is changed according to the entered operator command.

User response:

No action is required.

HVT032S MAXIMUM NUMBER OF nnnn 'stmt' STATEMENTS EXCEEDED

Explanation

During VTE server initialization, refreshing of one of the refreshable members or a utility run, VTE detected that a multiple handled initialization statement has been supplied more times than allowed. The following variables are in this message:

nnnn

Maximum number of times the initialization statement can be supplied.

stmt

The name of the initialization statement.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response:

Decrease the number of initialization statements, then restart the VTE server, retry the refresh or rerun the utility.

HVT037I *serverid* SERVER: CUU=*cuu1*–*cuu2*, ES=*esoteric*, GN=*generic*, DT=X'*devtype*', S=*simdev*, CHP=(*chpid*)

Explanation

This is an informative message that is issued by the VTE server during initialization. This message gives details about the virtual devices it handles. The following variables are listed in this message:

serverid

The VTE serverid, as specified in the IDENT initialization statement.

cuu1

The device number of the first virtual device in the range of virtual devices served by the VTE server.

cuu2

The device number of the last virtual device in the range of virtual devices served by the VTE server.

esoteric

The esoteric name of the virtual devices served by the VTE server. This value is similar to the ESOTERIC parameter value of the VDEVICE initialization statement.

generic

The generic name of the virtual devices served by the VTE server. This value is similar to the appropriate GENERIC value specified in HVTX002 exit.

devtype

The hexadecimal UCB device type of the virtual devices served by the VTE server. The first byte is similar to the appropriate UCBTBYT1 value specified in HVTX002 exit. The other three bytes are similar to 3480, 3490, 3590 or 3592 UCB device types, respectively.

simdev

The simulated tape device. Currently, it is one of the following four tape device simulations: 3480, 3490, 3590, 3592.

chpid

The list of CHPIDs (either existing or non-existing) attached to the VTE control unit.

System action:

None

User response:

None

HVT038W VIRTUAL DEVICES ATTACHED BY CHPID(S) DEFINED THRU HCD, VCHPID VALUES IGNORED

Explanation:

During VTE server initialization, the VTE server detected that its virtual devices are attached to a virtual control unit and CHPIDs through HCD.

System action:

The server ignores the VCHPID parameter's value(s) and uses only the HCD defined CHPIDs. The initialization process continues.

User response:

If your site needs the virtual devices to be attached to a virtual control unit and CHPIDs through HCD, this message can be ignored. Otherwise, remove the virtual devices definitions from HCD. The recommended method is to generate the VTE virtual

devices without control unit and CHPIDs, and have the CHPIDs defined through the VCHPID parameter.

HVT039I **HW: TYPE=type, MODEL=model, SERIAL=serial**
SW: NAME=name, VERSION=ver.rel.mod

Explanation

This is an informative message issued by the VTE server during initialization. This message gives details about the running hardware processor and the operating system version. The following variables are in this message:

type

Four-byte processor type number. If asterisks are displayed, the type value could not be obtained.

model

Four-byte processor model number. If asterisks are displayed, the model value could not be obtained.

serial

Six-byte processor serial number. If asterisks are displayed, the serial value could not be obtained.

name

Operating system name

ver

Operating system version number

rel

Operating system release number

mod

Operating system modification level

System action:

None

User response:

None

HVT040E **SEVERE I/O ERROR ON CUU=cuu,RC=rc (text)**

Explanation

An error was encountered while passing an I/O request from a user address space to the VTE address space. The following variables are in this message:

cuu

Specifies the device number of the virtual device to which the I/O was directed.

rc

Displays the return code associated with this message. Possible values are:

01	The internal table used for the I/O requests transfer is exhausted. This may occur when
----	---

	insufficient entries have been initially prepared in this table or too many I/Os are sent by MVS to the virtual device at once.
02	An internal error occurred while trying to locate the job's address space cross-memory environment parameters.
03	The VTE server's I/O interface encountered a shortage in its internal save area tables.

text

Provides a short explanation of the error.

System action:

The I/Os are not passed to the VTE server. The job that uses the virtual device is EQCed.

User response:

Collect and forward the appropriate documentation to customer support for technical assistance.

HVT042E **FRONT-END RTN=name NOT RESTORED (X'ouraddr'/X'isvaddr')**

Explanation

During VTE server shutdown, the STARTIO front-end removal component detected that the environment to be restored does not match with the structure that existed at the time the front-end was created. A common reason for this situation is that a third-party product has front-ended itself to the same environment. The following variables are listed in this message:

name

Lists the name of the STARTIO routine that cannot be restored.

ouraddr

Shows the address of the original STARTIO routine, as recorded by the VTE server at startup time.

isvaddr

Shows the new address that seems to have been planted by a third-party product.

System action:

Shut down of the server continues. The VTE STARTIO routines are not removed from the system and become orphan (that is, every I/O passed to them is passed back to the system). No other impact should be experienced by the system.

User response:

Browse the common storage pointed to by isvaddr and try to identify to whom this storage belongs. Then collect and forward the appropriate documentation to customer support for technical assistance.

HVT044I **ZIIP=(support**
SUPPORTED,status), O_CPU=cpu#,
Z_CPU=ziip#, ZIIP%=percent%

Explanation

Informative message issued by the VTE server during:

- Initialization (when ZIIP=ENABLE was specified on the OPTIONS initialization statement).
- Internal zIIP support status change (through the server operator commands).

The following variables are listed in this message:

support

Valid values are:

blank

The operating system supports zIIP.

NOT

The operating system does not support zIIP.

status

Valid values are:

ENABLED

The VTE internal zIIP support is enabled.

PROJECT

The VTE internal zIIP support is enabled due to the PROJECTCPU=YES specification in IEAOPTxx. The host does not have an online zIIP processor.

DISABLED

The VTE internal zIIP support is disabled.

cpu#

Total number of online processors (CP and zIIP).

ziip#

Total number of online zIIP processor(s). If asterisks are displayed, the operating system does not support zIIP and the number of zIIP processors could not be determined.

percent

Percent of zIIP-eligible work to be actually offloaded to zIIP processor(s).

System action:

None

User response:

None

HVT092E **MAXIMUM NUMBER OF amount**
ALLOWED VIRTUAL TAPES
EXCEEDED

Explanation:

During VTE server initialization or when updating the VDB-in-Storage table, VTE detected that the number of virtual tapes in the VDB exceeds its internal buffer designed to store them. The only variable in this

message is *amount*, which represents the maximum number of supported virtual tapes.

System action

The VDB-in-Storage is not updated. The following consequences can be expected:

- The VTE subsystem may not reallocate all needed real device allocations to virtual devices.
- Certain e-Vaulted virtual tapes may remain in an interim e-Vault status.

Except for the above, VTE continues to function properly. The message is repeated every five seconds and as long as the number of virtual tapes in the VDB exceeds the maximum allowed number.

User response:

If some of the virtual tapes in the VDB are not in use, remove them using the HVTUTAP utility in the HVTUTAP member in the SAMPLIB library.

HVT100I **text**

Explanation:

General message used to list initialization statements that have been read. The only variable in this message is *text*, which is the initialization statement that was read.

System action:

None

User response:

None

HVT102E **R/PFILT=r/pfilt_name BLOCK**
EXCEEDS THE MAXIMUM NUMBER
OF nnnn ALLOWED R/PPTRN
PATTERNS

Explanation

VTE detected that a given RFILT or PFILT block has exceeded the number of allowed patterns. This occurred during one of the following:

- During VTE server initialization.
- Enabling the dynamic reallocation facility (due to the 'F server, DYRON' operator command).
- Enabling the PAT facility (due to the 'F server, PATON', 'F server, PTBON' or 'F server, PTDON' operator commands).
- Refreshing the HVTRULxx or HVTPLxx members.

The following variables are in the message:

r/pfilt_name

Displays the RFILT or PFILT block name.

nnnn

Specifies the RFILT or PFILT block name.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the server shuts down.
- If the error occurs during enabling of the dynamic reallocation facility or the PAT facility, the enable fails.
- If the error occurs during the refresh of the HVTRULxx or HVTPLxx members, the refresh fails and the old existing values remain.

User response:

Decrease the number of patterns in the RFILT or PFILT block, then restart the VTE server, retry the enable or retry the refresh.

HVT103E R/PFILT=*r/pfilt_name* NOT FOUND

Explanation

VTE detected that a RULE or PRULE initialization statement parameter value refers to an RFILT or PFILT block that could not be found in the HVTRULxx or HVTPLxx members, respectively. This occurred during one of the following:

- During VTE server initialization.
- Enabling the dynamic reallocation facility (due to the 'F server,DYRON' operator command).
- Enabling the PAT facility (due to the 'F server,PATON','F server,PTBON' or 'F server,PTDON' operator commands).
- Refreshing the HVTRULxx or HVTPLxx members.

The only variable in this message is *r/pfilt_name*, which represents the referred RFILT or PFILT block name that could not be found.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the server shuts down.
- If the error occurs during enabling of the dynamic reallocation facility or the PAT facility, the enable fails.
- If the error occurs during the refresh of the HVTRULxx or HVTPLxx members, the refresh fails and the old existing values remain.

User response:

Correct the invalid RFILT or PFILT reference or add an RFILT or PFILT block that can be referenced by the RULE or PRULE initialization statement parameter value, respectively. Then restart the VTE server, retry the enable or retry the refresh.

HVT110E CHANNEL COMMAND X'*ccw*' NOT SUPPORTED

Explanation:

The VTE server encountered a channel command that is not supported. The only variable in this message is *ccw*, which represents the unsupported channel command.

System action:

The channel command is rejected. The VTE server continues normal processing, if possible.

User response

If the problem is re-creatable, turn on the GTF trace (by issuing the 'F server,GTFON' operator command), recreate the problem, then turn off the trace (by issuing the 'F server,GTOFF' operator command).

Collect and forward the appropriate documentation to customer support for technical assistance.

HVT111E INTERNAL ERROR CODE=(*module/rc*), VOLUME=*volume* (text)

Explanation

During regular work, one of the VTE server components or a utility encountered an unexpected condition. Most causes to these conditions are internal undetected product errors.

The following variables are listed in the message:

module

The VTE module in which the internal error occurred.

rc

An internal return code classifying the internal error.

volume

The mounted virtual tape volume serial number at the time the internal error occurred. If asterisks are displayed, the value could not be determined.

text

Short explanation of the error.

System action:

In most cases, the job using the virtual tape that was served at the time the internal error occurred is EQC-ed (Equipment Checked).

User response:

Collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT112E CHANNEL PROTECTION CHECK ON CUU=*cuu*, A=X'*addr*' K=X'*progkey*' S=X'*storkey*' P=*type*

Explanation

During a user storage access attempt, the VTE server detected a Channel Protection Check condition that prevents it from accessing that storage.

The following variables are listed in the message:

cuu

Specifies the device number of the virtual device to which VTE performed the I/O that caused the Channel Protection Check condition.

addr

Lists the starting or ending address of the user storage that caused the Channel Protection Check condition.

progkey

Lists the user program key used while accessing the storage. The value is a one-byte field in which the key is represented by the four left-justified bits.

storkey

Specifies the user storage key. The value is a one-byte field in which the key is represented by the four left-justified bits and the fifth bit represents the fetch protection status.

type

The type of protection being used:

- F – The storage is fetch protected.
- S – The storage is store protected.

System action:

The VTE server suppresses the I/O. The returned channel status is set to x'..1.' indicating Channel Protection Check. Most likely the application will terminate the I/O request and fail the job.

User response:

Make sure you specified the right storage address in the CCW or IDAW and that the storage is accessible in the program key in which the program executes.

HVT115E **VOLUME=volume (jobname/jobid)**
SMF RECORD WRITE FAILED,
RC=rc

Explanation

During a virtual tape unload process, the VTE server detected an error while writing the user SMF record. This error usually occurs as a result of environmental problems. The following variables are in the message:

volume

The virtual tape volume serial number on behalf of which the record write was attempted.

jobname

The name of the job that unloaded the virtual tape.

jobid

The jobid of the job that unloaded the virtual tape.

rc

Return code values lower than 80 are the SMFWTM system service return codes.

The only other return code is 80, which signifies that the SMF record to be written contains a null header. The HVT383W message may precede HVT115E.

System action:

The user SMF record is not written. Job processing continues.

User response

According to the SMFWTM system service return code, you may take the following steps:

- Allow IEFU83 exit to write the user record.
- Allow the user record write in the SMFHVTxx member.
- If there is a permanent environmental problem, consider disabling the user record write by specifying SMF=DISABLE in the OPTIONS initialization statement.

HVT126E **SSI func ERROR FOR jobname/**
jobid, RC=rc, RS=rs

Explanation

The VTE server encountered an error while using the IEFSSREQ system service in order to inquire the JES subsystem about a job's status or to purge a job's sysout.

This occurred during one of the following:

- Release of the PAT alias volumes (after the job using them finished).
- Inquiry of the VTE server on the HVTUFTP utility execution (during a type3 or type4 e-Vault process execution).
- Inquiry of the HVTUFTP utility on the VTE server status (during a type3 or type4 e-Vault process execution).
- Purge of the HVTUFTP utility started task sysout (after a type3 or type4 e-Vault process completed successfully).

The following variables are in this message:

func

Possible values are:

- STATUS – IEFSSREQ system service was used to obtain the job status.
- SYSOUT – IEFSSREQ system service was used to purge the job sysout.

jobname

Provides the name of the job whose inquire or sysout purge failed.

jobid

The job ID whose inquire or sysout purge failed.

rc

The return code from the IEFSSREQ system service used to access the JES subsystem. These return codes are listed in the IEFSSOBH macro.

rs

The reason code from the JES subsystem. These reason codes are listed in the IEFSSCS macro.

System action

System action depends on the value of *func*:

STATUS

PAT alias volumes used by this job are not freed. This can lead to a further shortage of PAT alias volumes which in turn can cause the PAT facility to be implicitly disabled.

Type3 or type4 e-Vaulted virtual tapes remain in an interim status. The e-Vault process resumes as soon as the problem is fixed.

SYSOUT

The HVTUFTP utility started task sysout is not purged. If this problem occurs too often, it can lead to spool space shortage.

User response:

Try to locate the error using the subsystem return code and reason code. If the problem is not located, collect and forward the appropriate documentation to customer support for technical assistance.

**HVT131W VIRTUAL TAPE UNLOAD FROM
CUU=*cuu* – CORRESPONDING
MOUNT NOT YET PROCESSED**

Explanation

During a virtual tape unload process, the VTE server noted that the corresponding mount process has not yet been performed. This situation is typical to slow systems that run IEFBR14-like jobs which allocate virtual tapes and unload them immediately without performing any I/O. In these slow systems, the VTE extended MCS console may process the mount message after the virtual tape is unloaded and the job ends.

This message is to be regarded as a warning message only. The only impact on the system is a possible annoying HVT319W message that should be replied to by the operator.

The only variable in this message is *cuu*, which represents the device number of the virtual device from which the virtual tape was unloaded.

System action:

The server delayed the unload process by 16 seconds; however, the VTE extended MCS console still did not process the mount message. After these 16 seconds, the server continued with the unload process (as not to obstruct the normal jobs flow). If the mount process will be performed after this unload completion, the next job's virtual tape mount on this virtual device may expect several HVT313W messages followed by I/O errors before its mount completes successfully.

User response

Check why the VTE extended MCS console did not process the mount message within the predefined 16 seconds period.

If the problem is not located, collect and forward the appropriate documentation to customer support for technical assistance.

**HVT200I CTC/VTE VER *version* UTILITY
utility STARTED**

Explanation

This is an informative message indicating that a Cloud Tape Connector (CTC) or Virtual Tape Emulation (VTE) utility has started. The following variables are in this message:

version

Indicates the version of the utility.

utility

Specifies the name of the utility.

System action:

None

User response:

None

HVT201I UTILITY *utility* ENDED, RC=*rc*

Explanation

This is an informative message indicating that a Virtual Tape Emulation (VTE) utility has ended. The following variables are in this message:

utility

Shows the name of the utility.

rc

Displays the completion code (return code) for this utility.

System action:

None

User response:

None

None

HVT202I *type* IN VDB = *count1* *type* IN LIST
 = *count2*

Explanation

This is an informative message issued by the HVTUREP utility at the end of the produced HVTU001 report, detailing the number of processed entries. The following variables are listed in this message:

type

Specifies the type of processed entry. Possible values are:

ACTIVE

Number of active virtual tapes.

ALLOC_GB

Total number of gigabytes allocated to the virtual tape data sets. Migrated, deleted or vaulted virtual tape data sets are not counted in this total.

ENTRIES

Total number of virtual tapes.

ERROR

Number of virtual tapes that are in error.

READ

Number of virtual tapes that are being read

S_LABEL

Number of virtual tapes that are standard-label tapes.

SCRATCH

Number of scratch virtual tapes.

USED_GB

Total number of gigabytes actually used by the virtual tape data sets. Migrated, deleted or vaulted virtual tape data sets are not counted in this total.

WRITE

Number of virtual tapes that are being written.

count1

Number of processed virtual tapes in the whole VDB (regardless any limitation imposed by other utility parameters).

count2

Number of processed virtual tapes (considering only the listed virtual tapes).

System action:

None

User response:

None

HVT220I

**JOURNAL FILE ALLOCATED AND
FORMATTED WITH *blk* BLOCKS
(*rec* RECORDS/BLOCK)**

Explanation

This is an informative message issued by the HVTUJFR utility indicating that a new journal file has been allocated and formatted. The following variables are listed in this message:

blk

Indicates the number of blocks allocated for the journal file.

rec

Indicates the number of journal records to be contained in one allocated block.

System action:

None

User response:

None

HVT221S

BAD JOURNAL FILE, RC=*rc* (*text*)

Explanation

On a request to open the journal file for either a read or write operation, the journal verification routine detected an error. As a result, the open failed. The following variables are listed in the message:

rc

Specifies the return code generated by this issue. Possible values are:

01

The opened file is not a journal file.

02

The journal file format is bad or the HVTUJFR utility failed before format completion.

text

Provides a short explanation of the error.

System action

- If the error occurs during an attempt to open the journal file on behalf of a job, that job is EQC-ed.
- If the error occurs in a utility run, that utility terminates.

User response:

Verify that the file allocated to DDname HVTJRN is indeed the journal file. If needed, reformat the journal file using the HVTUJFR utility in the HVTUJFR member in the SAMPLIB library.

HVT222E

***serverid* SERVER JOURNAL FILE
FULL**

Explanation:

On a request to open the journal file or after a write operation, the journal verification routine detected that the journal file was full. As a result, no component (except the HVTUVBK utility) can function. The only variable in this message is *serverid*, which represents the The VTE server ID, as specified in the IDENT initialization statement.

System action

- If the error occurs during accessing the journal file on behalf of a job, that job is EQCed.
- If the error occurs in a utility run (except HVTUVBK), that utility terminates.

User response:

Use the HVTUVBK job (from the SAMPLIB library) to back up the VDB and reset the journal file.

HVT223W	<i>serverid</i> SERVER'S JOURNAL FILE EXCEEDS THRESHOLD, <i>perc%</i> UTILIZATION
----------------	--

Explanation

Percent of used journal records is greater than the threshold value specified on the THRESH parameter of the JRN initialization statement. The following variables are listed in the message:

serverid

Indicates the VTE server ID, as specified in the IDENT initialization statement.

perc%

Percent of used journal records. This number is calculated as:

$$perc = (counted / total) * 100$$

counted

Specifies the number of journal records counted from the last VDB backup to the last written journal record.

total

Specifies the total number of records in the journal file.

System action:

The message is repeated on every journal file open and on each journal record write until the VDB is backed up and the journal file reset.

User response:

Use the HVTUVBK job (from the SAMPLIB library) to back up the VDB and reset the journal file.

HVT224I	JOURNAL FILE MARKED WITH <i>mode</i> OF BKP INDICATOR (RBA=X'<i>rba</i>)
----------------	---

Explanation

This is an informative message issued by the HVTUVBK utility in the HVTUVBK member in the SAMPLIB library, indicating that the last written journal block, which was written at the time the VDB backup took place, has been marked. The following variables are listed in the message:

mode

Designates whether this is the START or END of the last written block.

START

The last written block is temporarily marked. This is done by the HVTUVBK utility when it runs before the actual VDB backup starts.

END

The last written block is permanently set. This is done by the HVTUVBK utility when it runs after the actual VDB backup ends.

rba

Identifies the internal Relative Block Address (RBA) of the marked block.

System action:

None

User response:

None

HVT226I	VOLUME=<i>volume</i> RECORD SKIPPED, UPDATED LATER THAN THE RESPECTIVE JOURNAL RECORD (<i>date/time</i>)
----------------	---

Explanation

This is the informative message issued by the HVTUJRC utility in the HVTUJRC member in the SAMPLIB library, indicating that a specific journal record is already up to date in the VDB and does not need to be recovered. This situation is normal when a VDB recovery is based on a journal file containing records older than the last VDB update.

volume

The virtual tape volume serial number whose VDB record was just read from the journal file. If volume=*CNTL*, the record describes the VDB control record.

date

Date (0CYDDDDF) this record was written to the journal file.

time

Time (HHMMSSSTH) this record was written to the journal file.

System action:

The record is skipped and the next journal record is read.

User response:

None

HVT227S **VDB UNCOVERED PERIOD, LAST
VDB UPDATE=*date1/time1*
OLDEST JRN REC=*date2/time2***

Explanation

During VTE server initialization or a utility run, VTE detected that the oldest journal record is newer than the last updated VDB record.

This situation indicates that the VDB was restored from an old backup and a VDB recovery is required, but the journal file doesn't contain all the data required for the recovery (there is an uncovered period of time between the latest VDB record and the journal file).

This problem may have been caused by restoring a very old VDB file while the corresponding journal file records have already been overwritten.

The following variables are listed in the message:

date1

Last date (OCYYDDDF) the VDB was updated.

time1

Last time (HHMSSTH) the VDB was updated.

date2

Date (OCYYDDDF) of the oldest journal record.

time2

Time (HHMSSTH) of the oldest journal record.

System action

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.

User response

If a newer VDB file is available or an older journal file is available, perform the following steps:

- Restore a newer VDB file, then recover it using HVTUJRC utility in the HVTUJRC member in the SAMPLIB library.
- Restore an older journal file and:
 - Recover the VDB from it.
 - Recover the VDB from the latest journal file.

When finished, restart the VTE server or rerun the utility.

If a newer VDB file is not available or an older journal file is not available, perform the following steps:

- Perform full VDB recovery using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
- Define a new journal file using the HVTUJFR utility in the HVTUJFR member in the SAMPLIB library.

When finished, restart the VTE server or rerun the utility.

HVT228W **JOURNAL FILE IS EMPTY**

Explanation:

During an attempt to recover the VDB using the HVTUJRC utility in the HVTUJRC member in the SAMPLIB library, an empty journal file was detected.

System action:

The VDB recovery using the journal file is not performed.

User response:

Check why the journal file is empty. If the journal facility was disabled and no record was written to it, you may ignore this message.

HVT229W **VOLUME=*volume (date/time)*
DELETE FAILED, NOT FOUND IN
VDB**

Explanation

The HVTUJRC utility's attempt to delete a VDB record failed.

The requested record was not found in the VDB. Since HVTUJRC started recovery using journal records whose timestamp is a bit lower than the last VDB updated timestamp, some VDB records may already be updated. Because of this, the utility may try to delete an already deleted VDB record.

The above case is normal when the record's timestamp is earlier than the first recovered record timestamp (according to the first HVT230I message issued by the HVTUJRC utility run).

volume

The virtual tape volume serial number whose journal record could not be deleted.

date

Record date stamp (OCYYDDDF)

time

Record time stamp (HHMSSTH)

System action:

None

User response:

No action is required if the record timestamp is earlier than the first recovered record timestamp. Otherwise, collect and forward the appropriate documentation to customer support for technical assistance.

HVT230I	VOLUME=volume (date/time) action
Explanation Informative message issued by the HVTUJRC utility run for each processed record. The following variables are listed in this message:	
volume The virtual tape volume serial number whose record has just been processed. If volume=*CNTL*, the record describes the VDB control record.	
date Record's date stamp (0CYYDDDF)	
time Record's time stamp (HHMMSSSTH)	
action Specifies the action that occurred. Possible values are:	
ADDED The record has been added to the VDB.	
DELETED The record has been deleted from the VDB.	
UPDATED The record has been updated in the VDB.	
System action: None	
User response: None	

HVT231S	VDB LAST UPD TIME (date1/time1) LOWER THAN JRN (date2/time2), UJRC REQUIRED
Explanation During VTE server initialization or a utility run, VTE detected that the VDB and journal file timestamps do not coincide. This situation may be a result of restoring an old VDB without recovering its entries according to the journal file. The following variables are listed in the message:	
date1 Last date (0CYYDDDF) the VDB was updated.	
time1 Last time (HHMMSSSTH) the VDB was updated.	
date2 Last date (0CYYDDDF) the journal file was updated.	
time2 Last time (HHMMSSSTH) the journal file was updated.	

System action
<ul style="list-style-type: none"> • If the error occurs during VTE server initialization, the VTE server shuts down. • If the error occurs in a utility run, that utility terminates.
User response: Recover the VDB file using the HVTUJRC utility in the HVTUJRC member in the SAMPLIB library, then restart the VTE server or rerun the utility.

HVT232S	EMPTY VDB DETECTED, JOURNAL FILE MUST BE LOADED INTO IT
Explanation During VTE server initialization or a utility run, VTE detected that the VDB was empty while its corresponding journal file contained at least one record. This situation may be a result of creating a new VDB without recovering its entries according to the journal file.	
System action <ul style="list-style-type: none"> • If the error occurs during VTE server initialization, the VTE server shuts down. • If the error occurs in a utility run, that utility terminates. 	
User response Perform one of the following:	

<ul style="list-style-type: none"> • Restore the VDB from a backup and recover the VDB using the HVTUJRC utility in the HVTUJRC member in the SAMPLIB library. • If a backup of the VDB is not available, perform full VDB recovery using the HVTURCV utility in the HVTURCV member in the SAMPLIB library, and define a new journal file using the HVTUJFR utility in the UJFR member in the SAMPLIB library. 	When finished, restart the VTE server or rerun the utility.
--	---

HVT233S	serverid SERVER VDB DEFINED WITH INVALID PARAMETERS, RC=rc (text)
Explanation During VTE server initialization or a utility run, VTE detected that the VDB was defined with invalid parameters. The following variables are listed in the message:	

serverid

Provides the VTE server ID, as specified in the IDENT initialization statement.

rc

The return code generated by the issue. Possible values are:

01

The VDB has a wrong key length.

02

The key offset in the VDB is wrong.

03

The VDB record length is wrong.

04

The VDB is in Read_Only mode.

05

The VDB is not empty.

text

Provides a short explanation of the error.

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.

User response

The recommended action depends on the return code. See below for details.

Return codes 01, 02, 03

List the VDB file using the IDCAMS utility and compare its defined options with the options coded in the HVTUVSI job from the SAMPLIB library.

- If the VDB is empty, delete it and rerun the HVTUVSI job to redefine it properly.
- If the VDB is not empty, delete it and rerun the HVTUVSI job to redefine it properly. Then do one of the following:
 - Restore an older VDB backup and recover it from the journal file using the HVTUJRC utility in the SAMPLIB library.
 - Run the HVTURCV utility in the SAMPLIB library to recover all your virtual tapes.

Return code 04

Enter the HVTOPTx member in the SAMPLIB library, locate the VDB initialization statement and modify it to RMODE=RW.

When done, restart the VTE server or rerun the utility.

Return code 05

Probably user error. The HVTUVSI utility cannot initialize a non-empty VDB.

If the VDB should be initialized, use the HVTUVSI job in the SAMPLIB library to define a new VDB and to initialize it.

HVT240S**REQUESTED *type (typename)* NOT DEFINED FOR *serverid* SERVER****Explanation**

This message is generated for a couple of different reasons, which are described below.

- During VTE server initialization or when refreshing the HVTPOOLxx or HVTTRLxx members, VTE detected one of the following:
 - One or more TRULE initialization statement(s) did not specify existing TPOOL initialization statement(s).
 - One or more TPOOL initialization statement(s) did not specify existing DPOOL initialization statement(s).
 - One or more TPOOL initialization statement(s) LOCID value did not specify existing and valid VLTRMT initialization statement(s) or these VLTRMT initialization statements are not of type3.
- During a mount request of a scratch virtual tape, VTE detected that the requested tapepool was not found in any of the TPOOL initialization statements.

The following variables are used in this message:

type

The non-existent type is either a DISKPOOL or a TAPEPOOL, or is a "type3" remote off-site storage location (LOCID).

typename

Provides the name of the diskpool, tapepool, or the "type3" remote off-site storage location.

serverid

Provides the VTE server ID, as specified in the IDENT initialization statement.

System action

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs during the refresh of the HVTPOOLxx or HVTTRLxx members, the refresh fails and the old existing values remain.
- If the error occurred as a result of an attempt to locate a tapepool for a mounted virtual tape, that job is EQC-ed (Equipment Checked).

User response:

Check the involved initialization statements validity and fix them, then either restart the VTE server, retry the refresh or rerun the job.

HVT241S TAPEPOOLS RANGES OVERLAP
(poolname1/poolname2)

Explanation

During VTE server initialization or refreshing of the HVTPOLxx member, VTE detected that two or more ranges specified in the RANGEn parameter of the TPOOL initialization statements overlap. The following variables are listed in this message:

poolname 1

The name of the TPOOL initialization statement on which the first overlapping range has been detected.

poolname2

The name of the TPOOL initialization statement on which the second overlapping range has been detected.

System action

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs during the refresh of the HVTPOLxx member, the refresh fails and the old existing values remain.

User response:

Locate the respective TPOOL initialization statements and fix the RANGEn parameters values, then restart the VTE server or retry the refresh.

HVT242E VOLUME=volume NOT IN
REQUESTED TAPEPOOL
(poolname)

Explanation

As a result of message HVT317W requesting the operator to enter a new virtual tape volume serial number from a specific tapepool, the operator entered a virtual tape volume serial number from a different tapepool. The following variables are in this message:

volume

The virtual tape volume serial number entered by the operator.

poolname

The tapepool name whose range(s) should contain the entered virtual tape volume serial number.

System action:

The VTE server rejects the entered volume serial number and reissues message HVT317W.

User response

- Enter member HVTPOLxx in the SAMPLIB library and see what are the valid ranges for the given tapepool. Then reply again to message HVT317W, specifying a volume serial number from the correct tapepool.
- If all the volume serial numbers for the given tapepool exist already in the VDB enlarge an existing range (or define a new range of volume serial numbers) in the given tapepool. Then reply again to message HVT317W, specifying a volume serial number that has just been defined.

HVT243I VOLUME=volume RESET TO
TAPEPOOL=poolname

Explanation

This is an informative message issued by VTE server indicating either of the following:

- A scratch virtual tape with no associated tapepool has been mounted. However, as a result of the tapepool rules decision, its tapepool has been reset to a specific tapepool.
- A specific virtual tape has been mounted. In order to associate to it a diskpool, VTE assigns a tapepool according to its volume serial number range definitions.

The following variables are listed in this message:

volume

The mounted virtual tape volume serial number.

poolname

The tapepool name assigned to the mounted virtual tape.

System action:

None

User response:

None

HVT244I VOLUME=volume FORCED TO
DISKPOOL=poolname

Explanation

An existing virtual tape dataset to be allocated was found to reside on disk(s) not accessible with the UNIT parameter values of the DPOOL initialization statement pointed to by the appropriate tapepool (and listed in the preceding HVT315I message). VTE located another DPOOL initialization statement whose UNIT parameter values can be used to allocate the existing virtual tape dataset, and performed the allocation using these new values.

The need to force the diskpool may be a result of either of the following:

- The virtual tape dataset has been moved (by an independent utility) to another disk(s) included in another esoteric name or Storage Group.
- Modifications to the tapepool/diskpool references have been made in the HVTPOOLxx member so that VTE attempts to allocate existing virtual tape data sets using UNIT parameter values other than the values with which they were originally allocated.

The following variables are listed in this message:

volume

The mounted virtual tape volume serial number.

poolname

The name of the new diskpool forced to the mounted virtual tape.

System action:

The virtual tape dataset is allocated using the UNIT parameter values of the new diskpool (rather than the one listed in message HVT315I).

User response

- If the virtual tape dataset has been moved by an independent utility without your explicit specification (like an HSM recall), you may lose the tapepool/diskpool dependencies. In that case, you may instruct that utility to restore/recall the virtual tape data sets to their original unit.
- If you are aware that the virtual tape dataset has been moved to a different esoteric name or Storage Group from the one on which it had originally been allocated, you may ignore this message.

HVT246I ***tm-prod status, MVS MOUNT
MESSAGES WILL BE FURTHER
disp***

Explanation

Certain tape management systems that issue their own mount messages, do not suppress the equivalent MVS messages. VTE detects this situation and sets itself to ignore redundant mount messages. The detection is done at the following points:

- VTE is started after the tape management system is up. With the first tape management system mount message received by VTE, it detects the mount messages redundancy.
- The tape management system is brought up or down while VTE is up and functioning. VTE detects the modification in the tape management system status and sets itself accordingly.

In either case, VTE will ignore redundant mount messages or consider MVS mount messages, respectively.

The following variables are listed in this message:

tm-prod

Specifies the type of tape management system that is being handled. Valid types are CA-1 or TLMS

status

Indicates whether the current status of the tape management system, as identified by VTE, is now ACTIVE or INACTIVE.

disp

Indicates whether VTE will consider or ignore the MVS mount messages. Valid values are CONSIDERED or IGNORED.

System action

One of the following occurs:

- VTE ignores MVS mount messages replaced by the tape management system mount messages.
- VTE considers only the MVS mount messages.

User response:

None

HVT247S **DEFAULT TAPEPOOL NOT
DETECTED**

Explanation:

During VTE server initialization or refreshing of the HVTPOOLxx member, VTE could not detect the default tapepool definition. The default tapepool is defined as the TPOOL initialization statement whose RANGE1 value is (0,0).

System action

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of the HVTPOOLxx member, the refresh fails and the old existing values remain.

User response

Make sure that one of the TPOOL initialization statements contains a RANGE1=(0,0) parameter.

If you are a DFSMSrmm user, make sure that its TPOOL's NAME parameter value is similar to the default tapepool NAME value defined on the default DFSMSrmm VLPOOL initialization statement.

After fixing the TPOOL initialization statement, restart the VTE server or retry the refresh.

HVT249E **VOLUME=volume FUNCTION=func
ERROR, TYPE=type, RC=rc**

Explanation

An error was encountered by the VTE server or one of its utilities, during an attempt to perform one of the following:

- Inquire about certain compression type availability.
- Compress a data block to be written.
- Expand a data block just read.
- Build a compression dictionary.

The following variables are listed in the message:

volume

The virtual tape volume serial number whose data compression or expansion process encountered the error. If volume=*****, the volume serial number could not be determined.

func

The function can be any of the following values:

QUERY

The error occurred on VTE server or utility initialization while the RLE compress or expand facility was being tested for availability.

COMPRESS

The error occurred on an attempt to compress data.

EXPAND

The error occurred on an attempt to expand data.

BDIC

The error occurred on an attempt to build the compression dictionary.

type

01

The error occurred on an attempt to compress, expand or query using the RLE algorithm.

02

The error occurred on an attempt to compress or expand using the RLEfs algorithm.

03

The error occurred on an attempt to compress, expand or build a compression dictionary using the Lempel-Ziv algorithm.

rc

The value of the return code depends on the values of *func* and *type*.

If *type* = 01, this is the return code from the CSRCEsRV COMPRESS / EXPAND / QUERY system service, respectively. In addition, the following return codes are expected:

84

Invalid compress type

88

Invalid compress level

If *type* = 02 and *func* = COMPRESS, the following return codes are expected:

08

Unexpected return code from the internal RLEfs routine.

84

Invalid compress type

88

Invalid compress level

If *type* = 02 and *func* = EXPAND, the following return codes are expected:

08

Unexpected return code from the internal RLEfs routine.

16

Incorrect expanded string length

84

Invalid expand type

If *type* = 03 and *func* = BDICT, the following return codes are expected:

08

Error in the attempt to imbed the compression dictionary within the virtual tape dataset blocks.

12

Error when building the compression dictionary or a corresponding compression dictionary could not be built in the calling job address space.

Additional messages may be written on the SYSTSPRT sysout dataset.

If *type* = 03 and *func* = COMPRESS or EXPAND, this is the return code from the CSRCEsRV COMPRESS or EXPAND system service, respectively.

In addition, the following return codes are expected:

12

The CSRCEsRV system service abended.

84

Invalid expand type

88

Invalid compress level

System action

The system action depends on the function:

QUERY

RLE compression/expansion service is not available. Any attempt to compress data using the

RLE algorithm will fail and the data will be written uncompressed. Expansion of read data blocks is likely to fail.

COMPRESS/BDICT

The current data block that encountered the problem and all successive blocks of the current virtual tape will be written without compression.

EXPAND

The job whose data blocks cannot be expanded is EQC-ed. If the error occurs in a utility, various tape label type problems may occur when later accessing this virtual tape.

User response

The recommended operator action depends on the function:

QUERY

This may not be a problem as long as your processor does not support the RLE compress/expand feature.

COMPRESS/ EXPAND/BDICT

Investigate the problem according to the return code. If needed, collect and forward the appropriate documentation to customer support.

HVT250I **VOLUME=volume ELIGIBLE FOR
COMPRESSION, TYPE=type/level**

Explanation

This is an informative message issued by the VTE server during a virtual tape mount, indicating that the virtual tape dataset is eligible for compression. The message is not reissued when further job steps use virtual tapes passed to them by previous steps without an intervening unload/mount.

The following variables are listed in this message:

volume

The virtual tape volume serial number that is eligible for compression.

type

The compression type used for that virtual tape.

level

The compression level used for that virtual tape.

System action:

None

User response:

None

HVT260I **exit EXIT func comp, RC=X'rc',
RS=X'rs'**

Explanation

This is an informative message issued by the VTE server during initialization or as a result of activating or

deactivating the VTE subsystem (due to the 'F server, SSNON/SSNOFF' operator command). The message provides details about the special MVS IEFUJV and IEFDB401 exits that are dynamically added or deleted using the CSVDYNEX system service.

The following variables are listed in the message:

exit

This variable lists the type of exit that is being used.

SYS.IEFUJV or SYSSTC.IEFUJV

Specifies the MVS IEFUJV exit that is dynamically added or deleted.

SYSJES3.IEFUJV

This exit is used only in JES3 environments.

IEFDB401

This MVS exit is dynamically added or deleted.

func

This variable lists the type of function that is being used:

ADD

The exit is dynamically added to MVS.

DELETE

The exit is dynamically deleted from MVS.

comp

This variable lists the result that occurred when the function was performed:

WAS OK

The addition/deletion of the exit was OK.

FAILED

The addition/deletion of the exit failed.

rc

The hexadecimal return code from the MVS CSVDYNEX system service.

rs

The hexadecimal reason code from the MVS CSVDYNEX system service.

System action

If *comp*=WAS OK and *func*=ADD, the VTE subsystem is successfully activated.

For all other values of *comp* and *func*, the VTE subsystem deactivation is attempted. If the failure occurs during the VTE server's initialization phase, the VTE server shuts down.

User response

If the VTE subsystem activation fails due to a problem with the dynamic addition of the exit, try to locate the problem according to the CSVDYNEX system service *return code* and *reason code*.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT261E **serverid SERVER IEFUJV EXIT
VALIDATION ERROR
JOBNAME=jobname RC=rc**

Explanation

The IEFUJV exit (that is part of the VTE subsystem in JES3 environments) detected an error while analyzing and validating a job's requirements for dynamic reallocation or PAT eligibility. The message is issued only on the console (on behalf of the global processor or the C/I FSS) and does not appear in the job's log or VTE server's log.

The following variables are listed in the message:

serverid

This variable shows the VTE serverid (as specified in the IDENT initialization statement).

jobname

This variable lists the job name whose validation encountered the error.

rc

This variable lists the return code that was generated. A return code of 28 or 32 is related to PAT eligibility, and can be used for internal debugging purposes. All other return codes are related to dynamic reallocation.

System action

The system action depends on the return code:

28 or 32

The job is not tested for PAT eligibility and continues its normal processing.

Other

No dynamic reallocation is attempted for the job.

No 'defer' is forced for mounts on the VTE server's virtual devices. If the virtual devices are JES3-managed, MDS tries to mount the virtual tape (after issuing the IAT5210 message). Since the mount is not satisfied by VTE, MDS places the job on the Verify Queue where it waits forever, leaving the virtual device allocated.

User response

The recommended operator action depends on the return code:

28 or 32

None

Other

- Cancel the job so that it does not allocate the virtual device.

- For a temporary solution to the problem, explicitly specify an esoteric name (along with 'defer') in the job's JCL.

Collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT264I **jobname/jobid I/O PURGED,
CUU=cuu OPERATOR REPLY
DELETED**

Explanation

An application or MVS issued a PURGE for an I/O performed on a virtual device. As the PURGE request is processed by the VTE server, it detected that the job is waiting for an operator reply to complete the pending I/O. In order to finish the PURGE processing, the operator reply is deleted, and the pending virtual tape I/O is allowed to complete.

The following variables are used in the message:

jobname

The jobname of the job whose I/O is to be purged. If asterisks are displayed, the jobname could not be determined.

jobid

The jobid of the job whose I/O is to be purged. If asterisks are displayed, the jobid could not be determined.

cuu

The device number of the virtual device on which the I/O is pending.

System action:

The operator reply is deleted and the pending I/O is allowed to complete.

User response:

None

HVT265W **jobname/jobid DOING I/O, PURGE
PENDING, CUU=cuu – module/
comp**

Explanation

An application or MVS issued a PURGE for an I/O performed on a virtual device. As the PURGE request is processed by the VTE server, it detected that a virtual tape I/O command cannot be completed for a predefined amount of time. This message is repeated every 30 seconds and as long as the tape I/O is not completed. The message is displayed in the job's log and is replicated to the server's log.

There are several possible reasons for this problem:

- Heavy CPU load and a low VTE server priority.

- Very long command chain CCWs and a low VTE server priority.
- The disk that contains the virtual tape dataset or VDB is reserved or not responding. Thus, the VTE server cannot complete its I/O on the disk and cannot respond to the virtual tape I/O.
- A write is attempted from one system while the same virtual tape is read from another system.
- A compression dictionary is being built.
- High contention on the VDB.

The following variables are used in the message:

jobname

The jobname of the job whose I/O is to be purged. If asterisks are displayed, the jobname could not be determined.

jobid

The jobid of the job whose I/O is to be purged. If asterisks are displayed, the jobid could not be determined.

cuu

The device number of the virtual device on which the I/O is pending.

module

The server module that seems to have control when the PURGE pending condition was detected.

comp

The server component that seems to be stuck and prevents the current tape I/O to complete. Possible values are:

DSN-ALC

Allocation of a virtual tape dataset.

DSN-DAL

De-allocation of a virtual tape dataset.

DSN-OPN

Open of a virtual tape dataset.

DSN-CLS

Close of a virtual tape dataset.

VDB-RSV

RESERVE of the VDB.

VDB-OPN

Open the VDB.

VDB-CLS

Close the VDB.

VDB-CLT

Close TYPE=T the VDB.

VDB-PNT

Point to a record in the VDB.

VDB-GET

Get a record from the VDB.

VDB-PUT

Put a record in the VDB.

VDB-ERS

Erase a record from the VDB.

JRN-OPN

Open the journal file.

JRN-CLS

Close the journal file.

JRN-WRT

Write a record in the journal file.

CAT-CAT

Catalog a virtual tape dataset.

CAT-LOC

LOCATE a virtual tape dataset using the LOCATE system service.

CAT-CSI

LOCATE a virtual tape dataset using the IGGCSI00 system service.

DSK-OBT

OBTAIN a virtual tape dataset DSCB.

GRS-GQS

Inquire GRS using the GQSCAN system service.

XIO-cuu

Read/write a full track in a virtual tape dataset, where *cuu* is the disk device number on which the read/write is attempted.

blank

If blanks are displayed, no specific component was detected to have caused the PURGE delay.

System action:

The message is issued every 30 seconds for as long as the tape I/O is executed by the VTE server.

User response:

Check why the VTE server cannot complete its I/O to the disk and cannot respond to the virtual tape I/O, then take the appropriate actions. If needed, collect and forward the appropriate documentation to customer support for technical assistance.

HVT266I

***manager* MANAGER *func comp*,
RC=*rc* (X'*addrtn*)**

Explanation

This is an informative message issued by the VTE server during initialization, termination, or routine work. This message contains the ADD or DELETE details for either a VTE purger manager or a VTE ENF manager

The following variables are used in the message:

manager

Indicates the type of VTE manager (PURGER or ENF) that was added or deleted.

func

Indicates that the VTE purger or ENF manager been added (ADD) to MVS or deleted (DELETE) from MVS.

comp

Indicates that the addition or deletion of the purger or ENF manager was either successful (WAS OK) or failed (DELETE).

rc

The value of the return code depends on the type of VTE manager (PURGER or ENF). For a VTE purger manager, the return code is from the MVS SVCUPDTE system service. For a VTE ENF manager, the return code is from the MVS ENFREQ system service.

addrtn

The value of the address or token depends on the manager, PURGER or ENF. For a VTE purger manager, the address or token is set by the SVCUPDATE system service. For a VTE ENF manager, the address or token is set by the ENFREQ system service.

System action

The system action depends on the function and the completion code.

For *func* = ADD, if *comp* = WAS OK, the purger or ENF manager is added successfully. However, if *comp* = FAILED, the purger or ENF manager addition failed. As a result, the VTE server initialization fails.

For *func* = DELETE, if *comp* = WAS OK, the purger or ENF manager is successfully deleted. However, if *comp* = FAILED, the purger or ENF manager deletion failed, and the system action is as follows:

- For the purger manager, the VTE server retries its attempts to delete the purger manager and reissues the message every 10 seconds.
- For the ENF manager, the VTE server continues its shutdown process, but the ENF manager remains active.

User response

Check why the purger or ENF manager addition or deletion failed. If the purger manager cannot be deleted, the VTE server cannot terminate. In that case it cannot be canceled and if forced (FORCE), an IPL may follow.

Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT268E **I/O SEQUENCE LOST FOR**
jobname/jobid CUU=cuu,
LAST=X'last'
RECEIVED=X'received'

Explanation

The VTE server received an I/O command for a virtual device whose sequence number is out of order (i.e. a previous I/O command for that virtual device was missing). The VTE server waits ten seconds for the expected I/O command to arrive. As it has not yet arrived, the current command is disregarded (since a previous command is missing) and the job that sent that I/O command is EQC-ed. The following variables are listed in the message:

jobname

The jobname of the job that misses the I/O command.

jobid

The jobid of the job that misses the I/O command.

cuu

The device number of the virtual device for which the I/O command misses.

last

Hexadecimal value of the last I/O command sequence number for the virtual device.

received

Hexadecimal value of the received I/O command sequence number for the virtual device.

System action:

The job that misses the I/O is EQC-ed. Then, the virtual device I/O sequence number is reset (in order to accept other jobs on the failing virtual device). If the job wrote data on the virtual tape while the error occurred, this data may be corrupted.

User response:

Collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT292E **TAPES MEDIA MIXTURE FOR**
S.D=step.ddname, R=rvolume,
V=vvolume

Explanation

During the job initialization phase, VTE detected a specific tape volume list specified through a VOL=SER parameter, in which part of the tapes are real and part are virtual. The message lists a couple of adjacent tape volume serial numbers (as they appear on original VOL=SER list) for which the mixture was found.

The following variables are in this message:

step

The stepname containing the DD statement on which the mixed tape volume serial numbers were detected.

ddname

The DDname of the DD statement on which the mixed tape volume serial numbers were detected.

rvolume

The real tape volume serial number.

vvolume

The virtual tape volume serial number.

System action:

Scan of the specific tape volumes list continues in order to find and list other mixtures. The scan then continues with the next DD statement. At the end, the job fails due to a JCL error.

User response:

Investigate why the DD statement specified a mixture of real and virtual tapes. If these tapes are indeed needed, you may manually build two or more concatenated DD statements, each one specifying only real tapes or only virtual tapes.

**HVT293W 'UNLOAD *cuu*' OPERATOR
COMMAND SUPPRESSED BY
serverid SERVER**

Explanation

An 'UNLOAD *cuu*' operator command has been entered for a virtual device. This command is not really needed for unloading tapes from devices that have been deallocated, because they are unloaded by MVS when the device is needed by other tape mount or at step or job end. Due to its internal logic, VTE's processing of the unload operator command may interfere with some MVS internal processes. For this reason, VTE suppresses this unneeded operator command.

The following variables are listed in the message:

cuu

The device number for which the unload was attempted.

serverid

The VTE serverid (as specified in the IDENT initialization statement).

System action:

VTE suppresses the operator command.

User response:

None required. However, if you do need to issue a 'U *cuu*' operator command for a virtual tape previously mounted as a result of a 'M *cuu*, VOL=(SL, *volser*) ' operator command, issue the 'U *cuu*' operator command with six blanks separating the 'U' and the '*cuu*' operands.

**HVT294E *type* FAILED BY *serverid* SERVER
DUE TO DYNAMIC REALLOCATION
ERROR(S)**

Explanation

Errors were encountered while the VTE subsystem attempted to reallocate one or more DD statements or

a dynamically allocated device. This message is preceded by other messages detailing the encountered errors.

The following variables are in the message:

type

Indicates that the errors were encountered during static allocation (JOB) or during dynamic allocation (DYNALLOC).

serverid

Specifies the VTE serverid, as specified in the IDENT initialization statement.

System action

The system action depends on the type value:

JOB

Indicates the job subject to reallocation fails due to a JCL error.

DYNALLOC

Indicates the dynamic allocation of the device subject to dynamic reallocation fails. The return code from the MVS DYNALLOC service is 8, and the dynamic allocation fails.

User response:

Refer to the preceding messages explanations and act accordingly.

**HVT296E UNABLE TO BUILD INTERNAL
REALLOCATION JOB TABLE FOR
jobname/jobid, ENVIR=*envir*,
RC=*rc***

Explanation

The VTE subsystem failed to build the internal reallocation and PAT eligibility job table. This message is preceded by some internal informative messages, issued by the VTE subsystem, which give more details about the problem.

The following variables are listed in the message:

jobname

Indicates the jobname of the job that encountered the problem. If asterisks are displayed, the jobname could not be determined.

jobid

Indicates the jobid of the job that encountered the problem. If asterisks are displayed, the jobid could not be determined.

envir

Indicates that a problem was encountered while the VTE subsystem got control due to STATIC allocation (regular JCL) or DYNAMIC allocation. If asterisks are displayed, the environment (STATIC or DYNAMIC) could not be determined.

rc

Indicates the return code generated by this problem. Valid values are:

04

The VTE subsystem failed to get its needed storage.

08

Internal error (invalid internal requested function).

12

SWAREQ system service failed.

16

Internal error (gotten area to contain the job table is too small).

20

Internal error (locate the proper DD entry or DSNT entry failed).

System action

No dynamic reallocation and PAT eligibility checks that may be needed by the job are performed. That may lead to wrong tape device allocation (that is, virtual tapes may be allocated on real devices and vice-versa). Jobs that do not need any dynamic reallocation and are not PAT eligible are not affected by this problem.

In addition, in JES3 environments, no 'defer' is forced for mounts on the VTE server's virtual devices. If the virtual devices are JES3-managed, this leads to MDS trying to mount the virtual tape (after issuing the IAT5210 message). Since the mount is not satisfied by VTE, MDS places the job on the Verify Queue where it waits forever, leaving the virtual device allocated.

User response:

Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

**HVT297E REALLOCATION FAILED FOR
S.D=step.ddname, RC=rc (text)**

Explanation

The VTE subsystem could not decide whether or not to reallocate a given DD statement. This was due to contradictory conditions found at different points along its decision path.

The variables listed in this message are:

step

The stepname containing the DD statement for which the decision failed.

ddname

The DDname containing the DD statement for which the decision failed.

rc

Valid return codes are listed below:

01

The DD statement should not be reallocated (for example, it explicitly specifies an 'unknown' virtual tape), but according to the dynamic reallocation rules it should be reallocated (that is, the DD statement values match one of the INC type RULE initialization statements).

02

The DD statement should be reallocated (for example, it explicitly specifies a 'known' virtual tape), but according to the dynamic reallocation rules, it should not be reallocated (that is, the DD statement values match one of the EXC type RULE initialization statements).

03

The DD statement should be reallocated (for example, it explicitly specifies a 'known' virtual tape), and at the same time it should not be reallocated (that is, it is excluded due to special DD statements or a specific request coded on the DD statement).

text

Provides a short explanation of the error.

System action:

The job subject to reallocation fails due to JCL error.

User response:

Open the VTE subsystem internal trace (F server,WT00N=jobname) to view the contradictory decisions. Correct the RULE initialization statement(s) that caused the contradictory conditions, refresh the HVTRULxx member, then rerun the job.

**HVT298W S.D=step.ddname (INC) SUPP_ED
BY S.D=step.ddname (EXC)**

Explanation

Contradictory conditions were found when the VTE subsystem built the reallocation decision list. These contradictions occur when a DD statement in a VOL=REF group is eligible for reallocation and another DD statement in the same group is excluded (EXC) for reallocation. When such a condition is encountered, the entire VOL=REF group is not reallocated (that is, all already INCLUDED decisions are overridden by the EXCLUDE condition).

The following variables are listed in the message:

step

Indicates the stepname containing the DD statement to be reallocated and the stepname containing the DD statement to be excluded (EXC) from reallocation, respectively.

ddname

Indicates the DD statement to be reallocated and the DD statement to be excluded (EXC) from reallocation, respectively.

System action:

The DD statement is excluded from the reallocation table. Other DD statements may still be reallocated.

User response:

Verify that this problem is not caused by a user error in the RULE initialization statements definitions. If there is an error, you may need to fix the rules, then refresh the HVTRULxx member.

**HVT300S UNABLE TO OBTAIN
INFORMATION FOR type=name
(service), RC=rc, RS=rs**

Explanation

This message indicates that a problem occurred while using the EDTINFO system service to obtain information about the VTE virtual devices or the disks grouped under its diskpools. It is issued in one of the following processes:

- During VTE server initialization, in one of these cases:
 - VTE fails to obtain information about the VTE virtual device numbers grouped under its esoteric name.
 - VTE fails to obtain information about generic name of the VTE virtual device.
- During regular calculation of the total free space on non-SMS-managed disks used to store virtual tape data sets.
- During allocation of an existing virtual tape dataset that resides on a non-SMSmanaged diskpool.
- During calculation of a new virtual tape dataset extent size.

The variables listed in the message are described below:

type

Possible type values are:

ESOTERIC

The value for which the service failed is an esoteric name.

DEVTYPE

The value for which the service failed is a device type value used to obtain the virtual device's generic name.

name

The esoteric name or the hexadecimal device type value for which the EDTINFO system service failed.

service

Possible service values are:

RTNDEVN

The RTNDEVN option was used by EDTINFO to obtain the VTE virtual device numbers or the device numbers of the disks in the diskpool.

RTNUNIT

The RTNUNIT option was used by EDTINFO to obtain the VTE generic device name.

rc

This return code is generated after invoking the EDTINFO system service.

rs

This reason code is generated after invoking the EDTINFO system service.

System action

The system action depends on when the message is issued.

- If the message is issued during the VTE server initialization phase, the VTE server shuts down.
- If the message is issued during the free space calculation, the calculation will not be done and the non-SMS-managed disks eligible for virtual tape data sets allocation may become full without previous warning.
- If the message is issued during allocation of an existing virtual tape data set that resides on a non-SMS-managed diskpool, the error is ignored and another diskpool is looked for that matches the virtual tape data set location.
- If the message is issued during a virtual tape dataset extent size calculation, the job that requested this new extent is EQC-ed.

User response

Choose the appropriate action for the problem you are experiencing:

For the VTE server initialization problem:

- The ESOTERIC parameter of the VDEVICE initialization statement contains a value not yet defined to MVS. Either modify its value or define its value to MVS using the standard HCD process.
- The device type of the VTE devices is wrong. Collect and forward the appropriate documentation to IBM Technical Support.

For the free space calculation, existing virtual tape dataset allocation or extent size calculation problem:

- The UNIT parameter of one or more DPOOL or VLTRMT initialization statements contains a unit value not yet defined to MVS.

- The UNIT parameter of one or more DPOOL or VLTRMT initialization statements contain Storage Class definitions without an esoteric definition. However, the Storage Class ACS routine removed the Storage Class value. In this case, do the following:
 - For non-SMS diskpools – add an esoteric value to the UNIT parameter.
 - For SMS diskpools – correct the ACS routine so that it does not remove the Storage Class value.

When done, restart the VTE server or refresh the HVTPOLxx member.

HVT301I **SUBSYSTEM *subs* SSVT ->**
 X'ssvtaddr', EXECUTOR ->
 X'execaddr'

Explanation

This is an informative message issued by the VTE server during initialization or as a result of activating the VTE subsystem (due to the 'F server,SSNON' operator command), providing details about the VTE subsystem loading.

The variables listed in the message are described below:

subs

Lists the subsystem name as defined in the NAME parameter of the SUBSYS initialization statement.

ssvtaddr

Lists the address of the subsystem's SSVT in CSA.

execaddr

Lists the address of the subsystem's executor in ECSA.

System action:

None

User response:

None

HVT303I **SUBSYSTEM *subs* phase comp,**
 RC=rc

Explanation

This is informative message issued by the VTE server during server initialization, activating/deactivating the VTE subsystem (due to the 'F server,SSNON/SSNOFF' operator command) or refreshing the HVTRULxx member.

The variables listed in the message are described below:

subs

The subsystem name as defined in the NAME parameter of the SUBSYS initialization statement.

phase

The VTE subsystem is either activated (ACTIVATION) or deactivated (DEACTIVATION).

comp

The activation/deactivation of the subsystem was either successful (WAS OK) or failed (FAILED).

rc

The return code was generated by one of the actions in the message. Possible return code values are listed below.

00

The activation/deactivation was OK.

12

GETMAIN for one of the subsystem components (SSCT, SSVT, etc.) failed.

16

Subsystem activation requested, but the subsystem is already active.

20

Subsystem deactivation requested, but the subsystem is not active.

24

Subsystem executor or one of its subroutines was not found in the library pointed to by STEPLIB DDname.

36

An active subsystem has been detected, but it does not belong to VTE.

52

The dynamic addition of the VTE IEFUJV exit in an JES3 environment failed. HVT260I message that precedes this message gives more details about the error.

56

The partitioned dataset pointed to by STEPLIB DDname could not be opened in order to load the VTE subsystem modules.

84

The partitioned dataset pointed to by STEPLIB DDname could not be opened in order to load the VTE subsystem modules.

System action

The system action depends on the completion code:

WAS OK

The VTE subsystem is either activated or deactivated.

FAILED

The VTE subsystem is neither activated nor deactivated.

User response

The recommended operator action depends on the return code:

36

This message follows message HVT305S. Refer to message "HVT305S" for additional operator actions.

52

This message follows message HVT260I. Refer to message "HVT260I" for additional operator actions.

12, 24, 56, or 84

Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT304I *type REFRESH comp, RC=rc*

Explanation

This is an informative message issued during VTE server initialization, refreshing one of the refreshable members or a utility run. The variables listed in the message are described below.

type

Possible type values are:

POL

The tapepool/diskpool refresh was attempted.

RUL

The dynamic reallocation rules refresh was attempted.

PRL

The PAT rules refresh was attempted.

TRL

The tapepool rules refresh was attempted.

comp

The refresh was successful (WAS OK) or failed (FAILED).

rc

The return code generated by this issue is listed in the message. Possible values are:

00

The tapepool rules refresh was attempted.

28

One of the following occurred:

- One of the refreshable members was not found in the partitioned dataset pointed to by HVTSIN DDname.
- Initialization statement(s) found in one of the refreshable members were in error. The initialization statements listing on the HVTLIST sysout file provides more information about the error.

32

Dynamic reallocation rules or PAT rules refresh was requested, but the VTE subsystem is not active.

44

One of the following occurred:

- The first RULE or PRULE initialization statement in the HVTRULxx or HVTPLRLxx member, respectively, is not an INC type RULE or PRULE initialization statement.
- The first TRULE initialization statement in the HVTTRLxx member is not an INC type TRULE initialization statement.

48

One of the following occurred:

- At least one TPOOL initialization statement did not specify an existing DPOOL initialization statement.
- At least one TRULE initialization statement did not specify an existing TPOOL initialization statement.

52

Internal error attempting to refresh the HVTPLxx refreshable member.

56

One or more tapepool ranges overlap.

64

No TPOOL initialization statement defines the default tapepool. The default tapepool is defined as the TPOOL initialization statement in which RANGE1=(0,0).

68

A RULE or PRULE initialization statement parameter value refers to a non-existing RFILT or PFILT block, respectively.

72

The first RPTRN or PPTRN initialization statement in the HVTRULxx or HVTPLRLxx member is not preceded by an RFILT or PFILT initialization statement, respectively.

76

Maximum number of allowed patterns in an RPTRN or PPTRN initialization statement exceeded.

80

A HVTRULxx or HVTPLRLxx member refresh was attempted, but the corresponding DYR/PAT/PTB/PTD option was DISABLED.

84

A pool refresh operator command was issued while an immediate preceding pool refresh

operator command is still extracting information about the diskpools/tapepools.

88

At least one TPOOL initialization statement LOCID value does not specify an existing and valid VLTRMT initialization statement or that VLTRMT initialization statement is not of type3.

System action

The system action depends on the completion code.

WAS OK

The refresh succeeded.

FAILED

The system action is as follows:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old, previously existing values will remain.

User response

The recommended operator action depends on the return code.

28

Check the validity of the RULE, PRULE, TRULE, DPOOL or TPOOL initialization statements. Fix the initialization statements, then restart the server or retry the refresh. One of the following occurred:

32

Issue the 'F server,SSNON' operator command to activate the VTE subsystem, then retry the refresh.

44

Make sure the first RULE, PRULE or TRULE initialization statements are INC type initialization statements. Fix the initialization statements, then restart the VTE server or retry the refresh.

48

Make sure each TRULE initialization statement specifies an existing TPOOL initialization statement. Make sure each TPOOL initialization statement specifies an existing DPOOL initialization statement. Fix the initialization statements, then restart the VTE server or retry the refresh.

52

Collect and forward the appropriate documentation to customer support for technical assistance.

56

Fix the tapepool's ranges so that they do not overlap, then restart the VTE server or retry the refresh.

64

Define one TPOOL initialization statement with RANGE1=(0,0). This tapepool is considered to be the default tapepool. RMM users should make sure this tapepool's name matches the name specified on the RMM VLPOOL statement for the RMM default tapepool. Restart the VTE server or retry the refresh.

68

This message follows message HVT103E detailing the incorrect RFILT or PFILT name. Correct the invalid RFILT or PFILT reference or add an RFILT or PFILT block that can be referenced by the RULE or PRULE initialization statement parameter value, respectively.

72

Make sure the first RPTRN or PPTRN initialization statement is preceded by an RFILT or PFILT initialization statements, respectively.

76

Decrease the number of patterns in the RFILT or PFILT block, respectively.

80

Enable the option by issuing the appropriate 'F server,DYRON', 'F server,PATON', 'F server,PTBON' or 'F server,PTDON' operator command, then retry the refresh.

84

Wait a little, then reissue to operator command.

88

Make sure each TPOOL initialization statement that contains a LOCID parameter specifies an existing and valid VLTRMT initialization statement and that the VLTRMT initialization statement is of type3.

Fix the initialization statements, then restart the VTE server or retry the refresh.

HVT305S

ACTIVE SUBSYSTEM *subs* DOES NOT MATCH PRODUCT

Explanation

During VTE server initialization or VTE subsystem activation, VTE detected that a subsystem whose name matches the NAME parameter on the SUBSYS initialization statement already exists. However, it does not belong to the VTE product.

The only variable in this message is *subs*, which is the subsystem name as defined in the NAME parameter of the SUBSYS initialization statement.

System action

Possible system actions are listed below:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during VTE subsystem activation, the subsystem is not activated.

User response:

Make sure that the subsystem name specified under the NAME parameter of the SUBSYS initialization statement is correct, then restart the VTE server or retry the subsystem activation.

HVT306I **/PROC /STEP /DDNAME /UNIT/
DATACLAS/STORCLAS/MGMTCLAS
-> NEWUNIT**

Explanation

This is an informative message that details the successful dynamic reallocation of one or more allocations. This message serves as a general title to the reallocation messages that follow and are aligned according to this general title.

In JES2 environments (all types of allocations), the message is issued on the console and is written in the first job's JES file (JESMSG LG) of the job whose allocations have been dynamically reallocated.

In JES3 environments (static allocations), the message is issued on the console and is written in the third job's JES file (JESYSMSG) of the job whose allocations have been dynamically reallocated.

In JES2 environments (dynamic allocations), the message is issued on the console and is written in the first job's JES file (JESMSG LG) of the job whose allocations have been dynamically reallocated.

The following variables are in the message:

proc

The name of the step that called the procedure owning the DDname being reallocated.

step

The stepname owning the DDname being reallocated.

ddname

The DDname being reallocated.

unit

The original unit or device type that has been reallocated.

dataclas

In JES2 environments, this is the Data Class deduced from the SMS ACS routine or defined on the DDname being reallocated.

In JES3 environments, this is the Data Class defined on the DDname being reallocated.

storclas

In JES2 environments, this is the Storage Class deduced from the SMS ACS routine or defined on the DDname being reallocated.

In JES3 environments, this is the Storage Class defined on the DDname being reallocated.

mgmtclas

In JES2 environments, this is the Management Class deduced from the SMS ACS routine or defined on the DDname being reallocated.

In JES3 environments, this is the Management Class defined on the DDname being reallocated.

newunit

The unitname to which the reallocation has just been performed.

System action:

None

User response:

None

HVT307E **UNABLE TO RETRIEVE JOB
ALLOCATION PARAMETERS,
RC=X'rc', RS=X'rs'**

Explanation

During the dynamic reallocation process, the VTE subsystem failed to retrieve one or more of the original job's allocation parameters. This problem usually occurs as a result of an internal error.

The following variables are listed in the message:

rc

Internal return code describing the error. All the return codes are for internal problem determination purposes.

rs

Internal reason code describing the error. All the reason codes are for internal problem determination purposes.

System action

No dynamic reallocation is attempted for this job. In addition, in JES3 environments, no 'defer' is forced for mounts on the VTE server's virtual devices. If the virtual devices are JES3-managed, this leads to MDS trying to mount the virtual tape (after issuing the IAT5210 message).

Since the mount is not satisfied by VTE, MDS places the job on the Verify Queue where it waits forever, leaving the virtual device allocated.

User response

In JES3 environments, cancel the job so that it does not allocate the virtual device.

Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT308I **/PROC /STEP /DDNAME /VOLSER/
ALIAS/DSNAME**

Explanation

This is an informative message indicating that one or more virtual tape volumes are accessed through Parallel Access Tape (PAT) alias volumes. This message serves as a general title to the PAT alias volumes substitution detailed messages that follow and are aligned to this general title.

In JES2 environments (all types of allocations), the message is issued on the console and is written in the first job's JES file (JESMSG LG) of the job that accesses the virtual tape volumes through PAT alias volumes.

In JES2 environments (dynamic allocations), the message is issued on the console and is written in the first job's JES file (JESMSG LG) of the job that accesses the virtual tape volumes through PAT alias volumes.

The following variables are in the message:

proc

The name of the step that called the procedure owning the DDname using the PAT alias volumes.

step

The stepname owning the DDname using the PAT alias volumes.

ddname

The DDname using the PAT alias volumes.

volser

The first (or only) virtual tape volume serial number substituted to PAT alias volume. If the dataset being accessed resides on more than one virtual tape volume, these volumes are not listed (although they are accessed through PAT alias volumes as well).

alias

The first (or only) PAT alias volume who substituted the original virtual tape volume.

dsname

The dataset name of the dataset residing on the virtual tape.

System action:

None

User response:

None

HVT310E

**name type (COMP=comp)
RECOVERED FROM A Ssss-Uuuuu
ABEND**

Explanation

During regular work, one of the VTE server components or a utility encountered an abend and the recovery routine successfully recovered from it. The following variables are listed in the message:

name

If *type* is SERVER or UTILITY or IRB, this is the VTE serverid (as specified in the IDENT initialization statement).

If *type* is SUBSYS or IEFDB401, this is the VTE subsystem name (as specified in the SUBSYS initialization statement).

If *type* is EVARY, this name value is VTE.

If the *name* value is VTE for any other type, the VTE serverid name could not be determined yet..

type

Possible *type* values are listed below:

SERVER

The abend occurred in one of the VTE server's components.

UTILITY

The abend occurred in one of the offline utilities.

SUBSYS

The abend occurred in the VTE subsystem.

IEFDB401

The abend occurred in the MVS dynamic allocation exit used by VTE.

EVARY

The abend occurred in the MVS ENF listener used by VTE.

IRB

The abend occurred in the IRB routine that runs in the job address space.

comp

The component where the error occurred. Possible values are:

SBSMMGR

The subtask that manages the virtual tapes data sets.

SCHMGR

Schedule SRB manager

SCMPMGR

VTE data compress manager

SDSTCOMP

HVTUDST utility

SENF

VTE ENF manager

SERVER

Main VTE server's dispatcher

SFTPCOMP

HVTUFTP utility

SHVTCCW

HVTCCW routing that runs in an enclave SRB.

SHVTSEC

HVTSEC routine that runs in an enclave SRB.

SOCEMGR

The subtask that manages the OPEN/CLOSE of the virtual tape data sets.

SRCVCOMP

HVTURCV utility

SRDVMGR

The virtual tape data set access recovery manager.

SREALLOC

VTE subsystem

SSCRCOMP

HVTUSCR utility

STAPCOMP

HVTUTAP utility

STASKMGR

Main I/O subtasks manager

STBFMGR

The type 3 and type 4 e-Vault data mover manager.

STBVMGR

The type 1 and type 2 e-Vault data mover manager.

ST2TCOPY

The type 1 and type 2 e-Vault data copy.

SVLTMGR

The e-Vault logical layer manager.

SV2VCOMP

HVTUV2V utility

SXIOMGR

Backend I/O to disk manager

SXTTMGR

VTE server's EXCP manager

sss

System abend code

uuuu

User abend code

System action

The system action depends upon when the abend occurred. Possible actions are:

- If the abend occurred in the VTE server, the job whose I/O caused the problem is EQC-ed.
- If the abend occurred during the e-Vault process, the virtual tape being e-Vaulted is left in an interim e-Vault status. An attempt to resume the e-Vault process will be made later.
- If the abend occurred in the VTE subsystem, the original allocation is performed, but no reallocation is attempted.
- If the abend is retrievable, the VTE server or utility recovers from the abend and continues its process.
- If the abend is non-retrievable (e.g. a utility is cancelled by the operator), that utility terminates.

User response:

If the abend is unexplainable, collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT311E

func ERROR R15=*r15*,
ERROR=X'*error*',X'*info*',X'*smerr*',
DSN=*dsn*

Explanation

An attempt to dynamically allocate or de-allocate a virtual tape dataset (either new or existing) failed. An HVT998I message is written on the original job's first JES file (JESMSG LG), providing a detailed explanation of the error. The return and reason codes are listed in the "Requesting Dynamic Allocation Functions" chapter of the *MVS Authorized Assembler Services Guide*.

The following variables are listed in this message:

func

Possible values for *func* are the following:

- ALLOC – The error occurred on an attempt to allocate the virtual tape dataset.
- DEALLOC – The error occurred on an attempt to deallocate the virtual tape dataset.

r15

The error occurred on an attempt to allocate the virtual tape dataset.

error

The hexadecimal error code associated with the SVC99 failure.

info

The hexadecimal information code associated with the SVC99 failure.

smserr

The SMS reason code associated with the SVC99 failure.

dsn

The virtual tape dataset name that failed the allocation/de-allocation.

System action:

The job that requested the virtual tape is EQC-ed.

User response

Try to locate the problem using the allocation/deallocation information and error codes, then resubmit the failing job.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

**HVT312E VOLUME=volume ERROR WHILE
CONSTRUCTING THE VIRTUAL
TAPE DATASET NAME, RC=rc**

Explanation

An error was encountered while constructing the virtual tape dataset name. This error occurs when invalid values have been coded in the QUAL1, QUAL2 or QUAL3 parameters of the VDSN initialization statement. The following variables are listed in this message:

volume

The mounted virtual tape volume serial number.

rc

Possible return codes are:

05

Neither QUAL2 nor QUAL3 parameters of the VDSN initialization statement is %VOL.

06

The second qualifier is neither %VOL nor %TDSN.

07

The second qualifier is too long.

08

The third qualifier is too long.

System action:

The job requesting the virtual tape is EQC-ed.

User response:

Check that the QUAL1, QUAL2 or QUAL3 parameters coded in the VDSN initialization statement adhere to the requirements. Correct any errors, then bounce the VTE server.

**HVT313W serverid SERVER WAITING FOR
TAPE MOUNT ON CUU=cuu**

Explanation

A virtual device was issued a channel command but it has not yet been issued a mount request by MVS. VTE must receive a mount request from MVS in order to complete its internal mount processing. A common cause of this error is a problem with the VTE extended MCS console, a problem that prevents VTE from detecting console messages. Another cause can be that a mount message is not coded in the HVTMCSxx member, hence not recognized by VTE.

The following variables are listed in this message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

cuu

The device number of the virtual device on which the MVS mount request is missing.

System action

The VTE server waits for the MVS mount request. On pre-defined intervals of 16 seconds, the message is repeated.

If after a pre-defined interval of 96 seconds the MVS mount request has not been received, the job waiting for the mount is EQC-ed.

User response

See whether a previous problem occurred with the VTE extended MCS console.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

**HVT314W serverid type ACCESS TO file
DELAYED FOR jobname/jobid
HELD BY hjobname/htcb/hsysnme**

Explanation

The VTE server attempted to access the VDB/journal file (due to a tape mount or unload request) or a utility attempted to access the VDB/journal file for adding/deleting/ updating one or more entries. However, the VDB/journal file is ENQ-ed or RESERVE-d by another process (either another VTE server subtask or a background utility) running in this or another system that shares the same VDB/journal and the access is delayed for more than 30 seconds. The access is periodically retried and if unsuccessful, this message is reissued every 60 seconds.

Note:

- The ENQ or RESERVE QNAME is V_TAPE and the RNAME is the VDB or journal dataset name, respectively.

- Sites that share their VDB between two or more systems but do not employ a GRS product (or the VDB RESERVE is not included in the RESERVE conversion list), will not get HVT314W when the VDB is held by another system. In that case, HVT314W will be issued only if the VDB is held by a task in the same system.

The following variables are listed in this message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

type

Specifies the type of delay that has occurred. Possible values are:

- SERVER – The VTE server access is delayed.
- UTILITY – A utility access is delayed.

file

Possible file types are:

- VDB – Virtual Database file
- JRN – Journal file

jobname

The job name varies depending upon the type of access.

- If the access is performed by the VTE server on behalf of a job accessing a virtual tape, this is the name of the job.
- If the access is performed by the VTE server for its internal process, this is the name of the VTE server started task.
- If the access is performed by a utility, this is the name of the job running the utility.

jobid

The ID of the job listed in the *jobname* field.

hjobname

The name of the job that holds (has exclusive control over) the VDB or journal file. If asterisks are displayed, the jobname could not be determined.

htcb

The TCB address of the task that issued the ENQ or RESERVE service. If asterisks are displayed, the TCB address could not be determined.

hsysname

The system name of the system in which the holding job runs. If asterisks are displayed, the system name could not be determined.

System action

Possible system actions are:

- The VDB/journal file access is retried. The message for this specific jobname/jobid is issued every 60 seconds.
- A 'D GRS , C' system command is issued and its output is broadcasted to the server log. The command and the response are prefixed by a three-bytes sequence number that connects the command and the response.

User response

Try the following actions to resolve this issue:

- Check whether a VTE server or utility running in another system (sharing the same VDB and journal file) failed while leaving the VDB or journal file in a RESERVE state.
- Check whether an offline utility (like HVTUTAP) has been running for a long time, thus RESERVE-ing the VDB or the journal file.
- An increase in the VDBACC parameter value in the PERFORM initialization statement can reduce the frequency for which this messages occurs.

HVT315I

**VOLUME=volume MOUNTED ON
CUU=cuu jobname/jobid/pgm/
tapepool/diskpool**

Explanation

This is an informative message issued by the VTE server after it mounted a virtual tape on one of its virtual devices. This message follows the tapepool reset (if needed) and contains the reset tapepool name. The following variables are listed in the message:

volume

The mounted virtual tape volume serial number.

cuu

The device number of the virtual device on which the virtual tape was mounted.

jobname

The name of the job that mounted the virtual tape.

jobid

The jobid of the job that mounted the virtual tape.

pgm

The job step program name executed by the job.

tapepool

The tapepool name from which the scratch virtual tape has been selected or the tapepool name assigned to the specific (non-scratch) virtual tape.

diskpool

The diskpool name associated with the selected tapepool.

System action:

None

User response:

None

HVT317W *poolname* **POOL EXHAUSTED,
ENTER VSN FOR *jobname/jobid*,
R(RETRY), N (FAIL)**

Explanation

No more scratch virtual tapes are eligible for mount from a specific (or the default) tapepool. Virtual Tape Emulation detected this as a result of a scratch virtual tape mount request from a specific (or the default) tapepool. The following variables are listed in this message:

poolname

The tapepool name which does not contain any more scratch virtual tapes.

jobname

The jobname that requested the scratch virtual tape.

jobid

The jobid of the job that requested the scratch virtual tape.

System action:

The VTE server waits for the operator's reply.

User response

Perform one of the following:

- Reply with a new volume serial number (one to six characters) of a virtual tape from the specified tapepool. The VTE server automatically initializes the new virtual tape and presents it as a scratch mounted virtual tape.
- Reply R, in which case the VTE server re-attempts to find a scratch virtual tape from the specified tapepool. Note that prior to using this option you need to either:
 - Run the HVTUSCR utility in the HVTUSCR member in the SAMPLIB library, in order to return used virtual tapes to scratch status.
 - Run the HVTUTAP utility in the HVTUTAP member in the SAMPLIB library, in order to add new virtual tapes to the VTE VDB. These tapes should be added to the local tape management system as well.
 - Reply N, in which case the job that requested the scratch virtual tape is EQC-ed.

HVT318E **VOLUME LABELED *volume* IS
INVALID, RC=*rc* (*text*)**

Explanation

The operator reply to HVT317W message was wrong. The following variables are listed in the message:

volume

The virtual tape volume serial number entered by the operator as a reply to the HVT317W message.

rc

A return code of "01" indicates that the entered volume serial number exists already.

text

Short explanation of the error.

System action:

The VTE server reissues the HVT317W message.

User response:

Refer to message [“HVT317W” on page 515](#) for further action.

HVT319W **VOLUME=*volume* REJECTED BY
TMS. REPLY R (RETRY), N (FAIL)**

Explanation

The TMS could not accept a virtual tape mounted by VTE.

This may occur for one of these reasons:

- The request was for a scratch virtual tape, but the mounted virtual tape was not scratch.
- The request was for a scratch virtual tape from a specific tapepool, but the mounted virtual tape belonged to a different tapepool.

Other reasons can cause the TMS to reject the mounted virtual tape. These reasons are listed by the TMS messages in the job's log that precedes the HVT319W message.

VTE might have mounted a virtual tape that is not accepted by the TMS, for one of these reasons:

- The tapepool definitions (defined in the TPOOL initialization statements) do not coincide to the TMS tapepool definitions.
- The scratch virtual tapes range in the VDB does not coincide with the TMS tapes range due to an error, VDB restore without proper recovery, bad virtual tape ranges definitions, etc.

In most cases, the TMS unloads the virtual tape and requests a new one. In order to keep from mounting other virtual tapes (in a mount process loop), VTE stops the process and allows the operator to fix the incorrect definitions, before proceeding with the next mount.

The only variable in this message is *volume*, which represents the mounted virtual tape's volume serial number that was not accepted by the tape management system.

System action:

The VTE server waits for the operator's reply.

User response

Perform one of the following:

- Reply R, in which case the VTE server re-attempts the mount. Note that prior to using this option, you must locate the reason that the TMS did not accept the mounted virtual tape and fix it. That reason is displayed in preceding messages issued by the TMS.
- Reply N, in which case the job that requested the virtual tape is EQC-ed.

HVT320E **VSAM FUNCTION=*func* FAILED,
RC=*rc*, ERROR=X'*error*'**

Explanation

A VSAM error was encountered while opening or closing the VDB during a virtual tape mount or unload. A free text (prefixed by IKJ) describing the error follows this message and is written on the VTE server JESMSGLG file only.

The following variables are listed in the message:

func

The type of function, either OPEN or CLOSE, that has been attempted.

rc

The return code displays the contents of R15 after the VSAM function terminated.

error

The hexadecimal value of the ACB ERROR field.

System action

The system action depends on the value of *func*:

OPEN

If the error occurred while the VTE server was processing a virtual tape, the job using the virtual tape is EQC-ed. If the error occurred for a utility run, the utility terminates.

CLOSE

The virtual tape is unloaded, but it may remain 'in-use'.

User response

Try to locate the problem using the return and error codes and correct the cause of the error.

- If the error occurred during mount, rerun the job.
- If the error occurred during unload, the virtual tape should be recovered using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT321E **VSAM FUNCTION=*func* FAILED,
RC=*rc*, FDBK=X'*fdbk*', KEY=*key1*,
REC=*key2***

Explanation

A VSAM error was encountered while the VDB was being updated. A free text message (prefixed by IKJ) describing the error follows this message and is written on the VTE server JESMSGLG file only.

The following variables are listed in the message:

func

A function has been attempted. Possible function values are:

- GET
- PUT
- ERASE
- POINT

rc

The return code displays the contents of R15 after the VSAM function terminated.

fdbk

The hexadecimal value of the RPL's FDBK field.

key1

The key of the current record contained in the VDB.

key2

The actual key of the record to be accessed.

System action

The system action could be either of the following actions:

- If the error occurred while the VTE server was processing a virtual tape, the job using the virtual tape is EQC-ed.
- If the error occurred for a utility run, the utility terminates.

User response

Try to locate the problem using the return and fdbk codes and correct the cause of the error.

- If the error occurred during mount, rerun the job.
- If the error occurred during unload, the virtual tape remains in an 'in-use' status and it should be recovered using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT322E **VSAM FUNCTION=*func* FAILED,
RC=*rc*, R0=*r0***

Explanation

A VSAM error was encountered while manipulating internal control blocks for accessing the VDB. A free text message (prefixed by IKJ) describing the error follows this message and is written on the VTE server JESMSG LG file only. The following variables are listed in the message:

func

A MODCB function has been attempted.

rc

Contents of R15 after the VSAM function terminated.

r0

Contents of R0 after the VSAM function terminated.

System action

Possible system actions are:

- If the error occurred while the VTE server was processing a virtual tape, the job using the virtual tape is EQC-ed.
- If the error occurred for a utility run, the utility terminates.

User response

Try to locate the problem using the return code and R0 contents and correct the cause of the error. The action is based on the following errors:

- If the error occurred during mount, rerun the job.
- If the error occurred during unload, the virtual tape remains in an 'in-use' status. Recover the virtual tape using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

**HVT323W VOLUME=volume (jobname/jobid)
REQ=request STAT=status (sysid)
TO CANCEL WAIT REPLY 'N'**

Explanation

The VTE server attempted to mount a virtual tape on one of its virtual devices or to perform its first write on an already mounted virtual tape. During that attempt it encountered one of the following:

- The virtual tape is to be mounted for read access, but the same virtual tape is busy for one of these reasons:
 - It is being written by one of the systems that share the VTE complex.

- It is being concurrently read by four systems that share the VTE complex.

- The virtual tape is already mounted for read and the application needs to perform its first write. At that time, the same virtual tape is being read by at least one of the other systems that share the VTE complex.

The following variables are listed in the message:

volume

The virtual tape volume serial number for which the mount or write was attempted.

jobname

The jobname of the job that requested the virtual tape mount or write.

jobid

The jobid of the job that requested the virtual tape mount or write.

request

The type of request to be processed. Valid values are:

- READ – The virtual tape is to be mounted for read.
- WRITE – The mounted virtual tape is to be written the first block.

status

The type of request to be processed. Valid values are:

- READ – The virtual tape is currently read by one or more systems that share the VTE complex.
- WRITE – The virtual tape is being written by a system that shares the VTE complex.

sysid

The system-id of the system that is currently writing or of one of the systems that are currently reading the virtual tape.

System action:

The VTE server waits either for the virtual tape to be released by the other system(s) or for the operator's reply.

User response

Perform one of the following:

- Wait for the other system(s) to release the virtual tape, in which case the mount will be completed successfully or the job will be able to write on the virtual tape.
- Reply N, in which case the job waiting for the virtual tape mount will be EQC-ed.
- If a system went down while using the virtual tape and you do not expect it to come up again, recover

the virtual tape using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.

HVT324W **VOLUME=*volume* VIRTUAL TAPE
DATASET HELD BY *jobname*/
sysname, TO CANCEL WAIT REPLY
'N'**

Explanation

The VTE server tried to mount a virtual tape on one of its virtual devices. However, it found that the virtual tape dataset was allocated to another job or user.

The following variables are listed in this message:

volume

The virtual tape volume serial number whose associated virtual tape dataset allocation was attempted.

jobname

Jobname of the job that allocated the virtual tape dataset.

sysname

System name in which the job that allocated the virtual tape dataset runs.

System action:

The VTE server waits for either the dataset to be released or for the operator's reply.

User response

Perform one of the following:

- Check who is allocating the virtual tape dataset and take the required steps to release it.
- Reply N, in which case the job requesting the virtual tape will be EQC-ed.

HVT325W **MAX NUMBER OF CONCURRENT
maxtask I/O SUBTASKS
REACHED, *jobname*/*jobid*
DELAYED**

Explanation

During normal activity, the VTE server detected a temporary shortage in the number of idle I/O subtasks that are allowed to run concurrently. This number is set by the MAXTASK parameter of the PERFORM initialization statement. If that parameter is set to its maximum value (or 0), the condition that leads to this message never occurs.

The shortage of idle I/O subtasks may be the result of one of the following events:

- The VTE server has issued several WTORs to which the operator has never replied.
- One or more disks used for storing the virtual tape datasets did not complete the I/O sent to them by

the VTE server. This can be either because they are RESERVE-d by other system(s) or because they are not functioning properly.

- Internal VTE problems that cause active subtasks to become inactive.
- Poor performance of the server which causes the number of I/Os to be processed to be greater than the I/Os that are actually processed.

The following variables are listed in the message:

maxtask

The maximum number of I/O subtasks that are allowed to be active concurrently.

jobname

The jobname whose I/O is delayed.

jobid

The jobid of the job whose I/O is delayed.

System action:

The job's I/O is delayed until one of the maxtask I/O subtasks becomes idle and is ready to process it. Additional jobs may issue I/Os that will be put in a queue behind this job however the message will not be issued for them. Rather, they may issue the HVT265W message. The message is issued every 30 seconds until the shortage is relieved.

User response

Perform one of the following:

- Reply to the outstanding VTE server WTOR(s) (if any).
- Make sure all the VTE server's I/Os to disks ended successfully. Verify that the disks containing the virtual tape datasets are not RESERVE-d or otherwise encountered other I/O errors.
- If a VTE problem is suspected, you may need to bounce the VTE server.
- If the MAXTASK parameter of the PERFORM initialization statement is lower than the number of defined virtual devices, it is recommended to increase it.

If the problem is not located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT330I **VOLUME=*volume* TAPE VOLUME
ENTRY DELETE *comp*, RC=*rc***

Explanation

This is an informative message indicating that HVTUV2V utility attempted to delete a tape volume entry. The following variables are listed in the message:

volume

The virtual tape volume serial number whose associated tape volume entry delete was attempted.

comp

Either the delete was successful (WAS OK) or it was not successful (FAILED).

rc

Possible values for the return code are:

00

The delete was successful.

01

IDCAMS utility failed to delete the tape volume entry. The detailed IDCAMS messages precede this message.

System action:

If the output virtual tape status prior to the copy start was 'unknown', it is deleted. Otherwise, no action is taken.

User response

The recommended operator action depends on the return code:

00

None

01

Check the HVTOUT sysout file for the IDCAMS messages explaining the error and act accordingly.

HVT331E UNEXPECTED EOF OR LABEL ERROR ON VOLUME=volume

Explanation

During sequential tape read, an End-of-File has been encountered on the virtual tape dataset. However, this End-of-File is not reflected in the corresponding VDB entry. This situation may have been caused by one of the following:

- A previous abend of the job that created the tape dataset.
- A restore of an old virtual tape dataset without recovering its VDB entry.
- An internal VTE problem that caused the mishandling of the data during a previous virtual tape write.

The only variable in this message is *volume*, which represents the virtual tape's volume serial number that experienced the End-of-File.

System action:

The job whose virtual tape experienced the unexpected End-of-File is EQC-ed.

User response

Perform one of the following:

- In most cases, a recovery of the virtual tape (using the HVTURCV utility in the HVTURCV member in the SAMPLIB library) will suffice.
- Restore an older version of the virtual tape dataset and run the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
- Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.

HVT332E I/O ERROR ON VOLUME=volume, RC=rc (text)

Explanation

A logical I/O error was encountered while accessing a virtual tape dataset. The following variables are listed in the message:

volume

The virtual tape volume serial number whose associated virtual tape dataset experienced the I/O error.

text

Short explanation of the error.

rc

The possible values for the return code are described below.

Return Codes**01**

The virtual tape was marked 'unusable' (by the HVTURCV utility) and no data can be read.

02

The just read record eye catcher is invalid.

03

Bad block segmentation. Possible reasons:

- Expected first segment of a block or a tapemark but received another segment.
- Segment number sequence error.

04

Writing a block with length zero.

05

Record length error. Possible reasons:

- Read length does not match the length in the block header.
- Writing a tape block larger than the maximum supported size of 256K.
- A tapemark block length is not 0.

06

Block number incorrect. The read block number does not match the requested block number.

- 09** Skip block error (either forward or backward).
- 10** General DVD verification error.
- 11** RDJFCB system service error during virtual tape dataset volumes switch (in a multivolume dataset).
- 14** A scratch virtual tape with LABEL=(,NL) has been mounted. However, the mount was rejected because MVS assigned a special volume serial number (Lnnnnn) that differs from the one VTE originally assigned and allocated for that virtual tape.
- 15** An application requested the mount of a SL (standard label) virtual tape for input process. However, that virtual tape is NL (no label). As a result, MVS unloads this virtual tape and prepares to request again for mounting a SL (standard label) virtual tape. In order to avoid a mount/unload loop, the job is EQC-ed.
- 16** An application requested the mount of a NL (no label) virtual tape for input process. However, that virtual tape is SL (standard label). As a result, MVS unloads this virtual tape and prepares to request again for mounting a NL (no label) virtual tape. In order to avoid a mount/unload loop, the job is EQC-ed.
- 18** An application requested the mount of a new 'unknown' virtual tape for input process. MVS unloads this virtual tape and prepares to request for mounting the same virtual tape again. In order to avoid a mount/unload loop, the job is EQC-ed.
- 21** An unexpected virtual tape End-of-Volume has been encountered while writing the VOL1/HDR1/TM records of a new mounted scratch virtual tape label.
- 22** The job's buffer from which the server should retrieve the virtual tape data is not accessible. Possible reasons:
- The job's buffer is shorter than the requested length. Similar jobs accessing real tapes would encounter a SB00 abend.
 - The VTE server tried to access the buffer after the job ended. If so happened, this message is not to be considered as an error (as due to timing issues, MVS does not always wait for channel
- commands to finish before it removes the buffers from storage).
- An internal error prevented the VTE server from correctly interpreting the job's I/O buffer's address.
- 23** The job's buffer into which the VTE server should put the virtual tape data is not accessible. Possible reasons:
- The job's buffer is shorter than the requested length. Similar jobs accessing real tapes would encounter an SB00 abend.
 - The VTE server tried to access the buffer after the job ended. If this happened, this message should not be considered an error (as due to timing issues, MVS does not always wait for channel commands to finish before it removes the buffers from storage).
 - An internal error prevented the VTE server from correctly interpreting the job's I/O buffer address.
- 25** The expected tapemark after a forward/backward file was not detected.
- 26** A block to be written contains a 'Previous TTR' of 0.
- 30** An attempt was made to access a virtual tape block whose location is beyond the virtual tape dataset boundary.
- 32** A data chain CCW has been encountered in which the number of chained CCWs exceeds 1000. The VTE server cannot handle more than 1000 data chained CCWs.
- 34** During a 'Locate Block' process, a segment number greater than 1 has been read (while the first segment of a block was expected).
- 35** During a 'Locate Block' process, the read block number does not correspond to the expected block number.
- 36** During a 'Locate Block' process, a block whose 'Previous TTR' field is 0 has been read.
- 37** A datachained CCW has been encountered in which the command is x'0c' (Read Backward). The VTE server cannot handle a datachained CCW that contains the 'Read Backward' command.

- 39** An error occurred while translating a real address to a virtual address.
- 40** Attempt to write on a virtual tape through a PAT alias volume.
- 41** The server was bounced after it assigned the PAT alias volumes. Therefore, it cannot associate the virtual tape volume serial number to the PAT alias volume.
- 42** A job attempted to read or write user data from/to E/CSA areas. This can be either a deliberate attempt or a result of bad VTE server interpretation of user data.
- 43** One of the system control blocks from which the server picks up data used to move data back and forth from the job's address space to the server's address space failed validation. This makes any data movement impossible.
- 44** A job attempted to write data on a specific existing virtual tape that belongs to a Read_Only mode tapepool or the VDB is in Read_Only mode.
- 45** A job attempted to create a specific new virtual tape, but it belongs to a Read_Only mode tapepool or the VDB is in Read_Only mode.
- 46** A job attempted to create a non-specific (scratch) virtual tape, but it belongs to a Read_Only mode tapepool or the VDB is in Read_Only mode.
- 47** A job attempted to write data on a virtual tape that is Logically Write Protected.
- 48** A job is using compression type=3 to write data to a virtual tape; however, the compression dictionary copy that resides in the job's address space is damaged (overwritten, deleted, etc.). The already written data in this run is not valid.

Compressed data written on this virtual tape in previous runs is valid and can be accessed.
- 49** As a result of an ad-hoc mount of a new virtual tape, the new virtual tape had to be added to the VDB. However, the number of virtual tapes in VDB exceeded the allowed maximum.

System action:

The job accessing the virtual tape is EQC-ed.

User response

The recommended operator action depends on the return code. Possible return codes are:

- 01** Perform one of the following:
- Restore the virtual tape dataset from an old backup, then recover it using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
 - Scratch the virtual tape.
- 02, 03, or 06** Perform one of the following:
- Restore the virtual tape dataset from an old backup, then recover it using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
 - Scratch the virtual tape.
 - Re-init the virtual tape (using IEHINIT).
- 05** Decrease the written tape block size. The maximum supported tape block size is 256K.
- 14** Make sure the job requests a standard label scratch virtual tape.
- 05 or 16** Make sure that the virtual tape label type matches the label type specified in the appropriate job's JCL statement.
- 18** Make sure that the read virtual tape is a 'known' virtual tape.
- 32** Decrease the number of CCWs in the datachain. VTE supports up to 1000 data chain CCWs in one EXCP sent to virtual tapes.
- 04, 09, 10, 11, 21, 22, 23, 25, 26, 30, 34, 35, 36, 37 or 39** Collect and forward the appropriate documentation to IBM Technical Support for technical assistance.
- 40** Make sure this specific DD statement is not included in the PAT processing.
- 41** Rerun the job.
- 42** If the data is read/written deliberately from/to E/CSA, make sure no data is read/written from these areas.

If no data is read/written from/to E/CSA, rerun the job (as it may be a temporary error).

43

Rerun the job (as it may be a temporary error).

44 or 45

Choose a virtual tape from a Read_Write tapepool or set the VDB to Read_Write mode and rerun the job.

46

Set the job to mount a scratch virtual tape from a Read_Write tapepool or set the VDB to Read_Write mode and rerun the job.

47

The job/user is probably not authorized to write on the virtual tape. If write is needed, provide the user with a READ authorization to the IEC.TAPERING profile or use a different user who has such authorization.

48

The job using the virtual tape may have encountered a problem causing it to overwrite the compression dictionary.

Obtain a system dump of that job and of the server and forward them to IBM Technical Support for assistance.

49

Use the HVTUTAP utility to delete redundant scratch virtual tapes from the VDB then rerun the job.

HVT333I **VOLUME=volume SCRATCH RC=rc**
 (text) (DSN=dsn)

Explanation

This is an informative message, issued by the following components, indicating that a virtual tape has been scratched or marked for remote-scratch processing. The reasons vary by the utility that performed the scratch:

HVTUSCR utility

A virtual tape has been scratched during the normal virtual tape scratch process or has been marked for remote scratch process.

VTE server

A scratch can occur after a new ('unknown') virtual tape has been mounted for "read" access, then unloaded with an IEC512I message (I/O error). As a result, the virtual tape becomes 'known' for the period of time between the mount and the unload. After the unload, the virtual tape is scratched and its corresponding VDB entry is removed from the VDB (the virtual tape becomes 'unknown' again).

HVTUV2V utility

A scratch can occur after a new ('unknown') virtual tape was added to the VDB as a result of a 'real to virtual' copy, but the copy failed. In that case, the virtual tape is scratched and its corresponding VDB entry is removed from the VDB (the virtual tape becomes 'unknown' again). This is done to revert the virtual tape status (as it was before the copy started).

The variables in the message are described below.

volume

The virtual tape volume serial number to be scratched or marked for

remote-scratch.

rc

Possible return codes are listed below.

00

Scratch or remote-scratch was OK, or the tape is already scratch.

01

Scratch or mark for remote-scratch failed. Virtual tape is 'in-use' (either being used in other LPAR or left 'in-use' due to a previous failure).

02

Scratch or mark for remote-scratch failed. Virtual tape currently enqueued by MVS with SYSZVOLS major name.

03

Scratch failed. IDCAMS utility failed to delete the virtual tape dataset. The IDCAMS error messages follow.

04

Scratch or mark for remote-scratch failed. VSAM error on the VDB.

05

Scratch or mark for remote-scratch failed. Date/time that this virtual tape was deferred by the HVTUSCR utility is higher than the current date/time.

06

Scratch or mark for remote-scratch failed. Scratch is deferred as indicated in the DEFERBY parameter of the TPOOL initialization statement that this virtual tape is associated with. If the DEFERBY value is 9999, this tape will never be scratched.

07

Scratch or mark for remote-scratch failed. The virtual tape is reclaimed and the defer-scratch indications in the VDB are removed.

08

Scratch or mark for remote-scratch was OK. As a result of specifying FORCE=YES, a scratch occurred regardless of any defer conditions.

09

Scratch or mark for remote-scratch was OK. Scratch occurred due to defer-scratch time expiration.

11

Scratch or mark for remote-scratch failed. The virtual tape is in an interim e-Vault status.

16

Scratch failed. The virtual tape resides on a Read_Only mode tapepool or the VDB is in Read_Only mode.

18

Scratch or mark for remote-scratch failed. The tape is a valid active tape created after the active tapes report was created and therefore not included in the report.

This return code is valid for all SYNC types *except* TYPE=VOLSER, and will NOT cause a return code of 4.

19

Scratch or mark for remote-scratch failed. The specified tape was not found in the VDB.

This return code is valid *only* for SYNC TYPE=VOLSER.

text

Short explanation of the error. Possible explanations are listed below.

OK

Virtual tape successfully scratched.

OK REMOTE

Virtual tape successfully marked for remote-scratch.

OK – ALREADY SCRATCH

Virtual tape to be scratched is already scratch.

NOTOK - VOLUME IN USE

Virtual tape is marked 'in-use' (from this system or other system sharing the VDB) as a result of a system failure.

NOTOK - ENQUEUED BY MVS

Volume is being used by MVS (probably as part of a multivolume tape dataset).

NOTOK - IDCAMS FAILED

IDCAMS utility failed to scratch the volume.

PUT ERROR ON VDB

A VSAM error occurred while updating the VDB.

NOTOK - date/time

The date/time field (0CYDDDF/HHMMSS) is the time that the HVTUSCR utility first tried to scratch this virtual tape but has deferred the process. This date/time is higher than the current date/time.

NOTOK - DFR BY hhhh HH

Scratch is still deferred by *hhhh* hours.

NOTOK - RECLAIMED

The volume serial number was in a defer-scratch status. This reason is either:

- An explicit request to reclaim it has been encountered.
- The local TMS has moved the virtual tape to active status. As a result, VTE removed its defer-scratch status.

OK - FORCED

The virtual tape was successfully scratched (regardless of its defer-scratch status or appropriate DEFERBY value).

OK - FORCED REMOTE

The virtual tape was successfully marked for remote-scratch, regardless of its defer-scratch status or appropriate DEFERBY value.

OK - EXPIRED

The virtual tape was successfully scratched when the defer period expired.

OK - EXPIRED REMOTE

The virtual tape was successfully marked for remote-scratch when the defer period expired.

NOTOK - IN VLT PROCESS

The virtual tape is in an interim e-Vault status, that is, not 'local' or 'vaulted'.

NOTOK - RO TPOOL / VDB

The virtual tape resides on a Read_Only mode tapepool or the VDB is in Read_Only mode.

NOTOK – CR AFTER JOBST

The virtual tape was created after the job that runs the HVTUSCR utility produced the active tapes list and therefore may be active.

NOTOK – NOT IN VDB

The specific virtual tape was not found in the Virtual Tape Database (VDB).

dsn

The virtual tape dataset name. If dsn=NONAME*, a virtual tape did not have an associated virtual tape dataset on disk.

System action:

The virtual tape is processed as described in the message.

User response

The recommended operator action depends on the return code:

00 or 02

None

01

If the virtual tape is not used in any LPAR, run the HVTURCV utility in the HVTURCV member in the SAMPLIB library to recover the virtual tape.

03 or 04

Check the HVTOUT sysout file for messages explaining the error and proceed accordingly.

05

Check why the current date/time is lower than the time HVTUSCR first tried to defer the virtual tape scratch process. In any case, the virtual tape can be scratched using the FORCE=YES parameter.

11

Wait until the virtual tape is local or fully e-Vaulted, then retry the scratch.

16

Virtual tapes that reside on Read_Only mode tapepools or defined in a Read_Only mode VDB cannot be scratched. If you are aware of this virtual tape scratch and still want to scratch it, remove the Read_Only mode of the tapepool or the VDB and (provided the disks are not physically set to Read_Only mode), rerun the HVTUSCR utility.

19

Probable user error. Specify a virtual tape found in the VDB.

HVT335W NO TAPE IN VDB IS ACTIVE IN TMS. ALL TAPES WILL BE SCRATCHED. REPLY Y (CONT), N (TERM)

Explanation:

During the VTE Active Tapes Report analyze phase that precedes the actual virtual tapes scratch process, the HVTUSCR utility found that although the report contains active tapes, no one of them is a virtual tape. This condition may indicate that a wrong report has been provided, condition that further leads to a total scratch of all virtual tapes found in the VDB.

System action:

The HVTUSCR utility waits for the operator's reply.

User response

Verify that the provided report is correct and contains the right active tapes report, then choose one of the following:

- Reply N, in which case the HVTUSCR utility will not scratch any virtual tape and will end with a completion code of 8.
- Reply Y, in which case the HVTUSCR utility will continue its scratch process.

HVT336W jobname/jobid I/O FAILED DUE TO OPERATOR REQUEST

Explanation

An I/O request from the job failed because the operator replied 'N' to one of the HVT317W, HVT319W, HVT323W, HVT324W or HVT413W messages. The following variables are listed in the message:

jobname

The jobname of the job whose I/O failed.

jobid

The job ID of the job whose I/O failed.

System action:

The job accessing the virtual tape is EQC-ed.

User response:

None

HVT337S resource name IS ALREADY SERVED BY SERVER serverid

Explanation

During VTE server initialization, VTE found that one of its resources is already used by another VTE server. The following variables are listed in the message:

resource

Possible values are:

SERVERID

The VTE serverid (as specified in the IDENT initialization statement).

ESOTERIC

The VTE esoteric name (the ESOTERIC parameter value in the VDEVICE initialization statement).

SUBSYS

The VTE subsystem name (the NAME parameter value in the SUBSYS initialization statement).

QUAL1

The high level qualifier of the virtual tape dataset name (the QUAL1 parameter value in the VDSN initialization statement).

ALIASPFX

The Parallel Access Tape (PAT) alias volume's prefix (as specified in the PAT initialization statement).

VCHPID

One of the virtual CHPIDs (as specified in the VDEVICE initialization statement).

name

The name of the resource.

serverid

The ID of the VTE server that is up and using the resource.

System action:

The newly started VTE server shuts down.

User response:

If two or more VTE servers are to be active concurrently, they must use different resources. To ensure this, make the necessary adjustments to the incorrectly defined parameters, then restart the VTE server.

HVT341I text**Explanation:**

This is an informative message. The only variable is *text*, which represents that contents of the message.

System action:

None

User response:

None

**HVT342E jobname/jobid I/O FAILED BY
serverid SERVER - EQC
GENERATED****Explanation**

The VTE server has concluded that the I/Os on a problematic virtual device should not be allowed to complete due to one of the following:

- An internal problem that prevents an I/O from proper completion.
- An I/O error on the virtual tape dataset.
- An operator request.

VTE prevents the completion of this I/O by generating repeated EQC statuses to that I/O and to any other retry I/Os sent by the application (or by MVS). The following variables are listed in the message:

jobname

The jobname of the job whose I/O should fail.

jobid

The ID of the job whose I/O should fail.

serverid

The VTE serverid (as specified in the IDENT initialization statement).

System action:

Any I/O sent to the virtual device is responded to with an internally generated EQC status. These EQC

statuses persist until the virtual tape is unloaded from the virtual device.

User response:

Check the messages preceding or following this message in the VTE server's log or HVTSOUT sysout file and act accordingly

**HVT343W serverid SERVER STOP PENDING.
ENTER 'S DEALLOC' OPERATOR
COMMAND TO LET MVS VARY
DEVICES OFFLINE****Explanation:**

The operator has issued a 'P server' operator command in order to stop the VTE server. However, at least one virtual device was online. The VTE server varied offline all the virtual devices, and is now waiting for them to go offline, after which it will shut down. The only variable in this message is *serverid*, which is the ID of the VTE server (as specified in the IDENT initialization statement).

System action:

The VTEVTE server continues to serve the jobs that are running. When all the virtual devices have gone offline, the VTE server shuts down.

User response

Wait for the jobs using the virtual devices to terminate. If one or more virtual devices remain in pending offline status but are not allocated to any running job, issue the 'S DEALLOC' operator command. This will let the system take the virtual devices offline and let VTE shut down.

If immediate shutdown is required, issue the 'F server, FORCE' operator command.

**HVT344I serverid SERVER FORCE STOP
ACCEPTED****Explanation**

The operator issued the 'F server, FORCE' operator command in order to stop the VTE server unconditionally. The VTE server shuts down. All running jobs served by the VTE server remain in pending status and may have to be FORCE-d in order to cancel them.

The already written virtual tape datasets remain in an unpredictable status and the virtual tapes remain in 'in-use' status.

The only variable in this message is *serverid*, which is the ID of the VTE server (as specified in the IDENT initialization statement).

System action:

The VTE server shuts down unconditionally.

User response

No action is required, however, the operator must deal with virtual tapes that have been written during the shut down, in one of these ways:

- Recover them using the HVTURCV utility before their next use.
- Scratch them.

HVT345E **ALESERV FUNCTION *func* FAILED,**
RC=X'*rc*', TOKEN=X'*token*',
ENVIR=*envir*

Explanation

During VTE server initialization, job run or invocation of an independent component (dynamic reallocation, resource manager, etc.) VTE detected an error in establishing or removing the cross-memory environment required for moving data between the user address space and the VTE server address space.

The following variables are listed in this message:

fun

The type of function that was attempted. Possible values are:

ADD

The error occurred during establishment of the environment.

DELETE

The error occurred during removal of the environment.

EXTRACT

The VTE server failed to extract the token of its home address space.

rc

The hexadecimal return code from the ALESERV system service.

token

The hexadecimal value of the extracted token. If asterisks are displayed, the token could not be extracted.

envir

This explains where the error occurred. Possible values are:

SERVER

In the VTE server while establishing a cross-memory connection with a job using virtual tapes.

ENF

In the VTE Event Notification Facility exit while establishing a cross-memory connection with the **VTE** server.

IRB

In the VTE IRB routine that runs in the job address space while establishing a cross-memory connection with the VTE server.

IEFUJV

In the VTE IEFUJV exit while establishing a cross-memory connection with the VTE server. This error may occur in JES3 environments only.

SUBSYS

In the VTE subsystem while establishing a cross-memory connection with the VTE server.

PURGER

In the VTE I/O purger manager while establishing a cross-memory connection with the VTE server.

IEFDB401

In the VTE Dynamic Allocation exit while establishing a cross-memory connection with the VTE server.

System action

The system action depends on the value of *envir*. Possible values are:

SERVER

Possible actions are:

- If the error occurred during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurred during a job run, that job is EQC-ed.

ENF

The server may not recognize when a virtual device becomes ready which in turn may cause delays in virtual tapes mounts.

IRB

Possible actions are:

- Newly created virtual tape datasets on disk contain the literal TEMPORAR as their 2nd or 3rd qualifier.
- Jobs writing data to virtual tapes using compress type=3 may not be able to compress the data.
- Internal job's control blocks could not be accessed. HVT383W message is further expected.

IEFUJV

Possible actions are:

- No dynamic reallocation is attempted for the job's DD statements in JES3 environments.

- No 'defer' is forced for static mounts on the VTE server's virtual devices. If the virtual devices are JES3-managed, this leads to MDS trying to mount the virtual tape (after issuing the IAT5210 message).

Since the mount is not satisfied by VTE, MDS places the job on the Verify Queue where it waits forever, leaving the virtual device allocated.

SUBSYS

Possible actions are:

- No dynamic reallocation is attempted for the job's DD statements.
- No PAT-eligibility is set for the job's DD statements.
- In JES3 environments no 'defer' is forced for mounts on the VTE server's virtual devices. If the virtual devices are JES3-managed, this leads to MDS trying to mount the virtual tape (after issuing the IAT5210 message).

Since the mount is not satisfied by VTE, MDS places the job on the Verify Queue where it waits forever, leaving the virtual device allocated.

PURGER

I/Os to virtual devices (to be purged by the system) do not pass the server internal purge process. This may cause premature I/O completion which in turn may cause jobs abend.

IEFDB401

This action is similar to the action taken for envir=SUBSYS, this time for dynamically allocated virtual devices.

User response:

Try to locate the problem using the ALESERV system service return code and correct the cause of the error. Collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT346E **Func FUNCTION FAILED FOR DISKVOL=*diskvol*, RC=*rc*, RS=*rs***

Explanation

An error was encountered while collecting the free space information for one of the following:

- A disk volume in a non-SMS-managed diskpool.
- A newly initialized disk volume in an SMS-managed diskpool.

This may have been caused by an invalid VTOC, real I/O errors on the disk, etc

The following variables are listed in the message:

func

Possible values are:

LSPACE

The error occurred while using the LSPACE system service.

CVAFDIR

The error occurred while using the CVAFDIR system service.

diskvol

The disk volume serial number on which the error was encountered.

rc

R15 contents on return from the service.

rs

The reason varies. Possible reasons are:

- For func=LSPACE always 0.
- For func=CVAFDIR, the CVAFDIR service status code (CVSTAT).

System action:

The free space calculation for the non-SMS-managed diskpool that the problematic disk volume belongs to, may be wrong. This leads to further erroneous HVT347W messages to be issued for the diskpool.

User response:

Verify that an environmental error has not occurred. If no such error has occurred, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT347W **USED SPACE ON *type name* EXCEEDS THRESHOLD, *perc%* UTILIZATION (*plname*)**

Explanation

This warning message is issued at user-defined intervals, informing that the free space exceeds a user-defined threshold on a given diskpool residing on an esoteric name or Storage Group. The message is issued separately for each diskpool defined under the various DPOOL initialization statements in the HVTPOLxx member.

In addition:

- For SMS-managed diskpools, the calculation considers all online and ENABLE / QUIESCE_ALL / QUIESCE_NEW disks grouped under the ENABLE / QUIESCE_ALL / QUIESCE_NEW Storage Group the DPOOL initialization statement defines.
- For non-SMS-managed diskpools, the calculation considers all online, STORAGE and not converted to SMS disks grouped under the esoteric name the DPOOL initialization statement define.

The following variables are listed in the message:

type

Possible *type* values are:

UNITNAME

The disks are non-SMS-managed and are grouped under an esoteric name.

STORGRP

The disks are SMS-managed and are grouped under a Storage Group.

name

Name of the above type.

perc

The percent of used space from the total available space on the above esoteric name or Storage Group.

plname

Name of the diskpool that defines the above esoteric name or Storage Group.

System action

The message is issued periodically at intervals defined in the THRESH parameter of the appropriate DPOOL initialization statement.

The message is an Immediate Action message and it is DOM-ed in the following circumstances:

- Conditions that caused the issue of previous HVT347W messages do not exist anymore. The DOM becomes effective on the next threshold interval for the respective pool.
- User-defined threshold interval expires, conditions that caused previous messages issue still exist and new messages are to be issued.
- Pools refresh. If needed, they are re-issued according to the new pools definitions.
- The VTE server shuts down.

User response:

Take appropriate actions to free space from the disks in the above diskpool.

HVT348W **NUMBER OF ACTIVE TAPES
EXCEEDS THRESHOLD, perc %
UTILIZATION (plname)**

Explanation

This warning message is issued at user-defined intervals informing that the number of active virtual tapes in a given tapepool exceeds a user-defined value defined in the respective tapepool THRESH parameter. The message is issued separately for each tapepool defined under the various TPOOL initialization statements in the HVTPOOLxx member.

Since the data, according to which the threshold values are calculated, is extracted from the VDB-In-Storage, a VDB-In-Storage update (along with the HVT354I message) may be expected along with this message.

Due to the asynchronous nature of the calculation of the threshold values and the VDB-In-Storage update processes, the HVT348W message may reflect the data acquired in the previous threshold interval.

The following variables are listed in the message:

perc

The percent of active virtual tapes from the total available virtual tapes in the tapepool.

plname

Name of the tapepool for which the above percent was calculated.

System action

The message is periodically issued at intervals defined in the THRESH parameter of the appropriate TPOOL initialization statement.

The message is an Immediate Action message and it is DOM-ed in the following circumstances:

- Conditions that caused the issue of previous HVT348W messages do not exist anymore. The DOM becomes effective on the next threshold interval for the respective pool.
- User-defined threshold interval expires, conditions that caused previous messages issue still exist and new messages are to be issued.
- Pools refresh. If needed, they are re-issued according to the new pools definitions.
- The VTE server shuts down.

User response:

Take appropriate actions to relieve the potential lack of scratch virtual tapes in the above tapepool (either by running the daily TMS cleaning utility followed by the HVTUSCR utility or by adding more virtual tapes using the HVTUTAP utility).

HVT350E **UCBLOOK FAILED FOR
DISKVOL=diskvol, RC=rc, RS=rs
(dsn)**

Explanation

During allocation of an existing virtual tape dataset residing on a non-SMS-managed diskpool, the VTE server encountered an error in the UCBLOOK system service. The following variables are listed in the message:

diskvol

The first (or only) disk volume serial number the virtual tape dataset resides on, and whose device number could not be obtained.

rc

The return code for the UCBLOOK system service.

rs

The reason code for the UCBLLOOK system service.

dsn

The virtual tape data set name.

System action:

The error is ignored and the VTE server tries to locate another diskpool whose parameters can be used to allocate the virtual tape dataset.

User response:

Try to locate the problem using the return and reason code and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT351S

**SWAREQ FAILED, RC=rc,
FUNCTION=func**

Explanation

During VTE server initialization, a utility run, dynamic reallocation, PAT processing or SRB processing, VTE encountered an error in the SWAREQ system service. The following variables are listed in the message:

rc

The reason code from the SWAREQ system service.

func

The internal function invoked by the SWAREQ system service.

System action

The system action varies, depending upon how the error occurred. Possible system actions are the following:

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.
- If the error occurs in the dynamic reallocation processing, no dynamic reallocation is performed for the job that encounters the problem. That may lead to erroneous tape device allocation (that is, virtual tapes may be allocated on real devices and vice versa).
- If the error occurs during the PAT processing, one or more DD statements will not be eligible for PAT substitution.

User response:

Try to locate the problem using the return code and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT352E

**IGGCSI00 FAILED R15=r15,
RC=rc, RS=rs, ID=modid, F=flag,
T=xtype, etype=name**

Explanation

During the VTE server run, a utility run, or PAT processing, VTE encountered an error in the IGGCSI00 system service. The following variables are listed in the message:

r15

Specifies the R15 contents on return from the IGGCSI00 system service or indicates invalid returned entries. Possible R15 contents are:

4

When *r15* contains 4, the IGGCSI00 system service failed.

8

When *r15* contains 8, the IGGCSI00 system service failed. The reason code contents are listed in the *DFSMSdfp Managing Catalogs* manual (in the "Catalog Search Interface" chapter of the User's Guide).

99

When *r15* contains 99, the IGGCSI00 system service was OK, but specific extracted entries are in error. In that case, the scan of current catalog is stopped and the next catalog is scanned so no more error entries are listed.

The return and reason codes are documented in MVS message IDC3009I. Refer to the MVS publication for that description of a particular return and reason code.

rc

The return code was returned by the IGGCSI00 system service for the scanned entry.

rs

The reason code was returned by the IGGCSI00 system service for the scanned entry.

modid

The catalog module-id that detected the error. When *r15* contains 99, the *modid* value is not relevant and it contains asterisks.

flag

When *r15* contains 99, this is the erroneous entry flag in hexadecimal. When *r15* does not contain 99, this flag value is not relevant and it contains zero.

xtype

One or more types of entries that the IGGCSI00 system service was requested to scan. Valid types are:

- U – User catalog entries

- A – Non-VSAM entries
- H – GDG entries
- X – Alias entries

etype

Type of entry whose name is listed by the *name* variable.

A

The entry is a non-VSAM entry (either specific or generic) or a GDG entry. IGGCSI00 should have scanned it through the standard catalog search path.

U

The entry is a User Catalog. IGGCSI00 should have scanned all entries in that User Catalog.

E

The entry is a non-VSAM entry or a GDG entry. It has been successfully extracted by IGGCSI00, but it was found to have an error.

name

The entry name associated with the above type.

System action

Possible system actions are:

- If the error occurs in the VTE server, the server assumes some defaults and the job, in which the error occurred, continues processing
- If the error occurs in a utility run, that utility terminates.
- If the error occurs during the PAT processing, one or more DD statements will not be eligible for PAT substitution.

User response

Try to locate the problem using the return and reason codes, and correct the cause of the error. If a virtual tape remained in an interim e-Vault status, it can be moved back to a stable state ('local', 'vaulted' or 'safe') using the HVTURCV and HVTUSCR utilities.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT353S VIRTUAL DEVICE NUMBERS (*cuu1/cuu2*) NOT CONTIGUOUS

Explanation

During VTE server initialization, VTE detected that the device numbers of its virtual devices are not contiguous. The following variables are listed in the message:

cuu1

The device number of which the next is not contiguous.

cuu2

The device number that is not contiguous to the virtual device represented by *cuu1*.

System action:

The initialization phase fails and the VTE server shuts down.

User response

Make sure that the device numbers of the VTE virtual devices defined under the VTE esoteric name (ESOTERIC parameter of the VDEVICE initialization statement) are contiguous. If needed, make new HCD definitions for the VTE esoteric name, then ACTIVATE the new I/O configuration definition.

When finished, restart the VTE server.

HVT354I *serverid* SERVER VDB-IN- STORAGE UPDATE COMPLETED

Explanation

This informative message is issued after the virtual tape volumes list, located in the VTE server's private storage, has been updated. This update occurs in one of the following cases:

- During VTE server initialization.
- Whenever an 'unknown' virtual tape is added dynamically to the VDB (as a result of adding, recovering or receiving an e-Vaulted virtual tape) in any one of the VTE servers or utilities in a VTE complex.
- Whenever a 'known' virtual tape is deleted from the VDB in any one of the VTE servers or utilities in a VTE complex.
- Whenever a virtual tape is dynamically added by a job specifying a specific volume serial number in any one of the VTE servers or utilities in a VTE complex.
- Whenever a threshold interval associated with a tapepool has expired and only in the LPAR that encountered the threshold interval expiration (regardless of it being a member of a VTE complex).

Note: When multiple TPOOL initialization statements are defined (with associated THRESH parameters), only the shortest threshold interval will trigger the VDB-In-Storage update and therefore the message will be issued once, regardless of the number of defined tapepools.

The only variable listed in this message is *serverid*, which represents the VTE server ID (as specified in the IDENT initialization statement).

System action:

None

User response:

None

HVT355I **VOLUME=volume FUNCTION=func**
status, RC=rc (text)

Explanation

This is an informative message that is issued by the HVTUTAP utility during virtual tapes addition/ deletion to/from the VDB or attribute change.

The following variables are listed in the message:

volume

The virtual tape's volume serial number.

func

Possible functions are:

ADD

A new virtual tape has been added to the VDB.

DEL

A new virtual tape has been deleted from the VDB.

MNTSPEC

An existing virtual tape has been assigned the 'Mount-On-Specific-Request-Only' attribute.

MNTALL

An existing virtual tape has been made eligible for all types of mount.

status

The operation was either successful (WAS OK) or unsuccessful (FAILED).

rc

Possible return codes are:

00

The operation was successful.

01

The virtual tape to be added already exists.

02

A VSAM error occurred while accessing the VDB.

03

The virtual tape volume serial number to be deleted does not exist.

04

The virtual tape to be deleted is not scratch, so it cannot be deleted.

05

The virtual tape is in use, so it cannot be deleted or its attributes cannot be modified.

06

The virtual tape to be deleted is in vault process.

07

The virtual tape to be assigned the 'Mount-On-Specific-Request-Only' attribute has already been assigned this attribute.

08

The virtual tape to be made eligible for all types of mounts has already been assigned this attribute.

09

The virtual tape belongs to a Read_Only mode tapepool or the VDB is in Read_Only mode.

10

The maximum number of virtual tapes in VDB was exceeded.

text

If *status*=FAILED, this is a short explanation of the error. Otherwise, this field is blank.

System action

The system action depends on the value of *status* and on the return code:

WAS OK

The operation completed successfully. No further system action is required.

FAILED

For rc=01 through 09:

The operation for the current virtual tape stops. The utility process resumes with the next virtual tape (if any).

For rc=10:

The operation for the current virtual tape stops and the utility terminates.

User response:

Check the HVT SOUT sysout file for messages explaining the error and act accordingly.

HVT356S **UCBSCAN FAILED FOR CUU=cuu,**
RC=rc, RS=rs

Explanation

One of the following events occurred:

- During VTE server initialization, VTE failed to locate and verify its handled virtual devices.
- On an attempt to allocate a virtual tape dataset on a non-SMS-managed diskpool, the VTE server failed to obtain the list of eligible disks to contain the virtual tape dataset.
- During a periodical calculation of the free space on a non-SMS-managed diskpool or storage location non-SMS-managed disks, the VTE server failed to obtain the list of disks contained in the above group.

The following variables are listed in the message:

cuu

The device number of the virtual device or the device number of the disk device for which the UCBSCAN system service failed.

rc

The return code for the UCBSCAN system service.

rs

The reason code for the UCBSCAN system service.

System action

Possible system actions are:

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs during allocation of a virtual tape dataset, the job using that virtual tape is EQC-ed.
- If the error occurs during free space calculation, the free space calculation for the non-SMS-managed diskpool to which the problematic disk volume belongs, may be wrong. This leads to further erroneous HVT347W messages to be issued for the diskpool.

User response:

Try to locate the problem using the return and reason code and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT357E ***type name* DOES NOT CONTAIN ANY DISK ELIGIBLE FOR ALLOCATION (*poolname*)**

Explanation

On an attempt to allocate a new virtual tape dataset, the VTE server detected that no eligible disk exists in the selected diskpool. The disk may or may not be managed by SMS:

- A non-SMS-managed eligible disk is a disk that is online, mounted with the STORAGE attribute, and is not converted to SMS.
- An SMS-managed eligible disk is a disk that is online and ENABLE / QUIESCE_ALL / QUIESCE_NEW defined under a Storage Group that is ENABLE / QUIESCE_ALL / QUIESCE_NEW.

The following variables are listed in the message:

type

Possible *type* values are:

UNTIME

The non-eligible disks are non-SMS-managed disks that are grouped under an esoteric name.

STORGRP

The non-eligible disks are SMS-managed disks that are grouped under a Storage Group

name

Name of the above type.

poolname

Name of the diskpool that defines the above esoteric name or Storage Group.

System action:

The job requesting the virtual tape is EQC-ed.

User response

Perform the appropriate action:

- For a non-SMS-managed diskpool, make sure that you defined the right unit in the DPOOL initialization statement, and there is at least one online, STORAGE, and non- SMS disk in the esoteric name referred to in the message.
- For an SMS-managed diskpool, make sure that you defined the right Storage Class (and optionally the right Management Class) in the DPOOL initialization statement, and there is at least one online and ENABLE / QUIESCE_ALL / QUIESCE_NEW disk in the ENABLE / QUIESCE_ALL / QUIESCE_NEW Storage Group referred to in the message.

HVT358S ***serverid1* SERVER ALLOCATED ANOTHER ENVIRONMENT (*serverid2*) file FILE**

Explanation

During VTE server initialization or a utility run, VTE detected that the VDB, journal file or history file belongs to another VTE environment.

The following variables are listed in the message:

serverid1

The VTE serverid (as specified in the IDENT initialization statement).

serverid2

The serverid to which this VDB, journal file or history file belongs.

If *serverid2*='UNKNOWN', the respective control record could not be read in order to determine to whom the respective file belongs.

file

Possible *file* types are:

VDB

The non-matching file is the VDB.

JRN

The non-matching file is the journal file.

HST

The non-matching file is the history file.

System action

Possible actions are:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.

User response:

Make sure the VDB, the journal file or the history file is initialized properly (using the correct initialization statements). Make sure that the HVTVDB, HVTJRN or HVTHST DDnames in the VTE procedure refer to the right VDB, journal file or history file, respectively.

HVT359E ***component service FAILED, RC=rc, RS=rs, VOLUME=volume***

Explanation

An internal service or a system service invoked by a VTE server component failed. This message may be accompanied by other messages that describe the action performed when this error occurred.

The following variables are listed in the message:

component

The internal server component in control when the error occurred.

service

The internal service or system service invoked by the internal component.

rc

The service return code. For system services this is the value of register 15 when control returned to the invoking component. For internal services, the return code is for internal problem determination purposes.

rs

The service reason code. For system services this is the value of register 0 when control returned to the invoking component. For internal services the reason code is always 0.

volume

The virtual tape's volume serial number handled at the time the error occurred. If blanks are displayed, either:

- The virtual tape volume serial number could not be determined.
- The service was invoked by a general component not necessarily related to a virtual tape.

System action:

This error may be accompanied by other messages describing the error origin. In most cases, the current job is EQC-ed.

User response

For system services consult the appropriate IBM publication for the return and reason codes description and act accordingly. Additional messages may give further hints on the origin of the error.

If the action to be done is unclear or if the failing service is an internal service, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT360S ***serverid SERVER ALLOCATED VDB LEVEL(lvl1), LEVEL(lvl2) IS REQUIRED***

Explanation

During VTE server initialization or a utility run, VTE detected that the VDB release does not match the current product level. The following variables are listed in the message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

lvl1

The VDB level (as extracted from the VDB control record).

lvl2

The VDB level (as expected by the VTE server).

System action

Possible system actions are:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs in a utility run, that utility terminates.

User response:

Make sure the VDB is initialized using the load library of the correct product release or the product release is at its correct PTF level. Also ensure that the HVTVDB DDname in the VTE procedure refers to the right VDB.

HVT361E ***VOLUME=volume VERIFICATION FAILED RC=rc (text)***

Explanation

The virtual tape verification done after a virtual tape mount failed. The following variables are listed in the message:

volume

The volume serial number of the virtual tape that failed the verification.

rc

Possible return codes are:

02

The virtual tape dataset extents write level does not match the appropriate VDB write level.

04

No blocks on the virtual tape or the virtual tape has not been checkpointed even once.

05

The number of virtual tape dataset allocated or used tracks (as derived from the appropriate VDB entry) does not match the actual number of allocated or used tracks (as derived from the DVD).

06

The number of blocks (as derived from the appropriate VDB entry) does not match the actual number of blocks (as derived from the DVD).

07

Structural recovery attempt failed.

text

Short explanation of the error.

System action:

The job requesting the virtual tape is EQC-ed.

User response

Use the HVTURCV utility in the HVTURCV member in the SAMPLIB library to recover the virtual tape. If the problem persists, perform one of the following actions:

- Restore an older backup of the virtual tape dataset and recover it using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
- Re-init the virtual tape (using the IEHINIT utility).
- Scratch the virtual tape.

HVT362E **VOLUME=volume, I/O ERROR ON VIRTUAL TAPE DS, RC=rc X'status' X'sense' (text)**

Explanation

During execution of a channel command on one of the virtual tape datasets, an I/O error occurred. The return code gives additional information about the error. Return codes 1 to 10 are issued when a virtual tape dataset is read/written as a result of a job using the virtual tape.

The following variables are listed in the message:

volume

The volume serial number of the virtual tape whose associated virtual tape dataset encountered the problem.

rc

Possible return codes are:

01

Conversion of a relative TTR to absolute CCHHR address on an existing disk failed.

02

An EXCP executing a WRITE command on the dataset failed.

03

An EXCP executing a READ command on the dataset failed.

04

Conversion of a relative TTR to absolute CCHHR address on a new allocated disk failed.

05

An EXCP executing a READ command on the dataset that allocated a new extent on a new disk failed.

06

An EXCP executing a WRITE command on the dataset that allocated a new extent on a new disk failed.

status

The hexadecimal device status of the failing channel command (where applicable) or 0.

sense

The hexadecimal sense of the failing channel command (where applicable) or 0.

text

Short explanation of the error.

System action:

If the error occurs when accessing a virtual tape dataset, the job using the virtual tape is EQC-ed.

User response:

Try to locate the problem using the return code and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT363I **VOLUME=volume END-OF-VOLUME FORCED, RC=rc (text)**

Explanation

This is an informative message issued by the VTE server when it forces an EOVS and tape switch for a virtual tape that did not yet reach its maximum capacity (as defined in the TAPECAP parameter of the TPOOL initialization statement). Switching a virtual tape whose maximum capacity is reached is considered to be a normal event and no special message is issued for it. The rc value details the reason the EOVS force was initiated.

The following variables are listed in the message:

volume

The virtual tape volume serial number for which the EOVS was forced.

rc

Possible return codes are:

01

The current written disk extent is the last available disk extent for that virtual tape dataset in the diskpool, but the dataset needs more extents in order to reach the virtual tape maximum capacity. This situation can occur when the disks selected to contain the virtual tape dataset are almost full and could not contain 16 extents (as they should have). This tape EOVS force avoids an Sx37 abend that may occur when the server will attempt to acquire a new disk extent.

02

The number of blocks on the virtual tape exceeded the 4,194,304 limit. The EOVS force avoided a IEC537I message and job abend.

text

Short explanation of the error.

System action:

The virtual tape is unloaded and a new scratch virtual tape is mounted. The net result is that the virtual tape that encountered this situation contains less data than a regular virtual tape from the same tapepool.

User response:

None

HVT364S **EXTERNAL UTILITY *utility* FAILED, RC=*rc***

Explanation

One of the VTE utilities tried to use an external standard utility. However, it ended with a non-zero return code. The following variables are listed in the message:

utility

The external utility name that was invoked by the VTE utility.

rc

The return code returned generated by the above routine.

System action:

The VTE utility terminates.

User response

As this seems to be an environmental problem, investigate why the external utility failed. The external

utility may have issued additional messages on the SYSPRINT or SYSOUT sysout files.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT365S **MULTIPLE VIRTUAL TAPE DATASETS FOR VOLUME=*volume* (*dsn*)**

Explanation

During the recovery process, the HVTURCV utility attempted to locate the virtual tape dataset containing the data of the virtual tape being recovered. However, more than one dataset adhering to the virtual tape dataset's naming convention has been found. HVTURCV is unable to decide which is the right dataset containing the virtual tape dataset.

The following variables are listed in the message:

volume

The virtual tape volume serial number that has multiple virtual tape datasets.

dsn

The name of the dataset that seems to be the associated virtual tape dataset.

System action:

The HVTURCV utility is terminated. No virtual tape is recovered.

User response

Enter ISPF 3.4 screen and list all datasets whose dsname high-level qualifier is similar to the QUAL1 parameter's value as specified in the VDSN initialization statement.

Find all dataset names whose 2nd or last qualifier (depending on whether QUAL2 or QUAL3 parameters is %VOL) is Vvolser.

You probably have more than one dataset like this. Investigate why this happened, delete the unnecessary dataset(s), then rerun HVTURCV utility in the HVTURCV member in the SAMPLIB library.

If the QUAL1 parameter of the VDSN initialization statement is not dedicated to VTE use, you must consider renaming all non-virtual tape datasets so that QUAL1 will be dedicated to virtual tape datasets only.

HVT366S **INVALID report REPORT, RC=*rc* (*text*)**

Explanation

The scratch report of an external TMS did not contain the expected data. This was encountered when the HVTUSCR utility attempted to analyze the scratch report prior to the SYNC process.

The following variables are listed in this message:

report

Valid *report* types are:

CA1/TLMS

The report was produced by CA1 or CA-Dynam/TLMS.

CONTROLT

The report was produced by CONTROL-T.

RMM

The report was produced by DFSMSrmm.

ZARA

The report was produced by ASG-Zara.

FTP

The report was produced by FTP.

CTC

The report was produced by Cloud Tape Connector (CTC).

rc

Valid return codes are:

01

For all reports except RMM and FTP, the report does not contain a title recognized by HVTUSCR.

02

For all reports except FTP, the volsers in the report are not sorted in ascending order.

03

One of the following occurred:

- The report is null.
- No valid volume records were encountered.
- All volumes in TMS are scratch.
- No volume is in the FTP off-site storage location.

04

For the RMM report, the tape status in the RMM TCDB is not one of the expected statuses. The expected statuses are MASTER, SCRATCH, INIT, USER, ENTRY.

05

For all reports except RMM, the report does not reside in a temporary file. The report should have been created in a previous step of the job and passed to the HVTUSCR utility via a temporary file.

06

The RMM report produced by the EDGHSKP utility and analyzed by the HVTUSCR utility failed one of the following validity checks:

- The 'H' record was not detected.

- Unexpected date format on the 'H' record.
- Invalid date on the 'H' record.
- Non-numeric date on the 'H' record.
- Bad Julian date (if FORMAT=J) on the 'H' record.

11

The HVTUSCR utility could not access the remote off-site storage location because the e-Vault option is disabled (VLT=DISABLE in HVTOPTxx member).

12

The FTP program could not be invoked because one of its input files could not be allocated. Message HVT311E, which precedes this message, gives more details on the failure reason.

13

The record format of one of the FTP program files is not fixed.

14

The LOCID parameter value could not be located in any of the VLTRMT initialization statements.

15

The FTP off-site storage location contains more than 150000 datasets

16

One or more virtual tapes contain duplicate virtual tape datasets. Message HVT365S, which precedes this message, gives more details on the affected virtual tapes.

17

The FTP program invocation failed.

text

Short explanation of the error.

System action:

The HVTUSCR utility is terminated. No virtual tape is scratched.

User response

The recommended operator action depends on the return code:

01

Make sure the HVTUSCR utility's REPIN DDname points to the EARL, CTRRPT or ZARAUTL report originally provided with VTE.

For EARL reports that list the TMC or VMF contents, the title may be missing, indicating there are no active tapes at all; however, this condition is not considered to be an error condition. HVTUSCR utility will function normally and scratch virtual

tapes as soon as at least one active tape is found in the TMC/VMF.

02

Add an IEBGENER step after the HVTUSCR step in the HVTUSCR job and print the contents of the temporary dataset allocated to SORTOUT DDname. Make sure the report contains volsers in an ascending sorted order.

The following reasons can affect the list sorting:

- The report is indeed not sorted. In that case, check why the TMS listing utilities (EARL, CTTTRPT, EDGHSKP or ZARAUTL) did not sort it.
- The HVTUSCR utility failed to analyze the produced report correctly and extracted invalid data as valid volsers. In that case, collect and forward the appropriate documentation to IBM Technical Support for assistance.

03

Make sure the HVTUSCR utility's REPIN DDname points to the TMS utility (EARL, CTTTRPT, EDGHSKP or ZARAUTL) report originally provided with VTE and these utilities ended with a return code of zero.

04

Do one of the following:

- Make sure the HVTUSCR utility's REPIN DDname points to the EDGHSKP report originally provided with VTE and it ended with a return code of zero.
- If a new status (except MASTER, SCRATCH, INIT, USER, ENTRY) has been introduced by a new release of RMM, collect and forward the appropriate documentation to customer support for technical assistance.

05

Make sure the HVTUSCR utility's REPIN DDname points to a temporary file that contains the report created by the EARL, CTTTRPT, ZARAUTL or EDGHSKP utilities originally provided with VTE. Refer to the appropriate sample in the VTE SAMPLIB library for the right input to the HVTUSCR utility.

06

Make sure the HVTUSCR utility's REPIN DDname points to the EDGHSKP report originally provided with VTE and it ended with a return code of zero.

11

Specify HVT=ENABLE in the HVTOPTxx member.

12

Investigate the error according to the accompanying HVT311E and IKJ56228I messages. If the error is not detected, please collect and forward the appropriate

documentation to IBM Technical Support for assistance.

13

Please collect and forward the appropriate documentation to IBM Technical Support for assistance.

14

Probably user error. Make sure the specified LOCID parameter value matches the LOCID parameter of a VLTRMT initialization statements in the HVTVLTxx member.

15

Probably user error. Enter the FTP site and manually delete unneeded datasets.

16

Probably user error. Enter the FTP site and manually delete unneeded duplicate datasets.

17

The HVTTRC sysout file of the job contains the FTP program sysout. If the error is not located, please forward the appropriate documentation to IBM Technical Support for assistance.

HVT367S

*****ONE OF THE FIRST/LAST
VALUES IS BAD (first/last)**

Explanation

During VTE server initialization, refreshing of the HVTPOLxx member or a utility run, VTE detected that an initialization statement contains a bad volser range (for example, the first volser is higher than the last or the volser lengths are not equal).

The following variables are listed in the message:

first

The first volser supplied in the initialization statement.

last

The last volser supplied in the initialization statement.

System action

Possible system actions are:

- If the error occurs during VTE server initialization, the initialization phase continues to parse the remaining initialization statements, then the VTE server shuts down.
- If the error occurs during the refresh of HVTPOLxx member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminates.

User response:

Correct the wrong initialization statement, then restart the VTE server, retry the refresh or rerun the utility.

**HVT368I PROCESSING ENTRIES FROM
UCAT=*ucat***

Explanation

This is an informative message that is issued by the HVTUSCR utility after it completes the scanning of a given user catalog for cataloged datasets residing on virtual tapes. The message is issued when the utility runs with the TYPE=CATALOG option.

The only variable in this message is *ucat*, which represents the user catalog name.

System action:

None

User response:

None

**HVT369S INVALID RMM REPORT, RC=*rc*
(*reason*), RMM=*value1*,
USCR=*value2***

Explanation

The scratch report of RMM did not contain the expected data. This was encountered when the HVTUSCR utility attempted to analyze the scratch report prior to the SYNC process.

The following variables are listed in this message:

rc (return code)

07 – The RMM report that was produced by the EDGHSKP utility and analyzed by the HVTUSCR utility has failed.

reason

Valid reasons are:

- BACKDATE – The report was created before the start of the HVTUSCR utility run.
- BAD SID – The report was created in another system. (SID is system ID.)

value1 and value2

These are comparison values for the RMM report and for the HVTUSCR utility run.

- value1 – For BACKDATE, *value1* is the date and time of the RMM report creation. For BAD SID, it is the system identifier where the RMM report was created.
- value 2 – For BACKDATE, *value 2* is the date and time of the HVTUSCR utility run. For BAD SID, it is the system identifier where the HVTUSCR utility was run.

System action:

The HVTUSCR utility is terminated. No virtual tape is scratched.

User response

The recommended operator action depends on the reason:

- BACKDATE – Make sure the EDGHSKP utility runs as a preceding step in the same job that executes the HVTUSCR utility.
- BAD SID – Make sure the DFSMSrmm is running on the same system where the HVTUSCR utility runs. The DFSMSrmm SYSID (*system_name*) is defined in the DFSMSrmm EDGRMMxx parmlib OPTION SYSID operand.

**HVT370I VOLUME=*volume* START OF '*phase*'
PHASE FOR VDB UPDATE**

Explanation

This is an informative message issued by the HVTURCV utility on the start of each virtual tape recovery. The following variables are listed in the message:

volume

The virtual tape volume serial number being recovered.

phase

Valid *phase* values are:

CORRELATING

Both the virtual tape dataset and the VDB entry exist, their correlating process starts.

MOVING TO SCRATCH

The virtual tape dataset was not found while the VDB entry existed. Therefore the VDB entry will be set to mark a 'scratch' virtual tape.

CREATING NEW VDB ENTRY

The virtual tape dataset was found while its corresponding VDB entry misses. A new VDB entry is created to describe the virtual tape.

System action:

Processing continues.

User response:

None

**HVT371I INTERIM EXTENTS STATUS, RC=*rc*
(*text*)**

Explanation

This is an informative message that is issued during a virtual tape recovery (either from the VTE server or from the HVTURCV utility) describing the action that was taken. The following variables are listed in this message:

rc

Possible return codes are:

- 00** No errors in the structural recovery. The VDB will be updated, if required for VDB recovery.
- 01** Error trying to access the disks in order to build the current virtual tape dataset structure.
- 02** The first virtual tape dataset block was not a DVD block.
- 03** No blocks on the virtual tape or the virtual tape has not been check-pointed even once.
- 04** Structural recovery cannot be completed. Message HVT385E gives further details about the problem.
- 11** An attempt to recatalog the virtual tape dataset started. This is necessary when an IPL prevented the appropriate cataloging of the virtual tape dataset.
- 12** The attempt to recatalog the virtual tape dataset failed due to its inability to OBTAIN the dataset information from all disks. The recovery will attempt to continue, however it may fail later.
- 13** The attempt to recatalog the virtual tape dataset failed due to a CAMLST system service error. The recovery will attempt to continue, however it may fail later.
- 14** The attempt to recatalog the virtual tape dataset successfully completed.
- 15** The highest TTR (as reflected in the virtual tape dataset DVD) is higher than the highest allocated TTR. This can be a result of not being able to allocate all the disks that contain this virtual tape dataset extent.
- 21** The recovery process found this virtual tape to be in error. However, later on this virtual tape was found to reside in a vault location, so the error bit is reset (the virtual tape is safe in its vault location).
- 22** The recovery process found this virtual tape does not have an associated virtual tape dataset on disk and, therefore, should be

marked scratch. However, the tape is in an interim vault status and its associated virtual tape dataset is safe in a vault location. Therefore, the tape is marked as 'vaulted'.

- 23** The virtual tape dataset could not be accessed (for example, its recall from a real tape or its allocation failed).
- 24** The VDB should be updated after a virtual tape recovery built the internal VDB entry. However, the VDB is in Read_Only mode.
- 25** A new entry should have been added to the VDB. However, the number of virtual tapes in VDB exceeded the allowed maximum.

text
Short explanation of the error.

System action
The system action depends on the return code:

- For rc=01 through 24:
If the operation is invoked by the HVTURCV utility, the process resumes with the next virtual tape (if any).
- For rc=25:
If the operation is invoked by the HVTURCV utility, the utility terminates.

User response:
None

HVT372I	VIRTUAL TAPE DATASET / VDB ALREADY CORRELATED, NO VDB UPDATE WILL TAKE PLACE
----------------	---

Explanation:
During the recovery of a virtual tape, the HVTURCV utility found that the virtual tape dataset and its corresponding VDB entry are already correlated. This volume recovery is skipped.

System action:
None

User response:
None

HVT373I	UPDATE OF THE VOLUME ENTRY IN THE VDB comp
----------------	---

Explanation:
The HVTURCV utility finished the recovery of a virtual tape and tried to update the relevant volume entry in the VDB. The only variable listed in this message is *comp*, which indicates whether the VDB volume entry update was successful (WAS OK) or unsuccessful

(FAILED). If it failed, other messages will explain the failure reasons.

System action:

The recovery of the current virtual tape ends. Next virtual tape (if any) recovery starts.

User response:

If the VDB update failed, try to investigate the reason. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT374I VAULT STATUS SET TO *status*

Explanation

This is an informative message that is issued during a virtual tape recovery (when the virtual tape is in an interim e-Vault status), describing the new stable e-Vault status to which the virtual tape has been moved. The only variable in this message is *status*, which describes the status of the virtual tape. Possible values are:

VAULTED

The virtual tape has been moved to 'vaulted' status. This means that the virtual tape dataset exists only in the remote off-site storage location.

NULL

The virtual tape has been moved to 'no vault' status. This means that the virtual tape dataset does not exist in the offsite storage location. It should exist in the local location.

LOCAL

The virtual tape has been moved to 'local' status. This means that the virtual tape dataset exists in the local location.

SAFE

The virtual tape has been moved to 'safe' status. This means that the virtual tape dataset exists in the off-site storage location.

System action:

None

User response:

None

HVT375E *jobname/jobid* FAILED DUE TO INSUFFICIENT SPACE ON *type name (poolname)*

Explanation

While trying to extend a virtual tape dataset (as a result of a write request) the VTE server detected that no more space was available on the disks eligible for virtual tape dataset allocation in the specified diskpool.

The following variables are listed in the message:

jobname

The jobname of the job whose virtual tape dataset had to be extended.

jobid

The ID of the job whose virtual tape dataset had to be extended.

type

Possible *type* values are:

UNITNAME

The disks are non-SMS-managed disks and are grouped under an esoteric name.

STORGRP

The disks are SMS-managed disks and are grouped under a Storage Group.

name

Name of the above type.

poolname

Name of the diskpool that defines the above esoteric name or Storage Group.

System action

The job using the virtual tape is EQC-ed. When an application tries later to read the virtual tape, it will encounter an EOD status when attempting to read beyond the last written point.

Note that due to the inability to complete the last write requests, the virtual tape misses its trailer label.

User response:

Make sure that sufficient space exists on the diskpool eligible for virtual tape dataset allocation. You may need to add new volumes to that diskpool or run the daily scratch process more frequently.

HVT376E VOLUME=*volume* VIRTUAL TAPE DS ON DISK ALREADY EXISTS (*dsn*)

Explanation

When allocating a new virtual tape dataset (either as a result of a specific request or scratch request) the VTE server found that a similar virtual tape dataset already exists. This situation may happen for one of these reasons:

- The VDB has been updated manually and it does not reflect the real virtual tape status. For example:
 - A non-scratch virtual tape has been marked as representing a scratch virtual tape.
 - A VDB entry has been deleted.
- The newly allocated virtual tape dataset status (DISP) has been changed from MOD to NEW by an internal product or exit. New virtual tape datasets

are allocated with DISP=(MOD, CATLG) and should remain so.

The following variables are listed in the message:

volume

Specifies the volume serial number of the tape whose associated virtual tape dataset already exists.

dsn

Specifies the dataset name of the virtual tape whose allocation failed.

System action:

The job requesting the virtual tape is EQC-ed.

User response

Investigate the discrepancy that exists between the VDB and the relevant virtual tape dataset on disk, then based on what you find, perform one of the following:

- Recover the virtual tape dataset (in order to recreate the appropriate VDB entry) using the HVTURCV utility in the HVTURCV member in the SAMPLIB library.
- Manually delete the virtual tape dataset that is no longer needed.

Also check that an installation exit (like IEFDB401) or another product does not, inadvertently, modify the dynamic allocation text unit DALSTATS from MOD to NEW.

HVT377E **SMS REQ FAILED, V=volume, F=func, RC=rc, RS=rs, storclas/mgmtclas, type=name**

Explanation

An invalid return code has been received from the SMS subsystem on an attempt to perform one of these actions:

- Allocate a new or existing virtual tape dataset on SMS-managed disks.
- Locate a diskpool that contains an existing virtual tape dataset that has been moved from its original diskpool.
- Calculate the free space percentage on the SMS-managed disks that contain virtual tape datasets.
- Dynamically reallocate a dynamically allocated dataset.

The following variables are listed in this message:

volume

The volume serial number of the virtual tape whose associated virtual tape dataset allocation encountered the problem. If asterisks are

displayed, the volume serial number could not be determined.

func

The internal function that encountered the error. Possible values are:

GETCLAS

An attempt was made to obtain the SMS classes of an allocated virtual tape dataset or of a dynamically reallocated dataset.

VERCLAS

An attempt was made to validate an SMS class definition.

COUNT

An attempt was made to do one of the following:

- Count the number of SMS-managed disks eligible for the virtual tape dataset allocation.
- Calculate free space percentage on Storage Groups containing virtual tape datasets.

rc

Return code from the IEFSSREQ system service used to access the SMS subsystem. The return code values are listed in the IEFSSOBH macro.

rs

Reason code from the SMS subsystem. The SMS reason codes are listed in the *DFSMSdfp Diagnosis Reference* manual (in the "SMS Diagnostic Aids" chapter).

storclas

Storage Class value (or blank) used as input to the SMS subsystem.

mgmtclas

Management Class value (or blank) used as input to the SMS subsystem.

type

Type of entry that encountered the error:

BS

Base Configuration

DC

Data Class

DS

Data Set

MC

Management Class

SC

Storage Class

SG

Storage Group

VL

Volumes list in a given Storage Group

name

Name of the entry. If *type=DS*, only the first 24 characters of the dataset name are displayed. If asterisks are displayed, the entry name could not be determined or is not relevant.

System action

The system action varies, depending on when the error occurs in the process. Possible system actions are:

- If the error occurs during allocation of a new or existing virtual tape dataset, the job accessing the virtual tape is EQC-ed.
- If the error occurs during locating the diskpool that contains an existing virtual tape dataset that has been moved from its original diskpool, the error is ignored, and the locate resumes with the next diskpool.
- If the error occurs during calculation of free space percentage on SMS-managed disks that contain virtual tape datasets (diskpool), the calculation for the above group is not done. This may further lead to jobs abend, due to lack of disk space without prior notice.
- If the error occurs during the dynamic reallocation of a dynamically allocated dataset, the dynamic reallocation is not performed.

User response:

Try to locate the error using the SMS return code and reason code. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT378E BAD SMS CONSTRUCT
VOLUME=volume, type=name,
RC=rc (text)

Explanation

On an attempt to allocate a new or existing virtual tape dataset on an SMS-managed disk, the VTE server detected that one of the SMS classes assigned to the dataset contains unsupported options.

The following variables are listed in the message:

volume

The virtual tape volume serial number whose associated virtual tape dataset allocation encountered the problem.

type

Type of entry that encountered the error. Possible values are:

DC

Data Class

MC

Management Class

SC

Storage Class

name

Name of above entry.

rc

The return code generated by this error. Possible values are:

01

A Data Class is associated with the virtual tape dataset. No Data Class is supported for virtual tape datasets.

11

The Management Class has a value other than NOLIMIT for one of the following options:

- Expire after Days Non-usage
- Expire after Date/Days

The above options should have a value of NOLIMIT.

12

The Management Class 'Partial Release' option is YES. The 'Partial Release' option should have a value of NO.

21

The Storage Class 'Guaranteed Space' option is YES. The 'Guaranteed Space' option should have a value of NO.

System action:

The job accessing the virtual tape is EQC-ed.

User response:

Modify the SMS class option in error, then rerun the job.

HVT379E IGWASMS SERVICE FAILED
VOLUME=volume, RC=rc, RS=rs,
PD=prob1/prob2

Explanation

On an attempt to determine an existing virtual tape dataset SMS attributes, the IGWASMS system service encountered an error. The following variables are listed in this message:

volume

The volume serial number of the virtual tape whose associated virtual tape dataset encountered the error.

rc

The return code from IGWASMS system service. The return codes are listed in the *DFSMSdfp Advanced Services* manual.

rs

The reason code from IGWASMS system service.
The return codes are listed in the *DFSMSdfp Advanced Services* manual.

prob1

The first word of the returned prob_det field.

prob2

The second word of the returned prob_det field.

System action

The system action varies depending on why the problem occurred. Possible actions are:

- If the problem occurs while the VTE server allocates a virtual tape dataset, the job accessing the virtual tape is EQC-ed.
- If the problem occurs while HVTURCV tries to recover a virtual tape, the recatalog attempt of the virtual tape dataset is skipped.
- In rare cases (when a system crash occurred before the VTE server cataloged an allocated non-SMS-managed virtual tape dataset), this may result in a failure in the virtual tape recovery, making it 'unusable'. The recovery proceeds with the next virtual tape (if any).

User response:

Try to locate the error using the IGWASMS system service return code and reason code. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT380I

**VOLUME=*volume* STRUCTURAL
RECOVERY STARTED**

Explanation

One of the VTE components detected that an existing virtual tape dataset changed its disk location or layout. As a result, the internal structural recovery mechanism tries to re-adjust the internal dataset pointers in order to reflect the new structure. The structural recovery is invoked by the following components:

- VTE server, when a 'known' non-scratch virtual tape is mounted.
- HVTURCV utility, on each recovered virtual tape (even those that their VDB entry is correlated).
- E-Vault process, when an e-Vaulted or returned virtual tape is copied to its permanent diskpool.

The only variable listed in this message, *volume*, represents the volume serial number of the virtual tape.

System action:

None

User response:

None

HVT381I

**VOLUME=*volume* STRUCTURAL
RECOVERY FAILED**

Explanation

This is an informative message indicating that during a virtual tape and its associated virtual tape dataset verification, one of the VTE components detected that the virtual tape dataset changed its disk location or layout. However, the internal structural recovery process that was invoked to re-adjust the layout pointers failed. Messages HVT385E and HVT371I (appearing only in the HVTOUT sysout file) give more information about the error.

The structural recovery is invoked by the following components:

- The VTE server, when a 'known' non-scratch virtual tape is mounted.
- The HVTURCV utility, on each recovered virtual tape (even those with a VDB entry that is correlated).
- The e-Vault process, when an e-Vaulted or returned virtual tape is copied to its permanent diskpool.

The following variable is listed in the message:

volume

Specifies the volume serial number of the virtual tape.

System action

Possible system actions are:

- If the structural recovery is done by the VTE server, the job requesting the virtual tape is EQC-ed.
- If the structural recovery is done by the HVTURCV utility, the virtual tape is marked as 'unusable'. The utility itself resumes the recovery with the next requested virtual tape (if any)
- If the structural recovery is done by the e-Vault process, its process for the current virtual tape is suspended. The e-Vault process on the suspended virtual tape will be re-attempted later.

User response

If the structural recovery fails, do one of the following:

- If the virtual tape resides on a Read_Only mode tapepool or is defined in a Read_Only mode VDB, it cannot be recovered. In that case, you may want to remove the Read_Only mode of the tapepool or the VDB and (provided the disks are not physically set to Read_Only mode) retry the recovery.
- Restore an older backup of the virtual tape dataset and recover it using the HVTURCV utility in the HVTURCV member in the VTESAMPLIB library .

- Re-init the virtual tape (using the IEHINIT utility).
- Scratch the virtual tape.

HVT383W **VOLUME=volume INTERNAL CONTROL BLOCK (block) COULD NOT BE OBTAINED (jobname/jobid), RC=rc**

Explanation

During virtual tape mount or unload, the VTE server could not use cross-memory services to access the job's address space in order to obtain data from one or more of its internal control blocks. The most common cause of this situation are the following:

- The job is abending or is FORCE-d by the operator.
- The job is stuck.
- The job is 'logically swapped out' on a busy system (despite the fact that it is performing I/O).

Due to that condition, the cross-memory service cannot complete.

The following variables are listed in the message:

volume

The volume serial number of the virtual tape for which the control block obtain was attempted.

block

Possible values for the type of control block are listed below:

JFCB

The JFCB control block, which is mainly used to obtain the first tape dataset name and the label type, could not be obtained.

VTE uses the JFCB to obtain:

- The first tape dataset name for building the virtual tape dataset name.
- The optionally modified (by means of OPEN TYPE=J) first tape dataset name for the HVTUDST utility.
- The label type for verifying that the virtual tape label type is consistent with the requested type.
- The updated TRTCH value for deciding whether the virtual tape is eligible for compression.

JMR

The JMR control block could not be obtained. VTE uses the JMR to obtain the job's reader date and time and other constants used for building the VTE SMF record.

jobname

The jobname of the job whose control block could not be obtained.

jobid

The jobid of the job whose control block could not be obtained.

rc

Possible return code values are:

01, 04

The mount process could not access the job's JFCB control block.

02

The mount process could not access the job's JMR control block.

03

The unload process could not access the job's JFCB control block.

System action

The system action depends on the return code:

01

For new output virtual tapes, the respective virtual tape dataset name qualifier (defined in the VDSN initialization statement as QUALn=%TDSN) will be TEMPORAR.

For all virtual tapes, the HVT332E message with RC=15 or RC=16 (invalid label type) may be further expected. Also already EQC-ed (Equipment Checked) jobs may do more attempts to retry and recover from the server-initiated EQC condition.

02

The VTE created SMF record will contain a null header. Any write activity is suppressed and the HVT115E message issued.

03

Already EQC-ed jobs may do more attempts to retry and recover from the server-initiated EQC condition.

04

For output virtual tapes, virtual tape compression may not be initiated. Optionally, modified first tape dataset name (by means of OPEN TYPE=J) is not recorded.

In all cases, the job executes as directed by conditions that existed prior to that error detection, that is, in most cases the job will continue its normal execution.

User response:

None

HVT385E **VOLUME=volume STRUCTURAL type NOT OK, RC=rc, VSN=vsu (text)**

Explanation

A diagnostic message issued during virtual tape verification and structural recovery process (either by

the VTE server or the HVTURCV utility). This message may appear one or more times with different return codes (and texts), depending on the detected error condition. Note that message HVT371I may appear on the HVTOUT sysout file giving more details.

The following variables are listed in the message:

volume

The volume serial number of the virtual tape.

type

Possible *type* values are:

FORMAT

A diagnostic message issued during virtual tape verification and structural recovery process (either by the VTE server or the HVTURCV utility). This message may appear one or more times with different return codes (and texts), depending on the detected error condition. Note that message HVT371I may appear on the HVTOUT sysout file giving more details.

RECOVERY

An error was detected during the structural recovery.

rc

Possible return codes are:

01

Open of the virtual tape dataset on the specified disk failed.

02

Read of the virtual tape dataset on the specified disk failed.

03

Write to the virtual tape dataset on the specified disk failed.

04

Close of the virtual tape dataset on the specified disk failed.

05

The structural validation process found that the write level in the dataset extents on one of the disks (that is part of the multi-volume dataset) is different than the write level in the extents on the other disks.

07

The structural validation process found that one of the disks containing the dataset extents was out of the expected sequence.

08

The structural validation found one of the following errors:

- Not all disks on which the virtual tape dataset resides are on the same write level.

This might have been caused by restoring an old disk that contains a current dataset extent.

- The disks on which the dataset resides are not in an ascending collating sequence.

09

During the structural validation process, the block that contained the current volume description was not found in its expected location. This may be the result of restoring the dataset onto an unlike device not on a track by track basis.

10

The virtual tape structured recovery could not update the virtual tape dataset since it resides on a Read_Only mode tapepool.

dsn

The disk sequence number where the error occurred.

text

Short explanation of the error.

System action

Possible system actions are:

- If the error was encountered during virtual tape verification, no recovery is attempted.
- If the error was encountered during virtual tape recovery, the recovery fails.

In both cases, message HVT381I follows.

User response:

See message “HVT381I” on page 543 for the appropriate operator action.

HVT387S

**VOLUME=volume NO DPOOL
CONTAINS DISKVOL=diskvol FOR
DS=dsn**

Explanation

An existing virtual tape dataset could not be allocated with the UNIT parameter values of the DPOOL initialization statement pointed to by the appropriate tapepool (and listed in the preceding HVT315I message).

VTE tried to locate another DPOOL initialization statement whose UNIT parameter values could be used to allocate the existing virtual tape dataset, however, no such initialization statement was found.

The following variables are listed in the message:

volume

The volume serial number of the mounted virtual tape.

diskvol

The first (or only) disk volume serial number on which the virtual tape dataset resides.

dsn

The virtual tape dataset name.

System action:

The job whose virtual tape dataset allocation failed is EQC-ed by the VTE server.

User response

Check the TPOOL and DPOOL initialization statements:

- Make sure that the tapepool the virtual tape volume serial number belongs to points to a valid diskpool.
- Make sure the Storage Class associated with the existing virtual tape dataset is defined (or derived) from the UNIT parameter of one of the defined DPOOL initialization statements.
- Make sure that the UNIT parameters of one of the defined DPOOL initialization statements contain the disks volume serial numbers on which the virtual tape dataset resides.
- If the virtual tape dataset has been moved to disks other than those contained in the above diskpool (and no diskpool defines them) add a new DPOOL initialization statement to contain the above disks, then refresh the HVTPOOLxx member.

HVT388I **VOLUME=volume VIRTUAL TAPE
DS RECALL status FOR DSN=dsn**

Explanation

This is an informative message issued by the VTE server when it initiates the recall of a virtual

tape dataset. This virtual tape dataset has been previously migrated and needs to be recalled as a result of the associated virtual tape mount.

The following variables are listed in the message:

volume

The volume serial number of the virtual tape whose associated virtual tape dataset recall is initiated.

status

Possible *status* values are:

STARTED

The recall process started.

ENDED

The recall process ended successfully.

FAILED

The recall process failed.

dsn

The data set name of the recalled virtual tape.

System action:

VTE waits for the recall completion and the virtual tape mount is suspended. When the recall completes, the mount process resumes. If the recall fails, the job using the virtual tape is EQC-ed.

User response

The recommended action depends on the *status* value.

If the recall FAILED, review the disk manager (DFSMSHsm STC, FDRABR's SYNRECAL procedure or CA-Disk's DMSAR procedure) log for error messages and act accordingly. The HVT359E message may precede this message and give further details about the reason the recall failed.

HVT390I **VOLUME=volume MARK TO BE func
direction LOCID=locid IN
VSID=vsid, RC=rc (text)**

Explanation

This is an informative message issued by the HVTUSCR utility indicating the e-Vault status set for a virtual tape. The following variables are listed in the message:

volume

The volume serial number of the virtual tape to be marked for e-Vault or return.

func

Indicates whether the virtual tape is to be marked for e-Vault (VAULTED) or marked for return (RETURNED).

direction

Possible values are:

TO

Used in conjunction with *func*=VAULTED.

FROM

Used in conjunction with *func*=RETURNED.

locid

The location name this virtual tape is to be e-Vaulted to or returned from.

- If *locid*=*MAIN*, the true location name is blank.
- If *locid*=*RECRTD*, this is a temporary location name set by HVTURCV (in case the virtual tape has been recovered). The real location name will be reset as soon as the virtual tape will be e-Vaulted or returned.

vsid

The system identifier that is eligible to process the vault request. If the current system does not share a VTE complex, the *vsid* value represents the system identifier of the current system.

rc

The return code generated by the process.
Possible values are:

00

The e-Vault setting was OK.

01

The e-Vault setting failed. The virtual tape is 'in-use'.

04

The e-Vault setting failed. VSAM error on the VDB.

11

The e-Vault setting failed. The virtual tape is already in e-Vault process.

13

The e-Vault setting failed. The tape was e-Vaulted due to TMS settings.

14

The e-Vault setting failed. The locid is not known.

15

The e-Vault setting failed. The virtual tape is scratch.

16

The e-Vault setting failed. The virtual tape resides on a Read_Only mode tapepool or the VDB is in Read_Only mode.

17

The e-Vault setting failed. A system service failure occurred and a previous message gives more details.

text

Short explanation of the error.

System action:

The virtual tape is processed as described in the message.

User response

The recommended operator action depends on the return code:

00

None

01

If the virtual tape is not used, run the HVTURCV utility in the HVTURCV member in the SAMPLIB library to recover the virtual tape.

04

Check the HVTOUT sysout file for messages explaining the error and act accordingly.

11

Wait for the virtual tape to be fully e-Vaulted or returned, then repeat the action.

13

Possible actions include:

- None (as long as HVTUSCR gets its input from the TMS active tapes list).
- Try to return using the HVTUSCR utility with SYNC TYPE=VOLSER or by using the VTE Dialog.

14

Make sure one of the VTLRMT initialization statements defines the off-site remote storage location to which this virtual tape is to be e-Vaulted.

16

Virtual tapes that reside on Read_Only mode tapepools or are defined in a Read_Only mode VDB cannot be e-Vaulted. If you are aware of this virtual tape e-Vault and still want to e-Vault it, remove the Read_Only mode of the tapepool or the VDB and (provided the disks are not physically set to Read_Only mode), rerun the HVTUSCR utility.

17

The e-Vault setting failed. A system service failure occurred and a previous message gives more details.

If the action to be done is unclear, collect and forward the appropriate documentation to Technical Support for assistance.

HVT391I

VOLUME=volume VAULT
INDICATORS RESET RC=rc (text)

Explanation

This is an informative message issued by the HVTUSCR utility indicating that virtual tape e-Vault indicators have been reset as a result of running the utility with the TYPE=VFTP option.

The following variables are in this message:

volume

The volume serial number of the virtual tape whose e-Vault indicators have been reset.

rc

Return code

00

Indicators reset was OK.

01

Indicators reset failed. The virtual tape is 'in-use' (either being used in other LPAR or left 'in-use' due to a previous failure).

04

Indicators reset failed. VSAM error on the VDB.

- 16**
Indicators reset failed. The virtual tape resides on a Read_Only mode tapepool or the VDB is in Read_Only mode.
- 21**
The virtual tape e-Vault indicators are OK and do not need to be reset.
- 22**
Indicators reset was OK. In addition, the virtual tape dataset name in the VDB was updated according to the name in the remote off-site storage location due to the FORCE=YES specification.
- 23**
Indicators reset was OK. In addition, the *locid* value in the VDB was updated according to the name in the remote off-site storage location due to the FORCE=YES specification.
- 24**
Indicators reset failed. The virtual tape dataset name in the VDB and the remote off-site storage location do not match.
- 25**
Indicators reset failed. The locid in the VDB and the remote off-site storage location do not match.
- 26**
Indicators reset failed. The initial virtual tape status is invalid and cannot be reset by the HVTUSCR utility. Valid statuses are:
- Tape is scratch at local
 - Tape is active and located at local.
 - Tape is active and e-Vaulted.
- 28**
Indicators reset failed. The virtual tape is an 'unknown' virtual tape (not found in the VDB). In order not to add uncontrolled data to the VDB, the virtual tape is not automatically added.

text

Short explanation of the error.

System action:

The virtual tape is processed as described in the message.

User response

The recommended operator or user action depends on the return code:

- 00**
None

- 01**
If the virtual tape is not used in any LPAR, run the HVTURCV utility in HVTURCV member in the SAMPLIB library to recover the virtual tape.
- 04**
Check the HVTSOUT sysout file for messages explaining the error and act accordingly.
- 16**
Remove the Read_Only mode of the tapepool or the VDB (provided the disks are not physically set to Read_Only mode).
- 21**
None
- 22**
None
- 23**
None
- 24**
If the virtual tape dataset name in the VDB is indeed wrong and should be replaced by the name found in the remote off-site storage location, use the FORCE=YES parameter on the SYNC TYPE=VFTP statement.
- 25**
If the locid in the VDB is indeed wrong and should be replaced by the locid found in the remote off-site storage location, use the FORCE=YES parameter on the SYNC TYPE=VFTP statement.
- 26**
Run the HVTURCV utility in order to recover the virtual tape.
- 28**
If the virtual tape is a valid virtual tape, run the HVTUTAP utility in order to add it to the VDB.

HVT392I

**VOLUME=volume COPY FROM
source TO target STARTED**

Explanation

This is an informative message indicating that HVTUV2V utility started a new volume-to-volume copy. The following variables are listed in the message:

volume

The volume serial number of the real or virtual tape whose copy process was started.

source

The input tape is either a real tape (REAL) or a virtual tape (VIRTUAL).

target

The output tape is either a real tape (REAL) or a virtual tape (VIRTUAL).

System action:

None

User response:

None

HVT393I **VOLUME=volume COPY FROM**
source TO target comp, RC=rc (text)

Explanation

This is an informative message indicating that HVTUV2V utility ended a volume-to-volume copy. The following variables are listed in the message:

volume

The volume serial number of the real or virtual tape whose copy process was ended.

source

The input tape was either a real tape (REAL) or a virtual tape (VIRTUAL).

target

The output tape was either a real tape (REAL) or a virtual tape (VIRTUAL).

comp

The volume-to-volume copy was either successful (WAS OK) or was unsuccessful (FAILED). The failure return code are listed under the *rc* value.

rc

Possible return codes are:

00

The volume to volume copy was OK.

01

An I/O error occurred while rewinding the real tape.

02

An I/O error occurred while writing a tape mark on the real tape.

03

An I/O error occurred while writing a data block on the real tape.

04

An End-of-Volume was encountered while writing a data block on the real tape.

05

An I/O error occurred while reading a data block from the real tape.

11

An error was encountered while pseudo-mounting the virtual tape. This message may be preceded by other VTE messages indicating the cause of the error.

12

An error was encountered while writing a tapemark on the virtual tape. This message

may be preceded by other VTE messages indicating the cause of the error.

13

An error was encountered while writing a data block on the virtual tape. This message may be preceded by other VTE messages indicating the cause of the error.

14

An End-of-Volume was encountered while writing a data block on the virtual tape.

15

An error was encountered while reading a data block from the virtual tape. This message may be preceded by other VTE messages indicating the cause of the error.

21

The copy process was stopped by the operator.

22

An internal abend occurred in the utility.

23

An error occurred while accessing the VDB.

24

An attempt was made to overwrite an existing virtual tape.

25

An error was encountered when trying to delete the tape volume entry from the VOLCAT. The IDCAMS error messages preceded this message.

26

An attempt was made to overwrite a real tape while the input virtual tape does not exist.

31

An error was encountered while dynamically allocating the real tape. This message may be preceded by other VTE messages indicating the cause of the error.

32

An error was encountered while opening the real tape.

33

An error was encountered while calculating the read block length from a real tape.

System action:

The current volume-to-volume copy ends. The utility resumes with the next volume copy (if any).

User response

The recommended operator action depends on the return code:

01, 02, 03, or 05

Try to locate the problem using the HVT394E message issued earlier and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

04 or 14

The volume-to-volume process cannot split input tapes.

- If the output tape is real, provide longer tapes.
- If the output tape is virtual, increase the tape capacity of the tapepool to which this virtual tape belongs.

11, 12, 13, 15, or 23

Try to locate the problem using previous messages issued earlier and correct the cause of the error. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

24

If you want to overwrite an existing virtual tape, specify VREPLACE=YES in the utility's COPY initialization statement.

25

Inspect the IDCAMS error messages and correct the cause of the error. You may do one of the following:

- Manually delete the tape volume entry by using the following IDCAMS statement:

```
DELETE Vvolser VOLENT PURG
```

- Retry the copy. In that case, you must also specify VREPLACE=YES.

26

Probable user error. Make sure you specified the right volser and the right FROM parameter value, then retry the copy.

31 and 32

These return codes may indicate an environmental error.

Try to locate the problem using system messages that may have been issued earlier. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

33

If the input real tape block size is larger than 256K, HVTUV2V cannot copy it. Otherwise, this return code may indicate an internal error. If so, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT394E

**VOLUME=volume I/O ERROR ON
REAL CUU=cuu, ECB=X'ecb',
STATUS=X'status',
SENSE=X'sense'**

Explanation

During a volume-to-volume copy, HVTUV2V utility encountered an I/O error on the real tape. The following variables are listed in the message:

volume

The volume serial number of the real tape that encountered the error.

cuu

The device number of the real device that encountered the error.

ecb

The hexadecimal return code from the EXCP operation.

status

The hexadecimal device status of the failing channel command.

sense

The hexadecimal sense of the failing channel command.

System action:

Message HVT393I follows, indicating what operation was being performed while the I/O error occurred. Thereafter, the current volume-to-volume copy terminates. The utility resumes with the next volume copy (if any).

User response:

Try to locate the problem using the message values. If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT395W

**VOLUME=volume1 ON REAL
CUU=cuu, EXPECTED
VOLUME=volume2 R (RETRY) N
(FAIL)**

Explanation

During a real tape label verification performed before a volume to volume copy started, the HVTUV2V utility detected that the mounted real tape has a wrong volume serial number. The label verification is performed only when the specific copy request did not mention RBLP=YES.

The following variables are listed in the message:

volume1

The volume serial number of the real tape as detected by HVTUV2V. If the *volume1*=*****, HVTUV2V could not read the real tape label.

cuu

The device number the real tape is mounted on.

volume2

The volume serial number of the requested real tape.

System action:

The utility unloads the real tape and waits for the operator's reply.

User response

Perform one of the following:

- Reply R and mount the correct real tape. In that case, the copy of this tape will be reattempted.
- Reply N. In that case, the current volume to volume copy terminates. The utility resumes with the next volume copy (if any).

HVT400I **SERVER *serverid* TCP/IP
ACCEPTING CONNECTIONS ON
PORT *port***

Explanation

This is an informative message that is conditionally issued by the VTE server during its initialization only when the TCP option is enabled. This message indicates that the TCP/IP server component of the VTE server has successfully initialized and is ready to accept connections from any other TCP/IP client component of any other VTE server.

The following variables are listed in this message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

port

The TCP/IP port number VTE listens to (as specified in the PORT parameter of the TCP initialization statement).

System action:

None

User response:

None

HVT401E **SERVER *serverid* TCP/IP
FUNCTION=*func* FAILED, RC=*rc***

Explanation

An error was encountered by the VTE server or client component of the VTE server during execution of a TCP/IP function. The following variables are listed in the message:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

func

The TCP/IP function whose execution failed. The list of supported TCP/ IP functions is given in the *IBM Client/Server IP API Guide*.

rc

The return code given by the EZASMI system service in its ERRNO field. The ERRNO values are listed in the *IBM Client/Server IP API Guide* in chapter "Sockets ERRNOs – ERRNOs returned by TCP/IP for MVS".

If *rc*=999998, the function could not be passed to TCP/IP and VTE faked this return code.

System action

Possible system actions are:

- If the failing function is associated with basic environment establishment at VTE server initialization, the VTE server shuts down.
- If the failing function is associated with an e-Vault or return process, the involved virtual tape e-Vault or return process is suspended in that phase. Retry of the phase will be attempted as soon as the problem is solved.

User response

You can perform the following actions to address this issue:

- If the failing function is due to a connection lost, a reconnection attempt is automatically retried every 10 seconds.
- For any other cause, use the return code from the EZASMI system service and try to locate the problem.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT402I **SERVER *serverid* TCP/IP READY
TO CONNECT TO IP=*ipaddr*/*port***

Explanation

This is an informative message that is conditionally issued by the VTE server during its initialization only when the TCP option is enabled.

This message indicates that the TCP/IP client component of the VTE server has successfully initialized, and is ready to connect to the TCP/IP server component of the The VTE serverid (as specified in the IDENT initialization statement). server when it is ready to accept this connection.

The variables listed in this message are the following:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

ipaddr

The IP address of the VTE server's TCP/IP server component to which this client is to connect (as specified in the SERVERIP parameter of the TCP initialization statement).

port

The TCP/IP port number to which the VTE TCP/IP client component is to connect (as specified in the PORT parameter of the TCP initialization statement).

System action:

None

User response:

None

HVT403I **SERVER *serverid* CONNECTION TO IP=*ipaddr*/port ESTABLISHED**

Explanation

This is an informative message conditionally issued by the VTE server TCP/IP client or server component. This message indicates that a connection has been established between the TCP/IP server component running in one VTE server and a TCP/IP client component running in another VTE server. The connection is established at one of these times:

- On VTE server initialization.
- After a previous connection has failed.

The variables listed in this message are the following:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

ipaddr

The IP address of the VTE server's TCP/IP server/client component with which the connection has been established.

port

The TCP/IP port number over which the above connection has been established.

System action:

None

User response:

None

HVT404W **SERVER *serverid* CONNECTION TO IP=*ipaddr*/port LOST**

Explanation

A TCP/IP connection that existed between the TCP/IP client component of a VTE server and a server

component of another VTE server is lost. This may happen as a result of the following:

- A general TCP/IP failure.
- One of the VTE servers failure.
- A system crash in which one of the VTE servers run.

The variables listed in this message are the following:

serverid

The VTE serverid (as specified in the IDENT initialization statement).

ipaddr

The IP address of the VTE server's TCP/IP server/client component with which the connection was lost.

port

The TCP/IP port number to which the above connection was lost.

System action

All e-Vault or return processes for type4 e-Vault with that location are suspended until the connection is re-established.

- If the message is issued by the TCP/IP server component, it will continue to listen until the other side connects, then the connection will be re-established.
- If the message is issued by the TCP/IP client component, it will retry to connect every 10 seconds. When successful, the connection will be re-established.

When the connection is re-established, the suspended e-Vault or return processes will resume from their last suspended phase.

User response

Check why the TCP/IP connection went down. If the problem occurred due to an environmental cause, fix it and let VTE re-establish the connection.

If the problem cannot be located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT405I **VOLUME=*volume* process *direction* LOCID=*locid* status**

Explanation

This is an informative message issued by the VTE server whenever a virtual tape e-Vault or replication process starts or ends. The message is issued in the local location and, for type4 e-Vault, also in the remote off-site storage location.

The following variables are listed in the message:

volume

The virtual tape volume serial number processed by the e-Vault or replication process.

process

The process is either VAULT (e-Vault) or REPLICATION.

direction

Possible values are:

TO

The message is issued in the local location and it indicates that the e-Vault or replication of a virtual tape to a remote off-site storage location has started/ended.

FROM

The message is issued in the remote off-site storage location and it indicates that the e-Vault of a virtual tape from the local location has started/ended.

locid

In the local location, this is the remote off-site storage location to which this virtual tape is e-Vaulted or replicated. In the remote off-site storage location, this is the local location from which this virtual tape is e-Vaulted or replicated. If locid=*MAIN*, the location contains only blanks.

status

Indicates whether the e-Vault or replication process has STARTED or ENDED.

System action:

None

User response:

None

HVT406I **VOLUME=volume RETURN direction**
 LOCID=locid status

Explanation

This is an informative message issued by the VTE server whenever an e-Vaulted virtual tape return process starts or ends. The message is issued in the local location and for type4 e-Vault, also in the remote off-site storage location.

The following variables are listed in the message:

volume

The e-Vaulted virtual tape's volume serial number processed by the return process

direction

Possible values are:

TO

The message is issued in the remote off-site storage location and it indicates that the return of an e-Vaulted virtual tape to the local location begins/ends.

FROM

The message is issued in the local location and it indicates that the return of an e-Vaulted virtual tape from a remote offsite storage location starts/ends.

locid

In the local location, this is the remote off-site storage location from which this e-Vaulted virtual tape is returned. In the remote off-site storage location, this is the local location to which this e-Vaulted virtual tape is returned. If locid=*MAIN*, the location contains only blanks.

status

Indicates whether the return process has STARTED or ENDED.

System action:

None

User response:

None

HVT407E **VAULT/RETURN OF**
 VOLUME=volume FAILED IN
 PHASE=phase (text)

Explanation

During execution of one of the e-Vault phases (either for e-Vaulting, returning or remote-scratching a virtual tape), VTE encountered a problem that caused that phase to fail.

The following VTE Type 3, Type4, or Type5 e-Vault components may fail in the following phases:

Phase 51 - 60

Physical layer responsible for preparing and controlling the data flow over the FTP links.

Phase 61 - 70

Logical layer responsible for controlling the above FTP physical layer.

The following variables are listed in the message:

volume

The e-Vaulted virtual tape volume serial number processed by the return process for which the error was encountered.

phase

The phase number in which the error was encountered.

51

An error occurred while accessing the VDB by the physical layer responsible for preparing and controlling the data flow over FTP links.

- 52** An error occurred while obtaining the tapepool name to contain a returned virtual tape dataset from a non-MVS platform.
- 53** An error occurred while extracting and calculating the characteristics (size, unit, unit count, etc.) of a returned virtual tape dataset from a non-MVS platform.
- 54** An error occurred while remote-scratching a virtual tape from a non-MVS platform.
- 55** An error occurred while FTP-ing a virtual tape dataset to another MVS platform.
- 56** An error occurred while FTP-ing a virtual tape dataset to a non-MVS platform.
- 57** An error occurred while FTP-ing a virtual tape dataset from a non-MVS platform.
- 58** An error occurred while deleting a virtual tape dataset from a non-MVS platform (after its successful return).
- 59** An error occurred while locating an off-site storage location whose name is specified in the (*text*) message value.
- 61** An error occurred while deriving the remote site's dataset name of a virtual tape dataset to be FTP-ed to another MVS platform.
- 62** An error occurred while obtaining the remote site's diskpool name to contain a virtual tape dataset to be FTP-ed to that MVS platform.
- 63** An error occurred while extracting and calculating the characteristics (size, unit, unit count, etc.) of a virtual tape dataset to be FTP-ed to that MVS platform.
- 65** The virtual tape to be recovered is currently in use.
- 66** An error occurred while recovering a virtual tape just FTP-ed to another MVS platform.
- 67** While recovering a virtual tape just FTP-ed to another MVS platform, the e-Vaulted virtual tape had to be added to the VDB. However, the

number of virtual tapes in VDB exceeded the allowed maximum.

text

Short text describing the failing phase. For phase=59 it specifies the off-site storage location that could not be located.

System action

The system action depends on the *phase* value:

58

The virtual tape 'return' is considered OK, even though the delete of the remote copy failed. That file will be reused if the virtual tape will be e-Vaulted again.

Other

The e-Vault process for the current virtual tape is suspended. The process resumes with the next virtual tape from the phase where it left off. When all virtual tapes have been processed, the VTE server waits 10 seconds, then attempts to resume all previously failed virtual tapes from the phase where they left off.

User response

The recommended operator action depends on the phase in which the error occurred:

54, 55, 56, 57 or 58

Browse the sysout of the job that activated the FTP (listed in message HVT415E) and locate the exact cause of the FTP failure. One common cause for that failure is a mismatch between the HVTNETx member contents that defines the remote site and the appropriate VLTRMT initialization statement.

59

If the specified location is *RECRTD*, check whether the local TMS contains active virtual tapes that do not really exist and clear them. Then use the HVTURCV utility to clear the VDB.

For other locations, check why these locations were defined. If needed, clear them using the HVTURCV utility.

Other

If further server attempts fail, try to see why this happens. Most of the HVT407E phases messages are preceded by other messages that detail the error that led to the phase suspend.

If you are unable to locate the problem and the virtual tape remains in an interim vault status, follow the recovery steps described in [“Recovery considerations for type3 e-Vault” on page 200](#), [“Recovery considerations for type4 e-Vault” on page 201](#), or [“Recovery considerations for type5 e-Vault” on page 202](#).

**HVT408E INVALID CONTROLLING VAULT
INSTRUCTION IGNORED,
type=value**

Explanation

The e-Vault logical layer TCP/IP component encountered an invalid controlling instruction. This instruction may have been sent by another external component or may be a result of an internal VTE failure.

The following variables are listed in this message:

type

Possible values are:

VLIDENT

The special identifier used to distinguish the VTE TCP/IP data traffic is wrong.

VLTFUNC

The logical layer received an invalid VTE instruction.

value

The erroneous identifier or the invalid logical layer instruction received by VTE.

System action

The received logical layer instruction is disregarded. Process continues with next instruction. If the problem is recurrent, the e-Vault process may be suspended.

In any case, the e-Vault process will resume from the failing phase as soon as the problem is fixed.

User response

Check whether another TCP/IP application sends data on the TCP/IP port defined to VTE. If so, do one of the following:

- Change the VTE definition and bounce the VTE server.
- Change the other TCP/IP application definitions.

If it seems that bad data is sent/received by VTE, collect and forward the appropriate documentation to customer support for technical assistance.

**HVT409I EVAULT PROCESSING HAS BEEN
COMPLETED**

Explanation:

This is an informative message issued by the VTE server whenever the e-Vault processing has been completed, which means there are no virtual tapes to be vaulted in the VTE server's private storage.

System action:

None

User response:

None

HVT412S LOCID=locid HANDSHAKE FAILED

Explanation

During the type4 e-Vault handshake process that follows establishment of the TCP/IP connection, a mismatch has been detected between the initialization statements of the local location and the remote off-site storage location VTE servers.

Message HVT416I follows in the local location server only and lists the handshaked parameters.

That mismatch consists of one of the following:

For all e-Vault types:

- The remote off-site storage location VTE server does not specify the local location as its remote site (the remote VLTRMT LOCID value is not similar to the local VLTOWN OWNLOCID value).
- The remote off-site storage location VTE server location name is not defined as one of the recognized remote off-site storage locations (the remote VLTOWN OWNLOCID value is not similar to one of the local VTLRMT LOCID values).
- The e-Vault type (the VLTRMT TYPE value) in the local location does not match the remote off-site storage location TYPE value.

For type4 e-Vault:

The remote off-site storage location home IP address does not match the VLTRMT FTPIP value specified in the local location.

The only variable in this message is *locid*. If the message is issued in the local location, then this is the remote off-site storage location whose initialization statements do not match. If the message is issued in the remote off-site storage location, then this is the local location whose initialization statements do not match.

System action:

The handshake process fails and the e-Vault process does not start (or resume). A handshake is re-attempted every 10 seconds and, if successful, the e-Vault process starts (or resumes).

User response

Possible operator actions are listed below.

- Make sure both locations specify the right LOCID values (that is, each location identifies correctly its remote party).
- Make sure both locations specify a similar e-Vault type in their respective LOCID parameter.
- Make sure the IP address in the FTPIP parameter matches the remote off-site storage location IP.

HVT416I message issued in the local location server only can be used to detect the wrongly specified value.

After you fixed the values, bounce the VTE server (either in the local location or in the remote off-site storage location).

HVT413W **VOLUME=*volume* MOVING
BETWEEN STORAGE LOCATIONS
IN VSID=*vsid*, TO CANCEL WAIT
REPLY 'N'**

Explanation

As a result of a system-specific mount request on one of the VTE virtual devices, the VTE server found that the virtual tape to be mounted was either:

- Moving between storage locations (as a result of a previous e-Vault, e-Mig, replication or return requests) that did not yet complete.
- In an e-Vault remote off-site storage location.

If the virtual tape is moving between storage locations, VTE waits for it to reach that location and to enter a stable e-Vault status (i.e. 'local', 'vaulted' or 'safe').

If as a result of the previous move the virtual tape dataset is not available at the location that requested the mount, a return of that virtual tape is automatically initiated.

The job execution is suspended until the virtual tape dataset is available at the location that requested the mount. When available, the job execution is resumed.

The variables in this message are the following:

volume

Represents the volume serial number of the virtual tape whose mount was requested.

vsid

Specifies the system identifier of the system that should subsequently return the virtual tape. If the current system does not share a VTE complex, the *vsid* value represents the system identifier of the current system.

System action:

The VTE server waits for the operator reply and, in the meantime, proceeds with the virtual tape return. If no reply is received and the whole virtual tape is made available, the job execution resumes.

User response

Perform one of the following:

- Reply N in which case the job that requested the e-Vaulted virtual tape will be EQC-ed.
- Wait for the virtual tape return after which the job will resume execution.

HVT414E **VOLUME=*volume type* FAILED,
RC=*rc* (*text*)**

Explanation

During execution of one of the FTP related functions related to type3, type4, or type5 e-Vault, an error was encountered by a utility (either in its processing or in the external FTP/CTC service). The following variables are listed in the message:

volume

The virtual tape volume serial number associated with the virtual tape dataset for which the FTP/CTC service was invoked.

type

Possible values are:

FTP

The HVTUFTP utility encountered an error.

CTC

The HVTUCTC utility encountered an error.

rc

Possible return codes are:

01

The HVTUFTP utility failed to obtain its jobid or the procedure identifier containing the virtual tape to be FTP-ed.

The most probable cause to this error is a manual start of the HVTUFTP utility. HVTUFTP utility should be started only by the VTE server.

02

The HVTUFTP utility failed to find the VTE subsystem or the VTE subsystem is inactive.

03

The HVTUFTP utility failed to establish a cross-memory environment and connect itself to the server. HVT345E message precedes and gives more details on that problem.

04

The HVTUFTP utility failed to find the internal FTP-related table prepared by the server. That table contains the virtual tape details to be used by the FTP external service.

05

The HVTUFTP utility failed to dynamically allocate one of the data sets required by the external FTP service.

HVT311E message precedes and gives more details on that problem.

06

The HVTUFTP utility failed to derive the name of virtual tape dataset in the remote off-site storage location.

- 07**
The HVTUFTP utility failed to open one of the data sets allocated earlier for the FTP external service.

HVT010E message precedes and gives more details on that problem.
- 08**
The VTE server went down while the FTP external service FTP-ed a virtual tape dataset in behalf of the HVTUFTP utility.
- 09**
The FTP external service invoked by the HVTUFTP utility ended with a non-zero return code.
- 11**
The HVTUFTP utility failed to re-format a 'returned' virtual tape dataset from a non-MVS platform fixed format to its original undefined format.
- 13**
The HVTUFTP utility failed to LOCATE the local virtual tape dataset.
- 14**
The disks that contain the local virtual tape dataset and the disks to contain the virtual tape dataset (on the remote off-site storage location) are of unlike device type.
- 20**
The HVTUFTP utility abended.

text

Short text describing the error.

System action

For all above errors, the HVTUFTP utility passes the return code value to the VTE server which will re-issue it under the HVT415E message.

It then will suspend this virtual tape e-Vault process. The process will be re-attempted later from the phase it was left off.

User response

The recommended action depends on the return code:

01 or 04

If you started HVTUFTP utility manually, it won't find the proper environment in the VTE address space. HVTUFTP utility should be started only by the VTE server.

02

Make sure the VTE subsystem is active. If not, issue the 'F server,SSNON' operator command in order to activate it.

06

If the QUAL1 parameter value on the VDSN initialization statement on the remote off-site storage location is longer than the local site QUAL1 parameter value, make sure QUAL2 parameter value on the remote off-site storage location is %VOL.

09

HVTUFTP utility's SYSPRINT sysout file contains the external FTP service messages. Browse that file and try to locate the problem.

If the problem is not located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

14

Re-define the remote VTE diskpool to contain disks of the same device type as the local diskpool.

Other

If you are unable to locate the problem and the virtual tape remains in an interim vault status, follow the recovery steps described in ["Recovery considerations for type3 e-Vault" on page 200](#), ["Recovery considerations for type4 e-Vault" on page 201](#), or ["Recovery considerations for type5 e-Vault" on page 202](#).

If the problem is not located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

HVT415E

**VOLUME=volume type BY jobname/
jobid FAILED, RC=rc**

Explanation

During execution of type3, type4, or type5 e-Vault, the VTE server was notified of the utility failure. This message is issued by the server address space to alert about the utility failure and indicate the *jobname* and *jobid* of the failing started task.

The following variables are listed in this message:

volume

The virtual tape volume serial number associated with the virtual tape dataset for which the utility was invoked.

type

Possible values are:

FTP

The HVTUFTP utility failed.

CTC

The HVTUCTC utility failed.

jobname

Jobname of the started task. If asterisks are displayed, the task failed before the server could obtain its jobname.

jobid

Jobid of the started task. If asterisks are displayed, the task failed before the server could obtain its jobid.

rc

Possible return codes are:

99

The utility could not initiate itself or could not communicate with the server.

Other

All return codes (except 99) are similar to the HVT414E message return codes.

System action:

The virtual tape e-Vault process is suspended. The process will be re-attempted later from the phase it was left off.

User response

The recommended action depends on the return code:

99

Try to see why the utility failed before establishing connection with the server.

If the problem is not located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

Other

Act as described under the appropriate return code of the HVT414E message.

HVT416I **LL=local_loc RO=remote_own**
LO=local_own RL=remote_loc
LIP=local_ip RIP=remote_ip
LT=local_typ RT=remote_typ

Explanation

This message follows the HVT412S message when a type4 e-Vault handshake process fails. It details all parameter values that are checked during that handshake process. Since the connection establishment is not yet completed, the message is issued only in the local location server.

The following variables are listed in this message:

local_loc

The LOCID value (as specified in the VLTRMT initialization statement) in the local location.

remote_own

The OWNLOCID value (as specified in the VLTOWN initialization statement) in the remote off-site storage location.

local_own

The OWNLOCID value (as specified in the VLTOWN initialization statement) in the local location.

remote_loc

The LOCID value (as specified in the VLTRMT initialization statement) in the remote off-site storage location.

local_ip

The FTPIP value (as specified in the VLTRMT initialization statement) in the local location.

This value is displayed as one contiguous 8 hexadecimal digits string in which each 2 digits correspond to one IP qualifier.

remote_ip

The home IP address of the remote off-site storage location (as extracted by the GETHOSTID service).

This value is displayed as one contiguous 8 hexadecimal digits string in which each 2 digits correspond to one IP qualifier.

local_typ

The TYPE value (as specified in the VLTRMT initialization statement) in the local location.

remote_type

The TYPE value (as specified in the VLTRMT initialization statement) in the remote off-site storage location.

System action:

All taken system actions are described by HVT412S message.

User response

In order for the handshake process to succeed, all the following conditions must be true:

- local_loc = remote_own
- local_own = remote_loc
- local_ip = remote_ip
- local_typ = remote_typ

If any of the above conditions are not true, correct the erroneous parameter's value, then bounce the local location or remote off-site storage location server, respectively.

HVT420S **MEMBER member NOT FOUND IN**
dsn

Explanation

During VTE server initialization, refreshing of one of the refreshable members or a utility run, VTE could not find a member that contains required initialization statements.

The following variables are listed in this message:

member

The member name that could not be found.

dsn

The partitioned dataset name in which the member could not be found. If *dsn* is part of a concatenation sequence, *dsn* is the first dataset name of that sequence.

System action

Possible system actions are:

- If the error occurs during VTE server initialization, the VTE server shuts down.
- If the error occurs during the refresh of a refreshable member, the refresh fails and the old existing values remain.
- If the error occurs in a utility run, that utility terminated.

User response

Possible operator actions are:

- If the error occurs during VTE server initialization or utility run, make sure the appropriate parameter in the SYS initialization statement on the EXEC PARM field specifies a valid member suffix.
- If the error occurs during refresh of a refreshable member, make sure you typed the appropriate member suffix.

HVT421I **USING MEMBER *member* FROM *dsn***

Explanation

This is an informative message that is issued during VTE server initialization, refreshing of one of the refreshable members or a utility run.

The message indicates the members and partitioned dataset names the initialization statements are read from.

The following variables are listed in the message:

member

The member name that contains the initialization statements. If *member*=**SEQ**, the initialization statements were read from a sequential dataset (or a partitioned dataset that specified also a member name).

dsn

The partitioned dataset name containing above member.

If JES is displayed, the initialization statements were read directly from the JES input file (that is, from a //DDNAME DD * statement).

System action:

None

User response:

None

HVT430I ***cuu volume(aliasvol) jobname/jobid***
stat kb_in kb_out

Explanation

This is an informative message issued as a result of the 'F server,D=A' operator command. It details the status of all active virtual devices. The message is preceded by a general title that describes the contents of each field.

The following parameters are listed in this message:

cuu

The device number of the virtual device on which a virtual tape is currently mounted.

volume

The volume serial number of the virtual tape that is currently mounted on the virtual device.

aliasvol

If the mounted virtual tape is accessed through a PAT alias volume, this is its volume serial number. Otherwise, the value is identical to the volume value.

jobname

The jobname of the job that owns the mounted virtual tape.

jobid

The jobid of the job that owns the mounted virtual tape.

stat

The read/write status of the mounted virtual tape. Possible values are:

R

The virtual tape has done READ I/O operations only.

W

The virtual tape has done READ and WRITE I/O operations.

kb_in

Number of KB that have been read (or 0). This number is rounded down to the nearest integer.

kb_out

Number of KB that have been written (or 0). This number is rounded down to the nearest integer.

System action:

None

User response:

None

HVT431I **NO ACTIVE UNITS**

Explanation:

This is an informative message that is issued as a result of the 'F server,D=A' operator command when there are no active virtual devices.

System action:

None

User response:

None

HVT432I *type1=name1 VOL#=vol#*
 TRKS=trks#_trksm USED=percent
 % type2=name2

Explanation

This is an informative message that issued as a result of the 'F server,D=DSP' operator command.

It details the number of eligible disks, total number of tracks and the percentage of used space on the esoteric names or Storage Groups associated with diskpools defined in the HVTPOOLxx member. The displayed data is extracted from fields that were updated on the last threshold interval associated with each diskpool.

For non-SMS-managed diskpools, the displayed values are subject to the *volmask* sub-parameter specified in the DPOOL THRESH parameter.

The following variables are listed in the message:

type1

Possible values for *type1* are:

ES

The displayed data pertains to non-SMS-managed disks whose esoteric name is given by *name1* value.

SG

The displayed data pertains to SMS-managed disks whose Storage Group name is given by *name1* value.

name1

Name of the above esoteric name or Storage Group. If asterisks are displayed, the name could not be determined. This is either because the server did not yet complete the diskpool refresh (performed as a result of a refresh request) or because the Storage Class associated with the above SMS-managed disks group is invalid.

vol#

Number of eligible disk volumes in the above esoteric name or Storage Group.

- For SMS-managed disks, an eligible disk is a non-EAV or an EAV disk (regardless of the IGDSMSxx USEEAV setting) that is online, has a status of ENABLE / QUIESCE_ALL / QUIESCE_NEW and belongs to a Storage Group that is ENABLE / QUIESCE_ALL / QUIESCE_NEW.

- For non-SMS-managed disks, an eligible disk is a disk that is online, has a mount attribute of STORAGE and is not converted to SMS.

If asterisks are displayed, the number of eligible disk volumes could not be determined because the referred esoteric name or Storage Group is invalid.

trks#

Total number of tracks in the above esoteric name or Storage Group subject to the adjacent *trksm* multiplicity factor. If asterisks are displayed, the number of tracks could not be determined because the referred esoteric name or Storage Group is invalid.

Note:

1. If the operating system release is lower than z/OS 1.12, the *trks#* value does not include cylinder-managed space tracks even though EAV disks may exist in the diskpool.
2. If the operating system release is z/OS 1.12 or higher and EAV disks exist in the diskpool, the *trks#* value includes cylinder-managed space tracks even if EATTR=OPT is not specified.
3. If data on EAV disks is to be included, this occurs even when the USEEAV (YES) parameter is not specified in the PARMLIB IGDSMSxx member.

trksm

The *trks#* multiplicity factor. Valid values are K, M, and G as described below:

- K – *trks#* is expressed in units of Kilo_bytes (1024).
- M – *trks#* is expressed in units of Mega_bytes (1024*1024)
- G – *trks#* is expressed in units of Giga_bytes (1024*1024*1024)

percent

Percent of used space from the total available space on the above esoteric name or Storage Group. See the Notes on EAV disks below.

Note:

1. If the operating system release is lower than z/OS 1.12, the *percent* value is calculated based on the disks track-managed space only, even though EAV disks may exist in the diskpool.
2. If the operating system release is z/OS 1.12 or higher and EAV disks exist in the diskpool, the *percent* value is calculated based on both track and cylinder-managed space tracks, even though EATTR=OPT is not specified.
3. If data on EAV disks is to be included, this occurs even when the USEEAV (YES)

parameter is not specified in the PARMLIB IGDSMSxx member.

type2

The only value for *type2* is DPOOL, which specifies that disks are defined to VTE under a diskpool whose name is given by the *name2* value.

name2

Name of the above diskpool.

System action:

None

User response:

None

HVT433I **TAPES=tapes ACT=active (percent %)** **SCR=scratch TPOOL=plname**

Explanation

This is an informative message issued as a result of the 'F server,D=TSP' operator command. It details the number of active, percent of active, and number of scratch virtual tapes in tapepools defined in the HVTPOLxx member.

The displayed data is extracted from fields that were last updated on the expiration of the lowest threshold interval of all defined tapepools. This means that delays may be encountered between tapepool capacity or content change and the updated HVT433I message data display.

The following variables are listed in the message:

tapes

Total number of virtual tapes found in the tapepool.

active

Number of active virtual tapes in the tapepool.

percent

The percent of active virtual tapes from the total available virtual tapes in the tapepool. If the number of tapes is 0, the percent value is defined as 0.

scratch

Number of scratch virtual tapes in the tapepool.

plname

Name of the tapepool for which the above data was displayed.

System action:

None

User response:

None

HVT434I **TOTAL: TAPES=tapes ACT=active (perc%) SCR=scratch**

Explanation

This is an informative message issued as a result of the 'F server,D=TSP' operator command. It follows the HVT433I message and totals the number of virtual tapes in all categories and tapepools.

The following variables are listed in this message:

tapes

Total number of virtual tapes found in all tapepools.

active

Number of active virtual tapes in all tapepools.

perc

The percent of active virtual tapes from the total available virtual tapes in all tapepools.

scratch

Number of scratch virtual tapes in all tapepools.

System action:

None

User response:

None

HVT436E **VOLUME=volume (jobname/jobid)**
CUU=cuu TCB=X'tcb' KILL FAILED,
RC=rc (text)

Explanation

A KILL operator command was entered; however, due to bad values or internal restrictions it was not performed. The return code value gives further details.

The following variables are listed in this message:

volume

The mounted virtual tape volume serial number currently mounted on the specified (or derived) virtual device.

jobname

The jobname of the job currently using the specified (or derived) virtual device. If asterisks are displayed, the value could not be determined.

jobid

The jobid of the job currently using the specified (or derived) virtual device. If asterisks are displayed, the value could not be determined.

cuu

The explicitly specified (or derived) virtual device number. If asterisks are displayed, the value could not be determined.

tcb

The explicitly specified (or derived) TCB hexadecimal address. If asterisks are displayed, the value could not be determined.

rc

Possible return codes are:

- 01** The specified CUU or TCB parameter value is too long.
- 02** No value specified for the CUU or TCB parameter.
- 03** The specified TCB parameter's value is not hexadecimal.
- 04** The specified TCB parameter value is not valid or does not represent a server I/O subtask.
- 05** The specified TCB parameter value is of a server I/O subtask; however, this subtask currently does not perform any I/O.
- 07** The specified TCB and associated virtual device do not point to each other.
- 08** The specified CUU parameter value does not represent a virtual device number.
- 11** The explicitly specified (or derived) virtual device does not perform any I/O.
- 21** The specified TCB and associated virtual device do not point to each other.
- 22** The explicitly specified (or derived) TCB does not match the final server I/O subtask to be killed.
- 23** The server I/O subtask is waiting for an operator reply.

text

Short explanation of the error.

System action:

The KILL command is not performed.

User response

The recommended operator action depends on the return code:

- 01** Specify a valid CUU or TCB value.
- A valid CUU value is a 3 or 4 hexadecimal digits value.
 - A valid TCB value is a 6 or 8 hexadecimal digits value.

- 02** Specify a non-blank CUU or TCB value.
- 03** Specify a valid TCB value. A valid TCB value is a 6 or 8 hexadecimal digits value.
- 04** Specify a valid TCB value that represents a server I/O subtask. No other subtask types can be KILL-ed by this command.
- 05, 07, 11, 21 or 22** The server I/O subtask to be KILL-ed is not stuck (as its status changed since the command was entered). As such, you may need to reconsider the need to KILL it.
- 08** Specify a valid CUU value that represents a virtual device number.
- 23** Reply any pending WTOR before trying to KILL a server I/O subtask.

HVT437I **VOLUME=volume CUU=cuu**
TCB=X'tcb' (jobname/jobid)
KILLED

Explanation

This is an informative message that is issued by the VTE server indicating that KILL of a server I/O subtask has been successfully performed.

The following variables are listed in this message:

volume

The mounted virtual tape volume serial number currently mounted on the specified (or derived) virtual device.

cuu

The explicitly specified (or derived) virtual device number.

tcb

The explicitly specified (or derived) TCB hexadecimal address.

jobname

The jobname of the job currently using the specified (or derived) virtual device.

jobid

The jobid of the job currently using the specified (or derived) virtual device.

System action:

The server I/O subtask is killed. The I/O that was served by the subtask is EQC-ed.

User response:

None

HVT440W **DPOOL=poolname ES=name1**
 SG=name2 IGNORED, RC=rc (text)

Explanation

During the diskpools contents analyze phase, the HVTUDST utility found that either:

- Two or more diskpools define a similar esoteric name or Storage Group.
- A diskpool contains no eligible disks or the number of eligible tracks in the diskpool is 0.

The following variables are listed in the message:

poolname

The diskpool name whose esoteric name or Storage Group is similar to those of a previously defined esoteric name or Storage Group, or whosnumber of disks or tracks is 0.

name1

The esoteric name defined by the diskpool. If blanks are displayed, no esoteric name has been derived for the diskpool.

name2

The Storage Group defined by the diskpool. If blanks are displayed, no Storage Group has been derived for the diskpool.

rc

Possible return codes are:

01

The diskpool esoteric name or Storage Group is similar to those of a previously handled diskpool.

02

The number of eligible disks or tracks in the diskpool is 0.

text

Short explanation of the error.

System action

Since this diskpool does not have any valuable data (from the HVTUDST point of view), the diskpool is ignored by further de-stage calculations.

In addition, if rc=01 the lowest 'low threshold' (the 3rd sub-parameter of the THRESH parameter) of both diskpools is retained on the remaining diskpool.

User response:

Check the diskpool definition and contents. If the findings are as expected, this message can be ignored.

HVT442I **disp1 PATTERN disp2 –**
 QUAL=qual#, LEN=len,
 NAME=name

Explanation

This is an informative message issued by the HVTUDST utility during the consolidation phase indicating that a new user pattern has been added or an old user pattern has been deleted. This message may be followed by one or more HVT443I and HVT444I messages detailing the dataset records affected by this pattern addition or deletion.

The following variables are listed in this message:

disp1

Either an OLD or a NEW user pattern that needs to be handled has been encountered.

disp2

The user pattern has either been ADDED to or DELETED from the history file.

qual#

The qualifier number the user pattern refers to. If the value is 0, the user pattern is similar to the tape dataset generic name.

len

The length of the user pattern.

name

The user pattern or tape dataset generic name.

System action

HVT443I and HVT444I messages detailing the dataset records affected by the addition or deletion of this pattern may follow.

User response:

None

HVT443I **disp2 NAME=name triplet1 triplet2**
 triplet3

Explanation

This is an informative message issued by the HVTUDST utility during the consolidation phase indicating that a dataset record has to be either deleted or consolidated into another existing dataset record.

The following variables are listed in this message:

disp2

Possible values are:

ADD

The dataset record contents is consolidated into another existing dataset record.

DEL

The dataset record is deleted.

name

The generic name contained in the dataset record.

triplet1

The first observation of the consolidated dataset record in the format `dc_scr#_mnt#` where:

dc

Number of days from the tape creation to the observation event.

scr#

Number of scratch events that occurred that day.

mnt#

Number of mount events that occurred that day.

triplet2

Same as triplet1 but for the second observation.

triplet3

Same as triplet2 but for the third observation.
More observations may be present in the dataset record but they are not displayed here.

System action:

The dataset record is either deleted or consolidated into another existing dataset record.

User response:

None

HVT444I ******SUMM=name triplet1 triplet2 triplet3**

Explanation

This is an informative message issued by the HVTUDST utility during the consolidation phase after all matching dataset records (listed by the preceding HVT443I messages) have been consolidated. This message lists the resulting consolidated dataset record.

The following variables are listed in this message:

name

The generic name of the dataset record into which the previously dataset records have been consolidated..

triplet1

The first observation of the resulting dataset record in the format `dc_scr#_mnt#` where:

dc

Number of days from the tape creation to the observation event.

scr#

Number of scratch events that occurred that day.

mnt#

Number of mount events that occurred that day.

triplet2

Same as triplet1 but for the second observation.

triplet3

Same as triplet2 but for the third observation.
More observations may be present in the dataset record but they are not displayed here.

System action:

This dataset record replaces all dataset records that have been consolidated into it.

User response:

None

HVT446I **GENERIC=name VOLUME=volume event DC=dc disp2 comp, RC=rc (text)**

Explanation

This is an informative message that is issued by the HVTUDST utility during the history file update phase, indicating that a new observation has been recorded in an existing (or newly created) dataset record.

The following variables are listed in the message:

name

The generic name of the dataset record into which the new observation is recorded. If asterisks are displayed, the generic name could not be determined.

volume

The virtual tape volume serial number that currently contains the tape dataset whose observation is being recorded.

event

Possible recorded events are SCR (virtual tape scratch) or MNT (virtual tape mount).

dc

Number of days that passed from the virtual tape creation until this event.

disp2

The observation has been recorded into a specific type of dataset record. Possible values are UPD (existing dataset record) or ADD (newly created dataset record).

comp

The observation recording was successful (WAS OK) or unsuccessful (FAILED).

rc

Possible return codes are:

00

The observation recording was OK.

01

First 27 characters of the first tape dataset on the virtual tape could not be determined.

Therefore, the tape dataset name could not be built and the relevant dataset record could not be found.

This error may occur when a virtual tape has been recovered or when the virtual tape has been created before the first HVTUDST utility version implementation.

02

The virtual tape's creation date could not be determined. Therefore, the statistical calculations could not be done.

This error may occur when a virtual tape has been recovered or when the virtual tape has been created before the first HVTUDST utility version implementation.

03

Number of days that passed from the virtual tape creation until this event is greater than 365.

text

Short explanation of the error.

System action

The system action depends on the return code:

00

The observation recording was OK.

01 or 02

The observation was not recorded.

03

The observation is ignored.

User response:

None

HVT447E **VOLUME=volume MISMATCH,
RC=rc (text) VD=name**

Explanation

During the VDB and the various diskpools contents consistency verification, the HVTUDST utility found that there is a mismatch between a VDB entry and the associated virtual tape dataset on disk.

Common causes of this error are due to:

- Disk management errors—a virtual tape has more than one virtual tape dataset on disk or, in other words, more than one dataset adhering to the virtual tape dataset's naming convention has been found to match one VDB entry.
- The VDB entry indicates that it represents a scratch virtual tape while a corresponding virtual tape exists on disk.

The following parameters are listed in this message:

volume

The volume serial number of the virtual tape for which the mismatch was encountered.

- If *rc=01* or *rc=03*, this value is extracted from the VDB entry.
- If *rc=02*, this value is extracted from the virtual tape dataset name.

rc

Possible return codes are:

01

The virtual tape dataset pointed from the VDB entry does not exist on disk.

02

A virtual tape dataset on disk does not have a corresponding entry in the VDB.

03

The virtual tape dataset pointed from the VDB entry and the existing virtual tape dataset on disk do not have the same name.

text

Short explanation of the error.

name

The virtual tape dataset name.

- If *rc=01*, this value is extracted from the VDB entry.
- If *rc=02* or *rc=03*, this value is extracted from the virtual tape dataset.

System action:

The listed virtual tape dataset is not considered for de-stage. In the case of duplicate virtual tape datasets, if one of them really matches the VDB entry, it will be considered for de-stage.

User response:

Delete duplicate virtual tape datasets (if any), then run the HVTURCV in order to synchronize the VDB and diskpools contents.

HVT448W **LAST UTILITY RUN WAS LESS
THAN 23 HOURS AGO, NEW
HISTORY OBSERVATIONS NOT
ADDED**

Explanation

The HVTUDST utility was found to run more than once in a less than one day interval. Since the events observations are recorded on a daily basis, multiple runs of the utility in a less than one day interval would cause multiple observations to be arbitrarily recorded. Due to that, this run will not add any new observations to the history file dataset records.

The 23 hours is considered to mark one day because, due to various reasons, production sites jobs may run in advance or be delayed.

System action:

New history observations are not added or updated. All other utility's functions perform normally.

User response:

None

HVT450I ******DPOOL=poolname**
TRKS=trks1#_K USED=percent1%
LTHRESH=percent2%
DTRKS=trks2#_K

Explanation

This is an informative message that is issued by the HVTUDST utility after the tape datasets have been assigned grades, the virtual tape datasets have been grouped by diskpool residency, and de-stage selection is done. It details the diskpool de-stage requirements that will cause its usage to go below the user-specified threshold.

This message is followed by HVT451I message and is repeated once for each diskpool.

The following variables are listed in this message:

poolname

Name of the diskpool.

trks1#

Total number of tracks in the diskpool. This number is displayed in units of 1024.

percent1

Percent of used space from the total available space on the above diskpool.

percent2

Low threshold percentage value specified in the THRESH parameter of the diskpool's DPOOL initialization statement.

trks2#

Number of tracks that are to be de-staged from the diskpool in order to achieve the percent2 value. This number is displayed in units of 1024.

System action:

None

User response:

None

HVT451I **volume Z=size C=days1# S=days2#**
M=mount# G=grade text * D=dsn#
+ N=name

Explanation

This is an informative message issued by the HVTUDST utility after the tape datasets have been

assigned grades, the virtual tape datasets have been grouped by diskpool residency, and

de-stage selection is done.

It details each virtual tape volume serial number along with de-stage related values.

All virtual tape volume serial numbers associated with virtual tape datasets that reside on the same diskpool are grouped under a preceding HVT450I message.

The following variables are listed in the message:

volume

The volume serial number of the virtual tape for which the details are listed.

size

Size (allocated number of tracks) of the virtual tape dataset.

days1

Number of days that passed from the virtual tape creation day until the current day.

days2

Days to scratch - number of days from the current day to the predicted scratch day.

mount#

Total number of mount events from the current day to the predicted scratch day.

grade

Final calculated (or defaulted) grade.

text

Special causes that had influences on the grade calculation. The following causes (and grades) may be expected:

OK

The grade was calculated according to the standard formula.

IN

Grade=0. The tape dataset is explicitly included by the user.

EX

Grade=2147483647. The tape dataset is explicitly excluded by the user.

DP

Grade=2147483647. The diskpool the virtual tape dataset resides on could not be determined. Seems like HVTUDST runs before the HVTUSCR utility that should have scratched the virtual tape.

27

Grade=2147483646. First 27 characters of the tape dsnname could not be determined, so the generic name could not be derived.

CR

Grade=2147483646. Virtual tape creation date not set, grade could not be calculated.

HS

Grade=2147483646. No dataset record in the history file matches the tape dataset name or generic name. Seems like this is a new tape dataset.

DT

Grade=2147483646. Dates validation. Current date is lower than the virtual tape creation date.

OB

Grade=2147483646. Observations later than the current day do not exist. A full tape dataset cycle did not yet complete.

SC

Grade=2147483646. No observation that contains a scratch event exists. Seems like the scratch event from the previous cycle was not recorded.

SO

Grade=2147483646. The scratch day is the current HVTUDST running day. Seems like HVTUDST runs before the HVTUSCR utility that should have scratched the virtual tape.

M0

The grade was calculated according to the standard formula except that the number of mount events is set to 1 and 1 is subtracted from the final grade.

Possible values are:

An asterisk (*) indicates that the virtual tape dataset is selected for de-stage.

blank

If blank, the virtual tape dataset is not selected for de-stage.

dsn#

Number of different tape datasets grouped under the generic name.

+

The number of different tape dataset names grouped under the generic name (dsn#) exceeds the allowed maximum. That number is reduced to the displayed maximum.

blank

If blank, the dsn# value represents the real number of different tape dataset names grouped under the generic name.

name

The dataset name or generic name of the first tape dataset on the virtual tape.

System action:

None

User response:

None

HVT452W

**DE-STAGE JOB NOT PREPARED,
RC=rc (text)**

Explanation

During the de-stage job preparation phase, the HVTUDST utility found that the job preparation could not be completed. The return code value details the cause.

The following variables are listed in the message:

rc

Possible return codes are:

01

No virtual tape dataset found to be eligible for de-stage.

02

The skeleton member that fits the local storage management product could not be found. This member should be located in the SAMPLIB library as HVTSKxxx, where xxx is the PROD parameter value on the DRUN initialization statement.

03

The skeleton member that fits the local storage management product is empty.

04

The skeleton member that fits the local storage management product is bad. The expected REPEAT statements are missing.

05

HVTMIG DD statement open failed. The HVTUDST utility should have written de-stage job on it.

text

Short explanation of the error.

System action:

De-stage job is not submitted.

User response

Check why the de-stage job could not be prepared:

- If the diskpools space usage is lower than the predefined user threshold, the message can be ignored.
- If the skeleton is missing or corrupted, restore it from the initial product installation libraries.

- If the HVTMIG DD statement is bad, redefine it.

When completed, rerun the HVTUDST utility.

**HVT453I DE-STAGE JOB PREPARATION
WAS OK, *count* VIRTUAL TAPE
DATASETS INCLUDED**

Explanation

This is an informative message issued by the HVTUDST utility after it successfully prepared the destage job and wrote it to the dataset pointed to by HVTMIG DD statement.

The only variable in this message is *count*, which represents the number of virtual tape datasets included in the de-stage job.

System action:

The job is written to the dataset pointed to by the HVTMIG DD statement.

User response:

None

**HVT454W MONITORING *min#* MIN, *ds#1* OF
ds#2 DS NOT YET DE-STAGED,
PRESS ANY KEY TO STOP
MONITORING**

Explanation

The DRUN initialization statement for the HVTUDST utility specifies a WAIT parameter whose value is different from 0. In that case, the utility monitors the de-staging process.

This message is issued *nn* minutes after the de-staging process has started and each *nn* minutes thereafter (where *nn* is the value specified on the WAIT parameter of the DRUN initialization statement).

The following variables are listed in this message:

min#

Number of minutes that have passed since the HVTUDST utility started to monitor the de-staging process.

ds#1

Number of virtual tape datasets that are waiting to be de-staged.

ds#2

Original number of virtual tape datasets that were selected to be de-staged.

System action

HVTUDST continues to monitor the de-stage process. If any key is pressed, the monitoring stops and the utility completes, letting the de-staging process continue unmonitored. In that case, the utility completion code is 4.

If all virtual tape datasets are de-staged, the message is automatically deleted.

User response:

Press any key to allow the HVTUDST utility to complete without continuing to monitor the de-staging process. Otherwise, wait for all virtual tape datasets to be de-staged, in which case the message is automatically deleted.

**HVT455W HST FILE RESTORE *comp*, RC=*rc*
(text)**

Explanation

This message is issued by the HVTUDST utility after it restored the history file from its backup file. That restore was needed because the last HVTUDST utility run failed and the history file was left in an unpredictable status.

The following variables are listed in this message:

comp

The restore was either successful (WAS OK) or unsuccessful (FAILED).

rc

Possible return codes are:

00

The restore was OK.

01

The history file status is invalid. Valid statuses are FRE and UPD.

02

The history backup file status is invalid. The only valid status is BKP.

03

The history backup file does not belong to the current server environment.

System action

The system action depends on the return code:

00

Processing continues.

Other

HVTUDST utility processing stops.

User response

Check why the history or history backup file does not contain the expected data and take appropriate actions to fix it.

The status and environment name (the serverid) are located in the first file record (the control record) of each file, respectively.

If the problem is not located, collect and forward the appropriate documentation to IBM Technical Support for assistance.

**HVT500W VOLUME=*volume (state)* REJECTED
BY VTE EXIT#**

Explanation

A specific virtual tape mount was processed by a VTE component however a VTE exit rejected it.

The following variables are in the message:

volume

Indicates the mounted virtual tape volume serial number.

state

Indicates the state of the virtual tape. Valid values are:

KNOWN, SCRATCH

Defined in VDB and scratch.

KNOWN, ACTIVE

Defined in VDB and active.

UNKNOWN

Not defined in VDB.

UNKNOWN, OPER

Operator replied to HVT317W message with a specific volume serial number.

#

The VTE exit number.

System action

The system action depends on the environment from which the exit was invoked:

Server

This job is EQC-ed (Equipment Checked).

UV2V

The COPY of the specific virtual tape terminates and message HVT393I is issued. The utility then resumes with the next COPY initialization statement.

VAM

The OPEN function fails with a return code of 48. The application should then call the CLOSE function.

User response:

Provide another specific virtual tape. Inform the system programmer that supports the VTE exit or contact IBM Technical Support for assistance.

**HVT600E ISPF/SERVER ERROR, FUNC=*func*
ARG=*arg*, RC=*rc (text)***

Explanation

During an attempt to process a VTE Dialog request, the server or the VTE Dialog encountered an error.

The following variables are listed in the message:

func

Specifies the function code contained in the request.

arg

Specifies the first argument of the function code.

rc

Specifies the function return code. Values from 01 to 10 are issued by the server component. Values from 11 to 20 are issued by the VTE Dialog component.

- 01 – Invalid internal VTE Dialog function code.
- 02 – VSAM error while trying to point to an existing record. Additional messages may be displayed in the server log.
- 03 – VSAM error while trying to get an existing record. Additional messages may be displayed in the server log.
- 04 – Internal error.
- 05 – Internal error.
- 11 – Invalid internal VTE Dialog function code.
- 14 – Internal error.
- 15 – Internal error.
- 16 – Internal communication vector table entries are either:
 - Temporarily exhausted due to concurrent VTE Dialog users performing concurrent requests.
 - Permanently exhausted due to repeated errors that prevent entries reuse.

text

Short explanation of the error.

System action:

The VTE Dialog request fails and an appropriate message is issued to the initiating VTE Dialog user.

User response

The recommended action depends on the return code.

For rc = 01, 02, 03, 04, 05, 11, 14, 15

Collect and forward the appropriate documentation to Technical Support for assistance.

For rc = 16

Retry the request. If the problem persists, collect and forward the appropriate documentation to Technical Support for assistance.

HVT601E SSID *ssid*, RC=*rc*, (*text*)

Explanation

During an attempt to connect to the server, the VTE Dialog encountered an error. The message is contained in the ISPF long message text and is displayed on the respective panel when the VTE Dialog user presses PF1 as a result of the connect error.

The following variables are listed in this message:

ssid

The subsystem name of the VTE environment that should be connected.

rc

Return code. Valid values are:

- 01 – The VTE subsystem not found or not active.
- 02 – The VTE Dialog release and the VTE server release do not match.
- 03 – The VTE Dialog release and the VTE server release do not match.
- 04 – The VTE Dialog option is not enabled.
- 05 – The VTE Dialog PTF level and the VTE server PTF level do not match.

text

Short explanation of the error.

System action:

The connection is rejected and the message is moved to the ISPF long message text.

User response

The recommended action depends on the return code:

01

Check whether the SSID field value in panel PP02 matches the value of the VTE server subsystem name. Next, check whether the VTE server is active and whether its subsystem is enabled.

Depending on the findings, correct the error and retry the connection attempt.

02

The SSID field value in panel PP02 specifies the subsystem name of a VTE server of a different release. Correct the error and retry the connection attempt.

03

Check whether all load libraries concatenated to the TSO logon procedure are APF authorized. Next, check whether module HVTAPF is correctly added to the IKJTSOxx member and whether that member was refreshed after the VTE Dialog installation.

Depending on the findings, correct the error and retry the connection attempt.

04

Enable the VTE Dialog option by either issuing the 'F server,DLGON' operator command or by coding the DLG=ENABLE parameter on the OPTIONS initialization statement and bounce the VTE server.

05

This return code can mean one of the following:

- The VTE server was not bounced after the last PTF APPLY.
- The VTE Dialog and the VTE server use different load libraries.
- Any other issue that causes both environments not to be on the same PTF level.

HVT602I

**DIALOG ATTEMPT=userid/appl/
request(arg), RC=rc (text)**

Explanation

Informative message issued by the VTE server, listing all authorized requests made by VTE Dialog users. The message serves as an audit tool and is displayed on the server log.

If the user is not authorized for the request, the message is moved to the ISPF long message text and is displayed on the respective panel when the VTE Dialog user presses PF1 as a result of the request reject.

The following variables are listed in the message:

userid

Specifies the TSO userid of the VTE Dialog user.

appl

Specifies the VTE Dialog two-character application identifier under which the request was done.

request

Specifies the request name. It is identical to one of the request names specified in the HVTDLGxx member.

arg

Specifies the first request argument.

rc

Specifies the return code. Valid values are:

- 00 – Request granted.
- 01 – Request rejected, user not authorized for request.

text

Provides a short explanation of the error.

System action:

If the user is not authorized, the request is rejected.

User response:

For return code 01, if the user should not be authorized for the request (correctly rejected for this request) , no action is needed. However, if the user was rejected but should be allowed to be authorized for the request, update member HVTDLGxx in the SAMPLIB library and refresh it using the ' F server,DLG=xx ' operator command.

HVT990I *modname origin ptf modname
 origin ptf*

Explanation

This is an informative message that details each loaded module load point and PTF level. It is issued and printed on the HVTTRC sysout file by the server and all of its utilities after their initialization phase. The modules list is printed in two columns, in alphabetical order.

This message is preceded by a general title that describes each field name.

The following variables are listed in the message:

modname
Module name

origin
Module's load point address.

ptf
Module's last applied PTF.

System action:
None

User response:
None

HVT995I *step ddname volume+ r# a# m/n+
 i/e+ pth dsname*

Explanation

This is an informative message that is issued for each traced job whose jobname was specified in a preceding 'F server,WT00N=jobname' operator command. This message shows the dynamic reallocation decision path.

The following variables are listed in this message:

step
This message shows the dynamic reallocation decision path.

ddname
The DD statement currently traced.

volume
The virtual tape volume serial number that is one of the following:

- Explicitly specified on the DD statement.

- LOCATE-d in the system catalog. The LOCATE is performed only for the DD statements that refer to or allocate a cataloged dataset and are members of a VOL=REF group.

The volume is not displayed if a previous DD catalogs a GDG dataset (by means of GDG (+1)) and the current DD refers to it as GDG (+1).

Possible values for this character are:

+
A plus sign (+) indicates that the volume serial number is found in the VDB.

-
A minus sign (-) indicates that the volume serial number is not found in the VDB.

The following values are displayed only if the DD statement belongs to a UNIT=AFF chain.

An asterisk (*) looks like cataloged (that is, does not contain UNIT, VOL=SER or both), but not yet LOCATE-d.

:
A colon (:) can mean any of the following:

- Cataloged on a virtual device.
- Cataloged by a previous DD statement that contains a virtual device UNIT specification.
- Cataloged by a previous DD statement that contains an arbitrary specific UNIT to be reallocated to a virtual device.

--
Cataloged by a previous DD statement that contains another arbitrary specific UNIT that is not reallocated.

r#
The VOL=REF group internal sequential number. Each new group has a successive sequential number. The field is blank for DD statements that are not part of VOL=REF groups.

a#
The UNIT=AFF chain internal sequential number. Each new group has a successive sequential number. The field is blank for DD statements that are not part of UNIT=AFF chains. This value is shown after UNIT=AFF chains have been reorganized as a result of DD statement dynamic reallocation.

m/n
Possible values are:

MST

The DD statement **MUST** be reallocated (for example, due to the virtual tape volume serial number being a 'known' virtual tape).

NVR

The DD statement will **NEVER** be reallocated (for example, due to the virtual tape volume serial being an 'unknown' virtual tape).

MIX

The DD statement is marked for both **NEVER** reallocate and **MUST** reallocate.

+

Possible values are:

*

NEVER reallocate due to hardcoded conditions (like special DD statements, subsystem files or a specific request coded on the DD statement).

blank

MUST/NEVER reallocate due to another DD statement that belongs to the same **VOL=REF** group.

+

MUST/NEVER reallocate due to the current DD statement condition(s).

!

NEVER reallocate due to hyper 'NEVER-reallocate' specification.

i/e

Possible values are:

INC

The DD statement will be **INCluded** as a result of a dynamic reallocation rule definition.

EXC

The DD statement will be **EXCluded** as a result of a dynamic reallocation rule definition.

blank

No dynamic reallocation rule matches this DD statement.

+

Possible values are:

+

The **INClude/EXClude** decision was made according to dynamic reallocation rules that match this DD statement.

blank

The **INClude/EXClude** decision was made due to other **INCluded/EXCluded** DD statements that belong to the same **VOL=REF** group.

pth

Possible values are:

VLR

The DD statement is referred by other DD statements through **VOL=REF**. This specific DD statement specifies **VOL=REF=dsn** (where dsn is a cataloged dataset).

DSN

The DD statement is referred by other DD statements through **VOL=REF**. This specific DD statement specifies a cataloged dataset with no specific **VOL=SER** or **VOL=REF** parameter

dsname

The first 15 characters of the dataset name whose associated DD statement is referred by another DD statement with a **VOL=REF** parameter.

System action:

None

User response:

None

HVT998I *text*
Explanation:

This message follows message HVT311E and gives a more detailed explanation of the dynamic allocation error. The only variable is *text*, which is an explanation (prefixed by IKJ) of the dynamic allocation error.

System action:

See message HVT311E for system action.

User response:

See message HVT311E for operator action.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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:

*IBM Corporation
J64A/G4
555 Bailey Avenue*

San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM®, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered marks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at: <http://www.ibm.com/legal/copytrade.shtml>.

The Hitachi Control Platform (HCP) is a trademark of Hitachi Data Systems.

The Amazon Simple Storage Service (S3) and Amazon S3 are trademarks of Amazon Web Services.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

- abend errors [24](#)
- access list entries trace [311](#)
- ACTIVE (A) operator command [313](#)
- add virtual tapes [349](#), [351](#), [353](#), [356](#)
- adding cloud definitions [23](#)
- adding tapes
 - in a TMS [185](#)
- adding virtual tapes [341](#)
- AES256 encryption [27](#), [31](#), [42](#)
- ALCON/ALCOFF operator command [311](#)
- ALEON/ALEOFF operator command [311](#)
- alias
 - creating [15](#)
- allocation of space [26](#)
- Amazon Simple Storage Service (S3) [42](#)
- APF authorize SCUZLOAD [14](#)
- API function calls [379](#)
- API Key [39](#)
- APIB BUILD_CLOUD_LIST function [383](#)
- APIB COPY_TO_CLOUD function [384](#)
- APIB DELETE_FROM_CLOUD function [390](#)
- APIB FREE function [381](#)
- APIB GET_CLOUD_STATUS function [382](#)
- APIB Obtain function [381](#)
- APIB RESTORE_FROM_CLOUD function [386](#)
- ASG-Zara
 - defining tapepool rules [359](#)
 - defining tapepools [358](#)
- ASG-Zara database
 - ZARAUTL procedure [358](#)
- ASG-Zara environment
 - implementing VTE in [357](#)
- AT-TLS [17](#)
- AT-TLS parameters [377](#)
- authorities needed for Restore processing [14](#)

B

- backup
 - generations [24](#)
 - history data sets [24](#)
 - history tasks [24](#)
- backup filter criteria [23](#)
- Backup Filter Criteria options [48–50](#)
- backup history data set
 - include or exclude [97](#)
- backup history files [9](#)
- backup repository [87](#)
- BKP initialization statement [347](#)
- BMC CONTROL-T
 - implementing VTE [355](#)
- Bucket Name [31](#), [42](#)
- buffers [26](#)
- BUILD_CLOUD_LIST function [383](#)

C

- CA-1 TMC
 - adding virtual tapes [349](#)
 - defining virtual tapes [350](#)
- CA-Dynam/TLMS
 - defining tapepool rules [352](#)
- CA-Dynam/TLMS VMF
 - adding virtual tapes [351](#)
- CA-Dynam/TMLS
 - implementing VTE [351](#)
- Catalog to Cloud support [90](#)
- CCPON/CCPOFF operator command [311](#)
- CCW Summary trace [311](#)
- cell pool
 - extents [24](#)
 - size [24](#)
- certificates for clouds [17](#)
- changes to book [1](#)
- cloud copy methods [9](#)
- cloud data
 - restoring [83](#)
- cloud data sets
 - deleting [80](#), [86](#)
 - restoring [80](#), [84](#)
 - selection criteria [80](#)
 - viewing [80](#)
- cloud definitions [23](#)
- cloud name [27](#), [30](#), [31](#), [34](#), [37](#), [39](#), [42](#), [48](#)
- Cloud Server options [27](#), [30](#), [46](#), [47](#)
- Cloud Server Status [53](#)
- cloud type [27](#), [30](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- cloud types
 - FTP [34](#)
 - HCP [37](#)
 - S3 [42](#)
 - S3-compatible [31](#)
 - SFT [39](#)
- column display functions
 - about [59](#)
 - CEXPAND [70](#)
 - CFIX [60](#)
 - CLEFT [69](#)
 - CNUM [69](#)
 - COLS [69](#)
 - CORDER [62](#)
 - CREMOVE [68](#)
 - CRESET [68](#)
 - CRIGHT [69](#)
 - CSIZE [63](#)
 - CSORT [65](#)
 - fastpath SORT [66](#)
 - ICLEFT [69](#)
 - ICRIGHT [69](#)
 - restrictions [70](#)
- commands
 - delete data set [83](#)

- commands (*continued*)
 - exclude data set [83](#)
 - restore data set [83](#)
- Compare tapes [105](#)
- components of VTE [119](#)
- Compress [27](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- compressing
 - virtual tape data set [180](#)
- Configure
 - choose method [7](#)
- Configure product
 - APF authorize SCUZLOAD [14](#)
 - AT-TLS parameters [377](#)
 - Authorize TSO functions [14](#)
 - Copy ISPF members to system CLIST library [15](#)
 - Copy started task JCL and customize [16](#)
 - Create runtime libraries [12](#)
 - Create user parameter library [13](#)
 - Customize ISPF CLIST [15](#)
 - Customize parmlib options via ISPF [16](#)
 - Define a z/OS repository file [16](#)
 - Define started task to security system [17](#)
 - Establish security for parameter library [13](#)
 - Start the started task [20](#)
 - Summary of steps [12](#)
- configuring product [14](#)
- Connection Retry Attempts [27](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- Connection Retry Time [27](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- console commands
 - DISPLAY [71](#)
 - REFRESH [72](#)
 - RESTORE [72](#)
 - RESTORE CLD [73](#)
 - START [79](#)
 - STOP [79](#)
- Console commands
 - ZIIP [80](#)
- Container [39](#)
- CONTROL-T
 - defining tapepool rules [357](#)
- CONTROL-T MDB
 - adding virtual tapes [356](#)
- cookie policy [573](#)
- COPY initialization statement [344](#)
- COPY_TO_CLOUD function [384](#)
- copying a volume [343](#)
- CPU operator command [314](#)
- creating a cloud server definition [30](#)
- creating a new cloud filter [48](#)
- creating a new cloud server definition [27](#)
- creating cloud definitions [23](#)
- creating new cloud filter criteria [49](#)
- criteria for backups [23](#)
- criteria for selecting data sets [80](#)
- CSETUP [59](#)
- CTC virtual tape data set [317](#)
- customization
 - Dialog (ISPF panels) [224](#)
- customization of product [229](#)
- customization steps
 - Step 1: Hardware configuration [229](#)
 - Step 10: Add virtual tapes to VDB [250](#)
 - Step 2: Setup Steps [235](#)
 - Step 3: Security requirements [237](#)

- customization steps (*continued*)
 - Step 4: System settings [238](#)
 - Step 5: JES3 customization [242](#)
 - Step 6: e-Vault installation [246](#)
 - Step 7: Third-party product pre-reqs [246](#)
 - Step 8: Start VTE [249](#)
 - Step 9: Run the IVP jobs [249](#)
- CUZ#APIB macro
 - BUILD_CLOUD_LIST [383](#)
 - COPY_TO_CLOUD [384](#)
 - DELETE_FROM_CLOUD [390](#)
 - FREE [381](#)
 - GET_CLOUD_STATUS [382](#)
 - RESTORE_FROM_CLOUD [386](#)
- CUZ#APIB module [379](#)
- CUZ#APIB sample JCL [391](#)
- CUZ#APIT module [379](#)
- CUZ#APIT sample member [391](#)
- CUZ#INEX [16](#)

D

- Data recovery options [50](#)
- data restoration options [23](#)
- Data Retry [27](#), [31](#), [42](#)
- data set
 - Parmlib [22](#), [97](#)
- data set selection criteria [80](#)
- data sets on cloud
 - managing [80](#)
- Data Transfer Retry [27](#), [31](#), [42](#)
- database reports
 - printing [326](#)
- Dataset Analysis Report [106](#)
- Dataset Compare Report [105](#), [106](#)
- DD statement
 - LOGMSG [72](#), [113](#)
- de-stage priority [317](#)
- de-staging
 - virtual tape data set [171](#), [172](#)
- Debug
 - job name [24](#)
 - mode [24](#)
- default tapepool
 - \$\$DPLT [355](#)
- define tape pools [350](#)
- defining
 - tapepools [185](#)
- defining a cloud filter [48](#)
- defining a cloud server [27](#)
- defining cloud filter criteria [49](#)
- defining cloud servers [23](#)
- defining tapepool rules
 - for ASG-Zara [359](#)
 - to CA-1 [350](#)
 - to CA-DYnam/TLMS [352](#)
 - to CONTROL-T [357](#)
 - to DFSMSrmm [355](#)
 - to VTE [354](#), [356](#)
- defining tapepools
 - to ASG-Zara [358](#)
 - to CA Dynam/TMLS [352](#)
 - to CONTROL-T [356](#)
 - to DFSMSrmm [354](#)

- defining tapepools (*continued*)
 - to VTE [352](#)
- definitions
 - e-Vault [289](#)
- DELETE_FROM_CLOUD function [390](#)
- deleting a cloud filter [50](#)
- deleting a cloud server definition [47](#)
- deleting a new cloud filter [48](#)
- deleting a new cloud server definition [27](#)
- deleting data sets from the cloud [80](#), [86](#)
- deleting virtual tapes [341](#)
- DFSMSrmm
 - defining tapepool rules [355](#)
- DFSMSrmm TCDB
 - adding virtual tapes [353](#)
- diagnosing problems with trace facilities [311](#)
- Dialog
 - set default and enable [225](#)
- Directory [31](#), [34](#), [37](#), [39](#), [42](#)
- Directory path [27](#)
- Directory Path [30](#)
- disk space
 - managing [165](#)
- Disk storage [373](#)
- diskpool
 - reconfiguring [183](#)
- diskpool rules [309](#)
- diskpool utilization [313](#)
- diskpools
 - managing [165](#)
- diskpools infrastructure [268](#)
- DISPLAY (D) operator command [313](#)
- DISPLAY command [71](#)
- documentation changes [1](#)
- DPOOL initialization statement [269](#)
- DPTRN initialization statement [317](#)
- DRULE initialization statement [317](#)
- DRUN initialization statement [317](#)
- DSPACE (DSP) operator command [313](#)
- dynamic allocation trace [311](#)
- dynamic reallocation
 - decision logic [145](#)
 - examples [147](#)
 - rules [145](#)
- dynamic reallocation decision tree trace [311](#)
- dynamic reallocation logic [145](#)
- dynamic reallocation rules [282](#), [309](#)

E

- e-Vault
 - complex implementation [204](#)
 - implicit return [195](#)
 - off-site storage locations [195](#)
 - process flow [195](#)
 - processing interval [303](#)
 - remote search [195](#)
 - suspend processing [303](#)
 - types [191](#)
- e-Vault complex definitions [289](#)
- e-Vault virtual tape status change trace [311](#)
- editing a cloud filter [48](#), [49](#)
- editing a cloud server definition [27](#), [46](#)
- editing cloud definition [46](#)

- electronic
 - vaulting [191](#)
- enable/disable features [303](#)
- encryption [17](#)
- Encryption AES256 [27](#), [31](#), [42](#)
- equipment check [189](#)
- error messages [401](#)
- error retry
 - count [26](#)
 - interval seconds [26](#)
- errors and abends [24](#)
- evault type 3 [200](#)
- evault type 4 [201](#)
- evault type 5 [202](#)
- events
 - handling exceptional [189](#)
- example
 - Dataset Compare Report - Different [106](#)
 - Dataset Compare Report - Same [105](#)
- exclude history data set [97](#)
- Exec History Interval Min [24](#)
- exits
 - EXIT001 [361](#)
 - EXIT002 [362](#)
 - EXIT004 [365](#)
 - EXIT006 [366](#)
- extended MCS console support [253](#)

F

- File Transfer Protocol [34](#)
- filter
 - backup history data sets [97](#)
- filter criteria
 - catalog to cloud [90](#)
- filter type [48](#)
- Filter Type [48](#)
- FIND [55](#)
- FORCE operator command [300](#)
- FORM [55](#)
- FORMAT initialization statement [322](#)
- FREE function [381](#)
- freeing server resources [315](#)
- FTP [34](#)
- FTP Type3 e-Vault [191](#)
- FTP Type4 e-Vault [191](#)
- FTP virtual tape data set [321](#)
- FUNCTION
 - BUILD_CLOUD_LIST [383](#)
 - COPY_TO_CLOUD [384](#)
 - DELETE_FROM_CLOUD [390](#)
 - FREE [381](#)
 - GET_CLOUD_STATUS [382](#)
 - OBTAIN [381](#)
 - RESTORE_FROM_CLOUD [386](#)

G

- GDG base
 - non-SMS data sets [78](#)
- GDG parameters
 - LIMIT [74](#)
 - NOSCRATCH [74](#)

- GDG parameters (*continued*)
 - SCRATCH [74](#)
- general options for Parmlib [24](#)
- general parmlib options [23](#)
- generation data group (GDG)
 - SMS-managed GDG [74](#)
- generation data group (GDG) [75](#)
- generations [24](#)
- GET_CLOUD_STATUS function [382](#)
- GMT format [20](#)
- GTF-like trace [311](#)
- GTFOF/GTFOFF operator command [311](#)

H

- HCP cloud type [37](#)
- history data set filter [97](#)
- history data sets [16](#), [24](#), [97](#)
- history file backup [9](#)
- Hitachi Content Platform [37](#)
- HVTDLGxx member
 - set default suffix [225](#)
- HVTMCSxx member [253](#)
- HVTOPTxx member [256](#)
- HVTPOLxx member [268](#)
- HVTPRLxx member [277](#)
- HVTRULxx member [282](#)
- HVTSYSxx member [286](#)
- HVTTRLxx member [288](#)
- HVTUCTC utility [317](#), [321](#)
- HVTUDST utility [317](#)
- HVTUFTP utility [321](#)
- HVTUJFR utility [322](#)
- HVTUJRC utility [324](#)
- HVTURCV utility [325](#)
- HVTUREP utility [326](#)
- HVTUSCR utility
 - special processes [340](#)
- HVTUTAP utility [341](#), [358](#)
- HVTUV2V utility
 - special considerations [344](#)
- HVTUVBK utility [346](#)
- HVTUVSI utility [347](#)
- HVTVLtxx member [289](#)

I

- I/O buffers [26](#)
- IBM SoftLayer server [39](#)
- IDENT initialization statement [256](#)
- include history data set [97](#)
- Include/Exclude data sets list [16](#)
- initialization statement
 - TAP [341](#)
- initialization statements
 - BKP [347](#)
 - COPY [344](#)
 - DPOOL [269](#)
 - DPTRN [317](#)
 - DRULE [317](#)
 - DRUN [317](#)
 - FORMAT [322](#)
 - HVTDLGxx [226](#)

initialization statements (*continued*)

- HVTMCSxx member [253](#)
- HVTOPTxx [256](#)
- HVTPOLxx member [268](#)
- HVTPRLxx member [277](#)
- HVTRULxx member [282](#)
- HVTSYSxx member [286](#)
- HVTTRLxx member [288](#)
- HVTVLtxx member [289](#)
- IDENT [256](#)
- JRN [256](#)
- MCS [253](#)
- OPTIONS [257](#)
- PAT [259](#)
- PERFORM [260](#)
- PFILT [279](#)
- PPRTN [279](#)
- PRULE [280](#)
- REPORT [326](#)
- RFILT [283](#)
- RPTRN [283](#)
- RULE [284](#)
- SMF [262](#)
- SUBSYS [262](#)
- SYNC [332](#)
- syntax [252](#)
- SYS [287](#)
- TCP [262](#)
- TPOOL [271](#)
- VDB [263](#)
- VDEVICE [264](#), [267](#)
- VLROWN [290](#)
- VLTRMT [292](#)
- internal flow trace [311](#)
- interval for e-Vault [303](#)
- intialization statements
 - members [251](#)
- IP address [27](#), [30](#), [31](#), [34](#), [37](#), [39](#)
- ISPF CLIST [15](#)
- ISPF interface messages [401](#)
- ISPF members
 - copy to system CLIST library [15](#)
- ISPF panels
 - customizing dialog [224](#)
- IVP jobs [249](#)

J

- JCL
 - started task [16](#)
- JES3 environment [187](#)
- jobs
 - IVP [249](#)
- journal file
 - management [164](#)
- journal file format [322](#)
- journal file size
 - planning [322](#)
- JRN initialization statement [256](#)

K

- Key ID [31](#), [42](#)

KILL operator command [315](#)

L

legal notices

cookie policy [573](#)

notices [573](#)

trademarks [573](#)

LIMIT parameter [74](#)

line commands

delete data set [83](#)

exclude data set [83](#)

restore data set [83](#)

Line commands

Restore (R) [84](#)

links

non-IBM Web sites [574](#)

LOGMSG DD statement [72](#), [113](#)

M

Main Menu [21](#)

managing

disk space [165](#)

diskpools [165](#)

managing virtual tape library [155](#)

Max restore tasks [50](#)

MCS console support [253](#)

MCS initialization statement [253](#)

memory

cell extents [24](#)

cell pool size [24](#)

primary cells [24](#)

secondary cells [24](#)

messages

ISPF interface [401](#)

Write-to-Operator (WTO) [24](#)

methods of cloud copy [9](#)

methods of configuration [7](#)

migrating

virtual tape data set [171](#)

MODIFY commands [71](#)

Modify operator command [302](#)

MODIFY operator command [300](#)

N

Non-SMS managed data sets

rolling into GDG base [78](#)

NOSCRATCH parameter [74](#)

notices [573](#)

NROW [55](#)

O

OBTAIN function [381](#)

offline utilities

VTE procedure [297](#)

operation files [373](#)

operational issues [71](#)

operator commands

ACTIVE (A) [313](#)

ALCON/ALCOFF [311](#)

operator commands (*continued*)

ALEON/ALEOFF [311](#)

CCPON/CCPOFF [311](#)

CPU [314](#)

DISPLAY (D) [313](#)

DRYON/DRYOFF [303](#)

DSPACE (DSP) [313](#)

for tracing facilities [311](#)

FORCE [300](#)

GTFON/GTFOFF [311](#)

KILL [315](#)

MODIFY [300](#), [302](#)

PATON/PATOFF [303](#)

POL [309](#)

restrictions [316](#)

SSNON/SNNOFF [303](#)

START [299](#)

STOP [300](#)

TCPON/TCPOFF [311](#)

TRCON/TRCOFF [311](#)

TSPACE (TSP) [313](#)

unsupported [316](#)

VLTON/VLTOFF [311](#)

VLRET [303](#)

VLTSUS [303](#)

WTOON/WTOOFF [311](#)

options for configuring product [7](#)

OPTIONS initialization statement [257](#)

P

Parallel Access Tape (PAT)

rules [309](#)

Parameter library [13](#)

parameter library security [13](#)

parameters

LIMIT [74](#)

NOSCRATCH [74](#)

Roll into GDG Base [77](#)

SCRATCH [74](#)

Parmlib

Backup Filter Criteria

creating a new filter [49](#)

deleting cloud filter criteria [50](#)

editing cloud filter criteria [49](#)

Cloud Server options

delete a definition [47](#)

edit a definition [46](#)

general options [24](#)

history member [24](#)

Save/Update member [52](#)

Staging options [26](#)

PARMLIB

Backup Filter Criteria [48](#)

Data recovery options [50](#)

Parmlib data set [97](#)

Parmlib dataset [22](#)

Parmlib Dataset [52](#), [97](#)

Parmlib member [22](#), [97](#)

Parmlib Member [52](#), [97](#)

parmlib options

customizing via ISPF [16](#)

Parmlib options [21](#)

Parmlib Options [23](#)

- Part Size [27, 31, 42](#)
- Password [27, 30, 34, 37](#)
- PAT initialization statement [259](#)
- PAT rules [277](#)
- PERFORM initialization statement [260](#)
- PFILT initialization statement [279](#)
- PKDS Label [27, 31, 34, 37, 39, 42](#)
- POL operator command [309](#)
- pool size [24](#)
- Port [27, 31, 34, 37, 39, 42](#)
- PPTRN initialization statement [279](#)
- primary commands
 - refresh data set list [83](#)
 - RESTORE [86](#)
 - restore data set [83](#)
- printing database reports [326](#)
- PRINTX [55](#)
- PRL operator command [309](#)
- procedure
 - VTE server [297](#)
- Product configuration
 - Summary of steps [12](#)
- product customization
 - steps [229](#)
- product enhancements [1](#)
- product messages [401](#)
- Product settings [21](#)
- Protocol type [30](#)
- PROW [55](#)
- Proxy Server [31](#)
- PRULE initialization statement [280](#)

R

- re-catalog data set on cloud [90](#)
- recalling
 - virtual tape data set [171](#)
- reconfiguring diskpool [183](#)
- recovering
 - VDB [324](#)
 - virtual tapes [325](#)
- recovering virtual tapes [159](#)
- recovery considerations [200–202](#)
- recovery options [50](#)
- REFRESH command [72](#)
- refreshing
 - dynamic reallocation [309](#)
 - Parallel Access Tape (PAT) rules [309](#)
 - server parameters [309](#)
 - tapepool and diskpool rules [309](#)
- REP001 report [326, 328](#)
- REP002 report [326, 328](#)
- REPORT initialization statement [326](#)
- reports
 - Dataset Analysis Report [106](#)
 - Dataset Compare Report [105, 106](#)
 - printing [326](#)
 - REP001 [326](#)
 - REP002 [326](#)
- Reports
 - printing [328](#)
- repository
 - backup [87](#)
- Repository

- Repository (*continued*)
 - restoring [89](#)
- repository file [16](#)
- requirements
 - storage [371](#)
- resource utilization
 - tuning server [314](#)
- Restore (R) line command [84](#)
- restore alias [15](#)
- Restore alias [50](#)
- Restore authority [14](#)
- RESTORE CLD command [73](#)
- RESTORE command [72](#)
- Restore DASD unit [50](#)
- Restore data from cloud [83](#)
- restore options [23](#)
- RESTORE primary command [86](#)
- Restore tape unit [50](#)
- Restore to DASD [50](#)
- RESTORE_FROM_CLOUD function [386](#)
- restoring
 - SMS-managed GDG [75](#)
- restoring cloud data sets to tape [80](#)
- restoring data sets from the cloud to tape [84](#)
- restoring GDG's not managed by SMS [78](#)
- restrictions for operator commands [316](#)
- retention period [48, 50](#)
- Retry Attempts [27, 31, 34, 37, 39, 42](#)
- Retry Time [27, 31, 34, 37, 39, 42](#)
- return code [24](#)
- RFILT initialization statement [283](#)
- Roll into GDG Base parameter
 - non-SMS managed data sets [78](#)
 - values for SMS DASD data sets [77](#)
 - values for SMS Tape data sets [77](#)
- RPTRN initialization statement [283](#)
- RUL operator command [309](#)
- RULE initialization statement [284](#)
- rules
 - dynamic reallocation [282](#)
 - for tapepools [288](#)
 - PAT [277](#)
- Runtime libraries [12, 13](#)

S

- S3 cloud type [31, 42](#)
- S3-compatible cloud [31](#)
- sample JCL
 - CUZ#APIT [391](#)
- sample library
 - CUZ#APIT for API function calls [379](#)
- Save Parmlib Member options [52](#)
- SCRATCH parameter [74](#)
- scratch virtual tapes
 - controlling inventory [161](#)
- SCUZLOAD
 - APF authorization [14](#)
- SCUZSAMP
 - CUZJINCL [108](#)
- Secret Key [31, 42](#)
- secure data transmissions [17](#)
- security for parameter library [13](#)
- security for Restore processing [14](#)

- selection criteria for data sets [80](#)
- server CPU utilization [314](#)
- server definitions [47](#)
- Server definitions [27](#), [30](#)
- Server type [27](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- Server-side encryption [31](#)
- Server-side Encryption [42](#)
- servers
 - multiple in multiple systems [135](#)
 - multiple in single system [136](#)
- setting de-stage priority [317](#)
- Settings [21](#)
- Settings for products [23](#)
- SFT cloud type [39](#)
- sharing
 - virtual tapes [137](#)
- shutdown of VTE
 - forced [300](#)
 - normal [300](#)
 - unconditional [300](#)
- SMF initialization statement [262](#)
- SMF Record Type [24](#)
- SMS information [15](#)
- SMS storage class [26](#), [50](#)
- SMS-managed GDG
 - restoring [75](#)
- SORT [55](#)
- space allocation [26](#)
- staging alias [15](#)
- staging data [9](#)
- staging options [23](#)
- Staging options [26](#)
- START command [79](#)
- START operator command [299](#)
- started task
 - defining [17](#)
- started task JCL [16](#)
- starting product [297](#)
- starting VTE [249](#)
- statements
 - initialization [251](#), [252](#)
- status of cloud servers [53](#)
- STOP command [79](#)
- STOP operator command [300](#)
- stopping product [297](#)
- storage requirements
 - calculation examples [372](#)
 - estimating [372](#)
 - estimating total storage used [371](#)
 - for disks [373](#)
 - of VTE [371](#)
 - operation files [373](#)
- SUBSYS initialization statement [262](#)
- suspend e-Vault processing [303](#)
- SYNC initialization statement [332](#)
- synchronizing tape status [332](#)
- SYS initialization statement [287](#)
- system CLIST library [15](#)

T

- TAP initialization statement [341](#)
- Tape Compare utility [105](#)
- tape management system (TMS)

- tape management system (TMS) (*continued*)
 - common definitions [185](#)
- tape management systems
 - working with VTE [349](#)
- tape pools
 - defining [350](#)
- tape status
 - synchronizing [332](#)
- tapepool rules
 - defining for ASG-Zara [359](#)
 - defining for CA-1 [350](#)
 - defining for CA-Dynam/TLMS [352](#)
 - defining for CONTROL-T [357](#)
 - defining for DFSMSrmm [355](#)
- tapepool utilization [313](#)
- tapepools
 - defining [185](#)
 - defining for ASG-Zara [358](#)
 - defining for CA Dynam/TLMS [352](#)
 - defining for CONTROL-T [356](#)
 - defining for DFSMSrmm [354](#)
 - defining for VTE [352](#), [354](#), [356](#), [358](#)
 - defining to ASG-Zara [358](#)
- tapepools infrastructure [268](#)
- tapes
 - adding to VDB [358](#)
- TCP initialization statement [262](#)
- TCP/IP flow trace [311](#)
- TCP/IP Type5 e-Vault [191](#)
- TCPON/TCPOFF operator command [311](#)
- terminating I/O subtask [315](#)
- testing direct cloud write [9](#)
- Threads [27](#), [31](#), [42](#)
- Time-out of socket Sec [31](#), [39](#), [42](#)
- TPOOL initialization statement [271](#)
- tracing operations
 - access list entries trace [311](#)
 - CCW Summary trace [311](#)
 - dynamic allocation trace [311](#)
 - dynamic reallocation decision tree trace [311](#)
 - e-Vault virtual tape status change trace [311](#)
 - GTF-like trace [311](#)
 - internal flow trace [311](#)
 - TCP/IP flow trace [311](#)
- trademarks [573](#)
- transmission [17](#)
- TRCON/TRCOFF operator command [311](#)
- TRL operator command [309](#)
- troubleshooting problems [311](#)
- TSO functions [14](#)
- TSO setup [223](#)
- TSPACE (TSP) operator command [313](#)
- turn on/off dynamic reallocation [303](#)
- turn on/off Parallel Access Tape (PAT) [303](#)
- turn on/off subsystem [303](#)
- type of protocol [30](#)
- type of server [27](#), [31](#), [34](#), [37](#), [39](#), [42](#)
- Type3 e-Vault [191](#)
- Type4 e-Vault [191](#)
- Type5 e-Vault [191](#)

U

- unsupported operator commands

unsupported operator commands (*continued*)

- DS, P [316](#)
- MOUNT, VOL [316](#)
- SWAP [316](#)
- UNLOAD [316](#)

Update Parmlib Member options [52](#)

User ID [27](#), [30](#), [34](#), [37](#), [39](#)

User parameter library [13](#)

utilities

- HVTUCTC [317](#), [321](#)
- HVTUDST [317](#)
- HVTUFTP [321](#)
- HVTUJFR [322](#)
- HVTUJRC [324](#)
- HVTURCV [325](#)
- HVTUREP [326](#)
- HVTUSCR [332](#)
- HVTUTAP [341](#)
- HVTUV2V [343](#)
- HVTUVBK [346](#)
- HVTUVSI [347](#)
- offline [251](#), [317](#)
- Tape Compare [105](#)

V

vault

- traditional [191](#)

vaulting

- electronic [191](#)

VDB

- adding virtual tapes [358](#)
- adding virtual tapes to [349](#), [351](#), [353](#), [356](#)
- backing up [346](#)
- initialize new one [347](#)
- synchronizing with tape management system database [332](#)

VDB initialization statement [263](#)

VDB recovery [324](#)

VDEVICE initialization statement [264](#), [267](#)

verifying tape data sets [105](#)

verifying virtual tapes [158](#)

viewing cloud data sets [80](#)

virtual database

- integrity [162](#)
- verification [162](#)

virtual device activity [313](#)

Virtual Host URL Style [27](#), [31](#), [42](#)

virtual tape

- compressing data set [180](#)
- data set [156](#)
- database (VDB) [155](#)
- destaging data set [172](#)
- integrity and verification [158](#)
- library [155](#), [157](#)
- library, activity tracking [164](#)
- migrating data set [171](#)
- recovering [159](#)
- scratch inventory [161](#)
- sharing [137](#)
- sharing, examples [141](#)

virtual tapes

- adding [349](#), [351](#), [353](#), [356](#)
- adding or deleting [341](#)

virtual tapes (*continued*)

- adding to VDB [358](#)

- synchronizing status [332](#)

virtual tapes recovery [325](#)

VLTON/VLTOFF operator command [311](#)

VLTON initialization statement [290](#)

VLTRMT initialization statement [292](#)

volumes

- copying [343](#)

VTE

- and tape management systems [349](#)

- sample JCL [391](#)

VTE exits

- EXIT001 [361](#)

- EXIT002 [362](#)

- EXIT004 [365](#)

- EXIT006 [366](#)

VTE main storage [372](#)

VTE procedure [297](#)

VTE tape pools

- defining [350](#)

W

write directly to cloud [9](#)

Write-to-Operator (WTO) messages [24](#)

WTOON/WTOOFF operator command [311](#)

Z

ZARAUTL procedure for ASG-Zara [358](#)

ZIIP command [80](#)



Product Number: 5698-ABM

SC27-9403-03

