EMC® Disk Library for mainframe

DLm Releases 1.2, 2.x, 3.x, and 4.x

Command Processors and Utilities for z/OS Guide

P/N 302-002-164
REV 01

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Published February 2016

Contents

Preface

As part of an effort to improve and enhance the performance and capabilities of its product lines, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information about product features, refer to your product release notes. If a product does not function properly or does not function as described in this document, please contact your EMC representative.

Audience

This document is part of the EMC Disk Library for mainframe (DLm) documentation set, and is intended for use by system operators to assist them in their day-to-day operations.

Readers of this document are expected to be familiar with tape library operations and the associated tasks in the mainframe environment.

This document is applicable to DLm releases 1.2, 2.x, 3.x, and 4.x.

Related documentation

Related documents include:

- *EMC Disk Library for mainframe User Guide*
- *EMC Disk Library for mainframe Release Notes*

The EMC documents specified here and additional VNX information are available on http://support.emc.com. The Data Domain documents specified here and additional Data Domain information are available in the Data Domain portal: https://my.datadomain.com. Data Domain documentation is also available on the Data Domain documentation CD that is delivered with DD880/DD990.

Conventions used in this document

EMC uses the following conventions for special notices:

Note: A note presents information that is important, but not hazard-related.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment.

IMPORTANT

An important notice contains information essential to software or hardware operation.

WARNING

A warning contains information essential to avoid a hazard that can cause severe personal injury, death, or substantial property damage if you ignore the warning.

DANGER

A danger notice contains information essential to avoid a hazard that will cause severe personal injury, death, or substantial property damage if you ignore the message.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, notifications
Bold	Used in running (nonprocedural) text for: • Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, man pages

	Used in procedures for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • What user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis (for example a new term) • Variables
`Courier`	Used for: • System output, such as an error message • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for: • Specific user input (such as commands)
`Courier italic`	Used in procedures for: • Variables on command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means "or"
{ }	Braces indicate content that you must specify (that is, x or y or z)
...	Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support, product, and licensing information can be obtained as follows:

Product information — For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at:

http://support.emc.com

Technical support — Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

Your comments Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this document to: techpubcomments@EMC.com

1

Overview

This chapter provides an overview of the EMC Disk Library for mainframe (DLm) Command Processors and the DLm utilities for the z/OS environment.

Topics include:

Introduction to Command Processors and Utilities

EMC® provides a set of utilities and a UIM for the z/OS environments along with Disk Library for mainframe (DLm). It also provides a number of command processors that enable you to issue DLm commands and monitor DLm status from the mainframe.

DLm utilities

The DLm utilities for the z/OS environments are:

- GENSTATS - A utility that generates reports from VTE and VOLSER range statistics.
- DLMSCR - A scratch utility that sends VOLSER scratch requests to DLm.
- DLMCMD - A utility that allows the mainframe to send DLm commands.
- DLMLIB - A utility that is required to define scratch volumes to an MTL.
- DLMVER - A utility that reports the versions of all DLm mainframe utilities on themainframe and the z/OS release.
- DLMHOST - A host utility that provides z/OS Console Operation support.
- DLMWTOR - A utility that allows a WTOR to be issued to the z/OS Console with up to 10 lines of user defined text. This provides a means to require z/OS operator confirmation of a DLm-related change such as halting all DLm Tape activity.

DLm Command Processors

DLm Command Processors (CPxxx) enable you to:

- Issue DLm commands from the mainframe
- Monitor DLm status from the mainframe
- Return the output of DLm commands and tasks to the mainframe

The following command processors are provided with DLm:

- CP499: Provide DLm diagnosics.
- CP501: Execute VTE command.
- CP502: Obtain VTE log information.
- CP503: DLm Volume Inventory Report.
- CP504: Dlm VNX Replication Report.
- CP601: DLm Data Domain System Utilization report.
- CP602: DLm Data Domain Deduplication Report.
- CP603: DLm Data Domain Replication Report.
- CP998: Obtain DLm File System Activity Statistics.
- CP999: Obtain DLm VTE Activity Statistics.

Gathering and Reporting DLm Statistics Information

The DLm maintains activity information that records information such as the mounts, IOs, and unloads of tape volumes. This information can then be used for various DLm reports using the following steps:

1. Gathering DLm Statistics Information.

   DLm Statistics can be collected for individual virtual tape engines, (VTEs) using Command Processor 999, or for an entire tape library using Command Processor 998. Information regarding this step is documented in Chapter 3 and Chapter 4.

2. Reporting on DLm Activity.

   DLm activity reporting uses the gathered statistics information to generate activity reports. This step is performed by a DLm z/OS utility called GENSTATS. Information regarding this DLm utility is documented in Chapter 2.

There are a number of PROCs that consolidate the above 2 steps into one process for the gathering and reporting of DLm activity. These PROCs, (GENSTATP, GENSTAT1, GENSTATW, and GENSTATR), are documented in Chapter 5.

DLm Command Processor support

Releases before DLm 1.2 do not support Command Processors. Table 1 on page 12 identifies the CP*xxx* commands and the DLm 1.x and 2.x releases that support them.

Table 1 **CP*xxx* commands supported by DLm 1.x and 2.x releases**

CP*xxx* command	DLm release				
	1.2	2.1	2.2	2.3, 2.4 without DD	2.3, 2.4 with DD
CP499	No	No	No	No	No
CP501	Yes	Yes	Yes	Yes	Yes
CP502	Yes	Yes	Yes	Yes	Yes
CP503	Yes	Yes	Yes	Yes	Yes
CP504	No	No	Yes	Yes	Yes
CP601	No	No	No	No	Yes
CP602	No	No	No	No	Yes
CP603	No	No	No	No	Yes
CP998	No	Yes	Yes	Yes	Yes
CP999	No	Yes	Yes	Yes	Yes

Table 2 on page 12 identifies the CP*xxx* commands and the DLm 3.x and 4.x releases that support them.

Table 2 **CP*xxx* commands supported by DLm 3.x and 4.x releases**

CP*xxx* command	DLm release				
	3.0, 3.1, 3.2, 3.3, 3.5, 3.5.2 without DD	3.0, 3.1, 3.2, 3.3, 3.5.0, 3.5.2 with DD	3.4,3.5.1 without DD	4.x.x with DD	4.x.x without DD
CP499	No	No	No	Yes	Yes
CP501	Yes	Yes	Yes	Yes	Yes
CP502	Yes	Yes	Yes	Yes	Yes
CP503	Yes	Yes	Yes	Yes	Yes
CP504	Yes	Yes	Yes	Yes	Yes
CP601	No	Yes	No	Yes	No
CP602	No	Yes	No	Yes	No
CP603	No	Yes	No	Yes	No
CP998	Yes	Yes	Yes	Yes	Yes
CP999	Yes	Yes	Yes	Yes	Yes

Note: Only a single execution of a particular Command Processor can run at a time. DLm does not support multiple executions of the same Command Processor simultaneously.

Install the DLm Command Processors

In DLm releases 3.x and 4.x, EMC service personnel install the DLm Command Processors during initial setup and system upgrades for all models except DLm1000.

Appendix A, "Installation on DLm v1.2, v2.x, and DLm1000," provides instructions to install DLm Command Processors on the other releases of DLm.

Table 3 provides the DLm releases and the sections containing the corresponding CP installation instructions.

Table 3 **Command processor installation instructions**

DLm release and models	Location of installation instructions
Release 1.2 - all DLm models	"Install DLm Command Processors on DLm Version 1.2" on page 188
Release 2.x - all DLm models	"Install DLm Command Processors on DLm Versions 2.x" on page 182
DLm1000	"Installation on DLm1000" on page 191 **Note:** Command Processors are supported only on DLm1000 systems that utilize the pre-defined internal IP addresses for Data Domain Storage. Command Processors are NOT supported on DLm1000 systems that have customer-defined IP addresses for Data Domain Storage.

Downloading and using DLm z/OS components

There are two installation procedures for DLm Utilities and JCL for z/OS: one using SMP/E and one non-SMP/E.

The "SMP/E Installation" appendix in the *EMC Disk Library for mainframe User Guide* provides instructions to install using the DLm downloadable files designed for an SMP/E install.

If you need an SMP/E installation, follow the instructions in the *EMC Disk Library for mainframe User Guide* and skip steps 1 through 7 below. If you need to do a non-SMP/E installation, perform the following steps.

To use any of these utilities or the UIM:

1. Download the file DLMZOS-‹version number›.zip from the EMC support website:
   a. Go to http://support.emc.com.
   b. Select **Downloads**.
   c. Download DLm-3.x-and-4.x-Mainframe-Utilities-1.nn.zip.
   d. Unzip the DLMZOS-‹version number›.zip file.

2. Transfer the extracted DLMZOS-‹version number›.XMI file to the mainframe as follows:

   a. `ftp target_system_name`

      (Satisfy login requirements of the mainframe.)

   b. `quote site recfm=fb lrecl=80`

      `bin`

      `put DLMZOS-<version number>.xmi`

      `quit`

      The file will be placed on the host as 'uid.DLMZOS-‹version number›.xmi', where uid is the login user id used for the FTP. Alternatively, you may use put DLMZOS-‹version number›.xmi 'filename' to force a specific filename of your choice.

3. After transferring the file, use ISPF function 6 (Command Shell) and type:

   `receive indataset('uid.dlmzos.xmi')`

4. Type `da('DLMZOS.PGMS')` at the following prompt:

   `Enter restore parameter or delete or end.`

   uid.DLMZOS.PGMS is created for the following members:

   - CBDEC255—The unit data table for the UIM
   - CBDUC255—The UIM for the EMC DLm devices
   - DLMLIB—The utility required to add volumes to a DLm MTL
   - DLMSCR—The DLm scratch utility
   - DLMCMD—The DLm command utility
   - DLMVER—The DLm utility version reporting utility
   - GENSTATS—The report formatting utility
   - DLMHOST—The DLm utility that provides a command interface to VTEs and a mechanism to list selected VTE log messages.
   - DLMWTOR—A DLm utility that allows for placement of a z/OS Operator prompt jobstep to allow for operator verification of a job to proceed.

5. Download the file DLMZOS.JCL-‹version›.zip from the EMC support website (https://support.emc.com).

a. Go to http://support.emc.com.

b. Click **Downloads**.

c. In the **Find a Product** field, enter the name of the DLm model and click **»**.

This lists all the downloads for that product.

6. Unzip the DLMZOS.JCL-‹version›.zip file. It contains the following files:

   - emc.dlms‹version›.rexxcmd.xmi: XMIT file containing the REXX programs used by DLm Command Processors.
   - emc.dlms‹version›.samplibu.xmi: XMIT file containing samplib JCL members for DLm utilities.
   - emc.dlms‹version›.samplibc.xmi: XMIT file containing samplib JCL and PROC members used by DLm Command Processors.

7. Upload the above XMIT files to the mainframe.

   ```
   ftp target_system_name
   ```

   (Satisfy login requirements of the mainframe.)

   ```
   quote site recfm=fb lrecl=80
   ```

   ```
   bin
   ```

   ```
   put emc.dlms<version>.rexxcmd.xmi
   ```

   (The file will be placed on the host as'uid. dlms440.rexxcmd.xmi ', where uid is the login userid used for the FTP.)

   ```
   put emc.dlms<version>.samplibc.xmi
   ```

   (The file will be placed on the host as'uid. dlms440.samplibc.xmi ', where uid is the login userid used for the FTP.)

   ```
   put emc.dlms<version>.samplibu.xmi
   ```

   (The file will be placed on the host as'uid. dlms440.samplibu.xmi ', where uid is the login userid used for the FTP.)

   ```
   quit
   ```

   The following is a sample Unload of REXX and SAMPLIB upload:

   ```
   C:\temp140520>ftp 10.242.29.45
   Connected to 10.242.29.45.
   ```

```
220-FTPD1 IBM FTP CS V1R12 at H15, 10:06:44 on
2014-05-20.

220 Connection will close if idle for more than 5
minutes.

User (10.242.29.45:(none)): rgreen7

331 Send password please.

Password:

230 RGREEN7 is logged on. Working directory is
"RGREEN7.".

ftp> quote site recfm=fb lrecl=80

200-BLOCKSIZE must be a multiple of LRECL for RECFM FB

200-BLOCKSIZE being set to 6160

200 SITE command was accepted

ftp> bin

200 Representation type is Image

ftp> put emc.dlms440.rexxcmd.xmi

200 Port request OK.

125 Storing data set RGREEN7.EMC.DLMS440.REXXCMD.XMI

250 Transfer completed successfully.

ftp: 283840 bytes sent in 0.34Seconds
829.94Kbytes/sec.

ftp> put emc.dlms440.samplibc.xmi

200 Port request OK.

125 Storing data set RGREEN7.EMC.DLMS440.SAMPLIBC.XMI

250 Transfer completed successfully.

ftp: 100240 bytes sent in 0.43Seconds
233.66Kbytes/sec.

ftp> put emc.dlms440.samplibu.xmi

200 Port request OK.

125 Storing data set RGREEN7.EMC.DLMS440.SAMPLIBU.XMI

250 Transfer completed successfully.

ftp: 15920 bytes sent in 0.09Seconds 173.04Kbytes/sec.

ftp> quit

221 Quit command received. Goodbye.
```

```
C:\temp140520>
```

8. After transferring the files, use ISPF function 6 (Command Shell) and receive the uid.EMC.DLMS440.REXXCMD.XMI, uid.EMC.DLMS440.SAMPLIBC.XMI, and uid.EMC.DLMS440.SAMPLIBU.XMI datasets:

   a. Type the following to receive uid.EMC.DLMS440.REXXCMD.XMI:

   ```
   receive indataset('uid.EMC.DLMS440.REXXCMD.XMI')
   ```

   b. At the prompt `Enter restore parameters or delete or end`, type:

   ```
   da('uid.EMC.DLMS440.REXXCMD')
   ```

   EMC.DLMS440.REXXCMD will then be populated with the DLm REXX program library.

   c. Type the following to receive uid.EMC.DLMS440.SAMPLIBU.XMI.

   ```
   receive indataset('uid.EMC.DLMS440.SAMPLIBU.XMI')
   ```

   d. At the prompt `Enter restore parameters or delete or end`, type:

   ```
   da('uid.EMC.DLMS440.SAMPLIBU')
   ```

   EMC.DLMS440.SAMPLIBU is populated with the DLm samplib library to run the various DLm Utilities. See member $INDEXU for a list of its contents.

   e. Type the following to receive EMC.DLMS440.SAMPLIBC.XMI.

   ```
   receive indataset('uid.EMC.DLMS440.SAMPLIBC.XMI')
   ```

   f. At the prompt `Enter restore parameters or delete or end`, type:

   ```
   da('uid.EMC.DLMS440.SAMPLIBC')
   ```

   EMC.DLMS440.SAMPLIBC is populated with the DLm samplib library that contains PROCs and sample JCL to run DLm Command Processors. See member $INDEXC for a list of its contents.

9. Move the DLm PROCs to a common Proclib and customize them for your installation.

   Changes must be made for the following PROCs:

   - DLMCMDPR

- DLMCMDP1
- DLMCMD49
- GENSTATP
- GENSTAT1
- GENSTATW
- GENSTATR

BTILIB: Dataset name for the DLm utilities referenced above as DLMZOS.PGMS.

REXXLIB: Dataset name for the DLm REXX library referenced above as

EMC.DLMS440.REXXCMD.

There are other optional changes to the GENSTATP, GENSTAT1, and GENSTATW Procs. By default, these create a temporary dataset on DASD for the STATSFILE which contains all .vstats records obtained from the DLm. These Procs use a default SPACE allocation of SPACE=(CYL,(50,50)). This space allocation may be changed or the DD can be modified to use DLm storage such as the examples below.

Note: When storing the &STATFILE on DLm storage, you must specify an additional DLm unit in your override DD statement and it must be online. Below is an example using a non-MTL unit 4502:

- Changing the temporary STATFILE to use DLm storage:

```
//STEP4.PRINT  DD DSN=&STATFILE,DISP=(NEW,PASS),
//             UNIT=/4502,DCB=(LRECL=320,RECFM=FB,BLKSIZE=3200)
```

- Here is an example of overriding the DASD space allocation for the STATFILE DD, (Changing the primary and secondary space allocations to 100 cylinders each):

```
//STEP4.PRINT DD DSN=&STATFILE,
// DISP=(,PASS),SPACE=(CYL,(100,100)),
// UNIT=SYSDA,
// DCB=(LRECL=320,RECFM=FB,BLKSIZE=3200)
```

2

DLm Utilities for z/OS Environments

This chapter introduces the DLm utilities for z/OS environments.

Topics include:

DLm Utilities for z/OS Environments

This section describes each of the DLm utilities for z/OS environments:

- GENSTATS
- DLMSCR
- DLMCMD
- DLMLIB
- DLMVER
- DLMHOST
- DLMWTOR

GENSTATS utility

The GENSTATS utility generates reports on the tape mount and unmount statistics logged at the VTE level and at the VOLSER range level. It can selectively present:

- Daily and hourly throughput numbers
- Mount rates
- Concurrent tape drive usage details
- Compression ratio
- Average and slow mount response information

GENSTATS uses command processors CP998 and CP999 to summarize virtual tape activity. A GENSTATS job consists of two steps:

1. Execute a command processor which accesses the appropriate statistics file and writes the data to a statistics file.
2. Run GENSTATS to generate a report from the statistics file data.

"GENSTATS" on page 116 contains more information about GENSTATS.

GENSTATS reports

A **GENSTATS** report is generated by executing **CP998** or **CP999** to retrieve the desired statistics, and then running the **GENSTATS** utility to produce the report. "GENSTATS reports" on page 23 provides more information about these reports.

Reports that can be generated from the retrieved data are:

- **THROUGHPUT**

  Note: This is the default report.

  The THROUGHPUT report is comprised of two parts:

  - First part gives the daily/hourly throughput data for quickly identifying peak periods per day of the reporting period. Throughput numbers are displayed in GB/second.
  - Second part gives a throughput summary for the range of days covered.

- **MAXTAPES**

  Produces a report detailing the highest number of concurrent tapes in use per hour for each of the days processed. If no tapes are active during a particular hour, that cell shows '***'.

- **SLOWMOUNTS=count**

  Shows the average mount times for the entire day range specified. It lists the slowest 'count' mount times along with the related VOLSER, VTEname, Pathname, Date, and Time. Use this report to identify and investigate any peak or slow periods.

- **STILLINUSE**

  Lists any tape that was mounted but has no matching unmount record.

  When used with the SYNCTIME parameter, this report can also be used for DR Testing by identifying volumes that were mounted at a particular point in time.

- Detail

  Lists all DLm volsers with detail compression statistics.

Note: The examples below use a Unit, (F207), that is part of an MTL. This is the reason for the SMSHONOR clause in the unit specification.

GENSTATS Tapelib Report Examples

Example 1: Slow Mounts Report

Sample JCL:

```
//GO1      EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),                X
//              REPORT='SLOWMOUNTS=5'
```

Sample Output

```
GENSTATS VER 1.28  Z1CWRK PARMS: SLOWMOUNTS=5

--------------------------------------------------------------------------------
PATHS INCLUDED IN STATS  :

NODENAME   PATH
vte1dlm0   tapelibFLR/CEL1_FLRP_FS1
vte1dlm0   tapelibFLR/CEL1_FLRP_FS2
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
TOP 50   SLOWEST MOUNTS (SECS.):

NODENAME   TIME     VOLSER  DATE        TIME        PATH
vte1dlm0   0000.026 BG0332  2015/08/31  11:02:15  tapelibFLR/CEL1_FLRP_FS1
vte1dlm0   0000.028 BG0337  2015/09/01  11:14:09  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0   0000.029 BG0323  2015/08/28  16:59:43  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0   0000.033 BG0325  2015/08/28  17:08:15  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0   0000.041 BFLGR1  2015/08/26  16:19:44  tapelibFLR/CEL1_FLRP_FS2
```

Example 2: Throughput Report

Sample JCL:

```
//GO2      EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),                X
//              REPORT='THROUGHPUT'
```

Sample Output

Note: In the following report, some lines do not fit in the margins and are continued for up to 4 lines in the sample output below.

```
GENSTATS VER 1.28  Z1CWRK PARMS: THROUGHPUT

--------------------------------------------------------------------------------------------------------------
PATHS INCLUDED IN STATS  :

NODENAME   PATH
vte1dlm0   tapelibFLR/CEL1_FLRP_FS1
```

```
vte1dlm0   tapelibFLR/CEL1_FLRP_FS2
----------------------------------------------------------------------------------------------------


----------------------------------------------------------------------------------------------------
THROUGHPUT IN GB/HR      :
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|YY/MM/DD|OUT  00|OUT  01|OUT  02|OUT  03|OUT  04|OUT  05|OUT  06|OUT  07|OUT  08|OUT  09|OUT  10|OUT  11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|       |OUT  12|OUT  13|OUT  14|OUT  15|OUT  16|OUT  17|OUT  18|OUT  19|OUT  20|OUT  21|OUT  22|OUT  23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|        |IN   00|IN   01|IN   02|IN   03|IN   04|IN   05|IN   06|IN   07|IN   08|IN   09|IN   10|IN   11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|        |IN   12|IN   13|IN   14|IN   15|IN   16|IN   17|IN   18|IN   19|IN   20|IN   21|IN   22|IN   23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|15/08/26|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-----
|    |0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000011
MAXMNT/HR=000011
….

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/09/02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.62|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000079
MAXMNT/HR=000054
```

Example 3: Detail Report

Sample JCL:

```
//GO4      EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),                X
//              REPORT='DETAIL'
```

Sample Output

```
GENSTATS VER 1.28  Z1CWKB PARMS: DETAIL

       ------ Unload ----- ------ Channel ------ ---- File System ----   Duration Compression
Volser Date        Time      4K Writes   4K Reads  4K Writes   4K Reads       Time     Percent
------ ---------- -------- ---------- ---------- ---------- ---------- ----------  ----------
BG0300 2015/08/26 16:11:06 0000005121 0000000001 0000000303 0000000000 1068058565       94.1%
BFLGR1 2015/08/26 16:19:44 0000000001 0000000000 0000000001 0000000000 1068058576       00.0%
BFLGR1 2015/08/26 16:19:56 0000000000 0000000000 0000000001 0000000000 1068058576
BG0301 2015/08/26 16:19:56 0000000001 0000000001 0000000001 0000000000 1068058564       00.0%
….
BG0348 2015/09/02 11:52:15 0000005121 0000000001 0000000303 0000000000 1067614753       94.1%
BG0349 2015/09/02 11:53:00 0000010241 0000000001 0000000605 0000000000 1067614708       94.1%
BG0353 2015/09/02 11:53:01 0000005121 0000000001 0000000303 0000000000 1067614755       94.1%
BG0350 2015/09/02 11:53:36 0000015361 0000000001 0000000907 0000000000 1067614672       94.1%
BG0351 2015/09/02 11:54:14 0000020481 0000000001 0000001208 0000000000 1067614634       94.2%
BG0354 2015/09/02 11:54:15 0000010241 0000000001 0000000605 0000000000 1067614726       94.1%
BG0352 2015/09/02 11:54:41 0000025601 0000000001 0000001510 0000000000 1067614607       94.2%
BG0355 2015/09/02 11:55:10 0000015361 0000000001 0000000907 0000000000 1067614707       94.1%
BG0356 2015/09/02 11:55:46 0000020481 0000000001 0000001208 0000000000 1067614709       94.2%
BG0357 2015/09/02 11:56:05 0000025601 0000000001 0000001510 0000000000 1067614716       94.2%
```

```
-----------------------------------------------------------------------------------------------------------------------
PATHS INCLUDED IN STATS  :

NODENAME   PATH
vte1dlm0   tapelibFLR/CEL1_FLRP_FS1
vte1dlm0   tapelibFLR/CEL1_FLRP_FS2
-----------------------------------------------------------------------------------------------------------------------
```

GENSTATS VTE Report Examples

Example 1: Slow Mounts Report

Sample JCL:

```
//GO5      EXEC GENSTAT1,CMD=999,UNIT=(/F207,,,SMSHONOR),              X
//             REPORT='SLOWMOUNTS=5'
```

Sample Output

```
GENSTATS VER 1.28  Z1CWK4 PARMS: SLOWMOUNTS=5


NODENAME = vte1
-----------------------------------------------------------------------------------------------------------------------
TOP 50   SLOWEST MOUNTS (SECS.):

NODENAME   TIME    VOLSER  DATE       TIME
vte1     0000.063 DV0011  2015/07/28  11:19:16
vte1     0000.064 BFL503  2015/06/23  18:07:12
vte1     0000.065 BFL502  2015/08/04  15:54:00
vte1     0000.080 BFL501  2015/06/23  18:09:47
vte1     0000.085 BFLMT7  2015/08/25  15:17:19
-----------------------------------------------------------------------------------------------------------------------
```

Example 2: Throughput Report

Sample JCL:

```
//GO6      EXEC GENSTAT1,CMD=999,UNIT=(/F207,,,SMSHONOR),              X
//             REPORT='THROUGHPUT'
```

Sample Output

```
GENSTATS VER 1.28  Z1CWKA PARMS: THROUGHPUT


NODENAME = vte1
-----------------------------------------------------------------------------------------------------------------------
THROUGHPUT IN GB/HR     :
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|YY/MM/DD|OUT 00|OUT 01|OUT 02|OUT 03|OUT 04|OUT 05|OUT 06|OUT 07|OUT 08|OUT 09|OUT 10|OUT 11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|      |OUT 12|OUT 13|OUT 14|OUT 15|OUT 16|OUT 17|OUT 18|OUT 19|OUT 20|OUT 21|OUT 22|OUT 23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|      |IN  00|IN  01|IN  02|IN  03|IN  04|IN  05|IN  06|IN  07|IN  08|IN  09|IN  10|IN  11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|      |IN  12|IN  13|IN  14|IN  15|IN  16|IN  17|IN  18|IN  19|IN  20|IN  21|IN  22|IN  23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
```

```
|15/06/23|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/09/02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.62|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000091
MAXMNT/HR=000054

COUNT OF UNLOAD RECORDS       :0000966
COUNT OF UNLOADS W/OUTPUT STATS:0000515
COUNT OF UNLOADS W/INPUT STATS :0000005
MAXIMUM OUTPUT/HR             :000000.62G
MAXIMUM OUTPUT/DAY            :000000.62G
MAXIMUM INPUT/HR              :000000.02G
MAXIMUM INPUT/DAY             :000000.02G
MAXIMUM THROUGHPUT/HR         :000000.62G
MAXIMUM THROUGHPUT/DAY        :000000.62G
MAXIMUM MOUNTS/HR             :000096
MAXIMUM MOUNTS/DAY            :000213
EFFECTIVE COMPRESSION RATIO    :  16.1 TO 1
AVERAGE MOUNT RESPONSE (SECS.) :000.003
-------------------------------------------------------------------------------------------------------------
```

Table 4 **GENSTATS Throughput daily report fields**

Field	**Description**
YY/MM/DD	Date of report, shown in the corresponding cell below the header. In this example, August 26, 2015.
OUT *xx*	The output bucket, shown in the corresponding cell below the header. This is the total uncompressed GB written by the mainframe, during the specified hour. Hours 1–24 are represented by xx values 00–23.
IN *xx*	The input bucket, shown in the corresponding cell below the header. This is the total uncompressed GB read by the mainframe, during the specified hour. Hours 1–24 are represented by xx values 00–23.
MAXI+O/HR	Maximum uncompressed GB read and written by the mainframe in any one hour bucket. In this example, the total I/O of 2 GB was observed in hour 14 (2.3 GB OUT plus 0.0 GB IN).
TOT/DAY	The total uncompresssed GB read and written by the mainframe in the 24-hour period.
OUT/DY	The total uncompressed GB written by the mainframe in the 24-hour period.

Field	Description
IN/DY	The total uncompressed GB read by the mainframe in the 24-hour period.
MAXO/HR	Maximum uncompressed GB written by the mainframe in any one hour bucket. In this example, 2 GB was observed in the hour 14 bucket.
MAXI/HR	Maximum uncompressed GB read by the mainframe in any one hour bucket. In this example, 0 GB was observed in hour 14.
MOUNTS/DY	Total virtual tape mounts observed in the 24-hour period.
MAXMNT/HR	The maximum number of virtual tape mounts observed in any one hour period.

Table 5 **GENSTATS Throughput summary report fields**

Field	Description
COUNT OF UNLOAD RECORDS	Total virtual tape unloads during the range of days.
COUNT OF UNLOADS W/OUTPUT STATS	Total unloads during the range of days of virtual tapes that had been written to by the mainframe.
COUNT OF UNLOADS W/INPUT STATS	Total unloads during the range of days of virtual tapes that had been read by the mainframe.
MAXIMUM OUTPUT/HR	Maximum uncompressed GB written by the mainframe in any one hour bucket during the range of days.
MAXIMUM OUTPUT /DAY	Maximum uncompressed GB written by the mainframe in any one day during the range of days.
MAXIMUM INPUT/HR	Maximum uncompressed GB read by the mainframe in any one hour bucket during the range of days.
MAXIMUM INPUT/DAY	Maximum uncompressed GB read by the mainframe in any one day during the range of days.
MAXIMUM THROUGHPUT/HR	Maximum uncompressed GB read and written by the mainframe in any one hour bucket during the range of days.

Field	Description
MAXIMUM THROUGHPUT/DAY	Maximum uncompresssed GB read and written by the mainframe in any one day during the range of days.
MAXIMUM MOUNTS/HR	Maximum virtual tape mounts in any one hour bucket during the range of days.
MAXIMUM MOUNTS/DAY	Maximum virtual tape mounts in any one day during the range of days.
EFFECTIVE COMPRESSION RATIO	The bytes written by the mainframe divided by the bytes written to back-end file systems over the range of days. This is the compression ratio achieved by the one or more VTEs. If the back-end file system resides on Data Domain, this ratio will be 1:1 (no compression will be performed by the VTE).

Example 3: Detail Report

Sample JCL:

```
//GO8      EXEC GENSTAT1,CMD=999,UNIT=(/F207,,,SMSHONOR),                X
//              REPORT='DETAIL'
```

Sample Output

```
GENSTATS VER 1.28  Z1CWRK PARMS: DETAIL

        ------ Unload ----- ------ Channel ------ ---- File System ----   Duration Compression
Volser Date       Time      4K Writes   4K Reads  4K Writes   4K Reads       Time     Percent
 ------ ---------- -------- ---------- ---------- ---------- ---------- ----------  ----------
BFL503 2015/06/23 18:07:12 0000000001 0000000000 0000000001 0000000000 1071310800       00.0%
DV0001 2015/06/23 18:07:16 0000000001 0000000001 0000000001 0000000000 1071310796       00.0%
BFL503 2015/06/23 18:08:22 0000000001 0000000000 0000000001 0000000000 1071310800       00.0%
DV0002 2015/06/23 18:08:27 0000000001 0000000001 0000000001 0000000000 1071310795       00.0%
BFL501 2015/06/23 18:09:48 0000000001 0000000000 0000000001 0000000000 1071310799       00.0%
DV0005 2015/06/23 18:09:59 0000000001 0000000001 0000000001 0000000000 1071310789       00.0%
BFL501 2015/06/23 18:10:00 0000000000 0000000000 0000000005 0000000000 1071310799
DV0004 2015/06/23 18:11:33 0000005121 0000005121 0000000300 0000000000 1071310798       94.2%
BFL499 2015/07/28 11:08:38 0000000001 0000000000 0000000001 0000000000 1069485135       00.0%
BFL499 2015/07/28 11:08:51 0000000000 0000000000 0000000001 0000000000 1069485136
DV0003 2015/07/28 11:08:51 0000000001 0000000001 0000000001 0000000000 1069485123       00.0%
BFL499 2015/07/28 11:08:52 0000000000 0000000000 0000000001 0000000000 1069485136
FL499 2015/07/28 11:09:53 0000000001 0000000000 0000000001 0000000000 1069485135       00.0%
DV0008 2015/07/28 11:10:05 0000000001 0000000001 0000000001 0000000000 1069485124       00.0%
BFL499 2015/07/28 11:10:06 0000000001 0000000000 0000000001 0000000000 1069485136       00.0%
BFL499 2015/07/28 11:10:19 0000000000 0000000000 0000000001 0000000000 1069485136
DV0007 2015/07/28 11:10:19 0000000001 0000000001 0000000001 0000000000 1069485124       00.0%
….
BG0348 2015/09/02 11:52:15 0000005121 0000000001 0000000303 0000000000 1067614753       94.1%
BG0349 2015/09/02 11:53:00 0000010241 0000000001 0000000605 0000000000 1067614708       94.1%
BG0353 2015/09/02 11:53:01 0000005121 0000000001 0000000303 0000000000 1067614755       94.1%
BG0350 2015/09/02 11:53:36 0000015361 0000000001 0000000907 0000000000 1067614672       94.1%
BG0351 2015/09/02 11:54:14 0000020481 0000000001 0000001208 0000000000 1067614634       94.2%
BG0354 2015/09/02 11:54:15 0000010241 0000000001 0000000605 0000000000 1067614726       94.1%
BG0352 2015/09/02 11:54:41 0000025601 0000000001 0000001510 0000000000 1067614607       94.2%
BG0355 2015/09/02 11:55:10 0000015361 0000000001 0000000907 0000000000 1067614707       94.1%
BG0356 2015/09/02 11:55:46 0000020481 0000000001 0000001208 0000000000 1067614709       94.2%
BG0357 2015/09/02 11:56:05 0000025601 0000000001 0000001510 0000000000 1067614716       94.2%
BFL503 2015/09/02 11:58:32 0000000001 0000000000 0000000001 0000000000 1067614800       00.0%
BFL503 2015/09/02 11:58:44 0000000000 0000000000 0000000015 0000000000 1067614800
```

GENSTATS Report Filters

Any of the following optional parameters can be combined to filter the data in the report:

- **DATEEND=yy/mm/dd**

  Excludes statistical information for records dated later than the date specified.

- **DATESTART=yy/mm/dd**

  Excludes statistical information for records preceding the date specified.

- **DAYCNT=xxxx**

  By default, **GENSTATS** processes up to 128 days of data. This parameter can be used to increase the number of days in the report. Available in **GENSTATS** Versions 1.11 and higher.

- **DEVCNT=xxxx**

  By default, **GENSTATS** processes up to 2000 devices. This parameter can be used to increase the number of devices to be processed. Available in **GENSTATS** Versions 1.11 and higher.

- **PATHCNT=xxxx**

  By default, **GENSTATS** processes up to 512 paths. This parameter can be used to increase the number of paths to be processed. Available in **GENSTATS** Versions 1.11 and higher.

- **VOLCNT=xxxx**

  By default, **GENSTATS** processes up to 2200 VOLSERs. This parameter refers to a table containing VOLSERs that are considered active which are mounted and never unloaded. Available in **GENSTATS** Versions 1.11 and higher.

  IMPORTANT notes about VOLCNT=xxxx:

  The VOLCNT parameter determines the size of an in-memory table of VOLSERs in the GENSTATS execution. The default value for VOLCNT is 2200. When the number of VOLSER table entries needed in a

GENSTATS execution exceeds the VOLSER table, GENSTATS fails with the following message:

```
GEN450I VOLSER TABLE OVERFLOW
```

The above GEN450I message will include a count of the number of times this error was encountered. Adding this count with the VOLCNT used will provide a new recommended VOLCNT parameter value.

To avoid the above failure, it may be necessary to increase the VOLCNT parameter. Here are some ways to reduce GENSTATS processing:

- Limit the amount of data that GENSTATS processes in one execution. Parameters for the starting and ending dates can be specified for both the collection of performance data on the DLm, (Command Processor 998 and 999 processing), as well as runtime parameters to the GENSTATS execution.

  In addition, parameters can be specified to filter what performance data, (.vstats files), is collected. These filter-related parameters, (VPATH, VMASK, and VDEPTH), are specified in the GENSTATP, GENSTAT1, and GENSTATW PROCs. "GENSTAT1 procedure" on page 162, "GENSTATW/R procedure" on page 166, and "GENSTATP procedure" on page 155 provide more information.

- Another alternative is to sort the first 20 bytes of the input file into GENSTATS which will sort the input to GENSTATS in date-time sequence. For GENSTATS processing of EFS File Systems where EFSMOVE is set to RANDOM, this can significantly reduce CPU usage.

◆ **NODENAME=vtename**

Filters the input data to select a particular VTE. This parameter is used for **CP998** and ignored for **CP999**. An example is **NODENAME=VTE1**.

◆ **PATHNAME=name**

Filters the input data for a particular pathname or group of pathnames. This parameter is used for **CP998** and ignored for **CP999**.

The characters specified are matched against the pathnames of the input records. Only the characters specified need to match for **GENSTATS** to accept a record.

For example, if you specify **PATHNAME=tapelib/G**, and the VTE paths include tapelib/G0, tapelib/G1, and so on, all of the tapelib/G*x* records are processed.

- **SYNCTIME=hh:mm:ss**

  This parameter can be used with the STILLINUSE report to specify (along with **DATEEND**) the point at which the STILLINUSE check will be applied. In addition, tapes mounted after **SYNCTIME** will be listed for user evaluation purposes. Available in **GENSTATS** Versions 1.09 and higher.

- **VBELOW**

  Allocate VOLCNT table in 24-bit addressable storage. Default is to allocate VOLCNT table in 31-bit addressable storage.

DLMSCR Utility

DLm provides the DLMSCR utility that you can use with any of the major tape management systems to keep your TMS scratch status synchronized with the DLm scratch status.

You must link the DLMSCR utility as an authorized program into an authorized library under the name DLMSCR. EMC recommends that you use security software, such as Resource Access Control Facility (RACF), to restrict the use of DLMSCR to authorized users only.

EMC provides an example of the JCL required to link DLMSCR. The sample JCL file is found in the LNKSCR member of EMC.DLMS440.SAMPLIBU.

DLMSCR runs on the mainframe and sends volume scratch requests to DLm. As the TMS may dynamically release tapes back to scratch status, you must regularly run DLMSCR to maintain synchronization between the TMS catalog and DLm. To use DLMSCR, you must run a TMS scratch report and input that scratch report into DLMSCR. DLMSCR scans the scratch report for the DLm-owned volumes and sends the appropriate scratch requests to DLm.

Note: You can run only one DLMSCR at a time on a DLm. You must not run more than one DLMSCR simultaneously, whether on the same VTE or on different VTEs within the DLm.

EMC provides an example of the JCL required to run DLMSCR. The sample JCL is found in the RUNSCRA and RUNSCRB members of EMC.DLMS400.SAMPLIBU. RUNSCRB illustrates the use of the DEV= parameter. "Downloading and using DLm z/OS components" on page 14 provides download instructions.

DLMSCR parameters

The following table lists the DLMSCR parameters that may need to be specified.

Table 6 **Parameters in DLMSCR (page 1 of 3)**

Parameters	Specification
TYPE=x	Where x is used to select the tape management system. Valid types include RMM, TLMS, TMS, TSM, ZARA, CTLM, AFM, or CTLT. This is the only required parameter.
PREFIX=y	Where y is a string of prefix characters that limits processing to volumes whose VOLSER begins with the characters specified. Unless otherwise specified by the PREFIXLN parameter, the default prefix length is 2.
PREFIX=AAABAC	causes DLMSCR to process only volumes whose serial numbers begin with AA, AB, or AC. Coding this parameter prevents DLMSCR from trying to unnecessarily scratch volumes that are not stored on DLm. If no PREFIX is specified, DLMSCR processes the entire scratch list.
PREFIXLN=n	Where n can be a single digit between 1 and 5. This value replaces the default prefix length of 2 for the PREFIX= parameter. PARM='PREFIX=ABCD,PREFIXLN=1' causes DLMSCR to process only volumes whose serial numbers begin with A, B, C, or D.
NODSNCHK	DLm normally validates dataset names (dsname) if found in the scratch report as part of the scratch process. A scratch is not successfully completed if the dsname in the scratch report does not match the dsname in the HDR1 label on the volume being scratched. NODSNCHK prevents the data set name check from being performed and is not recommended for normal use.

Table 6 **Parameters in DLMSCR (page 2 of 3)**

Parameters	Specification
FREESPACE	By default, DLMSCR reclassifies volumes being scratched as eligible for scratch allocation requests, without freeing the space occupied by that volume. The FREESPACE parameter may be used to request that the space be freed. FREESPACE requires the volumes to already be in scratch status. Therefore to immediately free the space, DLMSCR must be run twice. The first execution must run without the FREESPACE parameter to scratch the volumes, and the second execution must run with the FREESPACE parameter to release the space. Keep in mind that DLm automatically frees the space of scratched volumes when it needs space. So, it is generally not necessary to run DLMSCR with the FREESPACE parameter.
FREEAFTERSCR	While the FREESPACE parameter requires that a volume already be in a scratched state, FREEAFTERSCR frees space from a volume immediately after DLMSCR has scratched it. Once FREEAFTERSCR frees the space associated with the execution of DLMSCR, the volume cannot be recovered if it was scratched by mistake.
NODATECHK	DLm normally checks the creation date of a tape volume and does not allow any volume to be created and scratched in the same 24-hour period. Setting this parameter allows volumes to be created and scratched on the same day. This parameter ignores the default date check in DLMSCR.
IGNLCSERR	This parameter ignores any errors reported by Library Call Subsystem (LCS) used by OAM with the MTL volumes. Normally, DLMSCR logs any error returned by LCS and stops processing scratch tapes when these errors occur. If this parameter is set, DLMSCR scratch processing continues even when the LCS errors are encountered.
ALLVOLS	This parameter allows scratch of volumes with report dsnames of hex zeros or blanks.
IGNLCSRC4	This allows DLMSCR processing to continue after receiving a return code of 4 from LCS processing, but terminates if the return code from LCS processing is greater than 4.
NOTCDB	This prevents DLMSCR from attempting any TCDB updates. This should be used only if the TMS already performs this function.

Table 6 **Parameters in DLMSCR (page 3 of 3)**

Parameters	Specification
NOTIFSCR	This prevents DLMSCR from attempting to change the TCDB use attribute to scratch if DLm reports that the VOLSER was already scratched.
TEST	This parameter allows for testing — no actual changes will be performed.
DEV=xxxx	This allows the specification of an offline virtual tape device and the elimination of the DLMCTRL DD statement. The MTL considerations for VTE drive selection in the *EMC DIsk Library for mainframe User Guide* contains an example.
USETMC	For CA-1 TMS environments only. This parameter enables DLMSCR to directly read the CA-1 Tape Management Catalog (TMC) to find DLm-resident VOLSERs which have been scratched and send the appropriate scratch requests to DLm for processing. Use of the USETMC option requires that the DLMSCR DD JCL statement point directly at the TMC or a copy of the TMC.
SYNC	For CA-1 TMS environments only. This parameter is valid only if specified along with the USETMC parameter. It enables DLMSCR to synchronize the status of the VOLSERs in the Tape Control Data Base (TCDB) and the DLm library with those in the CA-1 TMC.
TERMIFRS	This terminates DLMSCR processing if DLMSCR receives a warning return code, (4), from CBRXLCS and the reason code exceeds 5.

Scratch utility output files

The two scratch utility output files are:

- The DLMLOG file maintains a history of the results of each delete request. The file should have a logical record length (LRECL) of 133.

  If an error occurs during a scratch request (such as scratching a non-existent volume), the failure is recorded in the log file. The program continues with the next scratch request and result in a non-zero return code from the program execution.

- The DLMCTRL file allocates a DLm tape device for use as a control path to pass the scratch requests. If multiple tape libraries in the DLm are being used to contain the DLm virtual volumes, you must select a tape device address associated with the library in the DLm containing the volumes to be scratched. The device selected must be associated with a library and associated file systems that are READ/WRITE and not READ ONLY.

  The **DEV=xxxx** parameter allows an offline tape device to be used instead of coding the DLMCTRL DD statement. For example, see RUNSCRB in the sample JCL library.

DLMSCR report output messages

Note: ALL messages are preceded by MM/DD/YYYY HH:MM:SS VOLUME XXXXXX. The hex codes listed are the error codes that the VTE returns to DLMSCR when DLMSCR requests an action on a volume.

Table 7 **DLMSCR report output messages**

Hex Code	Volume Rejected Message	Comments
0x01	REQUEST REJECT - INVALID LOAD / DSPLY	Invalid data length (must be 17, 23, or 40 bytes).
0x02	REQUEST REJECT -ALREADY A SCRATCH	Volume already scratched.
0x05	REQUEST REJECT - INVALID VOLSER	Invalid VOLSER specified — The input volume serial number does not conform to the standard volume naming convention. Check the input TMS report.
0x06	REQUEST REJECT - VOLUME IN USE	Volume in use on the same or different VTE or there is a possibility of a STALE LOCK.
0x07	REQUEST REJECT - VOLUME NOT FOUND	Volume not found in file system. Make sure input tape device number points to correct tape library (i.e., /tapelibxxx).
0x08	REQUEST REJECT - I/O ERROR	I/O error has occurred during scratching process — refer to btilog for additional information.

Table 7 **DLMSCR report output messages**

Hex Code	Volume Rejected Message	Comments
0x09	REQUEST REJECT - VOLSER LOCKED	File is locked — volume might be in use by another tape drive on this VTE or another VTE or there is a possibility of a STALE LOCK.
0x0A	REQUEST REJECT - DIRECTORY PROBLEM A	Tape library directory is not accessible — Verify tape unit selected for utility points to correct tape library.
0x0B	REQUEST REJECT - DIRECTORY PROBLEM B	Tape library directory is not writeable — Verify that tape library is marked for Read/Write.
0x0C	REQUEST REJECT - INVAL/MISSING VOL1	Invalid or missing VOL1 label in volume — Using AWSPRINT verify volume VOL1 record has not been overwritten.
0x0D	REQUEST REJECT - VOLSER MISMATCH	Volume serial number from the TMS report and the VOL1 HDR do not match; VOL1 HDR might have been overwritten.
0x0E	REQUEST REJECT - INVAL/MISSING HDR1	Invalid or missing HDR1 label in volume — The HDR1 record is not in the correct format; over-layed because of error; use AWSPRINT utility to determine error.
0x0F	REQUEST REJECT - MISMATCHING DSNAME	Invalid or missing HDR1 label in volume — The HDR1 record is not in the correct format; over-layed because of error; use AWSPRINT utility to determine error.
0x10	REQUEST REJECT - INVALID DATE PASSED	TMS input report date does not match the execution date of DLMSCR.
0x11	REQUEST REJECT - CREATE DATE=TODAY	Date mismatch — DLMSCR, unless overridden using NODATECHK, defaults to no scratch same day.
0x12	REQUEST REJECT - FILE NOT WRITEABLE	File not writable — The file system directory is probably marked as Read Only. This might be target site.

Working with the DLm scratch utility

Note these considerations when working with the DLm scratch utility:

- The DLMSCR file must point to the scratch report that you have created using the appropriate TMS utility. Generate the scratch report with a logical record length (LRECL) of 133.
- To avoid any confusion, use a single job to generate a current scratch list file and run the DLMSCR utility against that file. This eliminates the possibility of accidentally running the DLMSCR program against an old scratch report and causing the TMS and DLm to be out of sync.
- DLm does not scratch a volume created on the current day unless NODATECHK is specified.

  Also, the utility does not run against a scratch report that was not created the same day.
- The Scratch utility uses the dsname information from the scratch report to verify the volumes being scratched. If the dsname written in the volume header does not match the dsname on the scratch report for that volume, the scratch request is rejected. This action can be overridden by NODSNCHK.
- After completing the DLMSCR utility, you can use or reuse tapes that the utility successfully scratched.

RMM considerations

Observe the following rules when using DLm with RMM:

- Predefine the DLm scratch volumes to RMM. If you have not predefined DLm VOLSERs as scratch in RMM, RMM rejects the new volumes, which results in an unsatisfied mount request on the mainframe. To resolve the unsatisfied mount, define the DLm volsers as scratch in RMM, and execute a LOAD command at the appropriate VT console to satisfy a stalled request.
- When defining a new DLm scratch tape to RMM, set the initialize option to no. If you select yes and RMM detects that the volume must be initialized (or EDGINERS is run), RMM sends a request to mount a 'blank' VOLSER on a DLm device. DLm is not automatically

ready as it cannot recognize which volume to mount. Consequently, you must use the LOAD command at the VT console to manually mount each volume being initialized.

- DLMSCR processes two types of RMM scratch reports:
  - The scratch report that EDGRPTD creates.
  - The scratch report that EDGJRPT creates using the EDGRRPTE exec (EDGRPT01).

    Use the DATEFORM(I) parameter when running EDGRPTD to create scratch reports to ensure the expected date format is used. When the REXX exec form is used,DLMSCR may not accept a user-tailored version of EDGRRPTE.

TMS considerations

DLMSCR expects Report-05, Report-06, or Report-87 to be used.

TLMS considerations

DLMSCR expects either the TLMS003 or the TLMS043 report as input.

TSM considerations

DLMSCR expects a Tivoli Storage Manager (TSM) Volume History Report to be used as input to the DLMSCR DD.

ZARA considerations

DLMSCR expects the LIST SCRATCH type of scratch report to be used as input from ZARA.

CA-1 considerations

Although there are various reports supported by TMS (CA-1), DLMSCR expects Report-05 or Report-06 or Report-87 to be used.

The report generation parameters should request the field DSN17 instead of the default DSN. (See PRIMUS EMC204223.) Otherwise, the report for multi-volume multi-file tapes will have the incorrect DSN for all but the first VOLSER. Those volumes with incorrect DSN will fail the DSN validity check performed by DLMSCR before scratching a tape.

Unique to CA-1 TMS environments only, DLMSCR supports the following 2 additional run time parameters:

- USETMC — When this parameter is specified, DLMSCR directly reads the CA-1 Tape Management Catalog (TMC) to find DLm-resident VOLSERs which have been scratched. A separate execution of the CA-1 scratch report utility (EARL) is not required. Use of the USETMC option requires that the DLMSCR DD JCL statement point directly at the TMC (or a copy of the TMC). DLMSCR scans the TMC and sends the appropriate scratch requests to the DLm for processing.

  Note: When using the USETMC option, DLMSCR sends a scratch request for any scratch volume (those that pass prefix filtering) it finds in the TMC. This might result in a large number of DLm500I messages followed by DLm524W messages being issued to the DLm VTE btilog whenever DLMSCR is run. This is normal. The DLm500I message indicates that the VTE application has received a request to scratch a VOLSER. The DLm524W message indicates that the VOLSER was already scratched.

- SYNC — This parameter is valid only if specified along with USETMC. The SYNC option causes DLMSCR to synchronize the Tape Control Data Base (TCDB) and the DLm library with the CA-1 TMC. The status of the VOLSERs in the TCDB and in the DLm library will be changed from active to scratch or from scratch to active as required to match the status of the CA-1 Tape Management Catalog (TMC).

  Note: When using the SYNC option DLMSCR sends an unscratch request for any active volume it finds in the TMC. This may result in a large number of DLm500I messages followed by DLm524W messages being issued to the VTE btilog whenever DLMSCR is run. This is normal. The DLm500I message indicates that the VTE application has received a request to unscratch a VOLSER. The DLm524W message indicates the VOLSER was already unscratched.

  EMC recommends that you use the SYNC option only periodically to clean up the occasional mismatch. Using SYNC frequently can impact performance.

  The below table, lists the differences between using SYNC and a normal scratch report.

Table 8 **Differences between SYNC and normal scratch report**

Using SYNC	Not using SYNC
Every VOLSER that passes filtering results in either a scratch or an unscratch request to the DLm. (About 200,000 requests as stated below)	Only the VOLSERS in the scratch list result in a call to the DLm. (# of calls = # of tapes in scratch status)
Every VOLSER that passes filtering, and the control device is in an MTL, results in a call to OAM's LCS process to verify the status unless NOTCDB is specified. (About 200,000 calls)	Only VOLSERS that successfully scratch, and the control device is in an MTL, result in a call to OAM's LCS process, unless NOTCDB is specified. (# of calls = # of tapes successfully scratched)

TMS users who use Scratch Pool Management and need to limit the eligible scratch volumes to a limited range of VOLSERs must install the TMS usermod CL05219 (CTSMSGEX exit). When this exit is linked into IGX00030, an IPL with CLPA is required to activate it. The exit causes the first 8 characters of the scratch poolname to be placed into the Load_Display mount message that is sent to the tape drive. This poolname can be defined as a scratch synonym so that the VTE application software can restrict the eligible scratch volumes to a specific prefix group.

DLMCMD utility

The DLMCMD utility allows you to execute DLm commands from the mainframe. You must link this utility as an authorized program to an authorized library under the name DLMCMD. EMC highly recommends that you use security software, such as RACF, to restrict the use of DLMCMD to authorized users only.

DLm Command Processors make use of DLMCMD using PROCs such as DLMCMDPR, DLMCMDP1, DLMCMD49, GENSTAT1, GENSTATW and GENSTATP. EMC recommends use of the above Command Processor PROCs to execute the various Command Processor reports and Commands.

EMC provides an example of the JCL required to run DLMCMD. The sample JCL is found in the RUNCMDA and RUNCMDB members of EMC.DLMS440.SAMPLIBU. RUNCMDB illustrates the use of the DEV= parameter. "Downloading and using DLm z/OS components" on page 13 provides download instructions.

How the DLm command utility works:

- The DLMCMD utility reads one or more DLm commands from the DLMCMD input file, and sends each command to DLm for execution.

  Note: The DLMCMD utility accepts input cards up to 256 characters in length. Continuation lines are not allowed.

  Indication of success or failure is logged to the file to which the DLMLOG DD statement points.

  Note: Any messages and other textual results of the command that display on the DLm Console are not returned to the host.

  DLMCMD does not respond to a mainframe command on the communication tape device until the VTE processing for that command is complete.

- Use the DLMCTRL file to allocate a DLm device for use as a control path for passing the DLm commands. You can use any available DLm virtual tape device as the DLMCTRL device. The "MTL considerations for VTE drive selection" section in the *EMC Disk Library for mainframe User Guide* provides information about working with a Manual Tape Library. The DEV=xxxx parameter allows an offline tape device to be used instead of coding the DLMCTRL DD statement. See RUNCMDB in the sample JCL library for an example.

- The DLMCMD DD statement should point to a list of DLm commands to be sent. The LRECL of DLMCMD cannot exceed 256. If possible, create it using the NONUM ISPF edit option to avoid sequence numbers at the end of the command line. This can optionally be an in-stream input file (DLMCMD DD *) of commands.
- The DLMLOG DD statement points to a sequential file for logging the DLMCMD results. This file should have a logical record length (LRECL) of 133. If an error occurs during command processing, the failure is recorded in the log file, and a non-zero return code from DLMCMD results. "DLMCMD error messages" on page 43 lists the possible error codes from DLMCMD.

  This is a sample DLMLOG output:

```
DLMCMD VER 1.0 DLMCTRL = EA82
2004/09/10 12:47:49 CMD ERR=FF: this is an invalid command
2004/09/10 12:47:49 CMD ISSUED: q all
2004/09/10 12:47:49 CMD ERR=FC: q xxxxxxxx
2004/09/10 12:47:49 CMD ISSUED: set size=2g dev=EA80
```

The two optional methods to pass commands to DLMCMD are:

- Use of PARM='WTOR' — Sends the message DLC070I, ENTER COMMAND, to the operator, who can reply with the command. The message is reissued after each command is accepted until END is entered as the reply. This method does not use the DLMCMD input file.

  For example:

```
//LOG EXEC PGM=DLMCMD,PARM='WTOR'
//DLMLOG DD DSN=DLM.LOGFILE,DISP=OLD
//DLMCTRL DD DSN=DLM.CTRL,UNIT=V3480,VOL=SER=BT9999, DISP=(,KEEP)
```

- Use of PARM='CMD='— Allows you to pass a single command on the EXEC card instead of using the DLMCMD input file. This method also allows you to call DLMCMD from another program, and pass the command as an entry parameter. For example:

```
//LOG EXEC PGM=DLMCMD,PARM='CMD=Q SPACE'
//DLMLOG DD DSN=DLM.LOGFILE,DISP=OLD
//DLMCTRL DD DSN=DLM.CTRL,UNIT=V3480,VOL=SER=BT9999, DISP=(,KEEP)
```

Note: If you experience issues on the VTE for the DLm device with the DLMCMD, check the /var/log/messages file for error messages.

- Use of PARM='TIMEOUT='— Allows you to specify the time in seconds, (nnn), that DLMCMD will wait for the command to be completed. If the command is not completed in the number of seconds specified, the DLMCMD job step will terminate with condition code of 12 and the following message will be written to the DLMLOG DD:

  DLM080I COMMAND TIMED OUT – TERMINATING

  For example:

```
//GO EXEC DLMCMD,TIMEOUT=600
```

In the above example, DLMCMD will terminate with a condition code of 12 if the job step has not completed in 600 seconds.

Note: You may also specify the maximum time that DLMCMD waits when using the RUNWAIT command. When both the TIMEOUT= parm is specified and the RUNWAIT command also specifies a time limit, then whichever time limit expires first will terminate the DLMCMD job step. If the DLMCMD job step is terminated as a result of the RUNWAIT time limit, then the DLMCMD job step will terminate with a condition code of 4 and in the DLMCMD DLMLOG DD output, you will see the following error message: CMD ERR=FB.

DLMCMD error messages

The below Error code table, lists the error message from the DLMCMD.

Table 9 **Error code from DLMCMD**

Error code	Description
0x01	Invalid data length (must be between 1 and 256 bytes).
0x02	DLm does not accept Host-initiated console commands. This error code is generated when the HOSTCOMMAND option is set to NO in the xmap file. To enable it, you must manually modify the xmap file.

Table 9 **Error code from DLMCMD**

Error code	Description
0xFF(-1)	A general syntax error occurred. (The DLm console error message "DLM891E: Invalid command syntax" was displayed.)
0xFC (-4)	An "E" level error other than general syntax error occurred. (A console error message other than DLM891E was displayed.)
0xFB (-5)	This value will be returned if the RUNWAIT command is executed using DLMCMD, and the command does not finish in the wait time specified.

DLMVER utility

The DLMVER utility reports the versions of:

- The DLm mainframe modules on the mainframe:
  - DLMCMD
  - DLMLIB
  - DLMSCR
  - DLMVER
  - GENSTATS
  - DLMWTOR

DLMVER Messages

The messages related to DLMVER are:

- DLV010I UTILITY VERSIONS ():
- DLV050I LOG FILE FAILED TO OPEN

DLMWTOR utility

DLMWTOR is a z/OS utility that provides a means to obtain a z/OS operator confirmation that a proposed action is to be performed. This utility provides a safe guard against unauthorized or mistaken job

executions. DLMWTOR is included with the DLMDRC, (DLm Disaster Recovery Client), feature of DLm. However, it can be used in conjunction with other DLm actions such as executing a DLm VTE command.

The following sample is provided with the SAMPLIBU samplib as member DLMWTOR2:

```
//JOBNAME JOB (MYJOB)
//*
//GO EXEC PGM=DLMWTOR
//STEPLIB DD DSN=user.LOADLIB,DISP=SHR
//SYSABEND DD SYSOUT=*
//WTOLINES DD *
wtor message lines- up to 10 lines
//*
```

To use the DLMWTOR utility, insert the above sample JCL into the job stream that you need to require an z/OS Operator confirmation. Modify the STEPLIB DD to point to the installed DLm LOADLIB. DLMWTOR does not have to be APF Authorized. Enter the operator prompt that you wish to specify. When invoked, DLMWTOR will present the message lines to the z/OS Console log. Up to 10 lines can be displayed.

The operator is instructed to reply YES or CANCEL. Replying YES will result in a condition code of 0 to be returned from the DLMWTOR job step. Replying CANCEL will result in a condition code of 8 to be returned.

Code subsequent job steps to execute only if a condition code of 0, (YES), is returned from the DLMWTOR job step.

Here is an example:

```
//GO EXEC PGM=DLMWTOR
//STEPLIB DD DSN=RGREEN7.DLM.LINKLIB,DISP=SHR
//SYSABEND DD SYSOUT=*
//WTOLINES DD *
DLMC800W DLMDR VG8HALT REQUESTED FOR'
IP: 99.999.99.99 PORT: 8888'
*-------------------------------------*
* ALL TAPE DRIVE ACTIVITY MUST BE *
* QUIECED PRIOR TO THIS ACTION. *
* *
* FAILURE TO QUIECE ALL TAPE ACTIVITY *
* CAN RESULT IN DATA LOSS AND/OR *
* UNPREDICTABLE RESULTS *
*-------------------------------------*
//*
//DLMHALT EXEC DLMDRC,IP='99.999.99.99',PORT=8888,
//  CMD=VG8HALT,PARMS='VG8-1,CHECK',
//  COND=(0,NE)
SDSF output example:
15.48.54 J0013777 IEF403I RGDLMDR2 - STARTED - TIME=15.48.54
15.48.54 J0013777 +DLMC800W DLMDR VG8HALT REQUESTED FOR' 703
703 IP: 99.999.99.99 PORT: 8888'
703 *-------------------------------------*
703 * ALL TAPE DRIVE ACTIVITY MUST BE *
703 * QUIECED PRIOR TO THIS ACTION. *
703 * *
703 * FAILURE TO QUIECE ALL TAPE ACTIVITY *
703 * CAN RESULT IN DATA LOSS AND/OR *
703 * UNPREDICTABLE RESULTS *
703 *-------------------------------------*
```

```
15.48.54 J0013777 @0026 DLMWI001 DO YOU WISH TO CONTINUE? REPLY YES OR CANCEL
15.50.46 J0013777 R 26,MAYBE
15.50.46 J0013777 @0027 DLMWE002 UNKNOWN ANSWER, REPLY YES OR CANCEL
15.50.55 J0013777 R 27,CANCEL
15.50.56 J0013777 RGDLMDR2 GO DLMWTOR 0008 2 0 0
15.50.56 J0013777 RGDLMDR2 GO IKJEFT01 FLUSH 0 0 0
15.50.56 J0013777 IEF404I RGDLMDR2 - ENDED - TIME=15.50.56
15.50.56 J0013777 RGDLMDR2 RC 0008 ET 00:02:01 IO 2 CP .0/ .0
15.50.56 J0013777 $HASP395 RGDLMDR2 ENDED
z/OS log output example:
$HASP373 RGDLMDR2 STARTED - INIT 6 - CLASS A - SYS H15
IEF403I RGDLMDR2 - STARTED - TIME=15.48.54
+DLMC800W DLMDR VG8HALT REQUESTED FOR' 703
IP: 99.999.99.99 PORT: 8888'
*-------------------------------------*
* ALL TAPE DRIVE ACTIVITY MUST BE *
* QUIECED PRIOR TO THIS ACTION. *
* *
* FAILURE TO QUIECE ALL TAPE ACTIVITY *
* CAN RESULT IN DATA LOSS AND/OR *
* UNPREDICTABLE RESULTS *
*-------------------------------------*
@0026 DLMWI001 DO YOU WISH TO CONTINUE? REPLY YES OR CANCEL
R 26,MAYBE
IEE600I REPLY TO 0026 IS;MAYBE
@0027 DLMWE002 UNKNOWN ANSWER, REPLY YES OR CANCEL
R 27,CANCEL
IEE600I REPLY TO 0027 IS;CANCEL
RGDLMDR2 GO DLMWTOR 0008 2 0 0
RGDLMDR2 GO IKJEFT01 FLUSH 0 0 0
IEF404I RGDLMDR2 - ENDED - TIME=15.50.56
```

In the above example, job RGDLMDR2 was submitted. The operator was instructed to reply YES or CANCEL to proceed. An incorrect Reply of MAYBE was entered which was not accepted. A reply of CANCEL was then entered and caused the job to terminate because of the condition code clause on the subsequent DLMHALT job step.

DLMHOST utility

DLMHOST is a host utility that provides z/OS Console Operation support. It is used to communicate between a single logical partition's (LPAR) operator console and the DLm.

To make use of the DLm z/OS Console operation, you must install the z/OS DLMHOST utility and then configure the individual VTEs to communicate with it. Using the DLm Console program, you can configure which type (informational, warning, or error) of messages and/or which specific DLm messages are sent over the channel to the mainframe. For details about configuring the messages, see EMC Disk Library for mainframe User Guide, chapter Sending messages to z/OS, Configuring messages and recipients.

The DLMHOST utility runs as a started task, and accepts commands from the operator. By default, DLMHOST uses Write-to-Operator-with-Response (WTOR) capabilities for sending DLm

commands. The requests to DLm are through WTOR, and the logging from DLm is through Write-to-Operator (WTO). Optionally, you may configure DLMHOST to use the z/OS MODIFY function in place of WTOR.

At startup, DLMHOST reads a configuration file that defines the VTEs to be supported as well as the device addresses, per VTE, to be used for communication and logging. Each DLm VTE will be identified with a unique name so that commands can be targeted to specific VTEs. A tape drive device address must be selected from each VTE's range of addresses that will be used as the command/communication path. A second device address is required on each VTE if you want DLm to send log messages to the z/OS console. These devices will not be eligible for allocation once DLMHOST has been started.

Only log messages that have passed message filtering will be received by the host.

It should be noted that, depending upon the filtering options set on the VTEs, there may be many log messages sent to the consoles. Optionally, DLMHOST supports a configuration option to send the messages to a host file instead of the operator's console.

DLMHOST installation

DLMHOST is only supported in a single Logical Partition (LPAR). You cannot connect multiple DLMHOST tasks running in multiple LPARs to the same DLm VTE.

DLMHOST is distributed in the 3.0 DLMZOS.XMI and the package is available on the EMC support website. “Downloading and using DLm z/OS components” on page 13 provides more details.

The DLMHOST utility must be linked as an authorized program into an authorized library under the name DLMHOST. EMC highly recommends that RACF be used to restrict the useof DLMHOST to authorized users only.

An example of the JCL required to link DLMHOST follows:

```
//L EXEC PGM=HEWL,PARM='MAP,LET,LIST,NCAL,AC=1'
//SYSLMOD DD DSN=USER.LIBRARY,DISP=SHR
//SYSUT1 DD DSN=&&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
// UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD SYSOUT=*
//DDIN DD DSN=DLM.MODULE,DISP=SHR
//SYSLIN DD *
INCLUDE DDIN(DLMHOST)
```

```
NAME DLMHOST(R)
/*
```

Running DLMHOST

Use the following JCL to execute DLMHOST:

```
//DLMSTEP EXEC PGM=DLMHOST,PARM='parameters'
//DLMCFG DD DSN=PARMLIB(nodecfg),DISP=SHR
//DLMLOG DD DSN=logfilename,DISP=SHR
//* THE FORMAT OF THE CONFIG FILE IS AS
//* FOLLOWS:
//* Col 1 -10 Nodename
//* Col 12-15 Command path device address
//* Col 17-20 Log path device address
//* Col 22-29 Console name
```

The parameters than can be specified are:

- DOCMDS — Requires the use of a DLMCMD DD card pointing to a file of commands that are to be processed during DLMHOST startup. The commands should be in the same format as used in modify or WTOR processing.

  Note: EOJ can be specified as the last command to terminate DLMHOST after a series of commands.

- NOLOG — Prevents DLMHOST from receiving continuous log data from any VTE. Set this parameter if you plan to use DLMHOST only to send commands from the z/OS Console to the DLm. Command responses will be returned even when NOLOG is specified.
- NOWTOR — Prevents DLMHOST from issuing the normally outstanding WTOR. When this parameter is specified, MDL commands can be issued using the z/OS MODIFY command as the method of communication in place of WTOR.
- LOGFILE — Causes any received log data from the DLm system to be recorded in the file pointed to by the DLMLOG DD card. When LOGFILE is specified, the log messages are not sent to any console via WTO. If LOGFILE is not specified, the DLMLOG DD card is not required in the JCL. The LOGFILE dataset should be an FB LRECL 133 file, and will be opened for extend each time the task is started.

- RESETLOG — Instructs DLMHOST to discard any DLm messages that had been issued prior to the startup of DLMHOST. The default behavior of DLMHOST is to issue any DLm messages that had been issued on the DLm prior to its startup. This parameter provides the ability to change this default behavior.
- MSGID1ST — By default, DLm messages begin with the date and time for the message. When this parameter is specified, the DLm messages will begin with a message id.
- MSGIDFMT — By default DLm message ids will end with a severity code of 'E', 'W', or 'I' for error, warning, and information messages, respectively. Specify this parameter when you wish to have the severity code occur in position 4 of the message id.

EMC provides sample JCL to run DLMHOST. Refer to the section on Downloading and using DLm z/OS components for information on how to download the sample JCL. This SAMPLIBU dataset includes a sample PROC member to run the DLMHOST utility, called DLMHOST.

This proc must be customized to point to the APF authorized load library that DLMHOST has been installed in. Also, one or more configuration statements must be completed for the DLMCFG DD. The customized proc should be placed into a PROCLIB that is searched when the START DLMHOST command is issued from the zOS Console.

DLMHOST configuration file

The configuration file pointed to by the DLMCFG DD card should be an FB LRECL 80 file that has a single record for each VTE to be supported. You can define up to 64 records.

Comments cards can be included in the input configuration file by placing an asterisk in column 1.

The layout of the configuration file records is as follows:

- Col. 1 - 10 : NODENAME — The name used by the mainframe operator to identify which with VTE to communicate.
- Col 12 - 15: CMDDEV — The 4-digit device address of the tape drive that is to be used for operator commands and responses. If this field is left blank, no operator commands can be sent to this nodename.

- Col 17 - 20: LOGDEV — The 4-digit device address of the tape drive that is to be used for logging activity whenever logging is active for this VTE. If left blank, no host logging can occur from the VTE.
- Col 22 - 29: CONSNAME — The console that log messages should be directed to if logging is active for this VTE. If this field is left blank, the log messages will go to all routcde=5 (tape library) consoles.

The following is sample JCL for DLMHOST DLMCFG within a 3-VTE configuration supporting both messaging and commands:

```
VTLNODE1 038E 038F TAPECON1
VTLNODE2 039E 039F TAPECON1
VTLNODE3 03AE 03AF TAPECON1
```

Using z/OS Console support

If DLMHOST is active and configured to receive DLm messages, it automatically forwards any message received to the appropriate Console or log file.

When DLMHOST is executed without the NOWTOR parameter, the following message is displayed on the z/OS Console:

```
DLM001I ENTER COMMAND, EOJ,OR ? FOR HELP
```

An outstanding Write to Operator Reply (WTOR) message will remain pending.

To send a command to DLMHOST, you need to know the message reply number from the WTOR. To determine the WTOR message number, enter the following z/OS command on the operator's console:

```
d r,l (or /d r,l from SDSF)
```

This command returns the reply message number for any outstanding WTORs on the system.

To issue a command to DLMHOST, enter the command using the WTOR message number in the following format:

```
msg#,COMMAND
```

where:

- **msg#** is the reply message number returned from the d r,l command.

- **COMMAND** is the DLMHOST command to be executed.

When DLMHOST is executed with the NOWTOR parameter, the following message is returned:

```
DLM002I jobname USE MODIFY TO ENTER COMMAND, EOJ, OR ? FOR HELP
```

Subsequently, you can use the z/OS Modify command to issue commands to DLMHOST using the 'jobname' indicated in the DLM002I message.

The format of the z/OS Modify command is:

```
F jobname,command
```

where:

- **jobname** is the job name of DLMHOST reported in the DLM002I message.
- **command** is the DLMHOST command to be executed.

DLMHOST commands

The following commands are recognized by DLMHOST:

- **CMD** — Sends a DLm command to a specific VTE. This command requires a nodename also be specified by using the NODE= parameter (or N=). A nodename of ALL can be specified to send the command to every VTE.

  All DLm Operator commands can be entered as parameters to this command.

  The following are examples of valid use of this command:

  ```
  CMD=Q SPACE,NODE=NODE1
  ```

  ```
  CMD=FIND VOL=000001,N=N1
  ```

- **STOPLOG** — Requests that DLMHOST stop logging VTE log messages for a specific VTE. This command requires that a nodename be specified by using the NODE= parameter (or N=). A nodename of ALL can be specified to stop host logging for all defined VTEs.

  For example:

  ```
  STOPLOG,N=ALL
  ```

```
STOPLOG,N=VTLNODE1
```

- **STARTLOG** — Requests that DLMHOST start host logging of VTE log data for a specific VTE. This command requires that a nodename be specified by using the NODE=parameter (or N=). A nodename of ALL can be specified to start logging for all defined VTEs.

  For example:

  ```
  STARTLOG,N=NODE2
  STARTLOG,NODE=ALL
  ```

- **STATUS** — Requests that DLMHOST display the current configuration and status of the command and logging functions. DLMHOST will issue this message followed by the status of each configured node:

  ```
  DLM2401 NODENAME CMDDEV LOGDEV CONSNAME
  ```

  A **y** or **n** next to the device address indicates whether the command/logging function is currently active or inactive for that node name, respectively.

  For example:

  ```
  DLM2401 NODENAME CMDDEV LOGDEV CONSNAME
  NODE1 038E Y 038F Y CON1
  NODE2 048E N 048F Y
  ```

- **EOJ** — Terminate the DLMHOST task.

- **HELP or ?** — Returns the DLM000I message with a list of the valid DLMHOST commands.

  The commands that DLM000I lists are:

  ```
  STARTLOG,N=nodename/ALL
  STOPLOG,N=nodename/ALL
  C=Command,N=nodename/ALL
  STATUS
  ```

WTOR command examples

When DLMHOST has been executed without the NOWTOR parameter, an outstanding WTOR message reply is used to send commands to DLMHOST. The following are valid examples of DLMHOST commands:

```
msg#,STATUS
msg#,C=Q SPACE,N=N1
msg#,STOPLOG,N=ALL
```

where msg# is the message number returned from the d r,l (/d r,l from SDSF).

The following are valid examples of the same DLMHOST commands when DLMHOST sent with a z/OS MODIFY command when DLMHOST has been executed with the NOWTOR parameter using the job name DLMHOST:

```
F DLMHOST,STATUS
F DLMHOST,C=Q SPACE,N=N1
F DLMHOST,STOPLOG,N=ALL
```

DLMLIB Utility

If your tape devices are defined in a Manual Tape Library (MTL), you must also define them in the mainframe's tape configuration database (TCDB). DLm provide the DLMLIB utility to do this. DLMLIB invokes IBM's LCS External Services (CBRXLCS) macro.

DLMLIB can be used to define either a single tape volume or a continuous range of tape volumes to an MTL.

EMC provides an example of the JCL required to run DLMLIB. The sample JCL file is found in the RUNLIB member of installed SAMPLIBU library. "Downloading and using DLm z/OS components" on page 13 provides download instructions and additional information.

When executing DLMLIB, input requests are specified in the INDD DD statement.

DLMLIB writes a log of its actions using the OUTDD DD statement. This log file lists the result of each cartridge entry request, including any error codes.

Note: Return codes and reason codes can be found in the chapter "OAM Diagnostic Aids, CBRXLCS Return and Reason Codes" of DFSMSdfp Diagnosis (GY27-7618).

DLMLIB command

Example 1:

Entering in a range of Scratch volumes to an MTL called DLM020. VOLSERs start at BG0500 and 90 are to be created.

Sample JCL:

```
//LIB      EXEC PGM=DLMLIB
//STEPLIB   DD  DISP=SHR,DSN=USER.DLM.LINKLIB
//OUTDD     DD  DISP=SHR,DSN=USER.DATALIB(D15083A)
//*             (RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUDUMP DD   SYSOUT=*
//INDD      DD  *
M  BG0500 DLM020   400090
```

Sample Output

```
DLMLIB  VER 4.03
2015/08/28  11:10:18    M  BG0500 DLM020   400090          VOLUME SUCCESSFULLY ENTERED
2015/08/28  11:10:18    M  BG0501 DLM020   4               VOLUME SUCCESSFULLY ENTERED
2015/08/28  11:10:18    M  BG0502 DLM020   4               VOLUME SUCCESSFULLY ENTERED
2015/08/28  11:10:18    M  BG0503 DLM020   4               VOLUME SUCCESSFULLY ENTERED
2015/08/28  11:10:18    M  BG0504 DLM020   4               VOLUME SUCCESSFULLY ENTERED
2015/08/28  11:10:18    M  BG0505 DLM020   4               VOLUME SUCCESSFULLY ENTERED
….
```

Example 2:

Changing the attribute of a VOLSER, (BG0500), to PRIVATE in an MTL called DLM020.

Sample JCL:

```
//LIB      EXEC PGM=DLMLIB
//STEPLIB   DD  DISP=SHR,DSN=USER.DLM.LINKLIB
//OUTDD     DD  DISP=SHR,DSN=USER.DATALIB(D15083B)
//*             (RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUDUMP DD   SYSOUT=*
//INDD      DD  *
P  BG0500 DLM020   4
```

Sample Output

```
DLMLIB  VER 4.03
2015/08/28  11:19:07    P  BG0500 DLM020   4           USE ATTRIBUTE CHANGED
```

Example 3:

Ejecting a VOLSER, (BG0502), from an MTL called DLM020:

Sample JCL:

```
//LIB      EXEC PGM=DLMLIB
//STEPLIB   DD  DISP=SHR,DSN=USER.DLM.LINKLIB
//OUTDD     DD  DISP=SHR,DSN=USER.DATALIB(D15083C)
//*             (RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUDUMP DD   SYSOUT=*
//INDD      DD  *
E  BG0502 DLM020   4
```

Sample Output

```
DLMLIB  VER 4.03
2015/08/28  11:23:18    E  BG0502 DLM020   4              EJECT SCHEDULED
```

3 Introduction to Command Processors

This chapter introduces the different Command Processors .

Topics include:

DLm Command Processors

This section describes the following DLm Command Processors:

- CP499: Provide DLm diagnosics.
- CP501: Execute VTE command.
- CP502: Obtain VTE log information.
- CP503: DLm Volume Inventory Report.
- CP504: Dlm VNX Replication Report.
- CP601: DLm Data Domain System Utilization report.
- CP602: DLm Data Domain Deduplication Report.
- CP603: DLm Data Domain Replication Report.
- CP998: Obtain DLm File System Activity Statistics.
- CP999: Obtain DLm VTE Activity Statistics.

DLm Command Processor support

Releases before DLm 1.2 do not support Command Processors. Table 10 on page 58 identifies the CP*xxx* commands and the DLm 1.x and 2.x releases that support them.

Table 10 **CP*xxx* commands supported by DLm 1.x and 2.x releases**

CP*xxx* command	DLm release				
	1.2	2.1	2.2	2.3, 2.4 without DD	2.3, 2.4 with DD
CP499	No	No	No	No	No
CP501	Yes	Yes	Yes	Yes	Yes
CP502	Yes	Yes	Yes	Yes	Yes
CP503	Yes	Yes	Yes	Yes	Yes
CP504	No	No	Yes	Yes	Yes
CP601	No	No	No	No	Yes

Table 10 **CP*xxx* commands supported by DLm 1.x and 2.x releases**

CP*xxx* command	DLm release				
CP602	No	No	No	No	Yes
CP603	No	No	No	No	Yes
CP998	No	Yes	Yes	Yes	Yes
CP999	No	Yes	Yes	Yes	Yes

Table 11 on page 59 identifies the CP*xxx* commands and the DLm 3.x and 4.x releases that support them.

Table 11 **CP*xxx* commands supported by DLm 3.x and 4.x releases**

CP*xxx* command	DLm release				
	3.0, 3.1, 3.2, 3.3, 3.5, 3.5.2 without DD	**3.0, 3.1, 3.2, 3.3, 3.5.0, 3.5.2 with DD**	**3.4, 3.5.1 without DD**	**4.x.x with DD**	**4.x.x without DD**
CP499	No	No	No	Yes	Yes
CP501	Yes	Yes	Yes	Yes	Yes
CP502	Yes	Yes	Yes	Yes	Yes
CP503	Yes	Yes	Yes	Yes	Yes
CP504	Yes	Yes	Yes	Yes	Yes
CP601	No	Yes	No	Yes	No
CP602	No	Yes	No	Yes	No
CP603	No	Yes	No	Yes	No
CP998	Yes	Yes	Yes	Yes	Yes
CP999	Yes	Yes	Yes	Yes	Yes

Note: Only a single execution of a particular Command Processor can run at a time. DLm does not support multiple executions of the same Command Processor simultaneously.

Command Processor 499 (CP499)

CP499 is a user controllable diagnostic aid. It enables the user to test the accessibility of the attached VTE components using a batch job. **CP499** sends output that includes a PASSED or FAILED condition and is able to customize a WTO to use with its automated operations. The user can list all the components attached to the VTE (via the /etc/hosts file) and ping a specific component through a specific IP address or component name.

Command Processor 501 (CP501)

CP501 enables the mainframe operator to pass any supported DLm Console command to a selected Virtual Tape Engine (VTE), and retrieves the output for use on the mainframe as desired. The *EMC Disk Library for mainframe User Guide* provides a complete list of DLm commands.

Multiple executions of **CP501** cannot run simultaneously. In DLm 3.0 and later, the **DLMHOST** utility resolves this issue and provides enhanced functionality. "DLMHOST utility" on page 48 provides information about **DLMHOST**.

Command Processor 502 (CP502)

CP502 enables the mainframe operator to obtain any or all of a set of useful log files from a selected VTE. The log files available are:

- btilog files
- /var/log/messages
- VTE xmap and ximl file
- Output of the command **ipmitool sel list**

Note: The log files used by Virtuent may vary from DLm releases.

Command Processor 503 (CP503)

CP503 produces a report of the DLm VOLSER inventory.

Command Processor 504 (CP504)

CP504 collects the replication statistics of all the DLm VNX file systems on which replication has been enabled. It includes all file systems on the base and expansion EMC VNX® systems. It generates three types of reports:

- QUICK
- SUMMARY
- VERBOSE

Command Processor 601 (CP601)

CP601 generates Data Domain system utilization statistics and makes the reports available to the mainframe.

Command Processor 602 (CP602)

CP602 generates a report that provides deduplication statistics for the entire Data Domain system. The statistics include:

- Size of incoming data
- Size of data stored on the disk
- Deduplication ratio

Command Processor 603 (CP603)

CP603 generates reports that provide the following Data Domain replication information:

- Replication configuration
- Replication status
- Replication statistics
- Replication performance statistics
- Replication historical data for the past 24 hours

Command Processor 998 (CP998)

CP998 captures statistics from back-end file systems that are specified by their VOLSER range and then invokes the **GENSTATS** utility to report DLm virtual tape activity. These reports present:

- Daily and hourly throughput numbers
- Mount rates
- Concurrent tape drive usage details
- Compression ratio
- Average and slow mount response information

Command Processor 999 (CP999)

CP999 captures statistics from a local VTE and then invokes the **GENSTATS** utility to report DLm virtual tape activity. These reports present the same statistics as **CP998** (listed above).

Creating custom Command Processors

The Command Processor infrastructure of DLm can be used by EMC Professional Services to implement customized Command Processor-like operations that can be initiated from the mainframe host. **CP001** to **CP498** are reserved for this purpose. Contact your EMC representative for further details.

4

Using Command Processors

This chapter introduces how Command Processors are invoked from the mainframe, the parameters that are provided, and sample JCL and reports.

Topics include:

Performing DLm Diagnostics: CP499

This job pings DLm components attached to this VTE. Use it as a batch component testing function.

Note: The parameters are case sensitive.

Submit **CP499** from the mainframe by using **DLMCMD49** PROC.

CP499 syntax

The **CP499** command syntax is:

```
//GO      EXEC  DLMCMD49,CMD=499,UNIT=UNITID,
//              CPARMS='CP499 Parameters'
```

Table 12 on page 64 describes the **CP499** parameters.

Table 12 **CP499 parameters**

Parameter	Description
-l all	Interrogates the /etc/host and lists its contents
-i xxx.xxx.xxx.xxx	Pings a specific address
-c ComponentName	Pings a component by name

Note: Default parameters are **-l all** for list all.

Note: CP499 processing is directed to a single DLm VTE. If you are using an esoteric for the DLm Unit, the esoteric should only include DLm Units for that VTE.

Example 1: The following example interrogates the /etc/host and lists its contents.

Sample JCL:

- The JCL below uses a DLm Unit, (F200), that is not in an MTL:

```
//GO      EXEC  DLMCMD49,CMD=499,UNIT=/F200,
//              CPARMS='-L'
```

- The JCL below a DLm Unit, (F207), that is in an MTL:

```
//GO      EXEC  DLMCMD49,CMD=499,UNIT=(/F207,,,SMSHONOR),
//              CPARMS='-L'
```

Sample output

```
********************************** TOP OF DATA **********************************
20150828-15:38:20-34918
EMC ADHOC Component List and Test Report (499) v4.4.0-r5798
Fri Aug 28 15:41:26 EDT 2015
================================================================================
499 Adhoc Verification tool
*Component List Report*                          ( options flags -l all)
  This Report Lists the DLm components that can be reached from the VTE that
  this CP499 utility is running on (Components that CP499 can validate)
--------------------------------------------------------------------------------
         Component                       IP Address
         ---------                       --------------
        dd1                              192.168.100.110
        dd1s-alias0                      192.168.203.10
        dd1s-alias1                      192.168.203.11
        dd1s-alias2                      192.168.203.12
        dd1s-alias3                      192.168.203.13
        dd2                              192.168.100.120
        dd2s-alias0                      192.168.204.10
        dd2s-alias1                      192.168.204.11
        dd2s-alias2                      192.168.204.12
        dd2s-alias3                      192.168.204.13
        localhost                        127.0.0.1
        switch10a                        192.168.100.240
        switch10b                        192.168.100.241
        switcha                          192.168.100.1
        switchb                          192.168.100.2
        vnx1                             192.168.100.205
        vnx1cs0                          192.168.100.200
        vnx1cs1                          192.168.100.201
        vnx1dm2-alias0                   192.168.201.20
        vnx1dm2-alias1                   192.168.201.21
        vnx1dm2-alias2                   192.168.201.22
        vnx1dm2-alias3                   192.168.201.23
        vnx1dm3-alias0                   192.168.201.30
        vnx1dm3-alias1                   192.168.201.31
        vnx1dm3-alias2                   192.168.201.32
        vnx1dm3-alias3                   192.168.201.33
        vnx1dm4-alias0                   192.168.201.40
        vnx1dm4-alias1                   192.168.201.41
        vnx1dm4-alias2                   192.168.201.42
        vnx1dm4-alias3                   192.168.201.43
        vnx1dm5-alias0                   192.168.201.50
        vnx1dm5-alias1                   192.168.201.51
        vnx1dm5-alias2                   192.168.201.52
        vnx1dm5-alias3                   192.168.201.53
        vnx1dm6-alias0                   192.168.201.60
        vnx1dm6-alias1                   192.168.201.61
        vnx1dm6-alias2                   192.168.201.62
        vnx1dm6-alias3                   192.168.201.63
        vnx1dm7-alias0                   192.168.201.70
        vnx1dm7-alias1                   192.168.201.71
        vnx1dm7-alias2                   192.168.201.72
        vnx1dm7-alias3                   192.168.201.73
        vnx1dm8-alias0                   192.168.201.80
        vnx1dm8-alias1                   192.168.201.81
        vnx1dm8-alias2                   192.168.201.82
        vnx1dm8-alias3                   192.168.201.83
        vnx1dm9-alias0                   192.168.201.90
        vnx1dm9-alias1                   192.168.201.91
        vnx1dm9-alias2                   192.168.201.92
        vnx1dm9-alias3                   192.168.201.93
        vnx2                             192.168.100.215
        vnx2cs0                          192.168.100.210
        vnx2cs1                          192.168.100.211
        vnx2dm2-alias0                   192.168.202.20
        vnx2dm2-alias1                   192.168.202.21
        vnx2dm2-alias2                   192.168.202.22
        vnx2dm2-alias3                   192.168.202.23
        vnx2dm3-alias0                   192.168.202.30
        vnx2dm3-alias1                   192.168.202.31
        vnx2dm3-alias2                   192.168.202.32
        vnx2dm3-alias3                   192.168.202.33
        vnx2dm4-alias0                   192.168.202.40
        vnx2dm4-alias1                   192.168.202.41
        vnx2dm4-alias2                   192.168.202.42
        vnx2dm4-alias3                   192.168.202.43
```

```
        vnx2dm5-alias0                  192.168.202.50
        vnx2dm5-alias1                  192.168.202.51
        vnx2dm5-alias2                  192.168.202.52
        vnx2dm5-alias3                  192.168.202.53
        vnx2dm6-alias0                  192.168.202.60
        vnx2dm6-alias1                  192.168.202.61
        vnx2dm6-alias2                  192.168.202.62
        vnx2dm6-alias3                  192.168.202.63
        vnx2dm7-alias0                  192.168.202.70
        vnx2dm7-alias1                  192.168.202.71
        vnx2dm7-alias2                  192.168.202.72
        vnx2dm7-alias3                  192.168.202.73
        vnx2dm8-alias0                  192.168.202.80
        vnx2dm8-alias1                  192.168.202.81
        vnx2dm8-alias2                  192.168.202.82
        vnx2dm8-alias3                  192.168.202.83
        vnx2dm9-alias0                  192.168.202.90
        vnx2dm9-alias1                  192.168.202.91
        vnx2dm9-alias2                  192.168.202.92
        vnx2dm9-alias3                  192.168.202.93
        vte0                            192.168.100.253
        vte0-rmm                        192.168.100.254
        vte1                            192.168.100.10
        vte1-rmm                        192.168.100.11
        vte2                            192.168.100.20
        vte2-rmm                        192.168.100.21
        vte3                            192.168.100.30
        vte3-rmm                        192.168.100.31
        vte4                            192.168.100.40
        vte4-rmm                        192.168.100.41
        vte5                            192.168.100.50
        vte5-rmm                        192.168.100.51
        vte6                            192.168.100.60
        vte6-rmm                        192.168.100.61
        vte7                            192.168.100.70
        vte7-rmm                        192.168.100.71
        vte8                            192.168.100.80
        vte8-rmm                        192.168.100.81
        vtea                            192.168.100.8
PASSED
```

Example 2: The following example Pings a specific address.

Sample JCL:

The JCL below a DLm Unit, (F207), that is in an MTL:

```
//GO      EXEC  DLMCMD49,CMD=499,UNIT=(/F207,,,SMSHONOR),
//              CPARMS='-I 192.168.100.20'
```

Sample output

```
EMC ADHOC Component List and Test Report (499) v4.4.0-r5798
Fri Aug 28 15:43:00 EDT 2015
================================================================================
Pinged 192.168.100.20 3 times successfully - Component is => vte2
PASSED
--------------------------------------------------------------------------------
================================================================================
```

Executing VTE Commands: CP501

Using **CP501**, you can pass a DLm Virtual Tape Operator console command to a VTE, and use the output on the mainframe as appropriate. The Virtual Tape Operator commands enable you to control or query virtual tape drives.

Following Command Processors are supported by **CP501**:

- **CLOSE VSTATS PATH**
- **DETAIL MOUNT**
- **EXPORT**
- **FIND**
- **HELP**
- **IMPORT**
- **INITIALIZE**
- **LOAD**
- **QUERY**
- **QUIESCE**
- **READY**
- **RESET CHANNEL ADAPTER**

Note: The **RESET CHANNEL ADAPTER** command is not supported for Gen 4 DLms.

- **REWIND**
- **ROTATE**
- **SAVE TRACE**
- **SET**
- **SHOW**
- **SNMP**
- **STOP CHANNEL ADAPTER**

Note: The **STOP CHANNEL ADAPTER** command is not supported for Gen 4 DLms.

- **UNLOAD**
- **UNQUIESCE**

- **UNREADY**

Note: *EMC Disk Library for mainframe User Guide* provides information about the Virtual Tape Operator commands.

Note: In DLm 3.2, the **DLMHOST** utility provides an enhanced version of the **CP501** functionality. The "DLMHOST utility" on page 48 provides information about **DLMHOST**.

Submit **CP501** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP501 syntax

The **CP501** command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=501,UNIT=unitid1,
//              VCMD='VTECommand'
//GO       EXEC DLMCMDPR,CMD=501,UNIT=unitid1,UNIT2=unit2,
//              VCMD='VTECommand'
```

CP501 parameters

Table 13 on page 68 defines the **CP501** parameters.

Table 13 **CP501 parameters**

Parameter	Description
DLm subcommand	Any DLm Console command listed above. See the Virtual Tape Operator command reference Appendix in the *EMC Disk Library for mainframe User Guide*.
CPARMS	VTE Command parameters.

DETAIL MOUNT Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1    EXEC  DLMCMDP1,CMD=501,UNIT=(/F207,,,SMSHONOR),              X
//              VCMD='DETAIL MOUNT'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 13:28:45 EDT 2015
================================================================================
08/18/2015 13:28:45 vte1 VT: DLm0409I: Command received: 'DETAIL MOUNT'
08/18/2015 13:28:45 vte1 VT: Devicename VOLSER Load Time          Duration Requestor Path
08/18/2015 13:28:45 vte1 VT:                  Last I/O Time              Tape Library Path
08/18/2015 13:28:45 vte1 VT: ---------- ------ ------------------- -------- ----------------
08/18/2015 13:28:45 vte1 VT: F200      BG0059 2015/08/18 13:28:44   1.0 s ¢intfc 2/#1101500:link 110:lpar 15!
08/18/2015 13:28:45 vte1 VT:                  2015/08/18 13:28:44         /tapelibFLR/CEL1_P1_FS7/
08/18/2015 13:28:45 vte1 VT: F201      BG0051 2015/08/18 11:21:26   2.1 h ¢Manual Mount!
08/18/2015 13:28:45 vte1 VT:                  -                          /tapelibFLR/CEL1_P1_FS1/
================================================================================
```

INITIALIZE Command

Example 1: Example using DLm Unit that belongs to an MTL:

Sample JCL:

```
//GO1    EXEC  DLMCMDP1,CMD=501,UNIT=(/F207,,,SMSHONOR),              X
//             VCMD='INITIALIZE',                                     X
//             CPARMS='VOL=BG3001 DEV=F200 COUNT=2 CLASS=20'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 10:54:04 EDT 2015
================================================================================
08/18/2015 10:54:04 vte1 VT: DLm0409I: Command received: 'INITIALIZE VOL=BG3001 DEV=F200 COUNT=2 CLASS=20'
08/18/2015 10:54:04 vte1 VT: DLm0869I: 1 Standard-label volume initialized in /tapelibFLR/CEL1_P1_FS3
08/18/2015 10:54:04 vte1 VT: DLm0869I: 1 Standard-label volume initialized in /tapelibFLR/CEL1_P1_FS13
================================================================================
```

HELP Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1    EXEC  DLMCMDP1,CMD=501,UNIT=(/F207,,,SMSHONOR),              X
//             VCMD='INITIALIZE',                                     X
//             CPARMS='VOL=BG3001 DEV=F200 COUNT=2 CLASS=20'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 10:58:21 EDT 2015
================================================================================
08/18/2015 10:58:22 vte1 VT: DLm0409I: Command received: 'HELP'
08/18/2015 10:58:22 vte1 VT:  Help Summary of Commands (in alphabetic order):
08/18/2015 10:58:22 vte1 VT:  CLOSE VSTATS PATH=xxx
08/18/2015 10:58:22 vte1 VT:  DETAIL MOUNT
08/18/2015 10:58:22 vte1 VT:  EXPORT from to COMPRESS!
08/18/2015 10:58:22 vte1 VT:  FIND VOL=volser ¢DEV=devicename!¢LOCAL!
08/18/2015 10:58:22 vte1 VT:  HELP ¢ command | message number | ABOUT !
08/18/2015 10:58:22 vte1 VT:  IMPORT from to ¢COMPRESS! ¢NORUN!
08/18/2015 10:58:22 vte1 VT:  INITIALIZE VOL=volser DEV=devicename ¢COUNT=count! ¢LABEL=S/A/N!
08/18/2015 10:58:22 vte1 VT:                    ¢SCRATCH=YES/NO! ¢ERRORS=nnnnn! ¢EPIC=NO/YES!
08/18/2015 10:58:22 vte1 VT:                     ¢DIR=subdirectory! ¢CLASS=storageclass!
08/18/2015 10:58:22 vte1 VT:  LOAD ¢VOL=!volser ¢ON! ¢DEV=!devicename|* ¢ UNLABELED | LABELED | NL | SL | AL !
¢PROTECTED!
08/18/2015 10:58:22 vte1 VT:  QUERY ¢¢DEV=!devicename¢+! | ALL! ¢MOUNTED!
08/18/2015 10:58:22 vte1 VT:    ¢CONFIG! | ¢CRITICAL! | ¢PATHS ¢ASSIGNED!! | ¢EFS! | ¢LABELS! ¢¢DEV=!devicename!
08/18/2015 10:58:22 vte1 VT:     ¢VERSION! | ¢SPACE ¢BYCLASS | BYDLM!! | ¢SCRATCHNAMES! | ¢COMPRESSION!
08/18/2015 10:58:22 vte1 VT:       ¢GR!
08/18/2015 10:58:22 vte1 VT:      ¢WARNING | RECOVER | RECOVERAMT!
08/18/2015 10:58:22 vte1 VT:      ¢STATISTICS | STATS!
08/18/2015 10:58:22 vte1 VT:       ¢WORM!
08/18/2015 10:58:22 vte1 VT:  QUIESCE ¢DEV=!devicename | ALL | *
08/18/2015 10:58:22 vte1 VT:  QUIESCE SCRATCH PATH=path
08/18/2015 10:58:22 vte1 VT:  QUIESCE TAPELIB PATH=path
08/18/2015 10:58:22 vte1 VT:  QUIESCE IMMEDIATE TAPELIB PATH=path
08/18/2015 10:58:22 vte1 VT:  READY ¢DEV=!devicename
08/18/2015 10:58:22 vte1 VT:  REWIND ¢DEV=!devicename
08/18/2015 10:58:22 vte1 VT:  RMLOCK volser
08/18/2015 10:58:22 vte1 VT:  ROTATE ¢ LOCAL | DRLOG !
08/18/2015 10:58:22 vte1 VT:  SAVE ¢TRACE!
08/18/2015 10:58:22 vte1 VT:  SCRATCHNAME ADD synonym¢=(prefix,prefix,...,CLASS=CLASSn,CLASS=CLASSn,...)!
08/18/2015 10:58:22 vte1 VT:  SCRATCHNAME DELETE synonym
08/18/2015 10:58:22 vte1 VT:  SET ¢TRACE=n! ¢SIZE=xxx! ¢PATH=pathname ¢NOCHECK!! ¢VOL=prefix!
08/18/2015 10:58:22 vte1 VT:      ¢WARNING=nn! ¢RECOVER=nn! ¢RECOVERAMT=nn!
08/18/2015 10:58:22 vte1 VT:     ¢HWCOMP=ON/OFF/FORCE! ¢HWDECOMP=ON/OFF/FORCE! ¢IDRC=ON/OFF/FORCE!
08/18/2015 10:58:22 vte1 VT:      ¢DEV=devicename|ALL!
08/18/2015 10:58:22 vte1 VT:      ¢TIMESTAMPS=NO/YES!
08/18/2015 10:58:22 vte1 VT:      ¢QSPACETIMEOUT=nn!
08/18/2015 10:58:22 vte1 VT:      ¢ERASEPOLICYTTL=nnnD/nnnH!
08/18/2015 10:58:22 vte1 VT:      ¢CRITICALPERCENT=nn}
08/18/2015 10:58:22 vte1 VT:      ¢CRITICALSCRATCH=YES/NO!
08/18/2015 10:58:22 vte1 VT:      ¢CRITICALEFSMOVE=YES/NO!
08/18/2015 10:58:22 vte1 VT:      ¢CRITICALRO=YES/NO!
08/18/2015 10:58:22 vte1 VT:      ¢CRITICALLEOT=YES/NO!
08/18/2015 10:58:22 vte1 VT:      ¢NONTRUNCATEWRITE=YES/NO!
08/18/2015 10:58:22 vte1 VT:      ¢RDC256K=ON/OFF!
08/18/2015 10:58:22 vte1 VT:      ¢RESCRATCHIFREAD=NO/YES!
08/18/2015 10:58:22 vte1 VT:      ¢PROTECTVOL1=NO/YES!
08/18/2015 10:58:22 vte1 VT:      ¢WRITEVERIFY=ON/OFF/CRC/DATA/HEADERS!
08/18/2015 10:58:22 vte1 VT:      ¢DDWRITEVERIFY=ON/OFF/CRC/DATA/HEADERS!
08/18/2015 10:58:22 vte1 VT:      ¢TESTCOMPRESSEDDATA=ON|OFF|CRYPT!
08/18/2015 10:58:22 vte1 VT:      ¢AMDD=ON|OFF!
08/18/2015 10:58:22 vte1 VT:      ¢FSALLOCATION=RR|SPACE!
08/18/2015 10:58:22 vte1 VT:  SHOW CHANNEL ADAPTERS
08/18/2015 10:58:22 vte1 VT:  SHOW DRIVE LIST
08/18/2015 10:58:22 vte1 VT:  SHOW REJECTED PATHS
08/18/2015 10:58:22 vte1 VT:  SNMP SHOW STATUS
08/18/2015 10:58:22 vte1 VT:  SNMP  ACKNOWLEDGE FAILURE
08/18/2015 10:58:22 vte1 VT:  STATISTICS SHOW
08/18/2015 10:58:22 vte1 VT:  STATISTICS INTERVAL=MINUTES/SECONDS
08/18/2015 10:58:22 vte1 VT:  UNLOAD ¢DEV=!devicename
08/18/2015 10:58:22 vte1 VT:  UNQUIESCE ¢DEV=!devicename | ALL | *
08/18/2015 10:58:22 vte1 VT:  UNQUIESCE SCRATCH PATH=path
08/18/2015 10:58:22 vte1 VT:  UNQUIESCE TAPELIB PATH=path
08/18/2015 10:58:22 vte1 VT:  UNREADY ¢DEV=!devicename
================================================================================
```

QUERY CONFIG Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//               VCMD='QUERY CONFIG'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 10:58:35 EDT 2015
================================================================================
08/18/2015 10:58:35 vte1 VT: DLm0409I: Command received: 'QUERY CONFIG'
08/18/2015 10:58:35 vte1 VT: DLm0102I: Configuration file is /etc/bti/xmap0
08/18/2015 10:58:35 vte1 VT: Current Values Are:
08/18/2015 10:58:35 vte1 VT: Index Devicename Type CU UA  Options
08/18/2015 10:58:35 vte1 VT: -----------------------------------
08/18/2015 10:58:35 vte1 VT:   00  F200       3590 00 00  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   01  F201       3590 00 01  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   02  F202       3590 00 02  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   03  F203       3590 00 03  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   04  F204       3590 00 04  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   05  F205       3590 00 05  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   06  F206       3590 00 06  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   07  F207       3590 00 07  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   08  F208       3590 00 08  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   09  F209       3590 00 09  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0A  F20A       3590 00 0A  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0B  F20B       3590 00 0B  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0C  F20C       3590 00 0C  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0D  F20D       3590 00 0D  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0E  F20E       3590 00 0E  PATH=/tapelibFLR/ SIZE=40G
08/18/2015 10:58:35 vte1 VT:   0F  F20F       3590 00 0F  PATH=/tapelibFLR/ SIZE=40G
================================================================================
```

QUIESCE Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO3       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//               VCMD='QUIESCE',                                      X
//               CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 10:59:06 EDT 2015
```

```
================================================================================
08/18/2015 10:59:06 vte1 VT: DLm0409I: Command received: 'QUIESCE DEVICE=F201'
08/18/2015 10:59:06 vte1 VT: DLm0879I: Device F201 quiesced
================================================================================
```

UNQUIESCE Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//              VCMD='UNQUIESCE',                                     X
//              CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:09:38 EDT 2015
================================================================================
08/18/2015 11:09:38 vte1 VT: DLm0409I: Command received: 'UNQUIESCE DEVICE=F201'
08/18/2015 11:09:38 vte1 VT: DLm0879I: Device F201 unquiesced
================================================================================
```

UNREADY Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//              VCMD='UNREADY',                                       X
//              CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:19:30 EDT 2015
================================================================================
08/18/2015 11:19:30 vte1 VT: DLm0409I: Command received: 'UNREADY DEVICE=F201'
08/18/2015 11:19:30 vte1 VT: DLm0830I: Device F201 made Unready
================================================================================
```

REWIND Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO2       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
//               VCMD='REWIND',                                     X
//               CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:20:10 EDT 2015
================================================================================
08/18/2015 11:20:10 vte1 VT: DLm0409I: Command received: 'REWIND DEVICE=F201'
08/18/2015 11:20:10 vte1 VT: DLm0424I: Device F201 Volume BG0051 rewound
================================================================================
```

READY Command

Sample: JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO3       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
//               VCMD='READY',                                      X
//               CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:20:33 EDT 2015
================================================================================
08/18/2015 11:20:33 vte1 VT: DLm0409I: Command received: 'READY DEVICE=F201'
08/18/2015 11:20:33 vte1 VT: DLm0331E: Device F201 no requester path to send Ready status
08/18/2015 11:20:33 vte1 VT: DLm0830I: Device F201 made Ready
================================================================================
```

UNLOAD Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO5       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
```

```
//              VCMD='UNLOAD',                                          X
//              CPARMS='DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:21:13 EDT 2015
================================================================================
08/18/2015 11:21:13 vte1 VT: DLm0409I: Command received: 'UNLOAD DEVICE=F201'
08/18/2015 11:21:13 vte1 VT: DLm0490I: Device F201 Volume BG0051 unloaded
================================================================================
```

LOAD Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO4       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
//              VCMD='LOAD',                                        X
//              CPARMS='VOL=BG0051 ON DEVICE=F201'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:21:26 EDT 2015
================================================================================
08/18/2015 11:21:26 vte1 VT: DLm0409I: Command received: 'LOAD VOL=BG0051 ON DEVICE=F201'
08/18/2015 11:21:26 vte1 VT: DLm0440I: Device F201 Volume BG0051 (/tapelibFLR/CEL1_P1_FS1/BG0051) mounted
08/18/2015 11:21:26 vte1 VT: DLm0331E: Device F201 no requester path to send Ready status
================================================================================
```

ROTATE Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO1       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
//              VCMD='ROTATE',                                      X
//              CPARMS='DRLOG'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:30:26 EDT 2015
================================================================================
08/18/2015 11:30:26 vte1 VT: DLm0409I: Command received: 'ROTATE DRLOG'
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS15/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS15
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS3/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS3
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS8/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS8
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS4/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS4
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS5/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS5
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS13/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS13
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS19/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS19
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS14/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS14
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS16/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS16
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS21/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS21
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS7/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS7
08/18/2015 11:30:26 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_FLRP_FS1/.vstats.vte1dlm0.tapelibFLR.CEL1_FLRP_FS1
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS10/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS10
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS12/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS12
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS22/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS22
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_FLRP_FS2/.vstats.vte1dlm0.tapelibFLR.CEL1_FLRP_FS2
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS18/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS18
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS9/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS9
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS20/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS20
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS2/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS2
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS6/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS6
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS1/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS1
08/18/2015 11:30:27 vte1 VT: DLm0059I: Rotating statistics file
/tapelibFLR/CEL1_P1_FS17/.vstats.vte1dlm0.tapelibFLR.CEL1_P1_FS17
================================================================================
```

SET Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO2       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),          X
//              VCMD='SET',                                         X
//              CPARMS='HWDECOMP=OFF'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:30:54 EDT 2015
```

```
================================================================================
08/18/2015 11:30:54 vte1 VT: DLm0409I: Command received: 'SET HWDECOMP=OFF'
08/18/2015 11:30:54 vte1 VT: DLm0013I: Hardware decompression changed from ON to OFF
================================================================================
```

SET Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO3       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//               VCMD='SET',                                          X
//               CPARMS='HWDECOMP=ON'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:31:06 EDT 2015
================================================================================
08/18/2015 11:31:07 vte1 VT: DLm0409I: Command received: 'SET HWDECOMP=ON'
08/18/2015 11:31:07 vte1 VT: DLm0013I: Hardware decompression changed from OFF to ON
================================================================================
```

SHOW Command

Sample JCL:

Example using DLm Unit that belongs to an MTL:

```
//GO4       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//               VCMD='SHOW',                                         X
//               CPARMS='CHANNEL ADAPTERS'
```

Sample output

The sample report output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:31:45 EDT 2015
================================================================================
08/18/2015 11:31:45 vte1 VT: DLm0409I: Command received: 'SHOW CHANNEL ADAPTERS'
08/18/2015 11:31:45 vte1 VT: DLm0056I: Channel driver version is 5.4.2-2237
08/18/2015 11:31:45 vte1 VT: DLm0075I: Interface #1: 198401 (0x030701) bus:7 slot:1 port:1 type:18 (QLE2562)
media:3 (FiCon)
08/18/2015 11:31:45 vte1 VT: DLm0076I: Interface #1: hardware s/n: BFD1243F14784
08/18/2015 11:31:45 vte1 VT: DLm0070I: Interface #1: TRANSX emulation version set to 3
08/18/2015 11:31:45 vte1 VT: DLm0081I: Interface #1: Current state: running; Desired state: running; Media Down,
Loop Down
08/18/2015 11:31:45 vte1 VT: DLm0075I: Interface #2: 198402 (0x030702) bus:7 slot:2 port:2 type:18 (QLE2562)
media:3 (FiCon)
08/18/2015 11:31:45 vte1 VT: DLm0076I: Interface #2: hardware s/n: BFD1243F14784
```

```
08/18/2015 11:31:45 vte1 VT: DLm0070I: Interface #2: TRANSX emulation version set to 3
08/18/2015 11:31:45 vte1 VT: DLm0081I: Interface #2: Current state: running; Desired state: running; Media Up,
Loop Up
================================================================================
```

SAVE TRACE Command

Sample JCL;

Example using DLm Unit that belongs to an MTL:

```
//GO1       EXEC DLMCMDP1,CMD=501,UNIT=(/F200,,,SMSHONOR),            X
//               VCMD='SAVE TRACE'
```

Sample output

The sample output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5778
Tue Aug 18 11:41:52 EDT 2015
================================================================================
08/18/2015 11:41:52 vte1 VT: DLm0409I: Command received: 'SAVE TRACE'
08/18/2015 11:41:53 vte1 VT: DLm0801I: Trace buffers saved to disk
08/18/2015 11:41:54 vte1 VT: Saving managed trace files /opt/bti/dumps/u_2015-08-18_114154.trc and
/opt/bti/dumps/u_2015-08-18_11415 4.tbl
================================================================================
```

Security concerns

The VTE Command Processing feature of CP501 allows any DLm command to be passed, hence establish appropriate security measures, for example RACF, to restrict its use to authorized personnel.

Obtaining DLm Log Information: CP502

CP502 enables the transfer of the contents of the following to a mainframe tape file:

- btilog files
- /var/log/messages
- VTE xmap and ximl file.

Note: This information may vary according to DLm model. For example, there is no ximl file on Gen 4 DLms.

- Output of command **ipmitool sel list**

Submit **CP502** from the mainframe by using either the **DLMCMDPR** or **DLMCMDP1** utility.

btilog files

Each VTE maintains log files of operations performed by the virtual tape emulation application. These log files are known as btilogs. These files are a primary source of information for operation sequencing, completion, and problem determination. The btilogs are text files that contain the detailed activity of all VTEs and are located in the /var/bti/log directory. The most recent btilog file is *btilog*.

The number of btilogs to be transferred is specified by the user. The default value is 2 (the current log file and its immediate predecessor) and the maximum value is 10.

/var/log/messages

This is the standard operating system log file, which contains messages from the system software, kernel messages, non-kernel boot issues, and general system messages. As this file can be very large, **CP502** copies only the last 1,000 lines to the mainframe flat file.

VTE xmap and ximl file

The VTE contains up to six virtual tape device configuration files—xmap0, xmap1, xmap2, and so on in the /etc/bti directory. Only one of them can be active at any given time. The /etc/bti/ximl file maintains an entry for the currently active xmap file.

Note: On Gen 4 DLms, there is no /etc/bti/ximl file and the only xmap file is xmap0 which is the current xmap file.

ipmitool sel list

When this operating system level command is executed, the VTE baseboard controller system event log (SEL) is transferred to the mainframe.

CP502 syntax

The **CP502** command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=502,UNIT=unitid1,
//              CPARMS='command parameters'
//GO       EXEC DLMCMDPR,CMD=502,UNIT=unitid1,UNIT2=unit2,
//              CPARMS='command parameters'
```

CP502 parameters

Table 14 on page 79 defines the command parameters.

Table 14 **CP502 parameters**

Parameter	Description
CPARMS	Enter an optional command specific parameters depending on the files to transfer: • L0 - btilog files requested (see Number of btilogs to collect) • L1 - (Default) btilogs and /var/log/message • L2 - btilogs, /var/log/message, /etc/bti/ximl and all the xmap files and output of command ipmitool sel list • The bumber of btilogs to collect: 2 to 10 files. The default value is 2.

Example 1:

Sample JCL:

The JCL below a DLm Unit, (F207), that is not in an MTL:

```
//GO2      EXEC DLMCMDPR,CMD=502,UNIT=/F200,UNIT2=F201,
//              CPARMS='L2 3'
```

Sample output

The sample output shown below is generated by the above example Command Processor.

```
EMC DLm VTEC Log Collection Tool (502) v4.4.0-r5798
Wed Sep  2 10:11:08 EDT 2015
================================================================================
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
BTI Log name : /var/bti/log/old-logs/btilog-20150831

08/30/2015 23:59:04 vte1 VTd: ***** Rotated Virtuent log *****
08/30/2015 23:59:05 vte1 VT: DLm0409I: Command received: 'q all'
08/30/2015 23:59:05 vte1 VT:
```

```
08/30/2015 23:59:05 vte1 VT: Version 7.40-15840, built Jul  2 2015 14:11:50
08/30/2015 23:59:05 vte1 VT: Model Family: dlm8100v
08/30/2015 23:59:05 vte1 VT: Licensed Features Enabled:
08/30/2015 23:59:05 vte1 VT:   EMC FLR Support
08/30/2015 23:59:05 vte1 VT:   EMC DDRL Support
08/30/2015 23:59:05 vte1 VT:   SCSI Tape Support
….
08/31/2015 11:41:18 vte1 VT: DLm0490I: Device F207 Volume BFL502 unloaded
08/31/2015 11:41:18 vte1 VT: DLm0405I: Device F207 Display: 'KBFL502 '
08/31/2015 23:59:01 vte1 VTd: ***** Rotating Virtuent log *****
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
BTI Log name : /var/bti/log/old-logs/btilog-20150901

08/31/2015 23:59:04 vte1 VTd: ***** Rotated Virtuent log *****
08/31/2015 23:59:04 vte1 VT: DLm0409I: Command received: 'q all'
08/31/2015 23:59:04 vte1 VT:
08/31/2015 23:59:04 vte1 VT: Version 7.40-15840, built Jul  2 2015 14:11:50
08/31/2015 23:59:04 vte1 VT: Model Family: dlm8100v
….
09/01/2015 14:15:19 vte1 VT: DLm0409I: Command received: 'RESET CHANNEL ADAPTER'
09/01/2015 14:15:19 vte1 VT: DLm0891E: Invalid command syntax
09/01/2015 23:59:01 vte1 VTd: ***** Rotating Virtuent log *****
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
BTI Log name : /var/bti/log/btilog

09/01/2015 23:59:04 vte1 VTd: ***** Rotated Virtuent log *****
09/01/2015 23:59:04 vte1 VT: DLm0409I: Command received: 'q all'
09/01/2015 23:59:04 vte1 VT:
09/01/2015 23:59:04 vte1 VT: Version 7.40-15840, built Jul  2 2015 14:11:50
09/01/2015 23:59:04 vte1 VT: Model Family: dlm8100v
09/01/2015 23:59:04 vte1 VT: Licensed Features Enabled:
….
09/02/2015 10:11:07 vte1 VT: DLm0405I: Device F207 Display: ' BG0340S'
09/02/2015 10:11:07 vte1 VT: DLm0550I: Command received from Host: device F207: cmd = 'RUNWAIT (600) 502 "L2" "3" "DEFAULT"
09/02/2015 10:11:07 vte1 VT: DLm0554I: "RUNWAIT(600) 502" command has started (RUN process #36)
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
Linux System Log name : /var/log/messages

Sep  1 10:09:36 vte1 sshd¢125940!: Did not receive identification string from 10.241.216.65
Sep  1 10:10:01 vte1 /usr/sbin/cron¢126349!: (root) CMD (/opt/EMC_TOOLS/bin/dlm_dd_report.pl >/dev/null 2>&1)
Sep  1 10:10:01 vte1 sudo:     root : TTY=unknown ; PWD=/root ; USER=root ; COMMAND=/usr/bin/vte_version
Sep  1 10:14:09 vte1 /etc/bti/userscripts/script501: EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5798
Sep  1 10:14:09 vte1 /etc/bti/userscripts/script501: CP501:DEBUG_INTERNAL_STATUS_CODE : 4
Sep  1 10:14:09 vte1 /etc/bti/userscripts/script501: Executing 'HELP' command...
….
Sep  2 10:11:08 vte1 /etc/bti/userscripts/script502: Command Param : L2 3 DEFAULT BFL502.FLAT
Sep  2 10:11:08 vte1 /etc/bti/userscripts/script502: Collecting BTI Log files.
Sep  2 10:11:08 vte1 /etc/bti/userscripts/script502: Collecting Linux System Log /var/log/messages.
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
Current XMAP File : /etc/bti/xmap0

# /etc/bti/xmap0
#
# Configuration file - automatically generated by apply script - do not modify
#
#
USERNAME          xmap0
DESCRIPTION
ERASEPOLICY       SPACE
ERASEPOLICY       365D
TRACELEVEL        1
FWPATHTRACE       0
WARNINGPERCENT    88
….
¢DEVICE! INDEX=0C ACTIVE=1 NAME=F20C TYPE=3590 CU=00 UNIT=0C PATH=/tapelibFLR IDRC=yes
¢DEVICE! INDEX=0D ACTIVE=1 NAME=F20D TYPE=3590 CU=00 UNIT=0D PATH=/tapelibFLR IDRC=yes
¢DEVICE! INDEX=0E ACTIVE=1 NAME=F20E TYPE=3590 CU=00 UNIT=0E PATH=/tapelibFLR IDRC=yes
¢DEVICE! INDEX=0F ACTIVE=1 NAME=F20F TYPE=3590 CU=00 UNIT=0F PATH=/tapelibFLR IDRC=yes
--------------------------------------------------------------------------------
#####Diagnostic-Data#####
ipmitool sel list Output

   1 | 02/22/2013 | 19:36:35 | Event Logging Disabled #0x07 | Log area reset/cleared | Asserted
   2 | 02/22/2013 | 20:56:55 | Power Unit #0x01 | Power off/down | Asserted
   3 | 02/22/2013 | 20:57:01 | Power Unit #0x01 | Power off/down | Deasserted
   4 | 02/22/2013 | 20:57:12 | System Event #0x83 | Timestamp Clock Sync | Asserted
   5 | 02/22/2013 | 20:57:12 | System Event #0x83 | Timestamp Clock Sync | Asserted
   6 | 02/22/2013 | 20:57:42 | Physical Security #0x04 | System unplugged from LAN | Asserted
   7 | 02/22/2013 | 20:58:00 | Physical Security #0x04 | System unplugged from LAN | Deasserted
   8 | 02/22/2013 | 20:58:40 | System Event #0x83 | OEM System boot event | Asserted
   9 | 02/22/2013 | 20:59:36 | Physical Security #0x04 | System unplugged from LAN | Asserted
   a | 02/22/2013 | 21:00:04 | Physical Security #0x04 | System unplugged from LAN | Deasserted
   b | 02/22/2013 | 21:19:00 | Physical Security #0x04 | System unplugged from LAN | Asserted
```

```
….
47d | 08/12/2015 | 20:26:19 | System Event #0x83 | Timestamp Clock Sync | Asserted
47e | 08/12/2015 | 20:26:19 | System Event #0x83 | Timestamp Clock Sync | Asserted
47f | 08/12/2015 | 20:27:54 | System Event #0x83 | Timestamp Clock Sync | Asserted
480 | 08/12/2015 | 20:27:54 | System Event #0x83 | Timestamp Clock Sync | Asserted
481 | 08/12/2015 | 20:29:00 | System Event #0x83 | OEM System boot event | Asserted

==============================================================================
```

Obtaining Virtual Tape Volume Information

This section describes how to obtain Virtual Tape Volume information reports:

- "Using CP503 command"
- "Using awsprint utility"

Using CP503 command

CP503 provides an output report of the DLm Virtual Tape Volume Inventory.

Submit **CP503** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP503 syntax

The **CP503** command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=503,UNIT=unitid1,
//                CPARMS='command parameters'
```

CP503 parameters

The next table defines the **CP503** parameters.

Table 15 **awsprint parameters**

Parameter	Description
-p	Required , if not using the DLMCMDPR or DLMCMDP1 PROC. Path to the virtual tape library. Typically, /tapelib

Note: When using the DLMCMDPR or DLMCMDP1 PROCs, this parameter is overridden by the TLIB parameter, which defaults to the Virtual Tape Library associated with the UNIT parameter.

Table 15 **awsprint parameters**

-s	Optional. parameters to select a specific volume or range of volumes. For example: • **-s** BT0010 — Lists only volume BT0010 • **-s** BT00* — Lists all volumes from BT0000 to BT0099 **Note:** *If **-s** is omitted, all volumes in the tape library are listed.
-<sort criteria>	Sort criteria. sort by: • -N — The filename. This is the default sort order • -V — The volume serial number • -D — Last modification date • -H — Tape Header1 (HDR1) dataset name for labeled volumes and data in the TAPE INFO field for unlabeled volumes • -I — Filename with scratch and unscratched tapes interspersed • -S — Size
-d	Display tape content details.
-a	Display all, scratch and non-scratch, volumes
-r	Recurse the directories
--less	Displays less information (by not opening the tape file). Tape label information, the VOL1 label, label type, and Data Set Name (DSN), are not to be displayed. Using this option speeds up the processing because each DLm Tape volume in the directories processed does not need to be opened, read and closed..

Example 1: Using CP503 to obtain the DLm Virtual Tape Volume Inventory with default reporting options.

Note: The following example uses a Unit, (F207), which belongs to an MTL.

```
//GO3      EXEC DLMCMDPR,CMD=503,UNIT=(/F207,,,SMSHONOR),UNIT2=F201
```

Sample output

```
Sorted by Name 09/14/2015  11:25:30 AM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details
   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO          TAPE SIZE   FLR EXP

     1   BG0301  BG0301 08/26/15 15:19 S HDR1MCMD.TEMPFILE.H01     32.7K
     2   BG0302  BG0302 08/26/15 15:21 S HDR1MCMD.TEMPFILE.H01     32.7K
     3   BG0304  BG0304 08/27/15 11:01 S HDR1MCMD.TEMPFILE.H01     32.7K
     4   BG0306  BG0306 08/28/15 15:41 S HDR1P499.TEMPFILE.H01     32.7K
     5   BG0308  BG0308 08/28/15 15:42 S HDR1P499.TEMPFILE.H01     32.7K
     6   BG0310  BG0310 08/28/15 15:42 S HDR1P499.TEMPFILE.H01     32.7K
```

```
     7    BG0312   BG0312 08/28/15 15:42 S HDR1P499.TEMPFILE.H01     32.7K
     8    BG0314   BG0314 08/28/15 15:43 S HDR1P499.TEMPFILE.H01     32.7K
     9    BG0316   BG0316 08/28/15 15:43 S HDR1P499.TEMPFILE.H01     32.7K
    10    BG0318   BG0318 08/28/15 15:44 S HDR1P499.TEMPFILE.H01     32.7K
    11    BG0320   BG0320 08/28/15 15:44 S HDR1P499.TEMPFILE.H01     32.7K
    12    BG0322   BG0322 08/28/15 15:45 S HDR1P499.TEMPFILE.H01     32.7K
```

Example 2: Using CP503 to obtain the DLm Virtual Tape Volume Inventory with the recursive option and to only include VOLSERs beginning in BG030.

```
//GO3     EXEC DLMCMDPR,CMD=503,UNIT=(/F207,,,SMSHONOR),UNIT2=F201,
//             CPARMS='-r -s BG030*'
```

Sample output

```
Sorted by Name  09/14/2015  11:29:40 AM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: BG030* Without Details
   NO.  NAME     VOLSER     DATE    TIME LBL   TAPE INFO          TAPE SIZE   FLR EXP

     1    BG0301   BG0301 08/26/15 15:19 S HDR1MCMD.TEMPFILE.H01     32.7K
     2    BG0302   BG0302 08/26/15 15:21 S HDR1MCMD.TEMPFILE.H01     32.7K
     3    BG0304   BG0304 08/27/15 11:01 S HDR1MCMD.TEMPFILE.H01     32.7K
     4    BG0306   BG0306 08/28/15 15:41 S HDR1P499.TEMPFILE.H01     32.7K
     5    BG0308   BG0308 08/28/15 15:42 S HDR1P499.TEMPFILE.H01     32.7K
*************************************************************************
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01
```

Example 3: Using CP503 to obtain the DLm Virtual Tape Volume Inventory recursive option, sorted by HDR1s.

```
//GO3     EXEC DLMCMDPR,CMD=503,UNIT=(/F207,,,SMSHONOR),UNIT2=F201,
//             CPARMS='-r -H'
```

Sample output

```
Sorted by HDR1  09/14/2015  11:36:30 AM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details
   NO.  NAME     VOLSER     DATE    TIME LBL   TAPE INFO          TAPE SIZE   FLR EXP

     1    FLR_Lo           06/24/15 08:19 D Directory                   80
     2   ~BG0366   BG0366 12/31/89 23:00 S HDR1000000000000000000      416
     3   ~BG0368   BG0368 12/31/89 23:00 S HDR1000000000000000000      416
     4   ~BG0370   BG0370 12/31/89 23:00 S HDR1000000000000000000      416
     5   ~BG0372   BG0372 12/31/89 23:00 S HDR1000000000000000000      416
     6   ~BG0374   BG0374 12/31/89 23:00 S HDR1000000000000000000      416
     7   ~BG0376   BG0376 12/31/89 23:00 S HDR1000000000000000000      416
     8   ~BG0378   BG0378 12/31/89 23:00 S HDR1000000000000000000      416
     9   ~BG0380   BG0380 12/31/89 23:00 S HDR1000000000000000000      416
```

Example 4: Using CP503 to obtain the DLm Virtual Tape Volume Inventory with the recursive and detail reporting option.

```
//GO3     EXEC DLMCMDPR,CMD=503,UNIT=(/F207,,,SMSHONOR),UNIT2=F201,
//             CPARMS='-r -d'
```

Sample output

```
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  11:38:19 AM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes With Details

   NO.  NAME    VOLSER    DATE   TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

*******************************************************************************
     1  BG0301  BG0301  08/26/15 15:19 S    HDR1MCMD.TEMPFILE.H01  32.7K
*******************************************************************************
DETAILS OF /tapelibFLR/CEL1_FLRP_FS1/BG0301

<EBCDIC LABELS>
VOL1BG0301                                                  TCSBT
HDR1MCMD.TEMPFILE.H01BG030100010001      0152380000000000000IBM OS/VS 370
HDR2U0000000000000RGDLMCMD/STEP3    P    0004B1
...Tape File #1: 3 x 80-byte blocks
   0 Blocks Compressed  3 Blocks not Compressed
<TM>
<TM>
EOF1MCMD.TEMPFILE.H01BG030100010001      0152380000000000000IBM OS/VS 370
EOF2U0000000000000RGDLMCMD/STEP3    P    0004B1
...Tape File #3: 2 x 80-byte blocks
   0 Blocks Compressed  2 Blocks not Compressed
<TM>
<TM>
<1 x 32500-byte AWSMAP>
End of tape.
*******************************************************************************
```

Example 5: Using CP503 to obtain the DLm Virtual Tape Volume Inventory with the recursive and --less reporting options.

```
//GO3     EXEC DLMCMDPR,CMD=503,UNIT=(/F207,,,SMSHONOR),UNIT2=F201,
//             CPARMS='-r --less'
```

Sample output

```
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  01:16:56 PM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE   TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BG0301          08/26/15 15:19                          32.7K
     2   BG0302          08/26/15 15:21                          32.7K
     3   BG0304          08/27/15 11:01                          32.7K
     4   BG0306          08/28/15 15:41                          32.7K
     5   BG0308          08/28/15 15:42                          32.7K
     6   BG0310          08/28/15 15:42                          32.7K
     7   BG0312          08/28/15 15:42                          32.7K
     8   BG0314          08/28/15 15:43                          32.7K
     9   BG0316          08/28/15 15:43                          32.7K
    10   BG0318          08/28/15 15:44                          32.7K
    11   BG0320         08/28/15 15:44                          32.7K
```

Note on CP503 Tape Inventory reports Command Processors typically store their output in special Virtuent files called Flat Files. These Flat Files have a name that begins with BFL and ends with .FLAT. Typically a CP503 execution will create a file called

BFL503.FLAT and it will be stored in the tapelib primary folder. These Flat Files as well as the File System folders are often seen in CP503 reports, prior to the report output containing the DLm Tape Volume inventory.

Using awsprint utility

You must establish a CLI session on a DLm VTE to use awsprint. **awsprint** is a virtual tape library utility that the mainframe user can execute from the shell on the VTE to review the contents of the virtual tape library. **awsprint** enables you to produce lists of the tapes in the virtual tape library. The output is either displayed directly on the VTE Linux shell or written to a FLAT file. You can view the file using any text editor.

awsprint syntax

The syntax to run the **awsprint** utility in a VTE Shell is:

```
awsprint -p/tapelib -sBT00* -N -d
```

Table 15 on page 81 defines the **awsprint** command parameters. Also, example 1 below shows the syntax and parameters available. This report is returned by awsprint when awsprint is invoked without any parameters.

Example 1: Executing awsprint with no options specified. This results in a report of the awsprint syntax and a description of its options.

```
vte1:~ # awsprint
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Usage: awsprint directory [-s pattern] [-NDVHIS] [-dar] [--less]
directory      Specifies the path where the AWS tapes are stored.
                This is a required parameter. For backwards compatibility,
                directory can be specified with a "-p" switch.
-s pattern     Option to select files by filename, up to six characters;
                or to select files by scratch filename, up to seven characters;
                an optional ending '*' can be used as a wildcard.
                The default is to select all files in the directory.
-N             Sort output by fileName.
-D             Sort output by Date,
-V             Sort output by VOL1 Volser.
-H             Sort output by HDR1 dataset name.
-I             Sort output by fileName, interspersing scratch and non-scratch
                files (which normally would collate into separate groups).
-S             Sort output by file Size.
                Only one sort option can be specified.
                The default sort sequence is -N, by fileName.
-r --recurse   Specifies that all directories should be recursed.
-a --all       Specifies that all (scratch and non-scratch) volumes
                are to be printed regardless of selection pattern.
-h --help      Displays this help information.
-V --version   Displays version number.
-d --details   Specifies that tape details should be printed.
-l --less      Specifies that tape label information (VOL1 label, label
                type, DSN) are not to be displayed. Using this option will
```

```
                greatly speed up the program as awsprint does not need
                to open, read, and close every file in the directory.
Output is written to the console. You can use redirection (">") to
   write the awsprint output to a file.

Usage example:
awsprint /tapelib -sBT00* -H >tapelist.txt
   Selects all tapes starting with BT00 in path /tapelib, sorted by the
   HDR1 dataset name. The output is redirected to the file tapelist.txt.
```

Example 2: Executing awsprint with tapelib folder specified. Note that BFL FLAT files are stored on this folder.

```
vte1:~ # awsprint /tapelibFLR/CEL1_FLRP_FS1
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:30:08 PM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BG0301  BG0301 08/26/15 15:19 S HDR1MCMD.TEMPFILE.H01    32.7K
     2   BG0302  BG0302 08/26/15 15:21 S HDR1MCMD.TEMPFILE.H01    32.7K
     3   BG0304  BG0304 08/27/15 11:01 S HDR1MCMD.TEMPFILE.H01    32.7K
     4   BG0306  BG0306 08/28/15 15:41 S HDR1P499.TEMPFILE.H01    32.7K
     5   BG0308  BG0308 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     6   BG0310  BG0310 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     7   BG0312  BG0312 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     8   BG0314  BG0314 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
     9   BG0316  BG0316 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
    10   BG0318  BG0318 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    11   BG0320  BG0320 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    12   BG0322  BG0322 08/28/15 15:45 S HDR1P499.TEMPFILE.H01    32.7K
    13   BG0324  BG0324 08/28/15 16:05 S HDR1MGDLMTL1.TEMP.H01    32.7K
    14   BG0326  BG0326 08/28/15 16:11 S HDR1MGDLMTL1.TEMP.H01    32.7K
    15   BG0328  BG0328 08/31/15 09:56 S HDR1MGDLMTL1.TEMP.H01    32.7K
    16   BG0330  BG0330 08/31/15 10:01 S HDR1MGDLMTL1.TEMP.H01    32.7K
    17   BG0332  BG0332 08/31/15 10:02 S HDR1MGDLMTL1.TEMP.H01    32.7K
    18   BG0334  BG0334 08/31/15 10:04 S HDR1MGDLMTL1.TEMP.H01    32.7K
    19   BG0336  BG0336 09/01/15 10:08 S HDR1MGDLMTL1.TEMP.H01    32.7K
    20   BG0338  BG0338 09/01/15 10:21 S HDR1MGDLMTL1.TEMP.H01    32.7K
```

Example 3: Executing awsprint with the recursive option.

```
vte1:~ # awsprint /tapelibFLR -r
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:31:35 PM
Path: /tapelibFLR
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BFL499          08/28/15 15:45 F FLAT FILE               2.3K
     2   BFL500          08/20/15 14:01 F FLAT FILE               2.2K
     3   BFL501          09/01/15 10:14 F FLAT FILE              18.0K
     4   BFL502          09/14/15 13:16 F FLAT FILE               144K
     5   BFL503          09/02/15 11:07 F FLAT FILE               145K
     6   BFL504          09/02/15 10:22 F FLAT FILE               2.6K
     7   BFL601          09/02/15 10:23 F FLAT FILE               4.9K
     8   BFL602          09/02/15 10:23 F FLAT FILE               8.0K
     9   BFL603          09/02/15 10:23 F FLAT FILE               9.5K
….

awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:31:35 PM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP
```

```
     1   BG0301  BG0301 08/26/15 15:19 S HDR1MCMD.TEMPFILE.H01    32.7K
     2   BG0302  BG0302 08/26/15 15:21 S HDR1MCMD.TEMPFILE.H01    32.7K
     3   BG0304  BG0304 08/27/15 11:01 S HDR1MCMD.TEMPFILE.H01    32.7K
     4   BG0306  BG0306 08/28/15 15:41 S HDR1P499.TEMPFILE.H01    32.7K
     5   BG0308  BG0308 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     6   BG0310  BG0310 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     7   BG0312  BG0312 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     8   BG0314  BG0314 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
     9   BG0316  BG0316 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
    10   BG0318  BG0318 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    11   BG0320  BG0320 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    12   BG0322  BG0322 08/28/15 15:45 S HDR1P499.TEMPFILE.H01    32.7K
    13   BG0324  BG0324 08/28/15 16:05 S HDR1MGDLMTL1.TEMP.H01    32.7K
    14   BG0326  BG0326 08/28/15 16:11 S HDR1MGDLMTL1.TEMP.H01    32.7K

….

awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:31:35 PM
Path: /tapelibFLR/CEL1_FLRP_FS2
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BG0300  BG0300 08/26/15 15:11 S HDR1G20A.GDG.G0068V00     1.2M
     2   BG0303  BG0303 08/26/15 15:20 S HDR1MCMD.TEMPFILE.H01    32.7K
     3   BG0305  BG0305 08/27/15 11:00 S HDR1RGREEN7.TEMPX        32.7K
     4   BG0307  BG0307 08/28/15 15:41 S HDR1P499.TEMPFILE.H01    32.7K
     5   BG0309  BG0309 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     6   BG0311  BG0311 08/28/15 15:42 S HDR1P499.TEMPFILE.H01    32.7K
     7   BG0313  BG0313 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
     8   BG0315  BG0315 08/28/15 15:43 S HDR1P499.TEMPFILE.H01    32.7K
     9   BG0317  BG0317 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    10   BG0319  BG0319 08/28/15 15:44 S HDR1P499.TEMPFILE.H01    32.7K
    11   BG0321  BG0321 08/28/15 15:45 S HDR1P499.TEMPFILE.H01    32.7K
    12   BG0323  BG0323 08/28/15 15:59 S HDR1MGDLMTL1.TEMP.H01    32.7K
    13   BG0325  BG0325 08/28/15 16:08 S HDR1MGDLMTL1.TEMP.H01    32.7K
    14   BG0327  BG0327 08/28/15 16:12 S HDR1MGDLMTL1.TEMP.H01    32.7K
    15   BG0329  BG0329 08/31/15 09:59 S HDR1MGDLMTL1.TEMP.H01    32.7K
    16   BG0331  BG0331 08/31/15 10:01 S HDR1MGDLMTL1.TEMP.H01    32.7K
….
```

Example 4: Executing awsprint with tapelib specified along with recursive and detail options. This results in a report of all tape volumes in a tapelib.

```
vte1:~ # awsprint /tapelibFLR -r -d
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:32:05 PM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes With Details

   NO.  NAME    VOLSER    DATE    TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

*************************************************************************
     1  BG0301  BG0301  08/26/15 15:19 S    HDR1MCMD.TEMPFILE.H01  32.7K
*************************************************************************
DETAILS OF /tapelibFLR/CEL1_FLRP_FS1/BG0301

<EBCDIC LABELS>
VOL1BG0301                                          TCSBT
HDR1MCMD.TEMPFILE.H01BG030100010001      0152380000000000000IBM OS/VS 370
HDR2U0000000000000RGDLMCMD/STEP3   P      0004B1
...Tape File #1: 3 x 80-byte blocks
   0 Blocks Compressed  3 Blocks not Compressed
<TM>
<TM>
EOF1MCMD.TEMPFILE.H01BG030100010001      0152380000000000000IBM OS/VS 370
EOF2U0000000000000RGDLMCMD/STEP3   P      0004B1
...Tape File #3: 2 x 80-byte blocks
   0 Blocks Compressed  2 Blocks not Compressed
<TM>
<TM>
<1 x 32500-byte AWSMAP>
```

```
End of tape.
*************************************************************************
     2  BG0302  BG0302  08/26/15 15:21 S    HDR1MCMD.TEMPFILE.H01  32.7K
*************************************************************************
DETAILS OF /tapelibFLR/CEL1_FLRP_FS1/BG0302

<EBCDIC LABELS>
VOL1BG0302                                              TCSBT
HDR1MCMD.TEMPFILE.H01BG030200010001      01523800000000000000IBM OS/VS 370
HDR2U000000000000RGDLMCMD/STEP3   P      0004B1
...Tape File #1: 3 x 80-byte blocks
   0 Blocks Compressed  3 Blocks not Compressed
<TM>
<TM>
EOF1MCMD.TEMPFILE.H01BG030200010001      01523800000000000000IBM OS/VS 370
EOF2U000000000000RGDLMCMD/STEP3   P      0004B1
...Tape File #3: 2 x 80-byte blocks
   0 Blocks Compressed  2 Blocks not Compressed
<TM>
<TM>
<1 x 32500-byte AWSMAP>
End of tape.
*************************************************************************
```

Example 5: Executing awsprint with recursive and less options.

```
vte1:~ # awsprint /tapelibFLR -r -l
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:32:19 PM
Path: /tapelibFLR
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE   TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BFL499         08/28/15 15:45                            2.3K
     2   BFL500         08/20/15 14:01                            2.2K
     3   BFL501         09/01/15 10:14                           18.0K
     4   BFL502         09/14/15 13:16                            144K
     5   BFL503         09/02/15 11:07                            145K
     6   BFL504         09/02/15 10:22                            2.6K
     7   BFL601         09/02/15 10:23                            4.9K
     8   BFL602         09/02/15 10:23                            8.0K
     9   BFL603         09/02/15 10:23                            9.5K
    10   BFL999         09/02/15 11:27                            593K
    11   BFLABC         08/25/15 11:01                            3.4K
    12   BFLBG1         09/01/15 10:29                           18.0K
    13   BFLBG6         08/24/15 14:33                             320
    14   BFLBG9         08/24/15 16:20                             320
    15   BFLBGA         08/25/15 11:05                            5.7K
    16   BFLGR1         08/27/15 11:01                            145K
    17   BFLM10         08/26/15 09:27                           15.5K
    18   BFLMT1         08/25/15 13:38                               0
    19   BFLMT2         08/25/15 14:00                               0
    20   BFLMT3         08/25/15 14:03                               0
    21   BFLMT4         08/25/15 14:04                               0
    22   BFLMT5         08/25/15 14:05                               0
    23   BFLMT7         08/25/15 14:17                               0
    24   BFLXXX         08/25/15 11:03                            9.5K
    25   CEL1_F         09/14/15 14:27                           10.0K
    26   CEL1_F         09/13/15 23:02                           10.0K
*************************************************************************
awsprint Version 7.40-15840, Compiled Jul  2 2015 14:12:01

Sorted by Name 09/14/2015  02:32:19 PM
Path: /tapelibFLR/CEL1_FLRP_FS1
Selection Criteria: All Tapes Without Details

   NO.  NAME    VOLSER    DATE   TIME LBL   TAPE INFO         TAPE SIZE  FLR EXP

     1   BG0301         08/26/15 15:19                           32.7K
     2   BG0302         08/26/15 15:21                           32.7K
     3   BG0304         08/27/15 11:01                           32.7K
     4   BG0306         08/28/15 15:41                           32.7K
     5   BG0308         08/28/15 15:42                           32.7K
     6   BG0310         08/28/15 15:42                           32.7K
     7   BG0312         08/28/15 15:42                           32.7K
```

```
 8   BG0314          08/28/15 15:43                      32.7K
 9   BG0316          08/28/15 15:43                      32.7K
10   BG0318          08/28/15 15:44                      32.7K
11   BG0320          08/28/15 15:44                      32.7K
12   BG0322          08/28/15 15:45                      32.7K
13   BG0324          08/28/15 16:05                      32.7K
14   BG0326          08/28/15 16:11                      32.7K
15   BG0328          08/31/15 09:56                      32.7K
16   BG0330          08/31/15 10:01                      32.7K
17   BG0332          08/31/15 10:02                      32.7K
18   BG0334          08/31/15 10:04                      32.7K
19   BG0336          09/01/15 10:08                      32.7
```

Obtaining VNX Replication Information: CP504

CP504 collects the replication statistics of all VNX/base and expansion VNX file systems on which replication has been enabled. DLm version 4.0 and later supports collecting replication statistics from the expansion VNX.

Submit **CP504** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP504 syntax

The **CP504** command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=504,UNIT=unitid1,
//              VCMD='report type'
//GO       EXEC DLMCMDPR,CMD=504,UNIT=unitid1,UNIT2=unit2,
//              VCMD='report type'
```

CP504 parameters

Table 16 on page 89 defines the **CP504** parameters.

Table 16 **CP504 parameter**

Parameter	Description
Report type - VCMD	QUICK, SUMMARY, and VERBOSE

Sample JCL:

CP504:

```
//GO4      EXEC DLMCMDP1,CMD=504,UNIT=(/F207,,,SMSHONOR),CMD2=BG1,
//              VCMD='QUICK'

//GO4      EXEC DLMCMDPR,CMD=504,UNIT=/F200,UNIT2=F201,
//              VCMD='QUICK'
```

```
//GO5     EXEC DLMCMDP1,CMD=504,UNIT=(/F207,,,SMSHONOR),CMD2=BG1,
//             VCMD='SUMMARY'

//GO5     EXEC DLMCMDPR,CMD=504,UNIT=/F200,UNIT2=F201,
//             VCMD='SUMMARY'

//GO6     EXEC DLMCMDP1,CMD=504,UNIT=(/F207,,,SMSHONOR),CMD2=BG1,
//             VCMD='VERBOSE'

//GO6     EXEC DLMCMDPR,CMD=504,UNIT=/F200,UNIT2=F201,
//             VCMD='VERBOSE'
```

QUICK report

The QUICK report displays the following replication statistics:

- Destination VOLSER Range
- Time elapsed since last sync (Minutes)
- Avg Transfer Rate (MB/s)

Sample output

Sample QUICK report is shown below:

```
DLMCMD1 Runtime Report            19 Sep 2013 3:15pm

    Parameters:

        Input: CMD(504) DEBUG() FILE(504)      PATH(DEFAULT) VCMD(QUICK) WAIT(600)

       COMMAND: 504
   VTE COMMAND: QUICK
          PATH: DEFAULT
          FILE: BFL504.FLAT
     TIMESTAMP: 20130919-15:15:00-52096
         DEBUG: Off
    Task Name:  RGCMDPRO
    Task ID     J0016132
    Task User:  RGREEN7

DLMCMD2 Runtime Report            19 Sep 2013 3:15pm

    Parameters:

          Input: CMD(504) DEBUG() TLIB(DEFAULT)       CPARMS()

        COMMAND: 504
     Cmd Parms:
  Tape Library: DEFAULT
         DEBUG:

 Built Command: RUNWAIT (600) 504 "QUICK" "DEFAULT" "BFL504.FLAT"
"20130919-15:15:00-52096"

DLMCMD   VER 4.11     DLMCTRL = 2100
2013/09/19 15:15:03 CMD ISSUED:  RUNWAIT (600) 504 "QUICK" "DEFAULT" "BFL504.FLAT"
"20130919-15:15:00-52096"
20130919-15:15:00-52096

EMC Replication Stats Collection Tool (504) v4.0.0-r3001
Thu Sep 19 15:23:49 EDT 2013
Report Type : QUICK
=========================================================
Base Unit
---------------------------------------------------------
Destination VOLSER         Time Since          Avg Transfer
```

```
Range                          Last Sync (Mins)    Rate (MB/s)
-------------------------------------------------------------
tapelibTGT_50                        7.67                1.58
tapelibTGT_51                       15.33                0.38
tapelibTGT_52                       70.83                0.50
tapelibTGT_53                       73.00                0.38
tapelibTGT_54                       10.33                0.35
tapelibTGT_55                        7.67                1.97
tapelibTGT_56                       11.67                0.09
tapelibTGT_57                       71.17                0.24
tapelibTGT_58                       75.50                0.38
tapelibTGT_59                       11.17                0.33
tapelibTGT_60                        7.67                0.09
tapelibTGT_61                        8.83                0.91
tapelibTGT_62                       69.33                0.39
tapelibTGT_63                       71.00                0.41
tapelibTGT_64                        9.50                0.03
tapelibTGT_65                        8.83                0.03
tapelibTGT_66                        7.33                0.33
tapelibTGT_67                       71.17                0.06
tapelibTGT_68                       80.33                0.27
tapelibTGT_69                        9.50                0.06
tapelibTGT_70                        7.00                0.52
tapelibTGT_71                        8.33                0.33
tapelibTGT_72                       74.83                5.21
tapelibTGT_73                       79.33                0.53
tapelibTGT_74                       15.50                0.34

Margin of error for Time Since Last Sync (seconds): 1

Expansion Unit
-------------------------------------------------------------
Destination VOLSER          Time Since          Avg Transfer
Range                      Last Sync (Mins)    Rate (MB/s)
-------------------------------------------------------------
tapelibTGT_50                        8.03                1.58
tapelibTGT_51                       15.70                0.38
tapelibTGT_52                       71.20                0.50
tapelibTGT_53                       73.37                0.38
tapelibTGT_54                       10.70                0.35
tapelibTGT_55                        8.03                1.97
tapelibTGT_56                       12.03                0.09
tapelibTGT_57                       71.53                0.24
tapelibTGT_58                       75.87                0.38
tapelibTGT_59                       11.53                0.33
tapelibTGT_60                        8.03                0.09
tapelibTGT_61                        9.20                0.91
tapelibTGT_62                       69.70                0.39
tapelibTGT_63                       71.37                0.41
tapelibTGT_64                        9.87                0.03
tapelibTGT_65                        9.20                0.03
tapelibTGT_66                        7.70                0.33
tapelibTGT_67                       71.53                0.06
tapelibTGT_68                       71.03                0.27
tapelibTGT_69                        9.87                0.06
tapelibTGT_70                        7.37                0.52
tapelibTGT_71                        8.70                0.33
tapelibTGT_72                       75.20                5.21
tapelibTGT_73                       79.70                0.53
tapelibTGT_74                       15.87                0.34

Margin of error for Time Since Last Sync (seconds): 11
```

SUMMARY report

The SUMMARY report displays the following replication statistics for each file system:

Report fields	Description
Report Execution Time	Time when the report was created
Replication session name	Name of the replication session
Source VOLSER	VOLSER range that is the source for this replication session
Destination VOLSER	VOLSER range on a remote DLm that is the destination for this replication session
Max out of Sync Time	Maximum time between data synchronization of the destination VOLSER range with the source VOLSER range
Last Data Sync Time	Time when the last data synchronization was done between the source and destination VOLSER ranges
Time elapsed since last sync (Minutes)	Time elapsed since the last data synchronization
Source DLm Status	Status of the source DLm
Destination DLm Status	Status of the destination DLm
DLm Network Status	Network status between the source and destination DLm systems
Source Network Interface	Network interface that is part of this replication session at the source DLm
Destination Network Interface	Network interface that is part of this replication session at the destination DLm
Avg Transfer Rate (MB/s)	Average data transfer rate for this replication session
Margin of error since last sync	Accuracy of the elapsed since last sync value in seconds

Sample output

Sample SUMMARY report is shown below:

```
DLMCMD1 Runtime Report            19 Sep 2013 2:51pm
    Parameters:
```

```
         Input: CMD(504) DEBUG() FILE(504)       PATH(DEFAULT) VCMD(SUMMARY)
WAIT(600)

       COMMAND: 504
   VTE COMMAND: SUMMARY
          PATH: DEFAULT
          FILE: BFL504.FLAT
     TIMESTAMP: 20130919-14:51:06-42025
         DEBUG: Off
    Task Name:  RGCMDPRO
    Task ID     J0016130
    Task User:  RGREEN7

DLMCMD2 Runtime Report            19 Sep 2013 2:51pm

    Parameters:

         Input: CMD(504) DEBUG() TLIB(DEFAULT)       CPARMS()

       COMMAND: 504
     Cmd Parms:
  Tape Library: DEFAULT
         DEBUG:

 Built Command: RUNWAIT (600) 504 "SUMMARY" "DEFAULT" "BFL504.FLAT"
"20130919-14:51:06-42025"

DLMCMD    VER 4.11     DLMCTRL = 2100
2013/09/19 14:51:10 CMD ISSUED:  RUNWAIT (600) 504 "SUMMARY" "DEFAULT"
"BFL504.FLAT" "20130919-14:51:06-42025"
20130919-14:51:06-42025

EMC Replication Stats Collection Tool (504) v4.0.0-r3001
Thu Sep 19 14:59:45 EDT 2013
Report Type : SUMMARY
============================================================
Base Unit
------------------------------------------------------------

Replication Name               : Rep-50
Source DLm Status              : OK
DLm Network Status             : OK
Destination DLm Status         : OK
Report Execution Time          : Thu Sep 19 14:59:46 EDT 2013
Last Data Sync Time            : Thu Sep 19 14:47:20 EDT 2013
Time Since Last Sync           : 12.43 minutes
Destination VOLSER             : tapelibTGT_50
Destination Network Interface : 10.60.2.30
Max out of Sync Time           : 10 (minutes)
Avg Transfer Rate              : 1.58 (MB/s)
============================================================

Margin of error for Time Since Last Sync (seconds): 1

Expansion Unit
------------------------------------------------------------

Replication Name               : Rep-50
Source DLm Status              : OK
DLm Network Status             : OK
Destination DLm Status         : OK
Report Execution Time          : Thu Sep 19 14:59:47 EDT 2013
Last Data Sync Time            : Thu Sep 19 14:47:20 EDT 2013
Time Since Last Sync           : 12.45 minutes
Destination VOLSER             : tapelibTGT_50
Destination Network Interface : 10.60.2.30
Max out of Sync Time           : 10 (minutes)
Avg Transfer Rate              : 1.58 (MB/s)
============================================================

Margin of error for Time Since Last Sync (seconds): 1
```

VERBOSE report

This detailed replication data report is typically generated by EMC Customer Support personnel to analyze or troubleshoot replication issues. **CP504** does not modify or format the data collected from the **nas_replicate** command. It presents the data as is to the mainframe. Therefore, it may not be comprehensive to anyone other than an EMC Customer Support representative. The details include:

- Report Execution Time
- ID
- Name
- Source Status
- Network Status
- Destination Status
- Last Sync Time
- Type
- Celerra Network Server
- Dart Interconnect
- Peer Dart Interconnect
- Replication Role
- Source file system
- Source Data Mover
- Source Interface
- Source Control Port
- Source Current Data Port
- Destination file system
- Destination Data Mover
- Destination Interface
- Destination Control Port

- Destination Data Port
- Max Out of Sync Time (minutes)
- Next Transfer Size (KB)
- Current Transfer Size (KB)
- Current Transfer Remain (KB)
- Estimated Completion Time
- Current Transfer is Full Copy
- Current Transfer Rate (KB/s)
- Current Read Rate (KB/s)
- Current Write Rate (KB/s)
- Previous Transfer Rate (KB/s)
- Previous Read Rate (KB/s)
- Previous Write Rate (KB/s)
- Average Transfer Rate (KB/s)
- Average Read Rate (KB/s)

Sample output

Sample VERBOSE report is shown below:

```
DLMCMD1 Runtime Report            19 Sep 2013 3:16pm

    Parameters:

          Input: CMD(504) DEBUG() FILE(504)       PATH(DEFAULT) VCMD(VERBOSE)
WAIT(600)

       COMMAND: 504
   VTE COMMAND: VERBOSE
          PATH: DEFAULT
          FILE: BFL504.FLAT
     TIMESTAMP: 20130919-15:16:33-99929
         DEBUG: Off
    Task Name:  RGCMDPRO
    Task ID     J0016132
    Task User:  RGREEN7

DLMCMD2 Runtime Report            19 Sep 2013 3:16pm

    Parameters:

          Input: CMD(504) DEBUG() TLIB(DEFAULT)       CPARMS()

        COMMAND: 504
     Cmd Parms:
  Tape Library: DEFAULT
         DEBUG:
```

```
 Built Command: RUNWAIT (600) 504 "VERBOSE" "DEFAULT" "BFL504.FLAT"
"20130919-15:16:33-99929"

DLMCMD   VER 4.11     DLMCTRL = 2100
2013/09/19 15:16:37 CMD ISSUED:  RUNWAIT (600) 504 "VERBOSE" "DEFAULT"
"BFL504.FLAT" "20130919-15:16:33-99929"
20130919-15:16:33-99929

EMC Replication Stats Collection Tool (504) v4.0.0-r3001
Thu Sep 19 15:25:22 EDT 2013
Report Type : VERBOSE
================================================================================
Base Unit
--------------------------------------------------------------------------------

ID                                = 232_FNM00091100069_0000_258_APM00093100507_0000
Name                              = Rep-59
Source Status                     = OK
Network Status                    = OK
Destination Status                = OK
Last Sync Time                    = Thu Sep 19 15:12:50 EDT 2013
Type                              = file system
Celerra Network Server            = g2e
Dart Interconnect                 = g2bS6_g2eS6
Peer Dart Interconnect            = 60003
Replication Role                  = source
Source file system                 = tapelibREP_59
Source Data Mover                 = server_6
Source Interface                  = 10.55.6.30
Source Control Port               = 0
Source Current Data Port          = 0
Destination file system            = 91
Destination Data Mover            = server_6
Destination Interface             = 10.60.6.30
Destination Control Port          = 5085
Destination Data Port             = 8888
Max Out of Sync Time (minutes)    = 10
Next Transfer Size (KB)           = 0
Current Transfer Size (KB)        = 0
Current Transfer Remain (KB)      = 0
Estimated Completion Time         =
Current Transfer is Full Copy     = No
Current Transfer Rate (KB/s)      = 0
Current Read Rate (KB/s)          = 0
Current Write Rate (KB/s)         = 0
Previous Transfer Rate (KB/s)     = 220
Previous Read Rate (KB/s)         = 3353
Previous Write Rate (KB/s)        = 192
Average Transfer Rate (KB/s)      = 343
Average Read Rate (KB/s)          = 14602
Average Write Rate (KB/s)         = 2574

...

ID                                = 262_FNM00091100069_0000_288_APM00093100507_0000
Name                              = Rep-74
Source Status                     = OK
Network Status                    = OK
Destination Status                = OK
Last Sync Time                    = Thu Sep 19 15:18:10 EDT 2013
Type                              = file system
Celerra Network Server            = g2e
Dart Interconnect                 = g2bS6_g2eS6
Peer Dart Interconnect            = 60003
Replication Role                  = source
Source file system                 = tapelibREP_74
Source Data Mover                 = server_6
Source Interface                  = 10.55.6.30
Source Control Port               = 0
Source Current Data Port          = 0
Destination file system            = 106
Destination Data Mover            = server_6
Destination Interface             = 10.60.6.30
Destination Control Port          = 5085
Destination Data Port             = 8888
Max Out of Sync Time (minutes)    = 10
Next Transfer Size (KB)           = 0
Current Transfer Size (KB)        = 0
Current Transfer Remain (KB)      = 0
```

```
Estimated Completion Time       =
Current Transfer is Full Copy   = No
Current Transfer Rate (KB/s)    = 0
Current Read Rate (KB/s)        = 0
Current Write Rate (KB/s)       = 0
Previous Transfer Rate (KB/s)   = 266
Previous Read Rate (KB/s)       = 742
Previous Write Rate (KB/s)      = 234
Average Transfer Rate (KB/s)    = 350
Average Read Rate (KB/s)        = 4707
Average Write Rate (KB/s)       = 254

Expansion Unit
------------------------------------------------------------
ID        232_FNM00091100069_0000_258_APM00093100507_0000
Name                            = Rep-59
Source Status                   = OK
Network Status                  = OK
Destination Status              = OK
Last Sync Time                  = Thu Sep 19 15:12:50 EDT 2013
Type                            = file system
Celerra Network Server          = g2b
Dart Interconnect               = g2eS6_g2bS6
Peer Dart Interconnect          = 60003
Replication Role                = destination
Source file system               = 66
Source Data Mover               = server_6
Source Interface                = 10.55.6.30
Source Control Port             = 0
Source Current Data Port        = 0
Destination file system          = tapelibTGT_59
Destination Data Mover          = server_6
Destination Interface           = 10.60.6.30
Destination Control Port        = 5085
Destination Data Port           = 8888
Max Out of Sync Time (minutes)  = 10
Next Transfer Size (KB)         = 0
Current Transfer Size (KB)      = 0
Current Transfer Remain (KB)    = 0
Estimated Completion Time       =
Current Transfer is Full Copy   = No
Current Transfer Rate (KB/s)    = 0
Current Read Rate (KB/s)        = 0
Current Write Rate (KB/s)       = 0
Previous Transfer Rate (KB/s)   = 220
Previous Read Rate (KB/s)       = 3353
Previous Write Rate (KB/s)      = 192
Average Transfer Rate (KB/s)    = 343
Average Read Rate (KB/s)        = 14602
Average Write Rate (KB/s)       = 2574
```

Obtaining Data Domain Information: CP601, CP602, and CP603

CP601

CP601 generates Data Domain system utilization statistics and makes the reports available to the mainframe.

A Data Domain system compresses data at two levels: global and local. Global compression compares received data with the data already stored on disks. Duplicate data need not be stored again, while data that is new is compressed before being written to disk. Compressing the new data is called local compression.

Global compression is the same as data deduplication while local compression is the same as hardware compression.

The data compression achieved through data deduplication and hardware compression is collectively referred to as total compression in this document.

Total compression = Data deduplication + hardware compression

The Data Domain system deduplicates incoming tape data before storing it onto the disk. In this process, the actual amount of data to be stored on the disk is much less than the incoming data. CP601 reports the storage usage statistics, which include the following:

- Total amount of incoming data from VTEs
- Total amount of data stored on the disk after total compression
- Total available space for data storage
- Estimated amount of storage that will be cleaned during next Data Domain clean operation

CP601 collects the system usage data and saves it in a FLAT file on the VTE in EBCDIC format. This file can be retrieved by the mainframe.

Submit **CP601** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP601 syntax

The CP601 command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=601,UNIT=unitid1
//GO       EXEC DLMCMDPR,CMD=601,UNIT=unitid1,UNIT2=unit2
```

CP601 parameters

CP601command parameters - **none**.

Sample JCL:

```
//GO7      EXEC DLMCMDP1,CMD=601,UNIT=(/F207,,,SMSHONOR),CMD2=BG1
//GO7      EXEC DLMCMDPR,CMD=601,UNIT=/F200,UNIT2=F201
```

Sample output

The CP601 example shown above produces a storage utilization report as shown in the below sample:

```
  EMC Data Domain Storage Utilization Report (601) v4.4.0-r5798
Wed Sep  2 10:12:44 EDT 2015
Report Type : SYSTEM
================================================================================
Data Domain system: DD-1
--------------------------------------------------------------------------------
Total data received (GiB)                              : 7.4
Total storage capacity of Data Domain system (GiB): 21265.3
Total storage used (post compression) (GiB)          : 7.9
Total storage available (GiB)                        : 21257.3
Use%                                                 : 0%
Cleanable storage (GiB)                              : 0.0

Note:  * Estimated based on last cleaning of 2015/09/01 06:02:01.

--------------------------------------------------------------------------------
================================================================================
```

Table 17 on page 99 defines the **CP601** storage utilization report fields.

Table 17 **CP601 report fields**

Field	Description
Total data received (GiB)	Total amount of data received by the Data Domain system.
Total storage capacity of Data Domain system (GiB)	Total storage capacity of the Data Domain system.
Total storage used (post compression) (GiB)	Total storage used after achieving data deduplication and hardware compression of the data.
Total storage available (GiB)	Storage available {Total storage capacity of Data Domain system - Total storage used (post compression) } This is the storage available for data post data deduplication and hardware compression. For example, if total available capacity is 100 GB then it can store 500 GB of received data if the size of received data is 100 GB after total compression.
Use%	% of storage used.
Cleanable storage (GiB)	Data Domain does not make space available as soon as a file is deleted. This space is made available only after a periodic cleanup is run on Data Domain. This value shows the estimated amount of storage that will be cleaned up during next Data Domain cleanup operation. This estimation is based on last cleanup operation.

CP602

The Data Domain system deduplicates incoming tape data before storing it to disk. As a result of this process, the data size to be stored on disk is much smaller than the received data size.

CP602 generates a storage system compression report that provides data compression statistics for the Data Domain system.

Submit **CP602** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP602 syntax

The **CP602** command syntax is:

```
//GO      EXEC DLMCMDP1,CMD=602,UNIT=unitid1
//GO      EXEC DLMCMDPR,CMD=602,UNIT=unitid1,UNIT2=unit2
```

CP602 parameters

CP602 parameters - **none**.

Sample JCL:

```
//GO8     EXEC DLMCMDP1,CMD=602,UNIT=(/F207,,,SMSHONOR),CMD2=BG1

//GO8     EXEC DLMCMDPR,CMD=602,UNIT=/F200,UNIT2=F201
```

Sample output

```
EMC Data Domain Storage System Compression Report (602) v4.4.0-r5798
Wed Sep  2 10:13:00 EDT 2015
================================================================================
Data Domain system: DD-1
--------------------------------------------------------------------------------


From: 2015-08-26 10:00 To: 2015-09-02 10:00

                    Pre-Comp    Post-Comp   Global-Comp   Local-Comp      Total-Comp
                       (GiB)        (GiB)        Factor       Factor          Factor
                                                                        (Reduction %)
----------------   --------   ---------   -----------   ----------   -------------
Currently Used:*        7.4         7.9             -            -     0.9x (-6.5)
Written:
  Last 7 days           0.0         0.0          0.2x         8.5x     1.6x (36.4)
  Last 24 hrs           0.0         0.0          0.2x         8.5x     1.6x (36.7)
----------------   --------   ---------   -----------   ----------   -------------
 * Does not include the effects of pre-comp file deletes/truncates
```

CP602 generates a storage system data compression report that provides data compression statistics for the Data Domain system, including:

- Data received since the system was installed

- Data received in the last 7 days
- Data received in the last 24 hours

Note: When a tape is deleted on the mainframe, Data Domain does not immediately delete the files; however, they become unavailable. They are deleted when the next scheduled Data Domain cleanup operation is run. The cleanup deletes the tapes and frees up storage. Therefore, data generated for the last 7 days and the last 24 hours does not include the result of file deletion and truncation.

Table 18 on page 101 defines the CP602 data compression report fields.

Table 18 **CP602 report fields**

Field	Description
Pre-Comp	Size of received data.
Post-Comp	Size of compressed data. Size of data after total compression (deduplication and hardware compression). This is the actual amount of storage taken up on disks.
Global-Comp Factor	Ratio between received data size and deduplicated data size. Indicates that deduplication reduced the data size by *X* times. Global compression factor = Pre-Comp/(Size after deduplication)
Local-Comp Factor	Ratio between deduplicated data size and total compressed data size (deduplication and hardware compression). Indicates that hardware compression reduced deduplicated data by *X* times. Local compression factor = (Size after deduplication)/Post-Comp
Total-Comp Factor	Ratio between received data size and total compressed data size (deduplication and hardware compression). Indicates that total compression reduced the received data size by *X* times. It also shows the % reduction in data, that is, data stored on disk is Y% less than the data received. Total compression factor in percent: Total-Comp Factor = ((Pre-Comp - Post-Comp)/Pre-Comp)*100.

CP603

This Command Processor reports the following Data Domain replication information:

- Replication configuration
- Replication status
- Replication statistics
- Replication performance statistics
- Historical replication data for the past 24 hours

Submit **CP603** from the mainframe by using either **DLMCMDPR** or **DLMCMDP1** utility.

CP603 syntax

The **CP603** command syntax is:

```
//GO       EXEC DLMCMDP1,CMD=603,UNIT=unitid1,
//            CPARMS='command parameters'
//GO       EXEC DLMCMDPR,CMD=603,UNIT=unitid1, UNIT2=unit2,
//            CPARMS='command parameters'
```

CP603 parameters

Table 19 on page 102 defines the **CP603** parameters.

Table 19 **CP603 parameters**

Parameter	Description
Option - CPARMS	Enter one of the following reports to be generated: • STATUS — Status report containing the current replication status. • STATS —Statistics report containing replication statistics. • PERF — Performance report containing replication performance statistics. • HIST — History report containing replication I/O information. • ALL — All of the above reports.

Sample JCL:

```
//G10      EXEC DLMCMDP1,CMD=603,UNIT=(/F207,,,SMSHONOR),CMD2=BG1,
//             CPARMS=STATUS
```

```
//G10      EXEC DLMCMDPR,CMD=603,UNIT=/F200,UNIT2=F201,
//              CPARMS=STATUS
```

Sample output

```
20150902-10:10:12-86950

EMC Data Domain Replication Statistics Report (603) v4.4.0-r5798
Wed Sep  2 10:13:18 EDT 2015
Report Type : REPLICATION STATUS
================================================================================
Data Domain system: DD-1
--------------------------------------------------------------------------------

REPLICATION CONFIG
--------------------------------------------------------------------------------
CTX   Source                          Destination                                 Connection Host and Port
Low-bw-optim   Encryption   Enabled
---   ---------------------------   -----------------------------------------   -------------------------
------------   ----------   -------
2   mtree://a.b/data/col1/MTBKP1  mtree://DD1.corp.emc.com/data/col1/MTBKP1  a.b*      (default)*   disabled
disabled     yes
---   ---------------------------   -----------------------------------------   -------------------------
------------   ----------   -------
* Used for recovery only.
================================================================================

REPLICATION STATUS
--------------------------------------------------------------------------------
CTX:                                  2
Mode:                                 destination
Destination:                          mtree://DD1.corp.emc.com/data/col1/MTBKP1
Enabled:                              yes
Low bandwidth optimization:           disabled
Replication encryption:               disabled
Replication propagate-retention-lock: disabled
Local filesystem status:              enabled
Connection:                           idle since Wed Sep  2 10:00:03
State:                                normal
Error:                                no error
Sync'ed-as-of time:                   Wed Sep  2 10:00
Current throttle:                     unlimited (Recovery only)

================================================================================
```

Table 20 on page 103 defines the REPLICATION CONFIG fields that appear in all of the **CP603** replication reports.

Table 20 **CP603 reports: REPLICATION CONFIG fields**

Field	Description
CTX	The context number for directory replication.
Source	The Data Domain system that receives data from backup applications.

Field	Description
Enabled	The replication process is yes (enabled and available to replicate data) or no (disabled and not available to replicate data).
Destination	The Data Domain system that receives data from the replication source Data Domain system.
Connection	A source Data Domain system connects to the destination Data Domain system by using the destination name as returned by the **hostname** command on the destination or by using a destination name or IP address and port given with the replication modify connection-host command. The destination **hostname** may not resolve to the correct IP address for the connection when connecting to an alternate interface on the destination or when a connection passes through a firewall.

Table 21 on page 104 defines fields in the **CP603** REPLICATION STATUS report.

Table 21 **CP603 REPLICATION STATUS report fields**

Field	Description
CTX	The context number for directory replication.
Mode	The role of the local system: source or destination.
Destination	The replication destination.
Enabled	The enabled state (yes or no) of replication for each replication pair.
Local file system status	The status (enabled or disabled) of the local file system.
Connection	The most recent connection date and time or connection state for a replication pair.
State	The state of the replication process.
Error	A listing of any errors in the replication process.
Sync'ed-as-of time	The time when the most recently replicated data on the destination was generated on the source. A value ‘unknown’ appears during replication initialization.
Current throttle	The current throttle setting.

Table 22 on page 105 defines fields in the **CP603** REPLICATION STATISTICS report.

Table 22 **CP603 REPLICATION STATISTICS report fields**

Field	Description
CTX	The context number for directory replication.
Destination	The replication destination.
Network bytes sent	The number of bytes sent over the network. Does not include TCP/IP headers. Includes internal replication control information, metadata, and file system data.
Pre-compressed bytes written to source	Indicates the number of bytes of pre-compressed data sent to the source.
Pre-compressed bytes sent to destination	Indicates the number of bytes of pre-compressed data sent to the destination.
Pre-compressed bytes remaining	The sum of the sizes of one or more files remaining to be replicated for this context. **Note:** This includes the *entire* logical size of the current file being replicated, so if a very large file is being replicated, this number may not change for a noticeable period of time—it changes only after the current file finishes.
Files remaining	The number of files that are not replicated.
Compression ratio	The ratio of bytes transferred to network bytes transferred.
Sync'ed-as-of time	The time when the most recently replicated data on the destination was generated on the source. A value ‘unknown’ appears during replication initialization.

Table 23 on page 106 defines fields in the **CP603** REPLICATION PERFORMANCE report.

Table 23 **CP603 REPLICATION PERFORMANCE report fields**

Field	Description
rctx://1	Indicates the interval in seconds that the performance data represents. In this example, rctx is 1 second (default is 2 seconds).
Pre-Comp (KB/s)	Virtual speed over the network. The number of kilobytes per second of pre-compressed data that were transferred over the network during the specified interval.
Network (KB/s)	Data speed over the network. The number of kilobytes per second of compressed data that were transferred over the network during the specified interval.
Streams	Number of Streams used for the Context (number ranges from 1 to 9). If the Context is configured and enabled, the streams column displays 1 because one stream is reserved. If all eight streams are used, it displays 9.
Busy Reading	When the source Data Domain system is actively replicating data, the percent of that active time that is spent reading the source data.
Busy Meta	When the source Data Domain system is actively replicating data, the percent of that active time that is spent processing overhead data on behalf of the source data.
Waiting Dest.	When the source Data Domain system is waiting on an event while replicating, the percent of that time that is spent waiting on the destination Data Domain system.
Waiting Network	When the source Data Domain system is waiting on a event while replicating, the percent of that time that is spent waiting on the network.

Table 24 on page 106 defines fields in the **CP603** REPLICATION HISTORY report.

Table 24 **CP603 REPLICATION HISTORY report fields**

Field	Description
Sync-as-of Time	The time when the most recently replicated data on the destination was generated on the source. A value 'unknown' appears during replication initialization.
Pre-Comp (KB) Remaining	The amount of pre-compression data that is not replicated.
Replicated (KB) Pre-Comp	The amount of pre-compressed data that is replicated.
Replicated (KB) Network	The amount of compressed data sent over the network.

Obtaining DLm virtual tape activity: CP998 and CP999

CP998

Command Processor **CP998** retrieves statistics from a VOLSER range and writes this data to a non-labeled tape file. The **GENSTATS** program is then run to generate a report from the non-labeled tape file data. "GENSTAT1 procedure" on page 162 provides information about the **GENSTATS** utility and reports.

Submit **CP998** from the mainframe by using either of the GENSTATP, GENSTAT1, or GENSTATW PROCs.

CP998 syntax

The **CP998** command syntax using the GENSTATP PROC is:

```
//GO EXEC GENSTATP,CMD=998,UNIT=Unit1id,UNIT2=Unit2id
```

CP998 parameters

Table 25 on page 107 defines the **CP998** parameters.

Table 25 **CP998 parameters**

Parameter	Description
Unit	Unit to transport results
Unit2	Unit to transport command
VPATH	Allows filtering paths containing .vstats files. Example: VPATH='/tapelibREP/5*' limits .vstats data to that found in the folders begining with '/tapelibREP/5'
VMASK	Wildcard mask to filter .vstats files.
VDEPTH	Numeric count of maximum number of subfolders to filter .vstats files.
Startdate	STRTMM/DD/YY STRTMM (2 digit month, default 01), STRTDD=DD (2 digit day, default 01). STRTYYYY=YYYY (4 digit year, Default 2000) This optional parameter excludes statistics data preceding the date specified. Use of this parameter can result in improved performance when dealing with large amounts of statistical data.
Enddate	ENDMM/DD/YYYY ENDMM (2 digit month, default 01), ENDDD=DD (2 digit day, default 01). ENDYYYY=YYYY (4 digit year, Default 2099) This optional parameter excludes statistics data after the date specified. Use of this parameter can result in improved performance when dealing with large amounts of statistical data..

Example 1:

SLOW MOUNTS command

Sample JCL:

```
//GO1     EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),            X
//            REPORT='SLOWMOUNTS=5'
```

Sample output

```
GENSTATS VER 1.28  Z1CWRK PARMS: SLOWMOUNTS=5

--------------------------------------------------------------------------------
PATHS INCLUDED IN STATS  :

NODENAME    PATH
vte1dlm0    tapelibFLR/CEL1_FLRP_FS1
vte1dlm0    tapelibFLR/CEL1_FLRP_FS2
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
TOP 50   SLOWEST MOUNTS (SECS.):

NODENAME   TIME     VOLSER  DATE        TIME       PATH
vte1dlm0  0000.026 BG0332  2015/08/31  11:02:15  tapelibFLR/CEL1_FLRP_FS1
vte1dlm0  0000.028 BG0337  2015/09/01  11:14:09  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0  0000.029 BG0323  2015/08/28  16:59:43  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0  0000.033 BG0325  2015/08/28  17:08:15  tapelibFLR/CEL1_FLRP_FS2
vte1dlm0  0000.041 BFLGR1  2015/08/26  16:19:44  tapelibFLR/CEL1_FLRP_FS2
EMC Data Domain St
```

Example 2:

THROUGHPUT command

Sample JCL:

```
//GO2     EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),              X
//             REPORT='THROUGHPUT'
```

Sample output

```
GENSTATS VER 1.28  Z1CWRK PARMS: THROUGHPUT

---------------------------------------------------------------------------------------------------------------
-----------
PATHS INCLUDED IN STATS  :

NODENAME   PATH
vte1dlm0   tapelibFLR/CEL1_FLRP_FS1
vte1dlm0   tapelibFLR/CEL1_FLRP_FS2
---------------------------------------------------------------------------------------------------------------
-----------

---------------------------------------------------------------------------------------------------------------
-----------
THROUGHPUT IN GB/HR      :
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|YY/MM/DD|OUT  00|OUT  01|OUT  02|OUT  03|OUT  04|OUT  05|OUT  06|OUT  07|OUT  08|OUT  09|OUT  10|OUT  11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|       |OUT  12|OUT  13|OUT  14|OUT  15|OUT  16|OUT  17|OUT  18|OUT  19|OUT  20|OUT  21|OUT  22|OUT  23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|       |IN   00|IN   01|IN   02|IN   03|IN   04|IN   05|IN   06|IN   07|IN   08|IN   09|IN   10|IN   11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|       |IN   12|IN   13|IN   14|IN   15|IN   16|IN   17|IN   18|IN   19|IN   20|IN   21|IN   22|IN   23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/08/26|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000011
MAXMNT/HR=000011
….

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/09/02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.62|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
```

```
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000079
MAXMNT/HR=000054
```

Example 3:

DETAIL command

Sample JCL:

```
//GO4     EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),              X
//               REPORT='DETAIL'
```

Sample output

```
GENSTATS VER 1.28  Z1CWKB PARMS: DETAIL

        ------ Unload ----- ------ Channel ------ ---- File System ----   Duration Compression
 Volser Date       Time      4K Writes   4K Reads  4K Writes   4K Reads       Time     Percent
 ------ ---------- -------- ---------- ---------- ---------- ---------- ----------  ----------
 BG0300 2015/08/26 16:11:06 0000005121 0000000001 0000000303 0000000000 1068058565       94.1%
 BFLGR1 2015/08/26 16:19:44 0000000001 0000000000 0000000001 0000000000 1068058576       00.0%
 BFLGR1 2015/08/26 16:19:56 0000000000 0000000000 0000000001 0000000000 1068058576
 BG0301 2015/08/26 16:19:56 0000000001 0000000001 0000000001 0000000000 1068058564       00.0%
….
BG0348 2015/09/02 11:52:15 0000005121 0000000001 0000000303 0000000000 1067614753       94.1%
BG0349 2015/09/02 11:53:00 0000010241 0000000001 0000000605 0000000000 1067614708       94.1%
BG0353 2015/09/02 11:53:01 0000005121 0000000001 0000000303 0000000000 1067614755       94.1%
BG0350 2015/09/02 11:53:36 0000015361 0000000001 0000000907 0000000000 1067614672       94.1%
BG0351 2015/09/02 11:54:14 0000020481 0000000001 0000001208 0000000000 1067614634       94.2%
BG0354 2015/09/02 11:54:15 0000010241 0000000001 0000000605 0000000000 1067614726       94.1%
BG0352 2015/09/02 11:54:41 0000025601 0000000001 0000001510 0000000000 1067614607       94.2%
BG0355 2015/09/02 11:55:10 0000015361 0000000001 0000000907 0000000000 1067614707       94.1%
BG0356 2015/09/02 11:55:46 0000020481 0000000001 0000001208 0000000000 1067614709       94.2%
BG0357 2015/09/02 11:56:05 0000025601 0000000001 0000001510 0000000000 1067614716       94.2%
-------------------------------------------------------------------------------------------------------------------
---------------------
PATHS INCLUDED IN STATS  :

NODENAME   PATH
vte1dlm0   tapelibFLR/CEL1_FLRP_FS1
vte1dlm0   tapelibFLR/CEL1_FLRP_FS2
-------------------------------------------------------------------------------------------------------------------
---------------------
```

Filtering .vstats files

The VPATH, VMASK, and VDEPTH parameters can be used to filter the .vstats files that Command Processor 998 processes. These parameters are optional and either one or all of the parameters can be used. Here is a description of how these parameters can be used for this purpose:

- VPATH: Assume that we have the following .vstats files and would like to limit the .vstats files to be processed to only the .vstats files belonging in the /tapelibREP/5* folders:

  `/tapelibREP/65/.vstats`

  `/tapelibREP/53/.vstats`

```
/tapelibREP/69/.vstats
/tapelibREP/62/.vstats
/tapelibREP/64/.vstats
/tapelibREP/51/.vstats
```

Specifying **VPATH=/tapelibREP/5*** would limit the number of .vstats files to the following list:

```
/tapelibREP/53/.vstats
/tapelibREP/51/.vstats
```

- VMASK: Assume that we have the following .vstats files and would like to limit the .vstats files to be processed to only the .vstats files for vte1:

```
/tapelibBG141106/CEL1_BG141106_FS4/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F4
/tapelibBG141106/CEL1_BG141106_FS1/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F1
/tapelibBG141106/CEL1_BG141106_FS3/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F3
/tapelibBG141016/CEL1_BG141016_FS1/.vstats.vte1.tapelibBG141016.CEL1_BG141016_F1
/tapelibBG141016/CEL1_BG141016_FS1/.vstats.vte2.tapelibBG141016.CEL1_BG141016_F1
```

Specifying **VMASK=vte1.*** would limit the .vstats files to the following list:

```
/tapelibBG141106/CEL1_BG141106_FS4/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F4
/tapelibBG141106/CEL1_BG141106_FS1/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F1
/tapelibBG141106/CEL1_BG141106_FS3/.vstats.vte1.tapelibBG141106.CEL1_BG141106_F3
/tapelibBG141016/CEL1_BG141016_FS1/.vstats.vte1.tapelibBG141016.CEL1_BG141016_F1
```

- VDEPTH: The VDEPTH parameter allows .vstats files to be filtered by limiting the depth of subfolders for the .vstats files. Assume that we have the following .vstats files:

```
/tapelibDDR/DD1_DD_FS1/.snapshot/scheduled-2014-07-11-07-00/.vstats.vte1.tapelib
DDR.DD1_FS1
/tapelibDDR/DD1_DD_FS1/.snapshot/scheduled-2014-07-11-07-00/.vstats.vte2.tapelib
DDR.DD1_FS1
/tapelibDDR/DD1_DD_FS1/.snapshot/scheduled-2014-07-11-13-00/.vstats.vte1.tapelib
DDR.DD1_FS1
/tapelibDDR/DD1_DD_FS1/.vstats.vte1.tapelibDDR.DD1_FS1
/tapelibDDR/DD1_DD_FS1/.vstats.vte2.tapelibDDR.DD1_FS1
```

Specifying **VDEPTH=4** will would limit the .vstats files to the following list:

```
/tapelibDDR/DD1_DD_FS1/.vstats.vte1.tapelibDDR.DD1_FS1
/tapelibDDR/DD1_DD_FS1/.vstats.vte2.tapelibDDR.DD1_FS1
```

Considerations for DLm1000 and DLm2100 with Data Domain

For DLm1000 and DLm2100 with Data Domain, ensure the following:

- The .vstats file exists in every /tapelib filesystem.

If the .vstats file does not exist in every tapelib file system:

1. Logon to a Linux terminal session on a VTE on the DLm using a tool such as PuTTY.

   Note: You must log in as **root**.

2. Create a .vstats file in each directory in the library. Enter the following command to create a .vstats file in a single directory:

```
touch /libraryname/mountpoint/.vstats
```

Where:

libraryname is the top level qualifier identifying the virtual library; for example, /tapelib.

mountpoint is the mount point used to mount the directory where the .vstats file is to be created.

For example:

```
touch /tapelib/DISK1/.vstats
```

Note: You must repeat this procedure to create a .vstats file in each of the directories you want your Virtuent controller to log statistics. Normally, if your library has four directories, you would execute four **touch** commands so that all directories would be logging statistics.

You only need to create .vstats from one DLm accessing the library. If you have more than one DLm controller sharing the virtual library you do not need to create a .vstats from each controller. You only need to create the control files from a single controller. All controllers will recognize the presence of the .vstats file(s) and perform the required logging.

- If there are two DLm1000/DLm2100 units sharing the same Data Domain system, the DLms must have unique controller names.

  If the controller names are not unique, set them using the DLm Console:

  1. Access the DLm Console:
     a. Open a web browser.
     b. Type the Management VTE IP address: https://*<ip_address>*

where *‹ip_address›* is the address of the VTE on the customer LAN. For example: https://192.168.1.1.

2. Enter the user ID and password. The default user id is **dlmadmin** and the default password is password.
3. Click the Network menu tab.
4. Change the name of the host in the Host name field under Global Options.
5. To save the configuration and install it, click the Configurations menu tab.
6. Click the Save changes button.
7. Click the Install button which will become active when you save the configuration.

IMPORTANT

When you install a configuration change, the DLm controller automatically restarts the Virtuent virtual tape application. Do not install any configuration change without first varying the DLm emulated tape drives offline on the mainframe to avoid failure of any mainframe applications actively using a DLm tape drive.

CP999

This Command Processor retrieves statistics files from a local VTE and writes this data to a non-labeled tape file. The **GENSTATS** program is then run to generate a report from the non-labeled tape file data. "GENSTATP procedure" on page 155 provides information about the **GENSTATS** utility and reports.

Submit **CP999** from the mainframe by using either of the GENSTATP, GENSTAT1, or GENSTATW PROCs.

CP999 syntax

The **CP999** command syntax is:

```
//GO EXEC GENSTATP,CMD=999,UNIT=Unit1id,UNIT2=Unit2id
```

CP999 parameters

Table 26 on page 114 defines the CP999 parameters.

Table 26 **CP999 parameters**

Parameter	Description
Unit	Unit to transport results
Unit2	Unit to transport command
Startdate	STRTMM/DD/YY STRTMM (2 digit month, default 01), STRTDD=DD (2 digit day, default 01). STRTYYYY=YYYY (4 digit year, Default 2000) This optional parameter excludes statistics data preceding the date specified. Use of this parameter can result in improved performance when dealing with large amounts of statistical data.
Enddate	ENDMM/DD/YYYY ENDMM (2 digit month, default 01), ENDDD=DD (2 digit day, default 01). ENDYYYY=YYYY (4 digit year, Default 2099) This optional parameter excludes statistics data after the date specified. Use of this parameter can result in improved performance when dealing with large amounts of statistical data..

Example 1:

SLOW MOUNTS

Sample JCL:

```
//GO5     EXEC GENSTAT1,CMD=999,UNIT=(/F207,,,SMSHONOR),            X
//             REPORT='SLOWMOUNTS=5'
```

Sample output

```
GENSTATS VER 1.28 Z1CWK4 PARMS: SLOWMOUNTS=5


NODENAME = vte1
--------------------------------------------------------------------------------
------------------------------------------------
TOP 50  SLOWEST MOUNTS (SECS.):

NODENAME  TIME   VOLSER DATE     TIME
vte1    0000.063 DV0011 2015/07/28 11:19:16
```

```
vte1   0000.064 BFL503  2015/06/23  18:07:12
vte1   0000.065 BFL502  2015/08/04  15:54:00
vte1   0000.080 BFL501  2015/06/23  18:09:47
vte1   0000.085 BFLMT7  2015/08/25  15:17:19
--------------------------------------------------------------------------------
----------------------------------------------------
```

Example2:

THROUGHPUT

Sample JCL:

```
//GO6     EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),             X
//                REPORT='THROUGHPUT'
```

Sample output

```
GENSTATS VER 1.28  Z1CWKA PARMS: THROUGHPUT

NODENAME = vte1
-------------------------------------------------------------------------------------------------------------
---------------------
THROUGHPUT IN GB/HR      :
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|YY/MM/DD|OUT  00|OUT  01|OUT  02|OUT  03|OUT  04|OUT  05|OUT  06|OUT  07|OUT  08|OUT  09|OUT  10|OUT  11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|       |OUT  12|OUT  13|OUT  14|OUT  15|OUT  16|OUT  17|OUT  18|OUT  19|OUT  20|OUT  21|OUT  22|OUT  23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|        |IN   00|IN   01|IN   02|IN   03|IN   04|IN   05|IN   06|IN   07|IN   08|IN   09|IN   10|IN   11|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|        |IN   12|IN   13|IN   14|IN   15|IN   16|IN   17|IN   18|IN   19|IN   20|IN   21|IN   22|IN   23|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/06/23|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.02|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|15/09/02|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.62|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
|    |0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|0000.00|
+--------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
 MAXI+O/HR=00000G  TOT/DAY=00000G  OUT/DY=00000G  IN/DY=00000G  MAXO/HR=00000G  MAXI/HR=00000G  MOUNTS/DY=000091
MAXMNT/HR=000054


COUNT OF UNLOAD RECORDS        :0000966
COUNT OF UNLOADS W/OUTPUT STATS:0000515
COUNT OF UNLOADS W/INPUT STATS :0000005
MAXIMUM OUTPUT/HR             :000000.62G
MAXIMUM OUTPUT/DAY            :000000.62G
MAXIMUM INPUT/HR              :000000.02G
MAXIMUM INPUT/DAY             :000000.02G
MAXIMUM THROUGHPUT/HR         :000000.62G
MAXIMUM THROUGHPUT/DAY        :000000.62G
MAXIMUM MOUNTS/HR             :000096
MAXIMUM MOUNTS/DAY            :000213
EFFECTIVE COMPRESSION RATIO    :  16.1 TO 1
```

```
AVERAGE MOUNT RESPONSE (SECS.) :000.003
-------------------------------------------------------------------------------------------------------------------------------
---------------------
```

Example 3: DETAIL

Sample JCL

```
//GO8     EXEC GENSTAT1,CMD=998,UNIT=(/F207,,,SMSHONOR),              X
//               REPORT='DETAIL'
```

Sample output

```
GENSTATS VER 1.28  Z1CWRK PARMS: DETAIL

        ------ Unload ----- ------ Channel ------ ---- File System ----   Duration Compression
 Volser Date       Time      4K Writes   4K Reads  4K Writes   4K Reads       Time     Percent
 ------ ---------- -------- ---------- ---------- ---------- ---------- ----------  ----------
 BFL503 2015/06/23 18:07:12 0000000001 0000000000 0000000001 0000000000 1071310800       00.0%
 DV0001 2015/06/23 18:07:16 0000000001 0000000001 0000000001 0000000000 1071310796       00.0%
 BFL503 2015/06/23 18:08:22 0000000001 0000000000 0000000001 0000000000 1071310800       00.0%
 DV0002 2015/06/23 18:08:27 0000000001 0000000001 0000000001 0000000000 1071310795       00.0%
 BFL501 2015/06/23 18:09:48 0000000001 0000000000 0000000001 0000000000 1071310799       00.0%
 DV0005 2015/06/23 18:09:59 0000000001 0000000001 0000000001 0000000000 1071310789       00.0%
 BFL501 2015/06/23 18:10:00 0000000000 0000000000 0000000005 0000000000 1071310799
 DV0004 2015/06/23 18:11:33 0000005121 0000005121 0000000300 0000000000 1071310798       94.2%
 BFL499 2015/07/28 11:08:38 0000000001 0000000000 0000000001 0000000000 1069485135       00.0%
 BFL499 2015/07/28 11:08:51 0000000000 0000000000 0000000001 0000000000 1069485136
 DV0003 2015/07/28 11:08:51 0000000001 0000000001 0000000001 0000000000 1069485123       00.0%
 BFL499 2015/07/28 11:08:52 0000000000 0000000000 0000000001 0000000000 1069485136
 BFL499 2015/07/28 11:09:53 0000000001 0000000000 0000000001 0000000000 1069485135       00.0%
 DV0008 2015/07/28 11:10:05 0000000001 0000000001 0000000001 0000000000 1069485124       00.0%
 BFL499 2015/07/28 11:10:06 0000000001 0000000000 0000000001 0000000000 1069485136       00.0%
 BFL499 2015/07/28 11:10:19 0000000000 0000000000 0000000001 0000000000 1069485136
 DV0007 2015/07/28 11:10:19 0000000001 0000000001 0000000001 0000000000 1069485124       00.0%
….
BG0348 2015/09/02 11:52:15 0000005121 0000000001 0000000303 0000000000 1067614753       94.1%
BG0349 2015/09/02 11:53:00 0000010241 0000000001 0000000605 0000000000 1067614708       94.1%
BG0353 2015/09/02 11:53:01 0000005121 0000000001 0000000303 0000000000 1067614755       94.1%
BG0350 2015/09/02 11:53:36 0000015361 0000000001 0000000907 0000000000 1067614672       94.1%
BG0351 2015/09/02 11:54:14 0000020481 0000000001 0000001208 0000000000 1067614634       94.2%
BG0354 2015/09/02 11:54:15 0000010241 0000000001 0000000605 0000000000 1067614726       94.1%
BG0352 2015/09/02 11:54:41 0000025601 0000000001 0000001510 0000000000 1067614607       94.2%
BG0355 2015/09/02 11:55:10 0000015361 0000000001 0000000907 0000000000 1067614707       94.1%
BG0356 2015/09/02 11:55:46 0000020481 0000000001 0000001208 0000000000 1067614709       94.2%
BG0357 2015/09/02 11:56:05 0000025601 0000000001 0000001510 0000000000 1067614716       94.2%
BFL503 2015/09/02 11:58:32 0000000001 0000000000 0000000001 0000000000 1067614800       00.0%
BFL503 2015/09/02 11:58:44 0000000000 0000000000 0000000015 0000000000 1067614800
```

5

Command Processor Procedures

This chapter describes the procedures for **DLMCMDPR**, **DLMCMDP1**, **GENSTATP**, **GENSTAT1**, **GENSTATW/R** utilities.

Topics include:

MTL Considerations for VTE Drive Selection

When a request is made for a tape drive defined in an MTL, the Automatic Class Selection (ACS) routines select the appropriate tape storage group for the library. Allocation subsequently chooses any available drive in that library.

This is not a problem if only one VTE is defined as part of the library. However, an MTL can span multiple VTEs for performance and failover considerations. In this case, targeting a specific VTE for batch utilities is required.

The SMSHONOR parameter is now supported in all supported z/OS releases. With this parameter, a particular Unit that is part of an MTL can be specified. For example:

```
//GO     EXEC DLMCMDP1,CMD=503,UNIT=(/4305,,,SMSHONOR)
```

Note: The SMSHONOR clause is not supported for the UNIT2 parameter in the DLMCMDPR and GENSTATP PROCs. (UNIT2 is the Unit parameter that must be offline.)

Use one of the following methods to enable a batch utility to communicate with a specific VTE:

- If the MTL spans across multiple VTEs, omit a single drive from the MTL in each VTE's device group.

  For example, consider a MTL defined with two VTEs, each configured with 64 devices:

  a. In each VTE, define 63 devices as MTL=YES in the HCD. One device would be MTL=NO in the HCD.

  b. Subsequently, use demand allocation in JCL to select the specific drive address that is outside the MTL.

- Define a separate MTL for each VTE to enable VTE selection:

  a. Similar to the previous method, define only 63 devices on each VTE as part of the same MTL.

  b. For each VTE, define a separate MTL (different LIB-ID) for the remaining device, as well as a new esoteric.

  c. Use ACS routines to select the appropriate library that limits the available drive selection to that one drive.

EMC recommends that you leave this drive offline to prevent inadvertent allocation by other jobs. One way to accomplish this is to *bookend* your jobs with steps to vary the device online and offline with an operator command utility program.

DLMCMDPR, DLMCMDP1 and DLMCMD49 procedures

These procedures simplify Command Processor usage by providing clearly defined parameters to be specified along with default values.

The **DLMCMDPR** and **DLMCMDP1** procedures are identical except for the usage of DLm units. The **DLMCMDP1** procedure only uses one DLm Unit which must be online. This is the preferred procedure but is slightly slower that the **DLMCMDPR** procedure.

The **DLMCMD49** procedure is used to ping the DLm components attached to the respective VTE. It is used as a batch component testing function.

Parameters

The following are the required parameters used by the procedures:

- **UNIT**: This specifies the DLm Unit to be used to obtain a Command Processor's output.

  When using the DLMCMDPR procedure, it is important that the Unit value points to the same DLm VTE as the **UNIT2** parameter. The **UNIT2** parameter is used to send a Command Processor command to the DLm for processing. The output of the Command Processor is stored on a local folder within the VTE. As a result, the **UNIT** and **UNIT2** point to the same Virtual Tape Controller, (VTE).

  The **UNIT** parameter can be specified as an esoteric Unit name or a specific Unit ID. Here are some examples of valid specifications:

  ```
  UNIT=DLM43XX
  UNIT=/4308
  UNIT=(/4400,,,SMSHONOR)
  ```

  In the first example above, an esoteric unit name is used that specifies a range of Units from 4300-43FF. In the second example a specific Unit, (4308), is specified. In the third example, a specific unit is also selected but the unit is in a MTL. The advantage of the

esoteric unit name is that a Unit can be allocated from a range of Units. When using an esoteric unit name for the **UNIT** parameter ensure that all Units within the esoteric unit name point to the same VTE and are defined to have the same Mount Point.

Please ensure that the Unit specified or at least one Unit for an esoteric unit name is online.

- **UNIT2**: This specifies the DLm Unit to be used to send a Command Processor command to the DLm.

  The **UNIT2** parameter must specify a specific Unit and that Unit should be offline. The Unit specified for **UNIT2** must be in sync with the **UNIT** parameter.

  Note: The SMSHONOR clause is not supported for the UNIT2 parameter.

  Please refer to the above description of the **UNIT** parameter for more information.

  Note: If the Unit specified by **UNIT2** is not offline, then message DLC182I will be issued and processing will terminate with either a non-zero condition code or an abend.

- **CMD**: This parameter specifies the Command Processor to be executed. The table below lists the available Command Processors:

Table 27 **Available Command Processors**

Value	Description
499	Batch diagnostic aid for component responsiveness
501	Execute Virtual Tape Operator command
502	Collect Virtual Tape Controller Logs
503	Tape Volume Library Report
504	VNX Replication Report
601	Data Domain System Utilization Report
602	Data Domain Deduplication Report
603	Data Domain Replication Report

- **VCMD**: This parameter is required for Command Processor 501. It specifies the Virtual Tape Operator command. Please refer to the Virtual Tape Operator command reference section in the *DLm User's Guide* and CP501 in Chapter 4, "Using Command Processors," of this guide for a description of available commands.

The following are the optional parameters used by the procedures:

- **PATH**: When this parameter is not specified, the **DLMCMDP***x* procedures will locate and use the correct folder for the DLm **UNIT** parameter. If overridden, note that this parameter must point to the same VTE and mount point of the **UNIT** and **UNIT2** parameters. Please refer to the above discussion on the **UNIT** parameter for more information on this requirement.
- **CMD2**: Command Processor output is stored on the DLm with a file name that begins with **BFL** and ends with `.FLAT`. For example, the output from executing Command Processor 502, (CMD=502), will result in the output stored on the DLm as file `BFL502.FLAT`. If the **CMD2** parameter is specified, its value will be used in the file name. For example, specifying **CMD2=ABC** will result in a file named `BFLABC.FLAT`.
- **CPARMS**: This parameter allows for command parameters to be specified for the 501 and 502 Command Processors.

Note: The default **CPARMS** value for Command Processor 502:
Collection type: L0
Number of btilogs to collect: 2
The above defaults specify that 2 btilog files are to be collected.

- **TLIB**: This parameter specifies the folder on the DLm where DLm volumes are stored. When this parameter is not specified, the Tape Library folder for the PATH parameter will be used.
- **WAIT**: The z/OS job that is executing the Command Processor will wait while the Command Processor is executing on the DLm. This parameter allows for a maximum wait time to be specified. The default is 600 seconds. If the Wait state time exceeds a z/OS maximum wait time for an IO to complete, the Unit specified in the **UNIT** parameter will become Boxed. If this happens, the Unit should be brought back Online to be usable.

- **BTILIB**: This parameter specifies the load library where the DLMCMD executable is stored. It is expected that this parameter will be initially set in the DLMCMDPR, DLMCMDP1, DLMCMD49, GENSTAT1, GENSTATR, and GENSTATW PROCs when the DLm product is installed and overridden if/when a different load library is to be used.
- **REXXLIB**: This parameter specifies the library where the DLMCMD1, DLMCMD2, and DLMCMD3 REXX programs are stored. It is expected that this parameter will be initially set in the DLMCMDPR, DLMCMDP1, DLMCMD49, GENSTAT1, and GENSTATW PROCs procedures when the DLm product is installed and overridden if/when a different REXX library is to be used.

Procedure listings

DLMCMDPR/DLMCMDP1 procedure listing

The following is the listing for DLMCMDPR. This applies to both DLMCMDPR and DLMCMDP1 procedures.

```
//*
//*********************************************************************
//*                                                                   *
//*   DLMCMDPR: EMC DISK LIBRARY COMMAND PROCESSOR                    *
//*                                                                   *
//*   FUNCTION: PROCESS DISK LIBRARY COMMAND                          *
//*                                                                   *
//*  SYMBOLICS:                                                       *
//*                                                                   *
//*        CMD: COMMAND TO PROCESS                                    *
//*       CMD2: COMMAND ID FOR BFLFILE                                *
//*       WAIT: OPTIONAL WAIT TIME IN SECONDS.  DEFAULTS TO 600       *
//*       UNIT: UNIT TO TRANSPORT RESULTS                             *
//*      UNIT2: UNIT TO TRANSPORT COMMAND                             *
//*     CPARMS: OPTIONAL COMMAND SPECIFIC PARAMETERS                  *
//*       PATH: BFL FILE PATH.     DEFAULTS TO UNIT PATH              *
//*       TLIB: TAPELIB DIRECTORY. DEFAULTS TO UNIT PATH              *
//*       VCMD: SUBCOMMAND. REQUIRED FOR SOME COMMANDS                *
//*      WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS                *
//*     WTOTXT: USER Defined WTO TEXT                                 *
//*        XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012           *
//*      DEBUG: OPTIONAL DEBUG OPTION, (A, C, E, F, I, L, or N)       *
//*     BTILIB: BTI LIBRARY WHERE DLMCMD EXECUTABLE RESIDES           *
//*    REXXLIB: REXX LIBRARY FOR DLMCMD REXX MODULES                  *
//*                                                                   *
//*                                                                   *
//*    EXAMPLE:                                                       *
//*                                                                   *
//*    - COMMAND 503: TAPE LIBRARY CONTENTS.                          *
//*                                                                   *
//*      Execute command 503 using 4305 unit for DLM channel IO.      *
//*                                                                   *
//*      //GO    EXEC DLMCMDPR,CMD=503,UNIT=/4305,UNIT2=4305         *
//*                                                                   *
//*                                                                   *
//*   COPYRIGHT (C) 2012 - EMC CORPORATION                            *
//*                                                                   *
//*********************************************************************
//*
//DLMCMDPR PROC CMD='',                                                X
```

```
//              CMD2=&CMD,                                                      X
//              UNIT=,                                                          X
//              UNIT2=,                                                         X
//              CPARMS='',                                                      X
//              PATH='DEFAULT',                                                 X
//              TLIB='DEFAULT',                                                 X
//              VCMD='',                                                        X
//              WAIT=600,                                                       X
//              WTOID='DLM8888E',                                               X
//              WTOTXT='An ERROR Has Occurred in last step',                    X
//              XRC='12',                                                       X
//              DEBUG='',                                                       X
//              BTILIB=USER.DLM.LOADLIB,                                        X
//              REXXLIB=USER.DLM.REXX
//*
//* STEP 1: BUILD COMMAND SYNTAX
//*
//STEP1   EXEC PGM=IRXJCL,
//             PARM='DLMCMD1 CMD(&CMD) DEBUG(&DEBUG) FILE(&CMD2)                X
//             PATH(&PATH) VCMD(&VCMD) WAIT(&WAIT)'
//SYSEXEC  DD  DISP=SHR,DSN=&REXXLIB
//DLMCMD   DD  DSN=&CMDFILE1,DISP=(NEW,PASS),UNIT=SYSALLDA,
//             SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//DLMTOKEN DD  DSN=BFL&CMD2..FLAT,DISP=OLD,UNIT=&UNIT,LABEL=(,NL),
//             DCB=(RECFM=FB,LRECL=320,BLKSIZE=320,OPTCD=Q),
//             VOL=SER=BFL&CMD2
//SYSTSPRT DD  SYSOUT=*
//*
//* STEP 2: ADD OPTIONAL PARAMETERS TO COMMAND SYNTAX
//*
// IF  STEP1.RC  = 0 THEN
//STEP2   EXEC PGM=IRXJCL,
//             PARM='DLMCMD2 CMD(&CMD) DEBUG(&DEBUG) TLIB(&TLIB)                X
//             CPARMS(&CPARMS)'
//SYSEXEC  DD  DISP=SHR,DSN=&REXXLIB
//DLMCMDI  DD  DSN=&CMDFILE1,DISP=(OLD,DELETE)
//DLMCMDO  DD  DSN=&CMDFILE2,DISP=(NEW,PASS),UNIT=SYSALLDA,
//             SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//SYSTSPRT DD  SYSOUT=*
//   ENDIF
//*
//* STEP 3: SEND COMMAND TO BE PROCESSED
//*
// IF  STEP2.RC  = 0 THEN
//STEP3   EXEC PGM=DLMCMD,PARM='DEV=&UNIT2'
//STEPLIB  DD  DISP=SHR,DSN=&BTILIB
//DLMCMD   DD  DSN=&CMDFILE2,DISP=(OLD,PASS)
//DLMLOG   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//   ENDIF
//*
//* STEP4: OBTAIN COMMAND OUTPUT
//*
// IF  STEP3.RC  = 0 THEN
//STEP4   EXEC PGM=IRXJCL,
//             PARM='DLMCMD3 DEBUG(&DEBUG)'
//SYSEXEC  DD  DISP=SHR,DSN=&REXXLIB
//BFLFILE  DD  DSN=BFL&CMD2..FLAT,DISP=SHR,LABEL=(,NL),UNIT=&UNIT,
//             DCB=(RECFM=FB,LRECL=264,BLKSIZE=264),VOL=SER=BFL&CMD2
//DLMCMD   DD  DSN=&CMDFILE2,DISP=(OLD,DELETE)
//SYSTSIN  DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//PRINT    DD  SYSOUT=*
//   ENDIF
// IF  STEP4.RC  > 0 THEN
//* WTO ERROR MESSAGE Generator
//*************************************************
//DLMWTO   EXEC PGM=IKJEFT01,DYNAMNBR=30,
// PARM='DLMWTOM WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC  DD   DSN=&REXXLIB,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//*************************************************
//   ENDIF
```

DLMCMD49 procedure listing

The following is the listing for DLMCMD49. The **DLMCMD49** procedure is used to ping the DLm components attached to the respective VTEs.

```
//*
//*****************************************************************
//*                                                               *
//*    DLMCMD49: EMC DISK LIBRARY COMMAND PROCESSOR               *
//*                                                               *
//*    FUNCTION: This job pings DLm components attached to this VTE *
//*              Used as a batch component testing function       *
//*                                                               *
//*   SYMBOLICS:                                                  *
//*                                                               *
//*         CMD: COMMAND TO PROCESS                               *
//*        CMD2: COMMAND ID FOR BFLFILE                           *
//*        WAIT: OPTIONAL WAIT TIME IN SECONDS.  DEFAULTS TO 600  *
//*        UNIT: UNIT TO TRANSPORT RESULTS                        *
//*      CPARMS: OPTIONAL COMMAND SPECIFIC PARAMETERS             *
//*        PATH: BFL FILE PATH.      DEFAULTS TO /tapelib         *
//*        TLIB: TAPELIB DIRECTORY.  DEFAULTS TO /tapelib         *
//*        VCMD: SUBCOMMAND. REQUIRED FOR SOME COMMANDS           *
//*    TEXTTRIG: TEXT VALUE FROM CP499 to Trigger a WTO Alert     *
//*         POS: THe position of the text value in CP499 output   *
//*         AMT: The number of times to see the text to Trigger   *
//*              The Alert                                        *
//*       WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS           *
//*      WTOTXT: USER Defined WTO TEXT                            *
//*         XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012      *
//*       DEBUG: OPTIONAL DEBUG OPTION, (A, C, E, F, I, L, or N)  *
//*      BTILIB: BTI LIBRARY WHERE DLMCMD EXECUTABLE RESIDES      *
//*     REXXLIB: REXX LIBRARY FOR DLMCMD REXX MODULES             *
//*                                                               *
//*    CPARMS are:                                                *
//*    '-l all' for list all                                      *
//*    '-i xxx.xxx.xxx.xxx' ping a specific ip address            *
//*    '-c ComponentName' ping a component by name                *
//*                                                               *
//*     EXAMPLE:                                                  *
//*                                                               *
//*     - COMMAND 499: Adhoc Component List/Test                  *
//*                                                               *
//*       Execute command 499 using 4305 unit for DLM channel IO. *
//*                                                               *
//*       //GO    EXEC DLMCMD49,CMD=499,UNIT=/4305                *
//*                                                               *
//*                                                               *
//*    COPYRIGHT (C) 2012 - EMC CORPORATION                       *
//*                                                               *
//*****************************************************
//*
//DLMCMD49 PROC CMD='',                                                 X
//              CMD2=&CMD,                                              X
//              UNIT=,                                                  X
//              CPARMS='-l all',                                        X
//              PATH='DEFAULT',                                         X
//              TLIB='DEFAULT',                                         X
//              VCMD='',                                                X
//              WAIT=600,                                               X
//              TEXTTRIG='FAILED',                                      X
//              POS=1,                                                  X
//              AMT=1,                                                  X
//              WTOID='DLM8888E',                                       X
//              WTOTXT='CP499 Detected a Failure',                      X
//              XRC='12',                                               X
//              DEBUG='',                                               X
//              BTILIB=USER.DLM.LOADLIB,                                X
//              REXXLIB=USER.DLM.REXX
//*
//* STEP 1: BUILD COMMAND SYNTAX
//*
//STEP1   EXEC PGM=IRXJCL,
//             PARM='DLMCMD1 CMD(&CMD) DEBUG(&DEBUG) FILE(&CMD2)        X
//             PATH(&PATH) VCMD(&VCMD) WAIT(&WAIT)'
//SYSEXEC  DD  DISP=SHR,DSN=&REXXLIB
//DLMCMD   DD  DSN=&CMDFILE1,DISP=(NEW,PASS),UNIT=SYSALLDA,
```

```
//               SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//DLMTOKEN DD    DSN=BFL&CMD2..FLAT,DISP=OLD,UNIT=&UNIT,LABEL=(,NL),
//               DCB=(RECFM=FB,LRECL=320,BLKSIZE=320,OPTCD=Q),
//               VOL=SER=BFL&CMD2
//SYSTSPRT DD    SYSOUT=*
//*
//* STEP 2: ADD OPTIONAL PARAMETERS TO COMMAND SYNTAX
//*
//  IF  STEP1.RC  = 0 THEN
//STEP2    EXEC PGM=IRXJCL,
//               PARM='DLMCMD2 CMD(&CMD) DEBUG(&DEBUG) TLIB(&TLIB)       X
//               CPARMS(&CPARMS)'
//SYSEXEC  DD    DISP=SHR,DSN=&REXXLIB
//DLMCMDI  DD    DSN=&CMDFILE1,DISP=(OLD,DELETE)
//DLMCMDO  DD    DSN=&CMDFILE2,DISP=(NEW,PASS),UNIT=SYSALLDA,
//               SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//SYSTSPRT DD    SYSOUT=*
//    ENDIF
//*
//* STEP 3: SEND COMMAND TO BE PROCESSED
//*
//  IF  STEP2.RC  = 0 THEN
//STEP3    EXEC PGM=DLMCMD
//STEPLIB  DD    DISP=SHR,DSN=&BTILIB
//DLMCTRL  DD    UNIT=&UNIT,
//               DISP=(,DELETE),DSN=&TEMPFILE
//DLMCMD   DD    DSN=&CMDFILE2,DISP=(OLD,PASS)
//DLMLOG   DD    SYSOUT=*
//SYSUDUMP DD    SYSOUT=*
//    ENDIF
//*
//* STEP4: OBTAIN COMMAND OUTPUT
//*
//  IF  STEP3.RC  = 0 THEN
//STEP4    EXEC PGM=IRXJCL,
//               PARM='DLMCMD3 DEBUG(&DEBUG)'
//SYSEXEC  DD    DISP=SHR,DSN=&REXXLIB
//BFLFILE  DD    DSN=BFL&CMD2..FLAT,DISP=SHR,LABEL=(,NL),UNIT=&UNIT,
//               DCB=(RECFM=FB,LRECL=264,BLKSIZE=264),VOL=SER=BFL&CMD2
//DLMCMD   DD    DSN=&CMDFILE2,DISP=(OLD,DELETE)
//SYSTSIN  DD    DUMMY
//SYSTSPRT DD    SYSOUT=*
//PRINT    DD    SYSOUT=*
//    ENDIF
//* CP499 Text Parser
//  IF  STEP4.RC  = 0 THEN
//******-------------------------------------------
//STEP5    EXEC PGM=IKJEFT01,DYNAMNBR=30,
//               PARM='DLMPARS TEXTTRIG(&TEXTTRIG) POS(&POS) AMT(&AMT)    X
//               WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC  DD    DSN=&REXXLIB,DISP=SHR
//INFILE   DD    UNIT=&UNIT,VOL=SER=BFL&CMD2,DISP=OLD,LABEL=(,NL),
//       DSN=BFL&CMD2..FLAT,
//       DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//SYSTSPRT DD    SYSOUT=*
//SYSTSIN  DD    DUMMY
//*-------------------------------------------------------
//    ENDIF
```

JCL samples

The following JCL samples are listed with the CP*xxx* command descriptions:

- “Sample CP501 JCL: RUN501” on page 72
- “Sample CP502 JCL: RUN502” on page 77
- “Sample CP601 JCL: RUN601” on page 96
- “Sample CP602 JCL: RUN602” on page 99

- "Sample CP603 JCL: RUN603" on page 115

Note: You can find the JCL samples in the DLMZOS.JCL file on EMC support.emc.com. See the $INDEX member of the sample JCL library for a description of each sample.

Running GENSTATS

The **GENSTATS** reports can be created by running any of the three sets of procedures listed in Table 28 on page 126:

Table 28 **GENSTATS procedure to create reports**

Procedure Name	Sample JCL	Description
GENSTATP	GENJCL1	This procedure runs the standard **GENSTATS** Proc with standard use of 2 tapes (1 online and 1 offline).
GENSTAT1	GENJCL2	This procedure uses a single tape drive (this can be used for an MTL environment by invoking the **SMSHONOR** parameter).
GENSTATW and **GENSTATR**	GENJCL3	This breaks the **GENSTATS** procedure into a data extract procedure and a data reporting procedure (**GENSTATW**, for write the extract file and **GENSTATR** for read the extract file and create the **GENSTATS** report). This enables the user to create a single extract file and run multiple reports against it, saving time by creating the extract file only once.

GENSTATP procedure

The **GENSTATP** procedure simplifies **GENSTATS** usage by providing clearly defined parameters to be specified along with default values.

Using the **GENSTATP** procedure has the following advantages:

- Some **GENSTATS** parameters require changes to multiple places in the JCL. This can be avoided by using the **GENSTATP** procedure which also provides clearly defined values for each parameter.
- Typical JCL streams using the **GENSTATP** procedure require only 1 or 2 JCL statements.

- The **GENSTATP** procedure builds the **GENSTATS** syntax and ensures that the output received from the DLm matches the command sent to the DLm. This is done by passing a unique timestamp along with the command sent to the DLm that is validated in the output response.
- Use of the **GENSTATP** procedure allows for installation wide parameters to be specified in one place, (the **GENSTATP** procedure), instead of replicating multiple jobstreams.
- The **GENSTATP** procedure eliminates possible user errors because it simplifies Command Processors into clearly defined parameters.

```
//*
//*****************************************************
//*                                                   *
//* GENSTATS: EMC Disk Library GENSTATS               *
//*                    *                              *
//* Function: Produce statistics reports that summarize*
//*       DLm virtual tape activity.                  *
//*                                                   *
//* Symbolics:                                        *
//*                                                   *
//*         CMD: Command to process. Defaults to 998  *
//*        CMD2: COMMAND ID FOR BFLFILE               *
//*      REPORT: Report type. Defaults to THROUGHPUT   *
//*        WAIT: Wait time in seconds. Defaults to 600 *
//*        UNIT: Unit to transport command results back*
//*       UNIT2: Unit to transport command to DLm     *
//*      STRTMM: START MM - Month (default 01)        *
//*      STRTDD: START DD - Day (default 01)          *
//*    STRTYYYY: START YYYY - Year(default 2000)      *
//*       ENDMM: END MM - Month (default 01)          *
//*       ENDDD: END DD - Day (default 01)            *
//*     ENDYYYY: END YYYY - Year(default 2099)        *
//*        PATH: BFL file path. Defaults to Unit Path *
//*       VPATH: Option .vstats path filter.          *
//*       WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS*
//*      WTOTXT: USER Defined WTO TEXT                *
//*         XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012*
//*      BTILIB: BTI library where DLMCMD load module resides*
//*     REXXLIB: REXX library for DLMCMD REXX modules *
//*         DBG: Optional debug option, (A, C, E, F, I, L, or N)*
//*                                                   *
//*                                                   *
//* Examples:                                         *
//*                                                   *
//* - Command 998: GENSTATS Report.                   *
//*                                                   *
//* Execute command 998 using DLM73590 units for DLM channel IO.*
//*                                                   *
//* //GO EXEC GENSTATP,CMD=998,UNIT=DLM73590,UNIT2=4309*
//*                                                   *
//*                                                   *
//* COPYRIGHT (C) 2012 - EMC CORPORATION              *
//*                                                   *
//*****************************************************
//*
//GENSTATP PROC CMD=998,                              X
//         CMD2=&CMD,                                 X
//         REPORT='THROUGHPUT',                       X
//         UNIT=,                                     X
//         UNIT2=,                                    X
//         WAIT=600,                                  X
//         PATH='DEFAULT',                            X
//         VPATH='',                                  X
//         STRTMM='01',                               X
//         STRTDD='01',                               X
//         STRTYYYY='2000',                           X
//         ENDMM='01',                                X
```

```
//         ENDDD='01',                                          X
//         ENDYYYY='2099',                                      X
//         WTOID='DLM8888E',                                    X
//         WTOTXT='An ERROR Has Occurred in last step',         X
//         XRC='12',                                            X
//         DEBUG='',                                            X
//         BTILIB=USER.DLM.LOADLIB,                             X
//         REXXLIB=USER.DLM.REXX
//*
//* STEP 1: BUILD COMMAND SYNTAX
//*
//STEP1 EXEC PGM=IRXJCL,
//       PARM='DLMCMD1 CMD(&CMD) DEBUG(&DEBUG) FILE(&CMD2) PATH(X
//       &PATH) WAIT(&WAIT) SD(&STRTMM/&STRTDD/&STRTYYYY) ED(X
//       &ENDMM/&ENDDD/&ENDYYYY)'
//SYSEXEC DD DISP=SHR,DSN=&REXXLIB
//DLMCMD DD DSN=&CMDFILE1,DISP=(NEW,PASS),UNIT=SYSALLDA,
//       SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//DLMTOKEN DD DSN=BFL&CMD2..FLAT,DISP=OLD,UNIT=&UNIT,LABEL=(,NL),
//       DCB=(RECFM=FB,LRECL=320,BLKSIZE=320,OPTCD=Q),
//       VOL=SER=BFL&CMD2
//SYSTSPRT DD SYSOUT=*
//*
//* STEP 2: ADD OPTIONAL PARAMETERS TO COMMAND SYNTAX
//*
// IF STEP1.RC = 0 THEN
//STEP2 EXEC PGM=IRXJCL,
//       PARM='DLMCMD2 CMD(&CMD) DEBUG(&DEBUG) VPATH(&VPATH)'
//SYSEXEC DD DISP=SHR,DSN=&REXXLIB
//DLMCMDI DD DSN=&CMDFILE1,DISP=(OLD,DELETE)
//DLMCMDO DD DSN=&CMDFILE2,DISP=(NEW,PASS),UNIT=SYSALLDA,
//       SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//SYSTSPRT DD SYSOUT=*
// ENDIF
//*
//* STEP 3: SEND COMMAND TO BE PROCESSED
//*
// IF STEP2.RC = 0 THEN
//STEP3 EXEC PGM=DLMCMD,PARM='DEV=&UNIT2'
//STEPLIB DD DISP=SHR,DSN=&BTILIB
//DLMCMD DD DSN=&CMDFILE2,DISP=(OLD,PASS)
//DLMLOG DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// ENDIF
//*
//* STEP4: OBTAIN COMMAND OUTPUT
//*
// IF STEP3.RC < 8 THEN
//STEP4 EXEC PGM=IRXJCL,
//       PARM='DLMCMD3 DEBUG(&DEBUG)'
//SYSEXEC DD DISP=SHR,DSN=&REXXLIB
//BFLFILE DD DSN=BFL&CMD2..FLAT,DISP=SHR,LABEL=(,NL),UNIT=&UNIT,
//       DCB=(RECFM=FB,LRECL=320,BLKSIZE=320),VOL=SER=BFL&CMD2
//DLMCMD DD DSN=&CMDFILE2,DISP=(OLD,DELETE)
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//PRINT DD DSN=&STATFILE,DISP=(,PASS),SPACE=(CYL,(50,50)),
//       UNIT=SYSDA,DCB=(LRECL=320,RECFM=FB,BLKSIZE=3200)
//DLMLOG DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//* RUN GENSTATS REPORT
//*
//GO EXEC PGM=GENSTATS,PARM='&REPORT'
//STEPLIB DD DISP=SHR,DSN=&BTILIB
//GENIN DD DSN=&STATFILE,DISP=(OLD,DELETE)
//SORTIN DD UNIT=SYSDA,SPACE=(CYL,(50,50))
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(50,50))
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
// ELSE
//* WTO ERROR MESSAGE Generator
//*****************************************************
//DLMWTO EXEC PGM=IKJEFT01,DYNAMNBR=30,
// PARM='DLMWTOM WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC DD DSN=&REXXLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY
//*****************************************************
// ENDIF
```

Required parameters

Here is a description of the parameters used by **GENSTATP**:

- **UNIT**: This specifies the DLm Unit to be used to obtain the **GENSTATS** work file that is read from the DLm.

  It is important that the Unit value is in sync with UNIT2 and PATH parameters as follows:

  - It points to the same DLm VTE as the UNIT2 parameter.The UNIT2 parameter is used to send a GENSTATS request to the DLm for processing. The GENSTATS work file is stored on a local folder within the VTE. As a result, the UNIT and UNIT2 point to the same Virtual Tape Controller, (VTE).
  - The UNIT parameter can be specified as an esoteric Unit name or a specific Unit ID. Here are some examples of valid specifications:

    ```
    UNIT=DLM43XX
    UNIT=/4308
    UNIT=(4300,,,SMSHONOR)
    ```

    In the first example above, an esoteric unit name is used that specifies a range of Units from 4300-43FF. In the second example a specific Unit, (4308), is specified. In the third example, a specific unit was also specified, however that unit was in an MTL. The advantage of the esoteric unit name is that a Unit can be allocated from a range of Units.

    When using an esoteric unit name for the UNIT parameter ensure that all Units within the esoteric unit name point to the same VTE and are defined to have the same Mount Point.

  - Ensure that the Unit specified or at least one Unit for an esoteric unit name is online.

- UNIT2: This specifies the DLm Unit to be used to send a GENSTATS request to the DLm.

  The UNIT2 parameter must specify a specific Unit and that Unit should be offline. The Unit specified for UNIT2 must be in sync with the UNIT parameter. Please refer to the above description of the UNIT parameter for more information.

  Note: The SMSHONOR clause is not supported for the UNIT2 parameter.

Note: If the Unit specified by UNIT2 is not Offline, then message GEN183I will be issued and processing will terminate with either a non-zero condition code or an abend.

- **CMD**: This parameter specifies the type of **GENSTATS** report to be generated. Available **GENSTATS** requests are listed in Table 29 on page 130.

Table 29 **Available GENSTATS requests**

Value	Description
998	Statistics report of the entire DLm volume inventory. **Note:** Optional parameters can be specified to filter the statistics data.
999	Statistics report from a local VTE

Optional parameters

- **REPORT**: This parameter specifies the type of **GENSTATS** Report requested. Valid values are THROUGHPUT, MAXTAPES, SLOWMOUNTS, DETAIL, and STILLINUSE. If not specified, the default is THROUGHPUT. Please refer to the description of Report Types in the **"GENSTAT1 procedure"** section for more information on this parameter.
- **CMD2**: The GENSTATS work file is stored on the DLm with a file name that begins with **BFL** and ends with .FLAT. For example, the output from executing **GENSTATS** Command 998, (CMD=998), will result in the output stored on the DLm as file **BFL998.FLAT**. If the **CMD2** parameter is specified, its value will be used in the file name. For example, specifying **CMD2=ABC** results in a file named **BFLABC.FLAT**.
- **STRTMM/DD/YYYY**: This parameter provides the ability to exclude **GENSTATS** data prior to a given start date. This parameter is in the format of Month, Day, Year. I.e. MM/DD/YYYY.

  Example:

```
STRTMM='11'
STRTDD='15'
STRTYYYY='2013'
```

- **ENDMM/DD/YYYY**: This parameter provides the ability to exclude **GENSTATS** data to a given end date. This parameter is in the format of Month, Day, Year. I.e. MM/DD/YYYY.

  Example:

  ```
  ENDMM='12'
  ENDDD='03'
  ENDYYYY='2013
  ```

- **WAIT**: The z/OS job that is executing the Command Processor will wait while the Command Processor is executing on the DLm. This parameter allows for a maximum wait time to be specified. The default is 600 seconds. If the Wait state time exceeds a z/OS maximum wait time for an IO to complete, the Unit specified in the **UNIT** parameter will become Boxed. If this happens, the Unit should be brought back Online to be usable.

- **BTILIB**: This parameter specifies the load library where the DLMCMD executable is stored. It is expected that this parameter will be initially set in the GENSTATP, GENSTAT1, GENSTATW, and GENSTATR PROCs when the DLm product is installed and overridden if/when a different load library is to be used.

- **REXXLIB**: This parameter specifies the library where the DLMCMD1, DLMCMD2, and DLMCMD3 REXX programs are stored. It is expected that this parameter will be initially set in the GENSTATP, GENSTAT1, GENSTATW, and GENSTATR PROCs when the DLm product is installed and overridden if/when a different REXX library is to be used.

There is a step at the end of the PROC which allows the user to create a customized WTO and MSGID, Error Message Text, and Return Code between 0000 - 9999.

- **WTOID**: is the message id completely up to the users discretion. The default value is 'DLM8888E'.

- **WTOTXT**: is the text of the WTO message and is completely at the users discretion. The default value is 'An ERROR Has Occurred in last step'.

- **XRC**: is the completion code desired for the WTO step between 0000-9999. The default value is '0012'.

- **VMASK**: This filter allows you to specify a mask for the vstats files. For example, on the following statement:

  my @vstat_list=`find $FindVPath -group vtape -name .vstats.$VMask 2› /tmp/err998.$pid | sort -u `;

  The VMASK filter used for the $VMask variable above defaults to '*'. Users can specify a different mask.

- **VDEPTH**: This filter provides the ability to specify a maximum depth of folders. For example, setting this value to 4 will only allow.vstats files that are direct subfolders of the /tapelib folder.

GENSTAT1 procedure

The **GENSTAT1** utility enables the user to generate statics reports that summarize DLM virtual tape activity. The GENSTAT1 PROC is identical to the GENSTATP PROC, except it only requires one DLm unit, which must be online.

Run the **GENSTATS** report using a single DLm unit during execution.

Note: GENSTAT1 has the same parameters and creates the same outputs as **GENSTATP**. The only difference is that it uses a single UNIT parameter can be specified.

```
//*
//*****************************************************
//*                                                   *
//*   GENSTAT1: EMC Disk Library GENSTATS             *
//*                                                   *
//* Function: Produce statistics reports that summarize*
//*           DLm virtual tape activity.              *
//*           Used for single unit, MTL and, SMSHONOR support*
//*                                                   *
//* Symbolics:                                        *
//*                                                   *
//*        CMD: Command to process. Defaults to 998   *
//*       CMD2: COMMAND ID FOR BFLFILE                *
//*     REPORT: Report type. Defaults to THROUGHPUT    *
//*       WAIT: Wait time in seconds. Defaults to 600 *
//*       UNIT: Unit to transport command results back*
//*       PATH: BFL file path. Defaults to Unit Path  *
//*      VPATH: Option .vstats path filter.           *
//*     STRTMM: START MM - Month (default 01)         *
//*     STRTDD: START DD - Day (default 01)           *
//*   STRTYYYY: START YYYY - Year(default 2000)       *
//*      ENDMM: END MM - Month (default 01)           *
//*      ENDDD: END DD - Day (default 01)             *
//*    ENDYYYY: END YYYY - Year(default 2099)         *
//*      WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS*
//*     WTOTXT: USER Defined WTO TEXT                 *
//*        XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012*
//*     BTILIB: BTI library where DLMCMD load module resides*
//*    REXXLIB: REXX library for DLMCMD REXX modules  *
//*        DBG: Optional debug option, (A, C, E, F, I, L, or N)*
//* *
//* *
```

```
//* NOTE:                                                                *
//* THIS SAMPLE SINGLE UNIT Command Processor, THIS IS ALSO USED         *
//* MTL SMS MANAGED UNITS - SMSHONOR WILL ALLOW CAUSE THE UNIT           *
//* ALLOCATION TO BE HONORED (INTRODUCED IN ZOS 1.11)                    *
//*                                                                      *
//* Examples:                                                            *
//*                                                                      *
//* - Command 998: GENSTATS Report.                                      *
//*                                                                      *
//*   Execute command 998 using DLM73590 units for DLM channel IO.       *
//*                                                                      *
//*   //GO EXEC GENSTAT1,CMD=998,UNIT=/XXXX                            *
//*   //GO EXEC GENSTAT1,CMD=998,UNIT=(/XXXX,,,SMSHONOR)               *
//*                                                                      *
//*                                                                      *
//* COPYRIGHT (C) 2012 - EMC CORPORATION                                 *
//*                                                                      *
//*********************************************************
//*
//GENSTAT1 PROC CMD=998,                                                 X
//              CMD2=&CMD,                                               X
//              REPORT='THROUGHPUT',                                     X
//              UNIT=,                                                   X
//              WAIT=600,                                                X
//              PATH='DEFAULT',                                          X
//              VPATH='',                                                X
//              STRTMM='01',                                             X
//              STRTDD='01',                                             X
//              STRTYYYY='2000',                                         X
//              ENDMM='01',                                              X
//              ENDDD='01',                                              X
//              ENDYYYY='2099',                                          X
//              WTOID='DLM8888E',                                        X
//              WTOTXT='An ERROR Has Occurred in last step',             X
//              XRC='12',                                                X
//              DEBUG='',                                                X
//              BTILIB=USER.DLM.LOADLIB,                                 X
//              REXXLIB=USER.DLM.REXX
//*
//* STEP 1: BUILD COMMAND SYNTAX
//*
//STEP1 EXEC PGM=IRXJCL,
//              PARM='DLMCMD1 CMD(&CMD) DEBUG(&DEBUG) FILE(&CMD2)    PATH(X
//              &PATH) WAIT(&WAIT) SD(&STRTMM/&STRTDD/&STRTYYYY)       ED(X
//              &ENDMM/&ENDDD/&ENDYYYY)'
//SYSEXEC DD DISP=SHR,DSN=&REXXLIB
//DLMCMD DD DSN=&CMDFILE1,DISP=(NEW,PASS),UNIT=SYSALLDA,
//              SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//DLMTOKEN DD DSN=BFL&CMD2..FLAT,DISP=OLD,UNIT=&UNIT,LABEL=(,NL),
//              DCB=(RECFM=FB,LRECL=320,BLKSIZE=320,OPTCD=Q),
//              VOL=SER=BFL&CMD2
//SYSTSPRT DD SYSOUT=*
//*
//* STEP 2: ADD OPTIONAL PARAMETERS TO COMMAND SYNTAX
//*
// IF  STEP1.RC  = 0 THEN
//STEP2   EXEC PGM=IRXJCL,
//              PARM='DLMCMD2 CMD(&CMD) DEBUG(&DEBUG) VPATH(&VPATH)'
//SYSEXEC  DD  DISP=SHR,DSN=&REXXLIB
//DLMCMDI  DD  DSN=&CMDFILE1,DISP=(OLD,DELETE)
//DLMCMDO  DD  DSN=&CMDFILE2,DISP=(NEW,PASS),UNIT=SYSALLDA,
//              SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//SYSTSPRT DD  SYSOUT=*
//   ENDIF
//*
//* STEP 3: SEND COMMAND TO BE PROCESSED
//*
// IF STEP2.RC = 0 THEN
//STEP3   EXEC PGM=DLMCMD
//STEPLIB  DD  DISP=SHR,DSN=&BTILIB
//DLMCTRL  DD  UNIT=&UNIT,
//              DISP=(,DELETE),DSN=&TEMPFILE
//DLMCMD   DD  DSN=&CMDFILE2,DISP=(OLD,PASS)
//DLMLOG   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
// ENDIF
//*
//* STEP4: OBTAIN COMMAND OUTPUT
//*
// IF STEP3.RC < 8 THEN
```

```
//STEP4    EXEC PGM=IRXJCL,
//              PARM='DLMCMD3 DEBUG(&DEBUG)'
//SYSEXEC  DD   DISP=SHR,DSN=&REXXLIB
//BFLFILE  DD   DSN=BFL&CMD2..FLAT,DISP=SHR,LABEL=(,NL),UNIT=&UNIT,
//              DCB=(RECFM=FB,LRECL=320,BLKSIZE=320),VOL=SER=BFL&CMD2
//DLMCMD   DD   DSN=&CMDFILE2,DISP=(OLD,DELETE)
//SYSTSIN  DD   DUMMY
//SYSTSPRT DD   SYSOUT=*
//PRINT    DD   DSN=&STATFILE,DISP=(,PASS),SPACE=(CYL,(50,50)),
//              UNIT=SYSDA,DCB=(LRECL=320,RECFM=FB,BLKSIZE=3200)
//DLMLOG   DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//*
//* RUN GENSTATS REPORT
//*
//GO       EXEC PGM=GENSTATS,PARM='&REPORT'
//STEPLIB  DD   DISP=SHR,DSN=&BTILIB
//GENIN    DD   DSN=&STATFILE,DISP=(OLD,DELETE)
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(50,50))
//SORTOUT  DD   UNIT=SYSDA,SPACE=(CYL,(50,50))
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//    ELSE
//* WTO ERROR MESSAGE Generator
//*****************************************************
//DLMWTO   EXEC PGM=IKJEFT01,DYNAMNBR=30,
// PARM='DLMWTOM WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC  DD    DSN=&REXXLIB,DISP=SHR
//SYSTSPRT DD    SYSOUT =*
//SYSTSIN  DD    DUMMY
//*****************************************************
//    ENDIF
```

Example

The below example generates a Throughput report with a single unit:

```
//JOBNAME  JOB (MYJOB)
//*
//*  NOTE:
//* THIS SAMPLE SINGLE UNIT Command Processor, THIS IS //* ALSO USED
//* MTL SMS MANAGED UNITS - SMSHONOR WILL ALLOW CAUSE
//* THE UNIT ALLOCATION TO BE HONORED (INTRODUCED IN ZOS //* 1.11)
//*
//GO1      EXEC GENSTAT1,CMD=998,UNIT=(/2100,,,SMSHONOR),
//              REPORT='THROUGHPUT'
//*GO1     EXEC GENSTAT1,CMD=998,UNIT=/2100,
//*            REPORT='THROUGHPUT'
```

Note: All **GENTATS** report outputs are the same and consistent for all versions of the **GENSTAT** procedures.

Handling large quantities of statistics

The GENSTATP, GENSTAT1, and GENSTATW PROCs write DLm statistics information to a temporary disk dataset. This dataset has the following SPACE parameter:

SPACE=(CYL,(50,50))

When processing large amounts of statistics information, either of the following actions must be taken:

- The space parameter must be overridden.
- The statistics data must be redirected to a DLm tape volume.

The following is an examples of redirecting the intermediate statistics file to a DLm tape volume:

```
//GO1      EXEC GENSTAT1,CMD=998,UNIT=/F200,
//              REPORT='THROUGHPUT'
//STEP4.PRINT  DD DSN=&STATFILE,UNIT=/F201,DISP=(NEW,PASS)
```

In the example above, the STEP4.PRINT DD is being overridden to store the temporary dataset called &STATFILE on a DLm tape volume using the DLm unit F201.

Additionally, the STEP4.PRINT DD can be overridden to allocate more DASD for the statistics dataset.

GENSTATW/R procedure

The **GENSTATS** processing can be separated using the **GENSTATW** and **GENSTATR PROCS** to form an extract and process set of procedures.

The **GENSTATW** and **GENSTATR** procedures give the user the flexibility to create the data needed to generate multiple reports concurrently. This saves the user's time, space, and cycles. For instance, when this is run as a single process, each report is considered as a single jobstream, and the statistics data had to be created each time. The GENSTATW and GENSTATR PROCs procedures enable you to extract the statistics data only once.

GENSTATW uses the same parameters as **GENSTATP** with the following additions:

- **RPTDSN** - This is the name of the extract report dataset that is passed to the **GENSTATR** for processing and report creation.
- **UNIT3**: It is used to store the report extract dataset to be used by **GENSTATR**.
- **DISP1=** This is the report dataset extract file disposition. The default is (NEW,CATLG,DELETE).

GENSTATW procedure

Use the **GENSTATW** procedure to write the statistics file.

```
//*
//*************************************************
```

```
//*                                                                     *
//*    GENSTATW: EMC Disk Library GENSTATS                              *
//*                                                                     *
//*    Function: Produce statistics report dataset in the DLMCMD3       *
//*              Step in the PRINT DD file. The will be passed to       *
//*              GENSTATR Proc that will produce the reports that       *
//*              will summarize the DLm virtual tape activity.          *
//*                                                                     *
//*   Symbolics:                                                        *
//*                                                                     *
//*         CMD: Command to process.  Defaults to 998                   *
//*        CMD2: COMMAND ID FOR BFLFILE                                 *
//*      RPTDSN: Report DSN output    No defaults                       *
//*        WAIT: Wait time in seconds.  Defaults to 600                 *
//*        UNIT: Unit to transport command results back                 *
//*       UNIT2: Unit to transport command to DLm                       *
//*       UNIT3: Unit to write report file                              *
//*       DISP1: The disposition of the RPTDSN Default (,CATLG,DELETE)  *
//*        PATH: BFL file path. Defaults to Unit Path                   *
//*       VPATH: Option .vstats path filter.                            *
//*      STRTMM: START MM - Month (default 01)                          *
//*      STRTDD: START DD - Day   (default 01)                          *
//*    STRTYYYY: START YYYY - Year(default 2000)                        *
//*       ENDMM: END   MM - Month (default 01)                          *
//*       ENDDD: END   DD - Day   (default 01)                          *
//*     ENDYYYY: END   YYYY - Year(default 2099)                        *
//*       WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS                 *
//*      WTOTXT: USER Defined WTO TEXT                                  *
//*         XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012            *
//*      BTILIB: BTI library where DLMCMD load module resides           *
//*     REXXLIB: REXX library for DLMCMD REXX modules                   *
//*         DBG: Optional debug option, (A, C, E, F, I, L, or N)        *
//*                                                                     *
//*    Function: Produce statistics report dataset in the DLMCMD3       *
//*              Step in the PRINT DD file. The will be passed to       *
//*              GENSTATR Proc that will produce the reports that       *
//*              will summerize the DLm virtual tape activity.          *
//*                                                                     *
//*    Examples:                                                        *
//*                                                                     *
//*    - Command 998: GENSTATS Report.                                  *
//*                                                                     *
//*      Execute command 998 using DLM73590 units for DLM channel IO.   *
//*                                                                     *
//*      //GO     EXEC GENSTATW,CMD=998,UNIT=DLM73590,UNIT2=4309,       *
//*      //             UNIT3='/4302',RPTDSN=HLQ.PASS.OURTPUT           *
//*                                                                     *
//*                                                                     *
//*    COPYRIGHT (C) 2012 - EMC CORPORATION                             *
//*                                                                     *
//*********************************************************
//*
//GENSTATW PROC CMD=998,                                                X
//             CMD2=&CMD,                                               X
//             UNIT=,                                                   X
//             UNIT2=,                                                  X
//             UNIT3=,                                                  X
//             WAIT=600,                                                X
//             PATH='DEFAULT',                                          X
//             VPATH='',                                                X
//             STRTMM='01',                                             X
//             STRTDD='01',                                             X
//             STRTYYYY='2000',                                         X
//             ENDMM='01',                                              X
//             ENDDD='01',                                              X
//             ENDYYYY='2099',                                          X
//             WTOID='DLM8888E',                                        X
//             WTOTXT='An ERROR Has Occurred in last step',             X
//             XRC='12',                                                X
//             DEBUG='',                                                X
//             RPTDSN=,                                                 X
//             DISP1='(,CATLG,DELETE)',                                 X
//             BTILIB=USER.DLM.LOADLIB,                                 X
//             REXXLIB=USER.DLM.REXX
//*
//* STEP 1: BUILD COMMAND SYNTAX
//*
//STEP1    EXEC PGM=IRXJCL,
//              PARM='DLMCMD1 CMD(&CMD) DEBUG(&DEBUG) FILE(&CMD2)  PATH(X
//              &PATH) WAIT(&WAIT) SD(&STRTMM/&STRTDD/&STRTYYYY)     ED(X
```

```
//                &ENDMM/&ENDDD/&ENDYYYY)'
//SYSEXEC  DD   DISP=SHR,DSN=&REXXLIB
//DLMCMD   DD   DSN=&CMDFILE1,DISP=(NEW,PASS),UNIT=SYSALLDA,
//                SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//DLMTOKEN DD   DSN=BFL&CMD2..FLAT,DISP=OLD,UNIT=&UNIT,LABEL=(,NL),
//                DCB=(RECFM=FB,LRECL=320,BLKSIZE=320,OPTCD=Q),
//                VOL=SER=BFL&CMD2
//SYSTSPRT DD   SYSOUT=*
//*
//* STEP 2: ADD OPTIONAL PARAMETERS TO COMMAND SYNTAX
//*
// IF  STEP1.RC  = 0 THEN
//STEP2    EXEC PGM=IRXJCL,
//                PARM='DLMCMD2 CMD(&CMD) DEBUG(&DEBUG) VPATH(&VPATH)'
//SYSEXEC  DD   DISP=SHR,DSN=&REXXLIB
//DLMCMDI  DD   DSN=&CMDFILE1,DISP=(OLD,DELETE)
//DLMCMDO  DD   DSN=&CMDFILE2,DISP=(NEW,PASS),UNIT=SYSALLDA,
//                SPACE=(TRK,(1)),DCB=(LRECL=160,RECFM=FB,BLKSIZE=3200)
//SYSTSPRT DD   SYSOUT=*
//   ENDIF
//*
//* STEP 3: SEND COMMAND TO BE PROCESSED
//*
// IF STEP2.RC = 0 THEN
//STEP3    EXEC PGM=DLMCMD,PARM='DEV=&UNIT2'
//STEPLIB  DD   DISP=SHR,DSN=&BTILIB
//DLMCMD   DD   DSN=&CMDFILE2,DISP=(OLD,PASS)
//DLMLOG   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//  ENDIF
//*
//* STEP 4: Command Processed and Store Genstats data in PRINT
//*
// IF STEP3.RC < 8 THEN
//STEP4    EXEC PGM=IRXJCL,
//                PARM='DLMCMD3 DEBUG(&DEBUG)'
//SYSEXEC  DD   DISP=SHR,DSN=&REXXLIB
//BFLFILE  DD   DSN=BFL&CMD2..FLAT,DISP=SHR,LABEL=(,NL),UNIT=&UNIT,
//                DCB=(RECFM=FB,LRECL=320,BLKSIZE=320),VOL=SER=BFL&CMD2
//DLMCMD   DD   DSN=&CMDFILE2,DISP=(OLD,DELETE)
//SYSTSIN  DD   DUMMY
//SYSTSPRT DD   SYSOUT=*
//PRINT    DD   DSN=&RPTDSN,DISP=&DISP1,
//                UNIT=&UNIT3,DCB=(LRECL=320,RECFM=FB,BLKSIZE=3200)
//DLMLOG   DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//*
//   ELSE
//* WTO ERROR MESSAGE Generator
//*****************************************************
//DLMWTO   EXEC PGM=IKJEFT01,DYNAMNBR=30,
// PARM='DLMWTOM WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC  DD   DSN=&REXXLIB,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//*****************************************************
//   ENDIF
```

Handling large quantities of statistics

The GENSTATW PROC writes DLm statistics information to a temporary disk dataset. This dataset has the following SPACE parameter:

SPACE=(CYL,(50,50))

When processing large amounts of statistics information, either of the following actions must be taken:

- The space parameter must be overridden.

- The statistics data must be redirected to a DLm tape volume.

The following is an examples of redirecting the intermediate statistics file to a DLm tape volume:

```
//GO1WR  EXEC GENSTATW,CMD=998,UNIT=/F200,UNIT2='F203',
//        UNIT3='/F201',RPTDSN=RGREEN7.D150210.STATFILE
```

In the example above, the UNIT3 parameter is used to specify a DLm unit to store the RGREEN7.D140210.STATFILE dataset on a DLm tape volume.

GENSTATR procedure

The **GENSTATR** procedure is used to report on the data extracted. The additional parameters are:

- **REPORT** - The **GENSTATS** report you wish to run. The default is THROUGHPUT.
- **RPTDSN** - The extract report dataset created by **GENSTATW**.
- **UNIT3** - The same unit that the report extract was created on in the **GENSTATW** run.
- **DISP1** - Disposition for the report extract dataset. The default is SHR.

```
//*
//*******************************************************
//*                                                     *
//*   GENSTATR: EMC Disk Library GENSTATS               *
//*                                                     *
//*   Function: Produce statistics reports that summarize*
//*             DLm virtual tape activity. The input file to the*
//*             GENSTATS 'GO' step GENIN DD was created by a prior*
//*             run of the GENSTATW proc.               *
//*                                                     *
//*  Symbolics:                                         *
//*                                                     *
//*      DISP1: Disposition of RPTDSN Defaults to SHR   *
//*      UNIT3: Unit to read  report file               *
//*     REPORT: Report type.         Defaults to THROUGHPUT*
//*     RPTDSN: Input file.          No default        *
//*      WTOID: USER DEFINED WTO MSGID 7-8 CHARACHTERS  *
//*     WTOTXT: USER Defined WTO TEXT                   *
//*        XRC: COMP CODE FOR WTO STEP, 0-9999 DEFAULT 0012*
//*     BTILIB: BTI library where DLMCMD load module resides*
//*    REXXLIB: REXX library for DLMCMD REXX modules    *
//*                                                     *
//*                                                     *
//*   Examples:                                         *
//*                                                     *
//*   - Command 998: GENSTATS Report.                   *
//*                                                     *
//*     Execute Thropughput Report                      *
//*                                                     *
//*     //GO1RE1  EXEC GENSTATR,RPTDSN=WRAFFL0.PASS.OUTPUT,*
//*     //             REPORT='THROUGHPUT',UNIT3='/2101'  *
//*                                                     *
```

```
//*                                                                     *
//*   COPYRIGHT (C) 2012 - EMC CORPORATION                              *
//*                                                                     *
//*********************************************************
//*
//GENSTATR PROC REPORT='THROUGHPUT',                                     X
//             RPTDSN=,                                                  X
//             WTOID='DLM8888E',                                         X
//             WTOTXT='An ERROR Has Occurred in last step',              X
//             XRC='12',                                                 X
//             UNIT3=,                                                   X
//             DISP1=SHR,                                                X
//             BTILIB=USER.DLM.LOADLIB,                                  X
//             REXXLIB=USER.DLM.REXX
//*
//* RUN GENSTATS REPORT
//*
//GO       EXEC PGM=GENSTATS,PARM='&REPORT'
//STEPLIB   DD  DISP=SHR,DSN=&BTILIB
//GENIN     DD  DSN=&RPTDSN,DISP=&DISP1,UNIT=&UNIT3
//SORTIN    DD  UNIT=SYSDA,SPACE=(CYL,(50,50))
//SORTOUT   DD  UNIT=SYSDA,SPACE=(CYL,(50,50))
//SYSPRINT DD   SYSOUT=*
//SYSOUT    DD  SYSOUT=*
// IF GO.RC > 0 THEN
//* WTO ERROR MESSAGE Generator
//*********************************************************
//DLMWTO    EXEC PGM=IKJEFT01,DYNAMNBR=30,
// PARM='DLMWTOM WTOID(&WTOID) WTOTXT(&WTOTXT) XRC(&XRC)'
//SYSEXEC  DD    DSN=&REXXLIB,DISP=SHR
//SYSTSPRT DD    SYSOUT=*
//SYSTSIN  DD    DUMMY
//*********************************************************
//   ENDIF
```

Examples

The below example is used to create the **GENSTATW** data in dataset HLQ.FILEPASS.OUTPUT and read to create a THROUGHPUT, MAXTAPES, and SLOWMOUNTS **GENSTATS** reports with the **GENSTATR**.

```
//JOBNAME  JOB (MYJOB)
//*
//GO1WR    EXEC GENSTATW,CMD=998,UNIT=/2100,UNIT2='210D',
//              UNIT3='/2101',RPTDSN=HLQ.FILEPASS.OUTPUT'
//GO1RE1   EXEC GENSTATR,RPTDSN=HLQ.FILEPASS.OUTPUT,
//                REPORT='THROUGHPUT',UNIT3='/2101'
//GO1RE2   EXEC GENSTATR,RPTDSN=HLQ.FILEPASS.OUTPUT,
//              REPORT='MAXTAPES',UNIT3='/2101'
//GO1RE3   EXEC GENSTATR,RPTDSN=HLQ.FILEPASS.OUTPUT,
//              REPORT='SLOWMOUNTS',UNIT3='/2101'
```

Note: All GENSTATS report outputs are the same and consistent for all versions of the GENSTAT procedures

6

Command Processor SAMPLIBC Members

This chapter provides details about the Command Processors **SAMPLIBC** members.

Topics include:

SAMPLIBC sample JCL members

The installed **SAMPLIBC** PDS contains sample JCL and PROCs for Command Processor usage. Some of these members have already been discussed in the Chapter 5, "Command Processor Procedures." The member $INDEXC contains an index of the **SAMPLIBC** members with a description of their use.

After installation, the PROCs should either be moved to a common PROCLIB or should have JCLLIB statement placed in each Command Processor Job Stream. Below is a sample JCL statement pointing to the PDS containing the installed Command Processor procedures:

```
//LIB JCLLIB ORDER=EMC.CP400.PROCLIB
```

After placing it in the PROCs in the common PROCLIB, there are common parameter replacements that help to avoid specifying them in individual Job Streams. Below is a list of the possible replacements:

- **BTILIB:** Installed Link Library containing the DLm utilities such as **DLMCMD**.
- **REXXLIB:** Installed REXX Library containing the REXX programs used by Command Processors.

Below is a list of the Command Processor Procedures:

- **DLMCMDPR**: Sample procedure to run DLm Command Processors.
- **DLMCMDP1**: Sample procedure to run DLm Command Processors for single unit use.
- **DLMCMD49**: Sample procedure to run DLm Command Processor 499. **CP499** is used to list / ping components attached to the VTE giving the ability for batch controlled additional diagnostic tool.
- **GENSTATP**: Sample procedure to run DLm **GENSTATS** reports.
- **GENSTAT1**: Sample procedure to run DLm **GENSTAT1** reports for single unit use.
- **GENSTATW**: Sample procedure to extract DLm statistics data.
- **GENSTATR**: Sample procedure to execute the GENSTATS DLm z/OS utility.
- **DLMTRPT**: Sample procedure to run Cataloged DLm Library Report

Command Processor sample JCL streams

After installation, the following **SAMPLIBC** members should be copied to another PDS for customization. The following changes must be considered:

- Include JCLLIB statement to point to the library containing the Command Processor Procedures.

Example

```
//LIB     JCLLIB  ORDER=EMC.CP400.PROCLIB
```

- Include JOB Card that conforms to the installation standards.
- Specify any procedure specific parameters. Refer to the Chapter 4, "Using Command Processors" and Chapter 5, "Command Processor Procedures" for further information.

Below is a list of the Command Processor Sample JCL members contained in the **SAMPLIBC** library:

- **DLMCMDJ1**: Sample JCL to run DLm Command Processors using **DLMCMDPR**.
- **DLMCMDJ2**: Sample JCL to run DLm Command Processors using **IEBGENER**.
- **DLMCMDJ3**: Sample JCL to run DLm Command Processors using **DLMCMDP1** for single unit use.
- **GENJCL1**: Sample JCL for running **GENSTATS** reports using **GENSTATP**.
- **GENJCL2**: Sample JCL for running **GENSTATS** reports using **GENSTAT1** for single unit use.
- **GENJCL3**: Sample JCL for running **GENSTATS** reports using **GENSTATW** and **GENSTATR** for processing multiple reports from a single input.
- **CP499J**: Sample JCL to run DLm Command Processor 499 using **DLMCMD49**.

7 Invoke Command Processors directly with JCL

There are a number of advantages in using the supplied procedures such as **DLMCMDP1** to invoke Command Processors. These include ease of use and elimination of many possible user input errors. This chapter provides a sample showing the method to invoke Command Processors directly with JCL and sample commands to execute various Command Processors.

Topics include:

Invoking Command Processors directly with JCL

The Command Processors can also be invoked with "open" JCL. Below is an example:

Examples:

The JCL below uses a DLm Unit, (F200), that is not in an MTL:

```
//GETCMD  EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=&CMDFILE,DISP=(NEW,PASS),UNIT=SYSALLDA,
//             SPACE=(TRK,(1)),DCB=(LRECL=80,RECFM=FB,BLKSIZE=3200)
//SYSUT1   DD  *
RUNWAIT (600)  501 "HELP" "/tapelibFLR" "BFLBG1.FLAT"  "TS2"
//*
//CMD1    EXEC PGM=DLMCMD
//STEPLIB  DD  DISP=SHR,DSN=USER.DLM.LINKLIB
//DLMCMD   DD  DSN=&CMDFILE,DISP=(OLD,PASS)
//DLMLOG   DD  SYSOUT=*
//DLMCTRL  DD  DSN=&TEMP,DISP=(,DELETE),UNIT=/F200
//STEP4   EXEC PGM=IRXJCL,COND=(0,NE),
//             PARM='DLMCMD3 DEBUG()'
//SYSEXEC  DD  DSN=USER.DLM.REXXCMD,DISP=SHR
//BFLFILE  DD  DSN=BFLBG1.FLAT,VOL=SER=BFLBG1,DISP=SHR,LABEL=(,NL),
//             UNIT=/F200,
//             DCB=(RECFM=FB,LRECL=264,BLKSIZE=264)
//DLMCMD   DD  DSN=&CMDFILE,DISP=(OLD,DELETE)
//SYSTSIN  DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//PRINT    DD  SYSOUT=*
```

- The JCL below uses a DLm Unit, (F207), that is in an MTL:

```
//GETCMD  EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=&CMDFILE,DISP=(NEW,PASS),UNIT=SYSALLDA,
//             SPACE=(TRK,(1)),DCB=(LRECL=80,RECFM=FB,BLKSIZE=3200)
//SYSUT1   DD  *
RUNWAIT (600)  501 "HELP" "/tapelibFLR" "BFL501.FLAT"  "TS2"
//*
//CMD1    EXEC PGM=DLMCMD
//STEPLIB  DD  DISP=SHR,DSN=USER.DLM.LINKLIB
//DLMCMD   DD  DSN=&CMDFILE,DISP=(OLD,PASS)
//DLMLOG   DD  SYSOUT=*
//DLMCTRL  DD  DSN=&TEMP,DISP=(,DELETE),UNIT=(/F207,,,SMSHONOR)
//STEP4   EXEC PGM=IRXJCL,COND=(0,NE),
//             PARM='DLMCMD3 DEBUG()'
//SYSEXEC  DD  DSN=USER.DLM.REXXCMD,DISP=SHR
//BFLFILE  DD  DSN=BFL501.FLAT,VOL=SER=BFL501,DISP=SHR,LABEL=(,NL),
//             UNIT=(/F207,,,SMSHONOR),
//             DCB=(RECFM=FB,LRECL=264,BLKSIZE=264)
//DLMCMD   DD  DSN=&CMDFILE,DISP=(OLD,DELETE)
//SYSTSIN  DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//PRINT    DD  SYSOUT=*
```

In the JCL examples above:

- The tape library for the virtual tape devices specified above is tapelibFLR. This value can be seen on the DLmConfig Device panel in the Tape Library column for the device.

Note: You cannot specify the DEFAULT value for tapelib when executing a Command Processor directly with JCL. Instead, you must use a Command Processor PROC such as DLMCMDPR or DLMCMDP1.

- The Command Processor command must be specified in the SYSUT1 DD of the GETCMD job step. In this example, '501'.
- You must supply a valid time-stamp or token. This is the parameter 'TS2' on the above JCL sample.

Note: The inclusion of the time-stamp parameter is critical to ensure that the Command Processor output is the result of this command.

- The Command Processor command file name parameter, (digits 4-6, or '501' in the above example) must match digits 4-6 of the VOLSER in the BFLFILE DD statement of the STEP4 job step. The VOLSER of the above example is 'BFL501'.

Sample Output

```
EMC DLm VTEC Command Processor Tool (501) v4.4.0-r5798
Tue Sep  1 10:29:03 EDT 2015
================================================================================
09/01/2015 10:29:04 vte1 VT: DLm0409I: Command received: 'HELP'
09/01/2015 10:29:04 vte1 VT:  Help Summary of Commands (in alphabetic order):
09/01/2015 10:29:04 vte1 VT:   CLOSE VSTATS PATH=xxx
09/01/2015 10:29:04 vte1 VT:   DETAIL MOUNT
09/01/2015 10:29:04 vte1 VT:   EXPORT from to COMPRESS!
09/01/2015 10:29:04 vte1 VT:   FIND VOL=volser ¢DEV=devicename!¢LOCAL!
09/01/2015 10:29:04 vte1 VT:   HELP ¢ command | message number | ABOUT !
09/01/2015 10:29:04 vte1 VT:   IMPORT from to ¢COMPRESS! ¢NORUN!
09/01/2015 10:29:04 vte1 VT:  INITIALIZE VOL=volser DEV=devicename ¢COUNT=count! ¢LABEL=S/A/N!
09/01/2015 10:29:04 vte1 VT:                    ¢SCRATCH=YES/NO! ¢ERRORS=nnnnn! ¢EPIC=NO/YES!
09/01/2015 10:29:04 vte1 VT:                     ¢DIR=subdirectory! ¢CLASS=storageclass!
09/01/2015 10:29:04 vte1 VT:   LOAD ¢VOL=!volser ¢ON! ¢DEV=!devicename|* ¢ UNLABELED | LABELED | NL | SL | AL !
¢PROTECTED!
09/01/2015 10:29:04 vte1 VT:  QUERY ¢¢DEV=!devicename¢+! | ALL! ¢MOUNTED!
09/01/2015 10:29:04 vte1 VT:     ¢CONFIG! | ¢CRITICAL! | ¢PATHS ¢ASSIGNED!! | ¢EFS! | ¢LABELS! ¢¢DEV=!devicename!
09/01/2015 10:29:04 vte1 VT:      ¢VERSION! | ¢SPACE ¢BYCLASS | BYDLM!! | ¢SCRATCHNAMES! | ¢COMPRESSION!
09/01/2015 10:29:04 vte1 VT:        ¢GR!
09/01/2015 10:29:04 vte1 VT:         ¢WARNING | RECOVER | RECOVERAMT!
09/01/2015 10:29:04 vte1 VT:         ¢STATISTICS | STATS!
09/01/2015 10:29:04 vte1 VT:         ¢WORM!
09/01/2015 10:29:04 vte1 VT:   QUIESCE ¢DEV=!devicename | ALL | *
09/01/2015 10:29:04 vte1 VT:   QUIESCE SCRATCH PATH=path
09/01/2015 10:29:04 vte1 VT:   QUIESCE TAPELIB PATH=path
09/01/2015 10:29:04 vte1 VT:   QUIESCE IMMEDIATE TAPELIB PATH=path
09/01/2015 10:29:04 vte1 VT:   READY ¢DEV=!devicename
09/01/2015 10:29:04 vte1 VT:   REWIND ¢DEV=!devicename
09/01/2015 10:29:04 vte1 VT:   RMLOCK volser
09/01/2015 10:29:04 vte1 VT:   ROTATE ¢ LOCAL | DRLOG !
09/01/2015 10:29:04 vte1 VT:   SAVE ¢TRACE!
09/01/2015 10:29:04 vte1 VT:   SCRATCHNAME ADD synonym¢=(prefix,prefix,...,CLASS=CLASSn,CLASS=CLASSn,...)!
09/01/2015 10:29:04 vte1 VT:   SCRATCHNAME DELETE synonym
09/01/2015 10:29:04 vte1 VT:   SET ¢TRACE=n! ¢SIZE=xxx! ¢PATH=pathname ¢NOCHECK!! ¢VOL=prefix!
09/01/2015 10:29:04 vte1 VT:       ¢WARNING=nn! ¢RECOVER=nn! ¢RECOVERAMT=nn!
09/01/2015 10:29:04 vte1 VT:       ¢HWCOMP=ON/OFF/FORCE! ¢HWDECOMP=ON/OFF/FORCE! ¢IDRC=ON/OFF/FORCE!
09/01/2015 10:29:04 vte1 VT:       ¢DEV=devicename|ALL!
09/01/2015 10:29:04 vte1 VT:       ¢TIMESTAMPS=NO/YES!
09/01/2015 10:29:04 vte1 VT:       ¢QSPACETIMEOUT=nn!
09/01/2015 10:29:04 vte1 VT:       ¢ERASEPOLICYTTL=nnnD/nnnH!
09/01/2015 10:29:04 vte1 VT:       ¢CRITICALPERCENT=nn}
```

```
09/01/2015 10:29:04 vte1 VT:      ¢CRITICALSCRATCH=YES/NO!
09/01/2015 10:29:04 vte1 VT:      ¢CRITICALEFSMOVE=YES/NO!
09/01/2015 10:29:04 vte1 VT:      ¢CRITICALRO=YES/NO!
09/01/2015 10:29:04 vte1 VT:      ¢CRITICALLEOT=YES/NO!
09/01/2015 10:29:04 vte1 VT:      ¢NONTRUNCATEWRITE=YES/NO!
09/01/2015 10:29:04 vte1 VT:      ¢RDC256K=ON/OFF!
09/01/2015 10:29:04 vte1 VT:      ¢RESCRATCHIFREAD=NO/YES!
09/01/2015 10:29:04 vte1 VT:      ¢PROTECTVOL1=NO/YES!
09/01/2015 10:29:04 vte1 VT:      ¢WRITEVERIFY=ON/OFF/CRC/DATA/HEADERS!
09/01/2015 10:29:04 vte1 VT:      ¢DDWRITEVERIFY=ON/OFF/CRC/DATA/HEADERS!
09/01/2015 10:29:04 vte1 VT:      ¢TESTCOMPRESSEDDATA=ON|OFF|CRYPT!
09/01/2015 10:29:04 vte1 VT:      ¢AMDD=ON|OFF!
09/01/2015 10:29:04 vte1 VT:      ¢FSALLOCATION=RR|SPACE!
09/01/2015 10:29:04 vte1 VT:  SHOW CHANNEL ADAPTERS
09/01/2015 10:29:04 vte1 VT:  SHOW DRIVE LIST
09/01/2015 10:29:04 vte1 VT:  SHOW REJECTED PATHS
09/01/2015 10:29:04 vte1 VT:  SNMP SHOW STATUS
09/01/2015 10:29:04 vte1 VT:  SNMP  ACKNOWLEDGE FAILURE
09/01/2015 10:29:04 vte1 VT:  STATISTICS SHOW
09/01/2015 10:29:04 vte1 VT:  STATISTICS INTERVAL=MINUTES/SECONDS
09/01/2015 10:29:04 vte1 VT:  UNLOAD ¢DEV=!devicename
09/01/2015 10:29:04 vte1 VT:  UNQUIESCE ¢DEV=!devicename | ALL | *
09/01/2015 10:29:04 vte1 VT:  UNQUIESCE SCRATCH PATH=path
09/01/2015 10:29:04 vte1 VT:  UNQUIESCE TAPELIB PATH=path
09/01/2015 10:29:04 vte1 VT:  UNREADY ¢DEV=!devicename
================================================================================
```

Appendix A

Troubleshooting Command Processors

This appendix provides troubleshooting information for command processors.

Topics include:

Troubleshooting Command Processors

DLm Command Processor processing is composed of both mainframe and DLm.

Command Processor Framework

The DLMZOS Utility DLMCMD is used to send a request to execute a Command Processor script on the DLm. DLMCMD uses a DLm Virtual Tape Drive unit to send the request to Virtuent on a DLm Virtual Tape Engine, (VTE). Virtuent then invokes the specified Command Processor on the VTE and passes it the parameters specified in the request.

Typically, the Command Processor script that runs on the DLm will receive input parameters that specify where to place its output. This is typically an output file whose name is in the format of BFLxxx.FLAT.

Typically, Command Processing mainframe processing includes a jobstep where the output from a Command Processor is retrieved. This is usually done by reading the BFLxxx.FLAT file referred to above. Special logic in Virtuent allows for reading these 'FLAT' files when a DD statement includes a DSN of BFLxxx.FLAT and a VOLSER of BFLxxx.

In the above framework there are a number of possible trouble spots:

- Communication errors and failures while sending Command Processor requests to the DLm.
- Command Processor script failures while executing on the DLm.
- User parameter errors.
- Failures to obtain Command Processor outputs.

DLm Command Processors include PROCs that include special processing to ensure that any errors from the above trouble spots are detected. These PROCs are DLMCMDP1, DLMCMDPR, GENSTATP, GENSTAT1, GENSTATW, and DLMCMD49. Information about these PROCs is contained elsewhere in this manual. However, each of these PROCs contain the following features that ensure that Command Processor failures are detected:

- Each Command Processor request sent to the DLm contains a unique token that is matched against the Command Processor output.

- Special processing is provided to ensure that the location for the Command Processor output is correct.
- Mainframe edit processing is performed on input user edits prior to being sent to the DLm.

A sample JCL member called **DLMCMDJ2** is provided for users who do not want to use the above PROCs. When the DLMCMDJ2 sample JCL member is used, one would change the 'TS' value to a unique token

What to do if errors occur

The following sections describe the steps to perform when an error does occur.

z/OS Based Verifications

If a Command Processor job failed, look thru all the output for any error messages. In addition, any communication problems might be indicated in the DLMLOG DD output created in the job step which executes the DLm z/OS utility DLMCMD.

DLMLOG Output

Within the job output for the failing job, the DLMLOG DD will contain a log of the DLMCMD processing. When an error has been returned by Virtuent, the error code will be displayed as ERR=xx, where the xx is the error code. Here are a list of possible error codes and their meanings:

Table 30 **DLMLOG Output Error Codes**

Error Code	Description	Corrective Action
FF	Unknown command or a command with invalid or wrong parameters passed.	Specify a correct Command Processor number and/or fix parameter list.
FC	Error encountered executing command.	Locate error in btilog and correct it. Note: The btilog can be found on the VTE of the DLm.
FB	Script did not complete within time specified.	Increase the RUNWAIT seconds.

Examples

This section provides some examples.

In the following example, the GENSTATP PROC is used to generate a Slow Mounts report but fails in the job step which invokes the DLm Command Processor. This failure occurs because the Command Processor script did not complete within the specified number of seconds, which is defaulted to 600 seconds.

The following is the example JCL for the failing job:

```
//GO1      EXEC GENSTATP,CMD=998,UNIT=/F200,UNIT2=F201,                X
//              REPORT='SLOWMOUNTS=5'
```

The following is example output:

```
DLMCMD   VER 4.12     DLMCTRL = F201
2015/09/02 13:13:44 CMD ERR=FB:  RUNWAIT (600) 998 "DEFAULT" "BFL998.FLAT" "startdate=01/01/2000"
"enddate=01/01/2099"  "20150902-13
2015/09/02 13:13:44    cont'd:  13:43-12072"
```

To resolve the above problem, the GENSTATP job step could be changed to increase the number of seconds to wait with the WAIT parameter. Here is an example of this with the changes in bold:

```
//GO1      EXEC GENSTATP,CMD=998,UNIT=/F200,UNIT2=F201,                X
//              REPORT='SLOWMOUNTS=5',WAIT=1200
```

DLm Based Verifications

If a Command Processor failure occurred during processing on the DLm, there are a number of files where error information can be found. The Command Processor output report should also be reviewed for possible error messages. Here are some possible locations for error information on the DLm:

VTCon Output

Current Virtuent output can be obtained from the Virtuent Console. This console output can be viewed in a CLI session on the VTE by entering vtcon at the command line or by clicking on the Console push button on the DLmConsole Status panel.

Message Logs

The btilog file on the VTE can be viewed for any Virtuent messages that have occurred. Below is an example message that can be found:

```
DLm556E  RUNWAIT process #xx has not finished in nn seconds
```

Note: With regards to the above message, even though the RUNWAIT may time out, the Command Processor script will still run to completion.

In addition, Command Processor messages can be found in the /var/log/messages file.

Both of the above message logs can also be retrieved using Command Processor 502, the DLmConsole System Status panel Log push button, and the DLmConsole Gather panel Gather push button

How to Terminate the Command Processor Process

When a RunWait command to execute a Command Processor has timed out, it may be necessary to terminate the associated Command Processor thread(s) that are still running on the VTE. The following commands require that you have established a CLI session on that VTE:

1. check to see if the command processor is currently running:

```
vte1:~ # ps -ef|grep script502
root      70833  56032  1 08:49 pts/5    00:00:00 script502 502 "L1" "2" "DEFAULT"
"BFL502.FLAT" "20150731-14:50:50-57843"
root      70916  58966  0 08:49 pts/7    00:00:00 grep script502
```

   In the example above, there is currently one process executing Command Processor 502. Its process ID is 70833.

2. Issue the following command to kill the above Command Processor process:

```
kill -9 70833
```

Communication Verifications

As described earlier in this section, Command Processor commands are sent from z/OS using the DLm z/OS Utility DLMCMD thru the FICON channel attached to the DLm VTE.

This is done using the UNIT or UNIT2 parameter of the Command Processor PROCs. With regards to these DLm Units:

- When using both the UNIT and UNIT2 parameters, ensure that the DLm Units are attached to the same VTE. Command Processors store their report output on local storage on the VTE.

  As a result, if the DLm Unit specified for the UNIT parameter is attached to one VTE while the DLm Unit specified for the UNIT2 parameter is attached to a different VTE, the Command Processor will fail because it cannot locate the correct output report. When using an esoteric for the UNIT parameter, ensure that the esoteric only includes DLm Units for the correct VTE.

- Ensure that the following Command Processors are run on the correct VTE. For example, Command Processor 501 issues VTE commands. As a result, you must ensure that you are running this Command Processor on the VTE that is to execute the VTE command. The following is a list of Command Processors that are VTE orientated:

Table 31

Command Processor	Description
499	Perform diagnostics
501	Execute VTE command
502	Obtain logs
999	Obtain VTE activity statistics

Note: Command Processor 503, (Virtual Tape Volume Information), may include local files for a VTE in its report. These files may include Command Processor output report files from previous Command Processor executions.

- When the UNIT2 parameter is used, ensure that this DLm Unit is offline. Bringing it online first, then offline will ensure that a path to the VTE has been established.
- Ensure that the DLm Unit specified in the UNIT parameter is online.