Dell EMC Mainframe Enablers
TimeFinder SnapVX and zDP
Version 8.4

Product Guide
REV 06
## CONTENTS

### Part 1  TimeFinder SnapVX

#### Chapter 1  Introduction

- Introduction to TimeFinder .......................................................... 16
- Mainframe Enablers and TimeFinder ........................................... 16
- Licensing ..................................................................................... 16
- Introduction to TimeFinder SnapVX ............................................. 17
- Backward compatibility to traditional TimeFinder products ........ 18

#### Chapter 2  Configuration

- TimeFinder SnapVX configuration layers ....................................... 20
  - Configuration Layer 1: EMCSNAPO site options ....................... 20
  - Configuration Layer 2: GLOBAL command parameters ............. 20
  - Configuration Layer 3: Parameters on other commands .......... 21
- Editing the EMCSNAPO macro ..................................................... 21
- EMCSNAPO site options .............................................................. 22
  - Summary .................................................................................. 22
  - ACTIVATE_SUBTASK# ............................................................... 24
  - ALLOW_FBA_META ................................................................. 24
  - ALLOW_LIGHTNING_DEVICE ................................................. 24
  - ALLOWZDP ............................................................................. 24
  - AUTOCREATE ......................................................................... 24
  - AUTOUNLINK .......................................................................... 25
  - BACKGRND ............................................................................ 25
  - CACHESYM ............................................................................. 25
  - CHECKBCV ............................................................................ 25
  - CHKONLIN ............................................................................. 25
  - CHKNSNZ ................................................................................ 25
  - CONDVOL ............................................................................... 25
  - CONSIST .................................................................................. 26
  - COPYVOL ................................................................................ 26
  - DEBUG_ERROR .......................................................................... 26
  - DEBUG_SDUMP ......................................................................... 26
  - DISPCUU ................................................................................. 26
  - EMUL_TYPE ............................................................................. 26
  - FREEUNLK ............................................................................... 26
  - GROUP_DSNAME ....................................................................... 27
  - MESSAGE ............................................................................... 27
  - REFVTOC ................................................................................ 27
  - REPLACE .................................................................................. 27
  - RESTORE_CREATE ................................................................. 27
  - SECURE .................................................................................... 27
  - SECURE_QUERY ....................................................................... 28
  - SHOWDEF ................................................................................ 28
  - SNAPSHOT_LIST ....................................................................... 28
  - SNAPSHOT_NAME .................................................................... 28
  - SOFTLINK ............................................................................... 28
  - SOFTRESTORE .......................................................................... 28
Chapter 3 Getting Started

TimeFinder SnapVX and TimeFinder(Clone Mainframe Snap Facility) ........................................... 32
Post-installation .......................................................................................................................... 32
Software interoperability considerations .................................................................................. 32
SIBBATCH ................................................................................................................................. 32
SRS ........................................................................................................................................ 32
FlashCopy ................................................................................................................................ 32
Starting ResourcePak Base ....................................................................................................... 33
Setting up security ..................................................................................................................... 33
Running TimeFinder(Clone Mainframe Snap Facility) .......................................................... 34

Chapter 4 Operations

Softlinked and hardlinked snapshots ....................................................................................... 36
Snapshot life cycle ................................................................................................................... 36
SnapVX group status transitions ............................................................................................ 37
Target volume track definition............................................................................................... 37
Monitoring SRP usage during linking .................................................................................... 38
Secure snaps ............................................................................................................................ 39
Summary of operations ........................................................................................................... 41
Managing snapshots ................................................................................................................ 41
Viewing snapshots .................................................................................................................. 41
Viewing groups ....................................................................................................................... 42
Fine-tuning snapshot processing ........................................................................................... 42
Linking/unlinking and copying ............................................................................................... 42
Setting up consistency ............................................................................................................. 43
Controlling device status ........................................................................................................ 44
Conditioning target volume ................................................................................................... 44
Miscellaneous SnapVX settings ............................................................................................. 45
Example of operations ............................................................................................................. 46
Defining groups of statements ............................................................................................... 47
Specifying the group dataset ................................................................................................... 47
Group dataset allocation example ........................................................................................... 48
Defining groups ........................................................................................................................ 48
Referencing other groups from within a group ....................................................................... 48
Using groups as arguments to SnapVX commands ............................................................... 49

Chapter 5 Command Reference

Syntax conventions .................................................................................................................. 52
SnapVX commands .................................................................................................................. 52
Contents

DEFINE VDG ................................................................. 147
DELETE TGT ................................................................. 150
DELETE VDG ................................................................. 151
GLOBAL ................................................................. 152
LINK VDG ................................................................. 153
MODIFY TGT ADD|REMOVE ........................................ 155
MODIFY VDG ADD|REMOVE ........................................ 156
MODIFY VDG OPTIONS ................................................ 158
PERSISTENT .......................................................... 160
QUERY FREE ........................................................... 161
QUERY TGT .............................................................. 162
QUERY VDG .............................................................. 164
RESTORE VDG .......................................................... 169
SECURE VDG ............................................................ 170
STOP_FREE TGT .......................................................... 171
TERMINATE VDG ....................................................... 172
UNLINK TGT .............................................................. 173
UNLINK VDG .............................................................. 174

Chapter 9  zDP ISPF Interface

Introduction ................................................................................. 176
Logging in to zDP .......................................................... 176
Logging out of zDP .......................................................... 176
Accessing zDP ISPF tools ................................................... 177
Setting session options and GLOBAL parameters ............... 178
Accessing zDP ISPF help ...................................................... 178
Monitoring and managing VDGs ...................................... 179
  Define VDG .......................................................... 180
  View SRP usage ......................................................... 181
  View VDG configuration parameters ........................ 182
  Modify VDG member ................................................ 182
  Query VDGs ............................................................ 183
  Set SMF parameters .................................................. 183
Managing VDG members ..................................................... 184
Managing target sets .......................................................... 185
  Define target set ......................................................... 186
  View target set storage details .................................. 187
  Modify target set member ........................................ 188
  Query target sets ....................................................... 188
  Link target set .......................................................... 189
  Unlink target set .......................................................... 189
Managing target set members ............................................ 190
Managing snapsets .......................................................... 191
  View VDG snapsets ................................................... 192
  Query snapsets .......................................................... 196

Chapter 10  SMF Records

Overview ..................................................................................... 198
SMF record format .......................................................... 198
SMF record header ........................................................... 198
Section header ............................................................... 199
Section descriptors .............................................................. 199
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EMCSNAP0 site options</td>
</tr>
<tr>
<td>2</td>
<td>Allowable SnapVX commands per snapshot state</td>
</tr>
<tr>
<td>3</td>
<td>SnapVX group actions and statuses</td>
</tr>
<tr>
<td>4</td>
<td>Managing snapshots</td>
</tr>
<tr>
<td>5</td>
<td>Viewing snapshots</td>
</tr>
<tr>
<td>6</td>
<td>Viewing groups</td>
</tr>
<tr>
<td>7</td>
<td>Fine-tuning snapshot processing</td>
</tr>
<tr>
<td>8</td>
<td>Linking/unlinking and copying</td>
</tr>
<tr>
<td>9</td>
<td>Setting up consistency</td>
</tr>
<tr>
<td>10</td>
<td>Controlling device status</td>
</tr>
<tr>
<td>11</td>
<td>Conditioning target volume</td>
</tr>
<tr>
<td>12</td>
<td>Miscellaneous SnapVX settings</td>
</tr>
<tr>
<td>13</td>
<td>Managing VDGs</td>
</tr>
<tr>
<td>14</td>
<td>Managing target sets</td>
</tr>
<tr>
<td>15</td>
<td>Managing snapsets</td>
</tr>
<tr>
<td>16</td>
<td>Setting VDG snapshot-related options</td>
</tr>
<tr>
<td>17</td>
<td>Setting SRP usage and RDP cache utilization parameters</td>
</tr>
<tr>
<td>18</td>
<td>Managing copy-once devices</td>
</tr>
<tr>
<td>19</td>
<td>Monitoring zDP status</td>
</tr>
<tr>
<td>20</td>
<td>zDP messaging and SMF records</td>
</tr>
<tr>
<td>21</td>
<td>Setting zDP runtime task options</td>
</tr>
</tbody>
</table>
As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information about product features.

Contact your Dell EMC representative if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. New versions of this document might be released on the Dell EMC Online Support website. Check the Dell EMC Online Support website to ensure that you are using the latest version of this document.

Purpose

This guide describes how to use TimeFinder SnapVX and Data Protector for z Systems (zDP).

Audience

This guide is intended for the host system administrator, system programmer, or operator who is evaluating, planning for, managing, or using TimeFinder SnapVX and zDP.

Related documentation

To access related documentation, go to the PowerMax and VMAX All Flash Technical Documentation webpage at:

www.dell EMC.com/en-us/documentation/vmax-all-flash-family.htm

The following documents provide information about Mainframe Enablers:

- Mainframe Enablers Release Notes
- Mainframe Enablers Installation and Customization Guide
- Mainframe Enablers Message Guide
- ResourcePak Base for z/OS Product Guide
- SRDF Host Component for z/OS Product Guide
- AutoSwap for z/OS Product Guide
- Consistency Groups for z/OS Product Guide
- TimeFinder SnapVX and zDP Product Guide
- TimeFinder/Clone Mainframe Snap Facility Product Guide
- TimeFinder/Mirror for z/OS Product Guide
- TimeFinder Utility for z/OS Product Guide
The following documents provide additional information:

- **PowerMax Family Product Guide**—Documents the features and functions of the PowerMax storage systems.
- **PowerMaxOS for PowerMax and VMAX All Flash Release Notes**—Describes new features and any known limitations.
- **VMAX All Flash Product Guide**—Documents the features and functions of the VMAX All Flash storage systems.
- **HYPERMAX OS for VMAX All Flash and VMAX3 Family Release Notes**—Describes new features and any known limitations.
- **E-Lab Interoperability Navigator (ELN)**—Provides a web-based interoperability and solution search portal. You can find the ELN at elabnavigator.EMC.com.

Conventions used in this document

Dell EMC uses the following type style conventions in this document:

- **Italic**
  - Used for:
    - Titles of publications referenced in text
    - Emphasis, for example, a new term

- **Courier**
  - Used for:
    - Command syntax and parameters
    - System output, such as messages

- **Courier bold**
  - Used for user input, for example: Reply **CONT**.

- **Courier italic**
  - Used for variables in command/parameter syntax and messages, for example: **DISPLAY** **ccuu**

- **Courier underline**
  - Underline indicates the default value, for example: **YES**|**NO**

- **< >**
  - Angle brackets enclose variables or explanatory text when it includes multiple words, for example: `<list of device numbers>`

- **[ ]**
  - Square brackets enclose optional values, for example: **DISPLAY [DETAIL]**

- **|**
  - Vertical bar indicates alternate selections (the bar means “or”), for example: **RUN|NORUN**

- **{}**
  - Braces are used together with the vertical bar (|) to indicate the start and end of alternate selections, for example: `{DEV `symdv#`|CUU **ccuu**}`

- **...**
  - Ellipses indicate nonessential information omitted from the example

Where to get help

**Product information**

For information about Dell EMC products, licensing, and service, go to [www.dellemc.com](http://www.dellemc.com) (registration required).

**Technical support**

To access the Dell EMC Online Support web site, go to [www.dell.com/support](http://www.dell.com/support) and search for your product. You will be redirected to the product support page, offering quick links to Documentation, Downloads, Advisories, and Knowledgebase for your...
product. The product support page also provides a link to the Service Center where you can create a service request, manage your service requests, and contact Dell EMC Customer Support either through Dell EMC Live Chat or using other options.

**Note:** To open a service request through Dell EMC Online Support, you must have a valid support agreement. Contact the Dell EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

eLicensing support

To activate your entitlements and obtain your license files, visit the Service Center as directed on your License Authorization Code (LAC) letter emailed to you.

For help with missing or incorrect entitlements after activation (that is, expected functionality remains unavailable because it is not licensed), contact your Dell EMC Account Representative or Authorized Reseller.

For help with any errors applying license files through Solutions Enabler, contact Dell EMC Customer Support.

If you are missing a LAC letter, or require further instructions on activating your licenses through the Online Support site, contact Dell EMC worldwide licensing team at licensing@emc.com or call:

- North America, Latin America, APJK, Australia, New Zealand: SVC4EMC (800-782-4362) and follow the voice prompts.
- EMEA: +353 (0) 21 4879862 and follow the voice prompts.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to: VMAXContentFeedback@emc.com
PART 1

TimeFinder SnapVX

Part 1 of this guide describes how to set up and configure TimeFinder SnapVX for z/OS.

Chapter 1, “Introduction,” provides an overview of TimeFinder SnapVX.

Chapter 3, “Getting Started,” describes SnapVX post-installation activities.

Chapter 2, “Configuration,” lists SnapVX configuration parameters.

Chapter 4, “Operations,” describes SnapVX operations.

CHAPTER 1
Introduction

This chapter covers the following topics:

◆ Introduction to TimeFinder ................................................. 16
◆ Introduction to TimeFinder SnapVX ..................................... 17
Introduction to TimeFinder

Dell EMC TimeFinder is a family of local replication products that allows you to non-disruptively create and manage point-in-time copies of data, enabling simultaneous action of business tasks that were previously sequential. For example, TimeFinder allows you to create a point-in-time copy of critical data while this data continues to be used in production operations.

The ability to access source data during the TimeFinder copy operation can increase the availability of the application. TimeFinder can also shorten backup windows, maintenance windows and improve service levels.

This manual provides the command and parameter details for using TimeFinder SnapVX, which is a TimeFinder product that supports the z/OS mainframe environment as one of the components of Dell EMC Mainframe Enablers.

Note: For a comprehensive description of traditional TimeFinder products, including TimeFinder/Clone and TimeFinder VP Snap, see the TimeFinder/Clone Mainframe Snap Facility Product Guide.

Mainframe Enablers and TimeFinder

TimeFinder is one of the Dell EMC Mainframe Enablers. The Dell EMC Mainframe Enablers allow you to monitor and manage the storage and include the following components:

- ResourcePak Base for z/OS
- SRDF Host Component for z/OS
- AutoSwap for z/OS
- Consistency Groups for z/OS
- TimeFinder SnapVX
- Data Protector for z Systems (zDP)¹
- TimeFinder/Clone Mainframe Snap Facility
- TimeFinder/Mirror for z/OS
- TimeFinder Utility

When you install the Dell EMC Mainframe Enablers product suite, you install ResourcePak Base, which runs in a persistent address space under z/OS, and includes the software for all the other components.

Licensing

See the following documents for information about licensing:

- Mainframe Enablers Installation and Customization Guide
- PowerMax Family Product Guide
- VMAX All Flash Product Guide

¹ zDP requires TimeFinder SnapVX but is a separately licensed product.
Introduction to TimeFinder SnapVX

Starting with HYPERMAX OS 5977, TimeFinder introduces TimeFinder SnapVX which provides the best aspects of the traditional TimeFinder offerings, together with increased scalability and ease-of-use.

SnapVX provides targetless snapshot capabilities. Instead of requiring target volumes to capture point-in-time images of source volumes, SnapVX introduces the new concept of creating a “snapshot” to create and access point-in-time images of volumes. SnapVX supports up to 256 snapshots per volume which effectively allows for the creation of 256 separate point-in-time images of every volume. Snapshots store changed tracks (deltas) directly in the Storage Resource Pool of the source device and use shared allocations to eliminate saving duplicate tracks between multiple snapshots. Users can assign names to individual snapshots and assign an automatic expiration date to each one.

**Note:** The PowerMax Family Product Guide and VMAX All Flash Product Guide provide additional information about SnapVX.

Linking snapshots to targets

With SnapVX, a snapshot can be accessed by linking it to a host accessible volume (known as a target volume). Up to 1024 target volumes can be linked to one or more snapshots of a single source volume. The 1024 links can all be to the same snapshot on a single source volume, or they can be multiple target volumes that are linked to multiple snapshots from the same source volume.

**Note:** A target volume may be linked only to one snapshot at a time.

You can link, unlink, relink snapshots to the same set of target volumes, which allows for flexibility in the point-in-time that is presented on a single set of target devices. These snapshots share allocations to the same track image whenever possible while ensuring they each continue to represent a unique point-in-time image of the source volume. Despite the space efficiency that is achieved through shared allocation to unchanged data, additional capacity is required to preserve the pre-update images of changed tracks that each point-in-time snapshot captures.

Cascading snapshots

Snapshots can be cascaded from linked targets, and targets can be linked to snapshots of linked targets. There is no limit to the number of levels of cascading, and the cascade can be broken.

COPY and NOCOPY mode

Targets can be linked in COPY mode to create full-copy clones. If an application is to find a particular point-in-time copy among a large set of snapshots, SnapVX enables you to link and relink until the correct snapshot is located.
If the linked target is in NOCOPY mode, a cascading snapshot is referred to as a dependent snapshot. A target device cannot be unlinked without removing dependent snapshots. Or, the snapshots should be removed in the reverse order of creation.

**Note:** See “Target volume track definition” on page 37 for information about how target volume tracks are defined.

### Backward compatibility to traditional TimeFinder products

TimeFinder SnapVX supports backward compatibility to traditional TimeFinder products, including TimeFinder/Clone, TimeFinder VP Snap, and TimeFinder/Mirror.

TimeFinder transparently converts the commands of the traditional TimeFinder products to SnapVX commands.

**Note:** Using TimeFinder SnapVX requires no change of scripts that use the traditional TimeFinder commands.

TimeFinder Emulation sessions and TimeFinder snapshots can coexist on the same device.
CHAPTER 2
Configuration

This chapter covers the following topics:

- TimeFinder SnapVX configuration layers .............................................................. 20
- Editing the EMCSNAPO macro ............................................................................ 21
- EMCSNAPO site options ..................................................................................... 22
TimeFinder SnapVX configuration layers

TimeFinder has three configuration layers, each of which can override the one above it:

1. EMCSNAPO site options
2. GLOBAL command parameters
3. Parameters on other commands

**WARNING**

Only the administrator for the site should set and change site options to avoid unforeseen complications and problems with TimeFinder processing and the expected results.

Configuration Layer 1: EMCSNAPO site options

The first configuration layer consists of the TimeFinder site options. You can accept the site option default values or you may permanently change the value to suit your requirements.

You can also use the Dell EMC REXX Interface to create scripts to influence standard Dell EMC processing or to perform related external user-defined processing. For more information, see the description of TimeFinder REXX exits in the TimeFinder/Clone Mainframe Snap Facility Product Guide.

The site-specific values are specified in the EMCSNAPO macro in the Mainframe Enablers SAMPLIB (SMP/E DDNAME: MFESAMP). You can change a value by adding the modification to a member in the RIMLIB and then running the EMCSNAPO statement.

The JCL in the RIMLIB specifies that the Mainframe Enablers SAMPLIB as the first dataset in the SYSLIB concatenation for the assembly.

See “EMCSNAPO site options” on page 22 and “Editing the EMCSNAPO macro” on page 21 for a list of the site options and a description on how to edit the EMCSNAPO macro.

Configuration Layer 2: GLOBAL command parameters

The second layer consists of the GLOBAL command parameters. The GLOBAL command parameters match many of the site options. GLOBAL command parameters override EMCSNAPO site options. If you need temporarily to change a site option value that has a matching GLOBAL parameter, you can set that GLOBAL parameter to the value you want to use.

GLOBAL parameter values apply only to commands that follow the GLOBAL command in the current job step. They do not affect any commands that precede them in the current job step. After that job step is over, TimeFinder uses the site option value again.

You can set multiple global commands within a job step. In each case, the GLOBAL command applies to the commands that follow unless overridden by another, later GLOBAL command.
Configuration Layer 3: Parameters on other commands

The third layer consists of the parameters associated with other TimeFinder commands. Many of the TimeFinder commands can take parameters available on the GLOBAL command statement. If you need to override an EMCSNAPO site option or a GLOBAL parameter, you can issue the equivalent parameter with that command.

The value you use is only in force for the duration of the operation of the command to which it is appended. After the command is finished, TimeFinder uses the GLOBAL parameter value (if one was set) or the site option value again.

Editing the EMCSNAPO macro

The EMCSNAPO macro lists all of the site options you can change. (Many of these site options are also parameters of the GLOBAL command.) For each site option, the macro lists:

- A short form of the site option name with the default value in the form:
  
  &SHORTFORM=DEFAULT

- The full name of the site option.

- A list of possible values.

For example, the macro listing for the AUTOMATIC_DEALLOC parameter is:

&AUTODEAL=YES, AUTOMATIC_DEALLOC (Y/N)

To change a parameter setting, you use the member #91SNPJB supplied in the RIMLIB and add the parameter to the EMCSNAPO statement and then run the job.

For example, the default value for the macro &AUTODEAL is YES. To change this setting to NO, you would add AUTODEAL=NO to the EMCSNAPO statement.

Note: The ampersand (&) is used only in the macro definition, but you do not use it when you are adding to the EMCSNAPO statement for execution.

Before:

EMCSNAPO DSECT=NO

After:

EMCSNAPO AUTODEAL=NO, DSECT=NO

The #91SNPJB executes the IBM assembler and the EMCSNAPO statement must follow the IBM assembler rules for coding. This especially applies to statement continuation, placing a character in column 72 to indicate continuation and continuing in column 16 of the next line. DSECT=NO must always be present.
For example:

```
  v - column 10
  v - column 16
  v - column 72
EMCSNAPO DATACLAS=ABC,
  X
MGMTCLAS=DEF,
  X
STORCLAS=GHI,
  X
DSECT=NO
```

**Note:** After a PTF has been applied that adds new GLOBAL parameters, add the new parameters to the EMCSNAPO statement in RIMLIB(#91SNPJB) and run the updated #91SNPJB job.

# EMCSNAPO site options

**Note:** This guide lists site options pertaining to SnapVX. For other TimeFinder site options, see the *TimeFinder/Clone Mainframe Snap Facility Product Guide*.

## Summary

Table 1 lists the EMCSNAPO site options and their default values together with the corresponding GLOBAL parameters.

### Table 1 EMCSNAPO site options (page 1 of 3)

<table>
<thead>
<tr>
<th>Page #</th>
<th>Site option (without the macro “&amp;” designation)</th>
<th>Site option default</th>
<th>Site option valid values</th>
<th>Site option name in QUERY GLOBAL output</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ACTIVATE_SUBTASK#</td>
<td>3</td>
<td>0-255</td>
<td>ACTIVATE_SUBTASK#</td>
</tr>
<tr>
<td>24</td>
<td>ALLOW_FBA_META</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>24</td>
<td>ALLOW_LIGHTNING_DEVICE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>24</td>
<td>ALLOWZDP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>24</td>
<td>AUTOCREATE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>AUTOUNLINK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>BACKGRND</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>CACHESYM</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>CHECKBCV</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>CHKONLIN</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>CHKSNSSZ</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>25</td>
<td>CONDVOL</td>
<td>ALL</td>
<td>ALL</td>
<td>DUMP</td>
</tr>
<tr>
<td>26</td>
<td>CONSIST</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>26</td>
<td>COPYVOL</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Page #</td>
<td>Site option (without the macro “&amp;” designation)</td>
<td>Site option default</td>
<td>Site option valid values</td>
<td>Site option name in QUERY GLOBAL output</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>26</td>
<td>DEBUG_ERROR</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>26</td>
<td>DEBUG_SDUMP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>26</td>
<td>DISPCUU</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>26</td>
<td>EMUL_TYPE</td>
<td>ALL</td>
<td>ALL</td>
<td>HARDLINK</td>
</tr>
<tr>
<td>26</td>
<td>FREEUNLK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>27</td>
<td>GROUP_DSNAME</td>
<td>n/a</td>
<td>dataset_name</td>
<td>GROUP_DSNAME(LOCAL)</td>
</tr>
<tr>
<td>27</td>
<td>MESSAGE</td>
<td>NONE</td>
<td>DISPLAY</td>
<td>PROMPT</td>
</tr>
<tr>
<td>27</td>
<td>REFVTOC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>27</td>
<td>REPLACE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>27</td>
<td>RESTORE_CREATE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>27</td>
<td>SECURE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>SECURE_QUERY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>SHOWDEF</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>SNAPSHOT_LIST</td>
<td>ALL</td>
<td>ALL</td>
<td>LINKED</td>
</tr>
<tr>
<td>28</td>
<td>SNAPSHOT_NAME</td>
<td>n/a</td>
<td>snapshot_name</td>
<td>NAME</td>
</tr>
<tr>
<td>28</td>
<td>SOFTLINK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>SOFTRESTORE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>SRDFA_RETRY</td>
<td>10</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>SRPPERCT</td>
<td>100</td>
<td>0-100</td>
<td>SRPPERCT</td>
</tr>
<tr>
<td>29</td>
<td>SRPMSGLVL</td>
<td>INFO</td>
<td>INFO</td>
<td>WARN</td>
</tr>
<tr>
<td>29</td>
<td>TERM_ALL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>TIMEOUT</td>
<td>15</td>
<td>0-128</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>29</td>
<td>VARYOFF</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
<tr>
<td>29</td>
<td>VARYON</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
</tbody>
</table>
**ACTIVATE_SUBTASK#**

See “ACTIVATE_SUBTASK#(nnn)” on page 52.

**Syntax**

```
ACTIVATE_SUBTASK#=nnn
```

Where:

```
nnn
```

The minimum number of storage systems. The default value is 3.

**ALLOW_FBA_META**

See “ALLOW_FBA_META((Yes|No))” on page 82.

**Syntax**

```
ALLOW_FBA_META={YES|NO}
```

**ALLOW_LIGHTNING_DEVICE**

See “ALLOW_LIGHTNING_DEVICE((Yes|No))” on page 82.

**Syntax**

```
ALLOW_LIGHTNING_DEVICE={YES|NO}
```

**ALLOWZDP**

See “ZDP((Yes|No))” on page 88.

**Syntax**

```
ALLOWZDP={YES|NO}
```

**AUTOCREATE**

See “AUTO_CREATE((Yes|No))” on page 91.

**Syntax**

```
AUTOCREATE={YES|NO}
```
**AUTOUNLINK**

See “AUTO_UNLink({Yes|No})” on page 52.

Syntax

\[
\text{AUTO\_UNLINK} = \{ \text{YES} | \text{NO} \}
\]

**BACKGRND**

See “BACKGROUNDCOPY({Yes|No|NOCOPYRD|VSE})” on page 83.

Syntax

\[
\text{BACKGRND} = \{ \text{YES} | \text{NO} | \text{NOCOPYRD} | \text{VSE} \}
\]

**CACHESYM**

See “CACHE_FULL_SYM({Yes|No})” on page 83.

Syntax

\[
\text{CACHESYM} = \{ \text{YES} | \text{NO} \}
\]

**CHECKBCV**

See “CHECKBCVHoldstatus({Yes|No})” on page 53.

Syntax

\[
\text{CHECKBCV} = \{ \text{YES} | \text{NO} \}
\]

**CHKONLIN**

See “CHECKONLINEpathstatus({Yes|No|NEVER})” on page 53.

Syntax

\[
\text{CHKONLIN} = \{ \text{YES} | \text{NO} | \text{NEVER} \}
\]

**CHKSNSZ**

See “CHECK_SNAPSHOT_SIZE({Yes|No})” on page 53.

Syntax

\[
\text{CHKSNSZ} = \{ \text{YES} | \text{NO} \}
\]

**CONDVOL**

See “CONDitionVOLUME({ALL|LaBeL|DUMP})” on page 54.

Syntax

\[
\text{CONDVOL} = \{ \text{ALL} | \text{DUMP} | \text{LABEL} \}
\]
**Configuration**

**CONSIST**

See “CONSISTENT({Yes|No})” on page 54.

Syntax

\[
\text{CONSIST} = \{ \text{YES} | \text{NO} \}
\]

**COPYVOL**

See “COPYVolid({Yes|No})” on page 55.

Syntax

\[
\text{COPYVOL} = \{ \text{YES} | \text{NO} \}
\]

**DEBUG_ERROR**

See “DEBUG({ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP})” on page 84, the ERROR option.

Syntax

\[
\text{DEBUG_ERROR} = \{ \text{YES} | \text{NO} \}
\]

**DEBUG_SDUMP**

See “DEBUG({ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP})” on page 84, the SDUMP option.

Syntax

\[
\text{DEBUG_SDUMP} = \{ \text{YES} | \text{NO} \}
\]

**DISPCUU**

See “DISPLAY_CUU({Yes|No})” on page 56.

Syntax

\[
\text{DISPCUU} = \{ \text{YES} | \text{NO} \}
\]

**EMUL_TYPE**

See “EMUL_TYPE({ALL|HARDLINK|SNAPVX})” on page 56.

Syntax

\[
\text{EMUL_TYPE} = \{ \text{ALL} | \text{HARDLINK} | \text{SNAPVX} \}
\]

**FREEUNLK**

See “FREE({Yes|No})” on page 57.

Syntax

\[
\text{FREEUNLK} = \{ \text{YES} | \text{NO} \}
\]
GROUP_DSNAME

See “GROUP_DATaset_name('dataset_name')” on page 84.

You can specify an unlimited number of dataset names. When allocated, they are concatenated in order.

You can override any group dataset specification in the site options table by using a different dataset specification in the GROUP_DSNAME(dataset_name) parameter of the GLOBAL command.

Syntax

GROUP_DSNAME(dataset_name, dataset_name, ...)

MESSAGE

See “MESsages({DISPLAY|PROMpt|NONE|DETAIL})” on page 59.

Syntax

MESSAGE={DISPLAY|PROMPT|NONE|DETAIL}

REFVTOC

See “REFVTOC({Yes|No})” on page 61.

Syntax

REFVTOC={YES|NO}

REPLACE

See “REPLace({Yes|No})” on page 62.

Syntax

REPLACE={YES|NO}

RESTORE_CREATE

See “REStore_create({Yes|No})” on page 92.

Syntax

RESTORE_CREATE={YES|NO}

SECURE

See “SECure({Yes|No})” on page 63.

Syntax

SECURE={YES|NO}
## Configuration

### Secure_Query

See “SECURE_Query(Yes|No)” on page 63.

**Syntax**

SECURE_QUERY = {YES | NO}

### ShowDef

See “SHOW_Tracks_to_be_defined(Yes|No)” on page 63.

**Syntax**

SHOWDEF = {YES | NO}

### Snapshot_List

See “SNAPSHOT_LIST(filter1[,filter2,...,filterN])” on page 63.

**Syntax**

SNAPSHOT_LIST = {ALL | LINKED | NOT_LINKED | SNAPSHOT | ACTIVATED | NOT_ACTIVATED | COPY | NOCOPY | DEFINED | NOT_DEFINED | SECURE | NOT_SECURE | ZDP | NOT_ZDP}

### Snapshot_Name

See “NAME(snapshot_name[%date[(4|6|8)][%time[(4|6)]])” on page 60.

**Syntax**

SNAPSHOT_NAME = snapshot_name

### SoftLink

See “SOFTlink(Yes|No)” on page 86.

**Syntax**

SOFTLINK = {YES | NO}

### SoftRestore

See “SOFTRestore(Yes|No)” on page 64.

**Syntax**

SOFTRESTORE = {YES | NO}

### Srdfa_Retry

See “SR DFA_CONSISTENT_RETRY(Yes|No|count)” on page 66.
**Syntax**

\[
\text{SRDFA\_RETRY} = (\text{YES} | \text{NO} | \text{count})
\]

**Note:** The default value is 10.

**SRPPERCT**

See “SRP\_PERCENT(srp-percentage,{INFO|WARN|ERROR|SKIP})” on page 87.

**Syntax**

\[
\text{SRPPERCT} = \text{srp-percentage}
\]

**Note:** The default value is 100.

**SRPMSGVL**

See “SRP\_PERCENT(srp-percentage,{INFO|WARN|ERROR|SKIP})” on page 87.

**Syntax**

\[
\text{SRPMSGVL} = \{\text{INFO} | \text{WARN} | \text{ERROR} | \text{SKIP}\}
\]

**TERM\_ALL**

See “TERM\_inate\_ALL(\{Yes|No\})” on page 111.

**Syntax**

\[
\text{TERM\_ALL} = \{\text{YES} | \text{NO}\}
\]

**TIMEOUT**

See “TIMEOUT(nnn)” on page 68.

**Syntax**

\[
\text{TIMEOUT} = \text{number\_of\_seconds}
\]

**VARYOFF**

See “VARY\_OFFline(\{AUTO|NEVER\})” on page 68.

**Syntax**

\[
\text{VARYOFF} = \{\text{AUTO} | \text{NEVER}\}
\]

**VARYON**

See “VARY\_ONline(\{AUTO|Yes|No\})” on page 69.

**Syntax**

\[
\text{VARYON} = \{\text{AUTO} | \text{NEVER}\} 
\]
Configuration

**VCLOSE**

See “VCLOSE({Yes|No})” on page 69.

**Syntax**

VCLOSE = (YES | NO)

**UNLNKAFT**

See “UNLINK_After_copy({Yes|No})” on page 92.

**Syntax**

UNLNKAFT = (YES | NO)

**WFDEF**

See “WAIT_FOR_Definition({Yes|No})” on page 69.

**Syntax**

WFDEF = (YES | NO)
CHAPTER 3
Getting Started

This chapter covers the following topics:

◆ TimeFinder SnapVX and TimeFinder/Clone Mainframe Snap Facility ................... 32
◆ Post-installation................................................................................................... 32
◆ Software interoperability considerations .............................................................. 32
◆ Starting ResourcePak Base.................................................................................. 33
◆ Setting up security............................................................................................... 33
◆ Running TimeFinder/Clone Mainframe Snap Facility ............................................ 34
TimeFinder SnapVX and TimeFinder/Clone Mainframe Snap Facility

SnapVX is a feature of TimeFinder/Clone Mainframe Snap Facility. As such, it uses the TimeFinder/Clone Mainframe Snap Facility program EMCSNAP and its global configuration parameters.

For information about other features of TimeFinder/Clone Mainframe Snap Facility, see the TimeFinder/Clone Mainframe Snap Facility Product Guide.

Post-installation

To use SnapVX, you need to install Mainframe Enablers and enable the TimeFinder/Clone component of TimeFinder/Clone Mainframe Snap Facility, as described in the Mainframe Enablers Installation and Customization Guide.

Software interoperability considerations

SIBBATCH

If you plan to run with full IBM SNAPSHOT compatibility, place the Dell EMC supplied SIBBATCH in the search list ahead of the IBM supplied SIBBATCH.

SIBBATCH uses the same DD statements as SNAPSHOT. If you want any TimeFinder functions, code the statement exactly as it is coded in TimeFinder. For instance, a GLOBAL statement could be added to the input stream.

SRS

If the installation uses the SRS (Space Management Software) from DTS Software, you should exclude TimeFinder from SRS recovery by coding the following rule:

```
DEFRULE NOEMC
IF PGM=EMCSNAP
THEN EXIT
```

FlashCopy

IBM 2105 and 2107 controllers are recognized, and IBM FlashCopy is automatically invoked as a datamover, if appropriate. TimeFinder recognizes FlashCopy V2 support and uses FlashCopy V2 to copy dataset extents.

TimeFinder recognizes whether a storage system is FlashCopy capable. It also recognizes when a FlashCopy session is active at the logical volume level.

FlashCopy and Snap sessions can coexist and the value in the site options table for &EMCDSSU_FLASH_SNAP is always used. This means that if the site options table has Snap as the preferred copy method (&EMCDSSU_FLASH_SNAP=SNAP) and a FlashCopy session already existed on the device, TimeFinder would use Snap.

**Note:** A FlashCopy target cannot be used as a SnapVX source.
Starting ResourcePak Base


By default, TimeFinder/Clone Mainframe Snap Facility expects that ResourcePak Base runs as a task named EMC$SCF. If your instance of ResourcePak Base has another name, specify it in the SCF$nnnn DD statement of the TimeFinder/Clone Mainframe Snap Facility job.

If ResourcePak Base is not running, issuing any TimeFinder/Clone Mainframe Snap Facility command results in an error message stating that EMCSCF is not available.

Setting up security

TimeFinder/Clone Mainframe Snap Facility uses SAF calls to validate access to resources. This feature is turned on by default. The source code for the TimeFinder/Clone Mainframe Snap Facility SAF interface routine is provided in the SAMPLIB library of ResourcePak Base, so that you can tailor it to your specific needs.

Note: The Mainframe Enablers Installation and Configuration Guide describes the Security Interface and the class and resource names used.
Running TimeFinder/Clone Mainframe Snap Facility

You normally run TimeFinder/Clone Mainframe Snap Facility as a batch job. The JCL is as follows:

```
//EMCSNAP EXEC PGM=EMCSNAP,REGION=0M
//STEPLIB DD DISP=SHR,DSN=ds_prefix.LINKLIB
//SYSABEND DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCERROR DD SYSOUT=* 
//SCF$nnnn DD DUMMY 
//QCINPUT DD *
```

Where:

- `ds_prefix` is the product dataset name prefix you specified during installation of Mainframe Enablers as described in the Mainframe Enablers Installation and Customization Guide.
- `nnnn` identifies the ResourcePak Base task that the TimeFinder/Clone Mainframe Snap Facility job runs against.
- The QCOUTPUT DD statement defines the QCOUTPUT report file described in the TimeFinder/Clone Mainframe Snap Facility Product Guide.
- When the QCERROR DD statement is specified, any error messages generated are written to the QCERROR file, as well as to the QCOUTPUT report file.
- The QCINPUT DD statement can reference a disk file: DISP=SHR, DSN=dsn.

**Note:** You can also use the DD statement //EMCGROUP DD to point to the statement group dataset. “Defining groups of statements” on page 47 discusses groups of statements.
CHAPTER 4
Operations

This chapter covers the following topics:

◆ Softlinked and hardlinked snapshots ................................................................. 36
◆ Snapshot life cycle ............................................................................................. 36
◆ Secure snaps ..................................................................................................... 39
◆ Summary of operations .................................................................................... 41
◆ Example of operations ..................................................................................... 46
◆ Defining groups of statements ......................................................................... 47
Softlinked and hardlinked snapshots

The TimeFinder term, **softlinked snapshot** or **soft snapshot**, refers to a copy that has been processed using the new “targetless” replication technology that allows you to perform a point-in-time copy using thin devices, without specifying a target destination device. The term “soft” refers to the fact the snapshot was created without an association to a target device. This snapshot can be linked and unlinked to multiple target devices.

**Note:** All softlinked snapshots require PowerMaxOS 5978 or HYPERMAX OS 5977.

The TimeFinder term, **hardlinked snapshot** or **hard snapshot**, refers to a snapshot that is created with a specific source and target device.

**Note:** For hardlinked snapshots, any VDEV operations are set to VSE.

With PowerMaxOS 5978, both softlinked and hardlinked snapshots can be created on the same device.

Snapshot life cycle

You create and manage SnapVX snapshots with the following commands:

- **CREATE**—A snapshot is created with the CREATE command.
- **ACTIVATE**—The point-in-time image of the source devices is saved to the created snapshot, once the snapshot is activated.
- **RENAME**—The snapshot can be renamed with the RENAME command.
- **LINK**—A created and activated snapshot is not usable for reads or writes until it is associated with a target device using the LINK command.
- **UNLINK**—Target devices are disassociated with a snapshot using the UNLINK command.
- **TERMINATE**—Snapshots are deleted with the TERMINATE command, provided they are not linked to a target device.

These new commands are full volume operations only. **Table 2** shows which commands can be used in specific snapshot states.

**Table 2** Allowable SnapVX commands per snapshot state

<table>
<thead>
<tr>
<th>SnapVX commands</th>
<th>NONE</th>
<th>CREATE</th>
<th>ACTIVATED</th>
<th>LINKED</th>
<th>UNLINKED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVATE</td>
<td></td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RENAME</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>
Table 2  Allowable SnapVX commands per snapshot state

<table>
<thead>
<tr>
<th>SnapVX commands</th>
<th>Snapshot states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NONE</td>
</tr>
<tr>
<td>LINK</td>
<td></td>
</tr>
<tr>
<td>UNLINK</td>
<td></td>
</tr>
<tr>
<td>TERMINATE</td>
<td></td>
</tr>
</tbody>
</table>

SnapVX group status transitions

Table 3 shows SnapVX group status transitions.

Note: “Defining groups of statements” on page 47 describes groups.

Table 3  SnapVX group actions and statuses

<table>
<thead>
<tr>
<th>Action</th>
<th>Initial status</th>
<th>Resulting status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE</td>
<td>INITIAL</td>
<td>INITIAL</td>
</tr>
<tr>
<td>CREATE</td>
<td>INITIAL</td>
<td>CREATED</td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>CREATED</td>
<td>PRESNAP, POSTSNAP, or ACTIVATE</td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>PRESNAP</td>
<td>ACTIVATE</td>
</tr>
<tr>
<td>LINK</td>
<td>PRESNAP, POSTSNAP, or ACTIVATE</td>
<td>LINKED</td>
</tr>
<tr>
<td>UNLINK</td>
<td>LINKED</td>
<td>UNLINKED</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>UNLINKED</td>
<td>INITIAL</td>
</tr>
<tr>
<td>QUERY</td>
<td>INITIAL</td>
<td>INITIAL</td>
</tr>
<tr>
<td>FREE</td>
<td>INITIAL</td>
<td>INITIAL</td>
</tr>
</tbody>
</table>

To view group status history, use the QUERY GROUP command.

Target volume track definition

After a link operation has completed, a background task is started to “define” each track on the target volume. The definition process changes each track in the target volume to point to the corresponding track of its linked snapshot.

Once a track has been defined, it will remain pointing to this new track even after the unlink process. The end result is that a fully defined target device, linked or unlinked, will look like the source device at the time the snapshot was created.

This definition process is not a physical copy operation, but rather a manipulation of in memory track pointers. If the link operation was done in MODE(NOCOPY), then the target and snapshot will be sharing physical tracks until a write is performed on the
target device. If the snapshot was linked in MODE(COPY), a subsequent background copy process will copy the snapshot data and assign these new unshared tracks to the target.

You can determine whether SnapVX will wait for all of the target tracks to be defined before unlinking the target device. To do this, use the WAIT_FOR_Definition parameter of the UNLINK (or TERMINATE) command.

**Monitoring SRP usage during linking**

SnapVX can monitor used capacity of the storage resource pool (SRP) the link target device is in and display an informational, warning, or error message when the user-defined capacity limit is exceeded.

To set up SRP capacity monitoring, use the SRP_PERCENT parameter of the GLOBAL command. You can specify the SRP used capacity threshold and select the severity level of the message to be displayed.

SRP summary information is displayed when SRP capacity monitoring is enabled, for example:

```
ESNP1L0I SRP# 000 USAGE INFORMATION FOR CUU:6400 SER#:000123456789
MHOP:FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ESNP1L0I CAPACITY/USED TRKS:196761600/7596656 SRP % USED
TOT/CKD/FBA:3%/4%/0% SNAPSHOT TRKS:193764 RESRVD CAP:10%
```

When the SRP is above the set capacity limit, a message similar to the following is displayed:

```
ESNP03GW TARGET SRP# 00000102 IS 40% FULL
```

If SRP processing is not required, you can skip it by specifying the SKIP option. When SKIP is specified, no SRP information is obtained or displayed.
Secure snaps

Secure snapshots is an optional setting for SnapVX targetless snapshots that prevent accidental or intentional deletion of snapshots. Terminating a secure snapshot while the retention period is in effect is not allowed. When the retention time is reached, the snapshot is automatically terminated only if there are no existing links or restore sessions. Extending the retention period is allowed but reducing it is not allowed. A non-secure snapshot can be converted to a secure snapshot, but a secure snapshot may not be converted to a non-secure snapshot. All SnapVX operations and rules for non-secure snapshots regarding restores, linked target operations, and automatic expiration also apply to secure snapshots.

⚠️ CAUTION ⚠️

When implementing secure snapshots, you should determine how many snapshots on a storage system must be secure. Consider using secure snapshots only on certain critical volumes, or only on a subset of the snapshots, to capture particular points-in-time that are critical to the business. Length of retention time should also be considered as part of the secure snapshot implementation. As always, proper planning and system sizing is crucial, no matter the types or number of snapshots that will exist in an environment.

Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See Knowledgebase article 498316 for more information.

Secure snapshots rules and allowable SnapVX operations:

- The secure expiration time is available at the snapshot level.
- The secure expiration time cannot be zero.
- If the secure expiration time expires and the snapshot has no links or restore session, the snapshot is automatically deleted by the system.
- If the secure expiration time expires and the snapshot has link to a device or has a restore session, the snapshot is automatically deleted by the system only after the linked target is unlinked or the restore session is terminated.
- If the secured snapshot expiration time expires and the snapshot is linked to a device or has a restore session, the snapshot is no longer secured and is identified as expired.
- A non-secure snapshot can be converted to secure snapshot by specifying a secure expiration time.
- A secure expiration time can only be extended forward in time and not set back in time from the current setting.
- A secure snapshot can be restored, renamed, linked, unlinked, set copy mode, and terminate a restore.
- A secure snapshot that fails can be terminated.

SnapVX operations not allowed on secure snapshots:

- Terminating the snapshot if it is activated
Operations

- Setting the secure expiration time back in time from the current setting

Secure snapshot rules and restrictions include:

- PowerMaxOS 5978 or HYPERMAX OS 5977 Q217SR is required to use secure snapshots.

- No new secure snapshots can be created when the SRP (Storage Resource Pool) is above the RC (Reserve Capacity).

- Secure snapshot rules and restrictions apply to both FBA and CKD devices.

- Secure snapshots are also unique in the handling of host writes and snapshots when SRP or Replication Cache limits are reached. Secure snapshots are intended for use in cases where Data Pointer-specific point-in-time images are considered the highest priority. When Reserved Capacity of an SRP is reached and a host write to a source volume requires a new allocation in the SRP, secure snapshots (unlike non-secure snapshots) will be allowed to grow past the Reserved Capacity limit of the SRP.

Preserving the secure snapshots will also take priority over host writes to those source volumes in the event the entire SRP runs out of available capacity. This is the intent of the secure snapshots. For example, if there were an intentional attack or a runaway application on a system that was using up all of the available capacity, preserving the secure snapshots will allow you to restore from the secure snapshots once the situation has been resolved.

Secure snapshots are shown in the QUERY SNAPSHOT MULTI_LINE_query(YES) output as follows:

```
ESNP1AAI SECURE SNAPSHOT: YES
```

To make a snapshot secure, set the SECure(YES) parameter of the ACTIVATE, CONFIG or GLOBAL command.

To set the Time to Live (TTL) value for a snapshot, specify the EXPIration parameter of the ACTIVATE or CONFIG command. Note that a secure snapshot must have a defined and non-zero TTL.

To view current TTL for a snapshot, check the “TIME TO LIVE” value in the QUERY SNAPSHOT MULTI_LINE_query(YES) output.
Summary of operations

Managing snapshots

Table 4 lists operations for managing snapshots.

Table 4  Managing snapshots

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create snapshot</td>
<td>• ACTIVATE command</td>
</tr>
<tr>
<td></td>
<td>• LINK command, AUTO_CREATE parameter</td>
</tr>
<tr>
<td>Activate snapshot</td>
<td>ACTIVATE command</td>
</tr>
<tr>
<td>Activate secure snapshot</td>
<td>• ACTIVATE command, SECure parameter</td>
</tr>
<tr>
<td></td>
<td>• CONFIG command, SECure parameter</td>
</tr>
<tr>
<td></td>
<td>• GLOBAL command, SECure parameter</td>
</tr>
<tr>
<td>Set snapshot expiration date</td>
<td>CONFIG command, EXPIRATION parameter</td>
</tr>
<tr>
<td>Rename snapshot</td>
<td>RENAME command</td>
</tr>
<tr>
<td>Link snapshot to device</td>
<td>LINK command</td>
</tr>
<tr>
<td>Unlink snapshot from device</td>
<td>UNLINK command</td>
</tr>
<tr>
<td>Delete snapshot</td>
<td>TERMINATE command</td>
</tr>
</tbody>
</table>

Viewing snapshots

Table 4 lists operations for viewing snapshots.

Table 5  Viewing snapshots

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>View snapshot(s)</td>
<td>QUERY_SNAPSHOT command</td>
</tr>
<tr>
<td>View only secure snapshots</td>
<td>• GLOBAL command, SECURE_Query parameter</td>
</tr>
<tr>
<td></td>
<td>• QUERY_SNAPSHOT command, SECURE_Query parameter</td>
</tr>
<tr>
<td></td>
<td>• &amp;SECURE_QUERY site option</td>
</tr>
<tr>
<td>View snapshots by emulation type</td>
<td>• GLOBAL command, EMUL_TYPE parameter</td>
</tr>
<tr>
<td></td>
<td>• QUERY_SNAPSHOT command, EMUL_TYPE parameter</td>
</tr>
<tr>
<td></td>
<td>• &amp;EMUL_TYPE site option</td>
</tr>
<tr>
<td>View snapshots by linking status</td>
<td>• GLOBAL command, SNAPSHOT_LIST parameter</td>
</tr>
<tr>
<td></td>
<td>• QUERY_SNAPSHOT command, SNAPSHOT_LIST parameter</td>
</tr>
<tr>
<td></td>
<td>• &amp;SNAPSHOT_LIST site option</td>
</tr>
<tr>
<td>Include detail query lines for each snapshot</td>
<td>• GLOBAL command, MULTI_LINE_query parameter</td>
</tr>
<tr>
<td></td>
<td>• QUERY_SNAPSHOT command, MULTI_LINE_query parameter</td>
</tr>
<tr>
<td>Include status summary for each snapshot</td>
<td>• GLOBAL command, COMPACT_query parameter</td>
</tr>
<tr>
<td></td>
<td>• QUERY_SNAPSHOT command, COMPACT_query parameter</td>
</tr>
</tbody>
</table>
Operations

Table 5  Viewing snapshots

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include CUU information</td>
<td>• QUERY_SNAPSHOT command, DISPLAY_CUU parameter</td>
</tr>
<tr>
<td>Sort snapshots by date</td>
<td>• QUERY_SNAPSHOT command, SORT parameter</td>
</tr>
<tr>
<td>View tracks to be freed up</td>
<td>• QUERY_FREE command</td>
</tr>
</tbody>
</table>

Viewing groups

Table 4 lists operations for viewing groups.

Table 6  Viewing groups

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>View contents of one or all groups</td>
<td>QUERY_GROUP command</td>
</tr>
<tr>
<td>Include syntax statements and/or recent action and resulting statuses</td>
<td>QUERY_GROUP command, LIST parameter</td>
</tr>
</tbody>
</table>

Fine-tuning snapshot processing

Table 7 lists operations for fine-tuning snapshot processing.

Table 7  Fine-tuning snapshot processing

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up snapshot post-processing</td>
<td>ACTIVATE command, POSTSNAP parameter</td>
</tr>
<tr>
<td>Set up snapshot pre-processing</td>
<td>ACTIVATE command, PRESNAP parameter</td>
</tr>
</tbody>
</table>

Linking/unlinking and copying

Table 8 lists operations for linking/unlinking and copying.

Table 8  Linking/unlinking and copying

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable/disable softlinking</td>
<td>• GLOBAL command, SOFTlink parameter &amp;SOFTLINK site option</td>
</tr>
<tr>
<td>Enable/disable soft restore</td>
<td>• GLOBAL command, SOFTRestore parameter &amp;SOFTRESTORE site option</td>
</tr>
<tr>
<td>Select background copy mode</td>
<td>• CONFIG command, MODE parameter • GLOBAL command, MODE parameter &amp;LINK command, MODE parameter</td>
</tr>
<tr>
<td>Link and overwrite data on existing target device</td>
<td>• GLOBAL command, REPLACE parameter &amp;REPLACE site option</td>
</tr>
<tr>
<td>Monitor target SRP used capacity when linking</td>
<td>GLOBAL command, SRP_PERCENT parameter</td>
</tr>
</tbody>
</table>
Table 8  Linking/unlinking and copying

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create snapshot of source when linking target back to source</td>
<td>LINK command, REStore_create parameter</td>
</tr>
</tbody>
</table>
| Unlink after full copy | • LINK command, UNLINK_After_copy parameter  
• GLOBAL command, UNLINK_After_copy parameter  
• &UNLNAKFT site option |
| Unlink after terminate | • GLOBAL command, AUTO_UNLink parameter  
• TERMINATE command, AUTO_UNLink parameter  
• &AUTO_UNLINK site option |
| Wait for target track definition before unlinking | • GLOBAL command, WAIT_FOR_DEFINITION parameter  
• UNLINK command, WAIT_FOR_DEFINITION parameter  
• TERMINATE command, WAIT_FOR_DEFINITION parameter  
• &WFDEF site option |
| Clear in-memory track pointers of a device | • FREE command  
• UNLINK command, FREE parameter  
• GLOBAL command, FREE parameter  
• TERMINATE command, AUTO_UNLink option with FREE parameter  
• &FREEUNLK site option |

Setting up consistency

Table 9 lists operations for setting up consistency.

Table 9  Setting up consistency

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Use ECA | • ACTIVATE command, CONSISTENT parameter  
• GLOBAL command, CONSISTENT parameter  
• &CONSIST site option |
| Set up consistency when using ECA | • ACTIVATE command, SRDFA_CONSISTENT_RETRY parameter  
• GLOBAL command, SRDFA_CONSISTENT_RETRY parameter  
• &SRDFA_RETRY site option |
| Set ECA timeout | • ACTIVATE command, TIMEOUT parameter  
• GLOBAL command, TIMEOUT parameter  
• &TIMEOUT site option |
## Controlling device status

Table 10 lists operations for controlling device status.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Ensure target device is offline before processing | - GLOBAL command, VARY_OFFline parameter  
- LINK command, VARY_OFFline parameter  
- UNLINK command, VARY_OFFline parameter  
- &VARYOFF site option                          |
| Ensure target device is online after processing  | - GLOBAL command, VARY_ONline parameter  
- LINK command, VARY_ONline parameter  
- UNLINK command, VARY_ONline parameter  
- &VARYON site option                            |
| Check target online/offline status of paths to other CPUs | - GLOBAL command, CHECKONLINEpathstatus parameter  
- LINK command, CHECKONLINEpathstatus parameter  
- &CHKONLIN site option                          |
| Make target device ready                        | - LINK command, READY parameter  
- UNLINK command, READY parameter  
-                                                     |

## Conditioning target volume

Table 11 lists operations for conditioning the target volume.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Condition new target volume                    | - GLOBAL command, CONDITIONVOLUME and COPYVOLID parameters  
- LINK command, CONDITIONVOLUME and COPYVOLID parameters  
- &CONDVOL and &COPYVOL site options              |
| Rebuild VTOC of the larger target volume       | - GLOBAL command, REFVTOC parameter  
- LINK command, REFVTOC parameter  
- &REFVTOC site option                             |
| Issue VCLOSE CAS for target volume             | - GLOBAL command, VCLOSE parameter  
- LINK command, VCLOSE parameter  
- UNLINK command, VCLOSE parameter  
- VCLOSE site option                                |
| Create a new volume ID                         | - LINK command, NEWVOLID parameter  
- UNLINK command, NEWVOLID parameter               |
## Miscellaneous SnapVX settings

*Table 12* lists operations to make miscellaneous SnapVX settings.

**Table 12** Miscellaneous SnapVX settings

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up subtasking</td>
<td>- <code>ACTIVATE</code> command, <code>ACTIVATE_SUBTASK#</code> parameter</td>
</tr>
<tr>
<td></td>
<td>- <code>GLOBAL</code> command, <code>ACTIVATE_SUBTASK#</code> parameter</td>
</tr>
<tr>
<td></td>
<td>- <code>&amp;ACTIVATE_SUBTASK#</code> site option</td>
</tr>
<tr>
<td>Set command processing type</td>
<td><code>GLOBAL</code> command, <code>TYPRUN</code> parameter</td>
</tr>
<tr>
<td>Allow or prohibit command execution</td>
<td>- <code>GLOBAL</code> command, <code>ALLOW_FBA_META</code> parameter</td>
</tr>
<tr>
<td>against FBA meta devices</td>
<td>- <code>&amp;ALLOW_FBA_META</code> site option</td>
</tr>
<tr>
<td>Improve job completion times for jobs</td>
<td>- <code>GLOBAL</code> command, <code>CACHE_FULL_SYM</code> parameter</td>
</tr>
<tr>
<td>with &gt;500 devices</td>
<td>- <code>CACHESYM</code> site option</td>
</tr>
<tr>
<td>Set up messaging</td>
<td>- <code>ACTIVATE</code> command, <code>MESSages</code> parameter</td>
</tr>
<tr>
<td></td>
<td>- <code>GLOBAL</code> command, <code>MESSages</code> parameter</td>
</tr>
<tr>
<td></td>
<td>- <code>&amp;MESSAGE</code> site options</td>
</tr>
<tr>
<td></td>
<td>- <code>GLOBAL</code> command, <code>MAXRC</code> parameter</td>
</tr>
<tr>
<td>Turn on/off debugging</td>
<td>- <code>GLOBAL</code> command, <code>DEBUG</code> parameter</td>
</tr>
<tr>
<td></td>
<td>- <code>&amp;DEBUG_SDUMP</code> site option</td>
</tr>
<tr>
<td></td>
<td>- <code>&amp;DEBUG_ERROR</code> site option</td>
</tr>
<tr>
<td>View global settings</td>
<td><code>QUERY GLOBAL</code> command</td>
</tr>
<tr>
<td>View site options</td>
<td><code>QUERY GROUP</code> command</td>
</tr>
</tbody>
</table>
Example of operations

This example shows how to create a consistent point-in-time snapshot, link it, unlink and terminate the snapshot.

Note: For detailed description of command use and syntax, see Chapter 5, “Command Reference.”

1. Create the snapshot:

   CREATE {
   NAME(TESTSNAPSHOT) -
   SOURCE(SYMDV#(1D0-1D2)) -
   LOCAL(UNIT(6200)) -
   }

   CREATE {
   NAME(TESTSNAPSHOT) -
   SOURCE(SYMDV#(1D3)) -
   LOCAL(UNIT(6200)) -
   }

2. Activate the snapshot and make it consistent:

   ACTIVATE (CONSISTENT(YES))

3. Link the volumes to target volumes:

   LINK {
   NAME(TESTSNAPSHOT) -
   SOURCE(SYMDV#(1D0-1D3)) -
   TARGET(SYMDV#(1D4-1D7)) -
   LOCAL(UNIT(6200)) -
   MODE(NOCOPY) -
   }

4. After you are done using the target volumes, you can remove the source to target link:

   UNLINK {
   TARGET(SYMDV#(1D4-1D7)) -
   LOCAL(UNIT(6200)) -
   }

5. If necessary, delete the snapshot on the source:

   TERMINATE {
   NAME(TESTSNAPSHOT) -
   SOURCE(SYMDV#(1D0-1D3)) -
   LOCAL(UNIT(6200)) -
   }
Defining groups of statements

TimeFinder SnapVX allows you to define a group of GLOBAL and LINK statements, store them in a group dataset, and then use that group as an argument to CREATE, ACTIVATE, CONFIG, LINK, QUERY SNAPSHOT, TERMINATE and UNLINK commands.

The steps in defining and using groups are as follows:

1. Specifying the group dataset to store the groups
2. Defining the groups
3. Using the groups as arguments to SnapVX commands

Specifying the group dataset

Before you define groups, you must define the group dataset. The group dataset can be a partitioned dataset (PDS) or partitioned dataset extended (PDS/E).

The group dataset must have the following characteristics:

- RECFM=FB
- LRECL=80
- BLKSIZE=8880 (Or any valid multiple of 80)

The amount of space needed is entirely dependent on expected usage. Secondary space is not recommended.

Two members are maintained for each group:

- The first member to hold the syntax
- The second member to hold the status and history

You should plan to use one directory block for every three groups. If the dataset fills, allocate a new dataset and copy all of the members from the old dataset into the new dataset.

You can specify the group dataset in three ways:

- Use the GROUP_DATaset_name parameter of the GLOBAL command.
- Use the DD statement //EMCGROUP DD to point to the group dataset.

  Note: You can concatenate multiple datasets together with EMCGROUP.

- Define GROUP_DSNAME in the site options table EMCSNAPO, as described in “GROUP_DSNAME” on page 27.

  Note: If you use a PDS, you are responsible for regularly compressing the dataset. If you use a PDSE, compressing is not necessary.

Each group stores two members into the group dataset. One contains the syntax or source statements. The other contains the group description and history.
Of the two members, one is all uppercase letters and the other is lowercase letters. The uppercase member contains the group history. The lowercase member contains the group syntax.

Each member has a cyclic redundancy check (CRC) calculated. If the CRC doesn’t match, the group is unusable. This process prevents the members from being directly changed without the status also being updated.

**Group dataset allocation example**

The following example allocates a group dataset.

```plaintext
//IEFBR14 EXEC PGM=IEFBR14
//GROUP DD DSN=group.dataset.name,DISP=(NEW,CATLG),
//UNIT=3390,SPACE=(CYL,(3,3,90)),VOL=SER=volser,
//DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
```

**Defining groups**

Use the DEFINE GROUP command to create or change a group definition. TimeFinder stores each group you define in a member in the group dataset under the group name you give it. TimeFinder performs simple syntax validation of the statements in the group when it adds the group to the group dataset library.

You cannot edit a group you have already created. Instead, issue the DEFINE GROUP command (with the REPLACE parameter) to re-specify all the commands within the group.

All commands that follow (until an END GROUP command is encountered) are considered to be part of the group definition. Only GLOBAL and LINK commands may be part of the group definition. The END GROUP command ends the group definition, and may be followed by other statements, including statements that refer to the group, or statements defining other groups.

The DELETE GROUP allows you to delete a group from the group dataset.

**Example**

```
DEFINE GROUP EMCZDP (REPLACE(YES))

GLOBAL
  AUTOMATIC_RELEASE(Y) CLEAN_DIFF(YES) -
  VARY_OFFLINE(AUTO) VARY_ONLINE(AUTO) CHKO(NEVER) -
  DEBUG(API)

LINK
  (SOURCE (SYMDV# (0100)) -
  TARGET (SYMDV# (0101)) -
  NAME(EMCZDP%TIME11% ) -
  LOCAL(UNIT(8000)) )

END GROUP
```
Referencing other groups from within a group

You can use %INCLUDE in a group definition to reference other groups in that definition. The %INCLUDE causes the indicated group to be brought in to replace the actual %INCLUDE statement. Nesting is allowed, but recursion is not.

The syntax for %INCLUDE is:

```
%INCLUDE GROUP(grpname)
```

Where:

```
grpname
```

The name of the group.

The group status is handled the same way as when the group is coded on the user commands. This means that all of the group initial statuses must be correct and all of the group statuses are updated after the contents are executed.

**Note:** You cannot reference a group that is defined or deleted in the current job step because a group reference is expanded at parse time, but the DEFINE GROUP (or DELETE GROUP) is performed at processing time. This means that the group reference gets the contents of the group at the beginning of the step, before the DEFINE GROUP or DELETE GROUP is processed. Separate the DEFINE GROUP or DELETE GROUP into a different job step. This ensures that the required group contents are used by the group reference.

The following example shows how you can nest multiple group definitions:

```
DEFINE GROUP A
  %INCLUDE GROUP(B)
  %INCLUDE GROUP(C)
  ...
END GROUP
DEFINE GROUP B
  ...
  %INCLUDE GROUP(D)
  ...
END GROUP
DEFINE GROUP C
  ...
END GROUP
DEFINE GROUP D
  ...
END GROUP
```
Operations

Using groups as arguments to SnapVX commands

After you define such a group, you can handle the devices as a group in an argument of the GROUP parameter with the CREATE, ACTIVATE, CONFIG, LINK, QUERY SNAPSHOT, TERMINATE and UNLINK commands. The GROUP parameter allows you to specify multiple groups at once.

When TimeFinder encounters the GROUP parameter, it retrieves the definition for the group from the group library, along with the statements and parameters it contains. TimeFinder checks the current status of the group to ensure that the operation is appropriate for the group at that time. TimeFinder then performs the requested operation on all appropriate devices in the group.
CHAPTER 5
Command Reference

This chapter covers the following topics:

- Syntax conventions ................................................................. 52
- SnapVX commands ................................................................. 52
- Grouping commands .............................................................. 115
Syntax conventions

- Conventions listed in “Conventions used in this document” on page 10
- Keywords appear in uppercase (for example, CREATE). They must be spelled exactly as shown.
- For easy reference, keywords can be supplemented by lowercase letters to form a meaningful word (for example, SORuce). When typing a command, use only CAPITALIZED characters of any keyword.
- Aside from the characters listed in “Conventions used in this document” on page 10, you must type all other characters that are shown in the syntax statements.
- The optional parameters must be separated from each other by a blank space. If a parameter has YES and NO keywords, you can substitute ON for YES and OFF for NO.

SnapVX commands

Use SnapVX commands to create and manage snapshots.

**Note:** If the command spans multiple lines, add a dash (-) at the end of each command line except the last one.

Common parameters

This section describes parameters that are common to multiple SnapVX commands.

**ACTIVATE_SUBTASK# (nnn)**

Sets the minimum number of storage systems being activated to invoke the subtasking feature.

The subtasking feature assigns one subtask for each storage system to minimize the ECA window when multiple syscalls are required. When the ECA window is opened, all the subtasks are posted to perform the ACTIVATE at the same time. As each subtask completes, it posts to the main task. Once all subtasks have completed, the main task closes the ECA window, and the subtasks is terminated.

The number value may be set anywhere from 0 to 255. The default value is 3. Zero effectively turns off the feature. When subtasking is used, there is one subtask attached for each storage system.

The ACTIVATE_SUBTASK# parameter has a matching site option, ACTIVATE_SUBTASK#.

**AUTO_UNLink({Yes|No})**

After the TERMINATE command executes, automatically performs an UNLINK command to release the snapshot thin device from the source device.

The AUTO_UNLink parameter has a matching site option, &AUTO_UNLINK.
CHECKBCVHoldstatus({Yes|No})

When set to Yes, SnapVX checks for existing TF/Mirror holds on target devices before further processing. The default value is Yes.

**Note:** The *TimeFinder/Mirror for z/OS Product Guide* discusses TF/Mirror device holds.

Aliases of CHECKBCVHoldstatus include CHECK_BCV_hold_status, CHKBCVhold, CHECK_DEVice_hold_status, CHKDevicehold, CBHS, CHECK_TARGET_hoId_status, CHECK_TGT_hoId_status, CHECKTARGETholdstatus, CHECKTGTHoldstatus, CHKTGThold, CTHS.

The CHECKBCVHOLDSTATUS parameter has a matching site option, &CHECKBCV.

CHECKONLINEpathstatus({Yes|No|NEVER})

The CHecKOnlinepathstatus parameter checks to see if paths from other CPUs to the target device are offline or online before performing a snapshot:

- **Yes** *(Default)* Specifies that if there are paths to the target device that are online to other CPUs, issue an error message and do not snap to the target volume.
- **No** Specifies that if there are paths to the target device that are online to other CPUs, issue a warning message and proceed with the snap to the target volume.
- **NEVER** Specifies that for all commands, no check or report is issued to indicate that the device is online to other systems.

The CHeckOnlinepathstatus parameter has a matching site option, &CHKONLIN.

CHECK_SNAPSHOT_SIZE({Yes|No})

Determines whether the current size of the source device or the snapshot size is compared with the target device size when the LINK command is processed:

- **Yes** Compare the snapshot size (cylinders).
- **No** *(Default)* Compare the current size of the source device (cylinders).

CHECK_SNAPSHOT_SIZE(YES) is useful in cases where the snapshot was created before dynamic expansion of the source device.

**Note:** The *ResourcePak Base for z/OS Product Guide* discusses Dynamic Volume Expansion (DVE).

CHECK_SNAPSHOT_SIZE(YES) may impact performance. It is recommended to use CHECK_SNAPSHOT_SIZE(NO) if the snapshot and the source volume are expected to be the same size.

The CHECK_SNAPSHOT_SIZE parameter has a matching site option, &CHKSNSZ.

COMPACT_query({Yes|No})

The COMPACT_query parameter adds status fields at the end of each line to provide a single-line summary of important snapshot information:

- **Yes** Display status fields for each snapshot
- **No** *(Default)* Do not display status fields
“Compact query” on page 98 provides additional information.

CONDitionVOLUME({ALL|LaBeL|DUMP})

The CONDITIONVOLUME parameter is used with COPYVOLID(NO) to condition the new target volume so that the target volume can remain online with its original volser, or with the newvolid, if specified:

- **ALL** *(Default)* Specifies that the label, VTOC, VTOCIX and VVDS of the target volume are to be conditioned so that the volume can remain online with its original volser. Datasets on this volume may be cataloged or re-cataloged with no problem.

- **DUMP** Specifies that the label of the target volume is updated so that it retains the original and the copied volser. No changes are made to the copied VTOC, VTOCIX, and the VVDS.

- **LaBeL** Only the label of the target volume is to be retained and no changes are made to the copied VTOC, VTOCIX and VVDS. The VTOC, VTOCIX and VVDS are the same as the original source volume. This is equivalent to an ICKDSF REFORMAT command with the VOLID parameter.

The CONDITIONVOLUME parameter only applies to locally addressable volumes. CONDITIONVOLUME is ignored if you specify it on actions with the SYMDV# or LOCAL or REMOTE parameters.

When you specify COPYV(N) and CONDVOL(ALL), the following additional changes are made after successful completion of snapshot processing:

- If a VTOC index and VVDS are present and active on the target volume, TimeFinder/Clone Mainframe Snap Facility updates any records for the VTOC index and VVDS files to reflect the new names of these files. VTOC index names have the form SYS1.VTOCIX.volser and VVDS names have the form SYS1.VVDS.Vnnnnn. The volser portion of these names is the same as the volser of the target volume.

- If the volser begins with a numeric character, the default name for the VTOC index is SYS1.VTOCIX.Vnnnnn, where nnnnn is the final five characters of the target volume volser.

- TimeFinder/Clone Mainframe Snap Facility updates the VTOC records for the VTOC index and VVDS, if present, to reflect the new names for these files with the same naming conventions as for the VTOC index updates.

The CONDITIONVOLUME parameter has a matching site option, &CONDVOL.

CONSISTENT({Yes|No})

Determines whether you use Enginuity Consistency Assist (ECA) for consistent snapshot operations:

- **Yes** Use ECA for consistent SNAP VOLUME operations.
- **No** *(Default)* Do not use ECA for consistent SNAP VOLUME operations.

You cannot use the CONSISTENT parameter unless you have installed the TF/Consistency Group Licensed Feature Code.

The CONSISTENT parameter has a matching site option, &CONSIST. It can also be set as a GLOBAL parameter. For the duration of the ACTIVATE command, CONSISTENT overrides any value set by the GLOBAL command CONSISTENT parameter or by the &CONSIST site option.
COPYVolid({Yes|No})

The COPYVolid parameter determines whether the source volume volser is copied to the target volume:

Yes  (Default) Specifies that the volser of the source volume is to be retained in the snap and the target volume is to be made unavailable to the host (varied offline).

No Specifies that the original volser of the target volume is to be retained and the target volume is to be made available to the host (varied online).

The COPYVOLID parameter only applies to locally addressable volumes. COPYVOLID is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters.

When you specify COPYV(N) and CONDVOL(ALL), the following additional changes are made after successful completion of LINK processing:

- If a VTOC index and VVDS are present and active on the target volume, TimeFinder updates any records for the VTOC index and VVDS files to reflect the new names of these files. VTOC index names have the form SYS1.VTOCIX.volser and VVDS names have the form SYS1.VVDS.Vnnnnn. The volser portion of these names is the same as the volser of the target volume.

- If the volser begins with a numeric character, the default name for the VTOC index is SYS1.VTOCIX.Vnnnnn, where nnnnn is the last five characters of the volser of the target volume.

- TimeFinder updates the VTOC records for the VTOC index and VVDS, if present, to reflect the new names for these files with the same naming conventions as for the VTOC index updates.

The COPYVolid parameter has a matching site option, &COPYVOL.

DEVice({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(count)|ALL})

The DEVICE parameter defines the devices to query by using the internal PowerMax/VMAX device numbers. The resulting output includes only the devices specified. The default is no devices.

- You can specify a single PowerMax/VMAX device number:
  
  DEVICE(symdv#)

- You can also specify a range of PowerMax/VMAX device numbers. You can write a range in three ways:

  - Specify the lowest device in the range and the highest device in the range separated by a dash:
    
    DEVICE(lowsymdv#-highsymdv#)

  - Specify the lowest device in the range and the highest device in the range separated by a colon:
    
    DEVICE(lowsymdv#:highsymdv#)
– Specify the starting device in the range and a count value (in parentheses) that indicates how many additional devices there are between that device and the end of the range. The count value includes the lowest and the highest devices.

\[
\text{DEVICE}(\text{symdv#}(\text{count}))
\]

For most commands, you only use the LOCAL, REMOTE, and CONTROLLER parameters when you use the SYMDV# parameter. This is because SYMDV# identifies only a device, not its location.

The QUERY commands are different. Because the QUERY commands use a storage system as a target, you can use the LOCAL, REMOTE, and CONTROLLER parameters with them without a SYMDV# parameter being present. The various filtering parameters each QUERY command can take let you isolate the particular devices on which you want to report.

- You can also specify ALL to show information about all devices.

\[
\text{DISPLAY_CUU}((\text{Yes}|\text{No}))
\]

Determines whether to include CUU information in QUERY SNAPSHOT and QUERY FREE output:

- Yes Display CUUs for all devices in the query range
- No (Default) Do not display CUUs. ‘N/A’ is shown instead with gatekeeper volser.

Query commands can take a long time to complete when querying a large range of devices. Much of this time is spent obtaining CUU information for each device in the query range. DISPLAY_CUU(NO) bypasses this process and speeds up the command significantly.

The DISPLAY_CUU parameter is also available as a site option, &DISPCUU.

\[
\text{EMUL_TYPE}((\text{ALL}|\text{HARDLINK}|\text{SNAPVX}))
\]

The EMUL_TYPE parameter is used to restrict the list of snapshots to one or more emulation types.

- ALL (Default) List snapshots of all emulation types.
- HARDLINK List hardlink snapshots.
- SNAPVX List SnapVX (softlink) snapshots.

The EMUL_TYPE parameter is also available as a site option, &EMUL_TYPE.

\[
\text{EXPIration}(\text{days})
\]

Associates a “time-to-live” value with the snapshot. The expiration value is specified as the number of days from 0 to 400 (decimal).

The expiration time specified is relative to the command execution time, not the snapshot creation time. Specifying a value of 0, or not specifying a value at all causes the snapshot to never expire.

If SECure(YES) is specified, the expiration value for the snapshot must be non-zero.
FREE({Yes|No})

The FREE parameter determines whether to clear the in-memory track pointers of a device:

Yes: Free the tracks that were linked to this device. The device will no longer be able to access the snapshot data that it was linked to.

No: (Default) Do not free the tracks. Even after an unlink operation, the device will continue to have access to the snapshot data that it was linked to; these tracks will only be freed when no snapshot or device is sharing them.

If the target device to be unlinked is the last device or snapshot sharing these tracks, they will be returned to the free track list in the Storage Resource Pool, and this space will be freed up for future allocations.

Tracks that are allocated for snapshots and devices are only freed when no device or snapshot is sharing them. A snapshot unshares tracks only when it is terminated. Multiple snapshots can share track allocations, so all snapshots sharing a track must be terminated (and no other source or target devices are sharing the track) before the track will be returned to its pool and the space freed up and ready for reallocation.

FREEing an SRDF device or a device with any session is blocked.

The target device must be offline, or CHECKONLINEPATHSTATUS(NO) must be specified before FREEing.

The FREE parameter has a matching site option, &FREEUNLK.

GROUP(grpname[, grpname, ...])

Specifies one or more groups of TimeFinder statements that contain information about devices on which you want to perform the operation. The grpname values are the names you have previously defined with the DEFINE GROUP command. You can specify up to 127 group names, separated by commas.

When TimeFinder encounters the GROUP parameter, it retrieves the definition for the group from the group library, along with the statements and parameters.

TimeFinder checks the current status of the group (by querying all of the storage systems and devices involved) to ensure that the operation is appropriate for the group at this time. The requested operation is then performed on all appropriate devices in all groups named as if they were a single group.

LIST([[[NO]STATements] [[NO]HISTORY]])

The LIST parameter is used with QUERY GROUP commands to list or not list syntax statements ([NO]STATements) and/or recent action and resulting statuses ([NO]HISTORY) associated with a specified group:

NOSTAtements QUERY GROUP does not list the syntax statements associated with the specified group.

STAtements QUERY GROUP lists the syntax statements associated with the specified group.

NOHISTORY QUERY GROUP does not list the recent action and resulting statuses associated with the specified group.

HISTORY QUERY GROUP lists the recent action and resulting statuses associated with the specified group.
Default value

None

LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxx[| name]})] |
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxx[| name]})] |
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxx[| name]})] |
CONTROLLER({[xxxxxxx-]xxxx[| name]})
})

The LOCAL parameter identifies a gatekeeper in the local storage system that allows access to devices in that local storage system.

**Note:** You cannot use the LOCAL and REMOTE parameters in the same command.

One of the following values must be present: UNIT, VOLUME, or DDNAME. UNIT and VOLUME can be specified together, or DDNAME may be used instead.

CONTROLLER({[xxxxxxx-]xxxx[| name]})

Specifies the local storage system.

The purpose of specifying the CONTROLLER subparameter inside the LOCAL parameter is to verify the serial number of the storage system where the action is to take place.

You may specify a five-digit (xxxxx) or a 12 digit (xxxxxxx-xxxxx) serial number. Or, you may specify a logical storage system name if you previously defined that name to ResourcePak Base.

If the logical storage system name is simple in format (single-word string, all upper case and no more than 64 characters), you can specify the storage system name without quotation marks.

If the logical storage system name is mixed case or contains spaces, you must enclose it in single quotation marks.

DDNAME(ddname)

Identifies the DD statement that refers to the gatekeeper.

UNIT(cuu)

Specifies the unit address of the gatekeeper.

**Note:** The z/OS device number cannot be the unit address of a VDEV.

VOLUME(volser)

Specifies the volser of the gatekeeper.

**Note:** VOLUME(volser) allows only a single device.
MESsages({DISplay|PROmpt|NONE|DETAIL})

Controls console messages before and after the activation of a snap volume:

- **DETAIL**
  Displays console messages before and after processing, and also adds message EQCA9211. There will be one statement for each storage system where SRDF/A is suspended.

- **DISplay**
  Display console messages before and after.

- **NONE**
  *(Default)* No console messages.

- **PROmpt**
  Display WTOR on console and wait for reply before proceeding.

Aliases for MESsages include: MSGs.

The MESsages parameter has a matching site option, &MESSAGE. It can also be set as a GLOBAL parameter. For the duration of the ACTIVATE command, MESsages overrides any value set by the GLOBAL command MESsages parameter or by the &MESSAGE site option.

MODE({COPY|NOCOPY|NOCOPYRD})

Specifies when the background copy from source to target occurs:

- **COPY**
  *(Default)* The source to target background copy should begin immediately after the snap is issued.

- **NOCOPY**
  The background copy task is not initiated. As a result of the LINK operation, new track allocations for the target device will only occur when tracks on the target device are modified.

- **NOCOPYRD**
  With PowerMaxOS 5978 and HYPERMAX OS 5977, NOCOPY and NOCOPYRD are not distinguished.

**Note:** See also “COPY and NOCOPY mode” on page 17 for general information about COPY and NOCOPY modes. “Target volume track definition” on page 37 provides an explanation of how target volume tracks are defined.

The NOCOPY (or NOCOPYRD) setting cannot be used when restoring a snapshot with the LINK command. If the NOCOPY (or NOCOPYRD) value is in effect, SnapVX ignores it and uses MODE(COPY) to proceed with the restore.

MULTI_LINE_query({Yes|No})

The MULTI_LINE_query parameter displays additional detail lines beyond the single summary line for each snapshot:

- **Yes**
  Display multiple lines of information for each snapshot.

- **No**
  *(Default)* Display only the single summary line of information for each snapshot.

“Multiline query” on page 98 provides additional information.

Aliases for this parameter include MULTILINEquery and MLQ.
NAME(snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])

Specifies the 1-32 character snapshot name. snapshot_name allows upper and lower case alpha, numerics, hyphens (-) and underscores (_). If the name contains hyphens, enclose it in single quotes. Embedded spaces are not allowed. Periods are allowed only if when GLOBAL ZDP(YES) is set.

**Note:** If the NAME(snapshot_name) parameter is specified in the GLOBAL command, it is not required in the CREATE command.

The snapshot name must be unique per source device. Specifying the NAME command only, as a GLOBAL parameter, will allow for only 1 snapshot per source device until the snapshot name is changed.

To ensure a unique snapshot name per source device, append the snapshot_name with date and time variables:

- %date%—Substitutes the current date in MM_DD_YYYY format.
- %date4%—Substitutes the current date in MMDD format.
- %date6%—Substitutes the current date in MMDDYY format.
- %date8%—Substitutes the current date in MMDDYYYY format.
- %time%—Substitutes the current time in HH_MM_SS format.
- %time4%—Substitutes the current time in HHMM format.
- %time6%—Substitutes the current time in HHMMSS format.

The NAME parameter has a matching site option, &SNAPSHOT_NAME.

NEWVOLID(volser)

Creates a new volume ID using the specified volser name.

The NEWVOLID parameter only applies to locally addressable volumes. NEWVOLID is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters.

READY( {Yes | No} )

Determines whether the target device is made ready to the host:

- Yes *(Default)* The target device is made ready to the host.
- No The target device is made not ready to the host.

The Ready status indicates the devices on a channel are available for any valid operation.

If a query command is issued immediately after you execute a LINK with the READY parameter set (YES) or cleared (NO), the status of the target device may show AVAILB for up to 5 seconds.
REFVTOC({Yes|No})

Determines whether to run ICKDSF automatically when the target volume is larger than the source volume:

Yes  ICKDSF automatically runs when the target volume is larger than the source volume.

No  (Default) ICKDSF does not automatically run when the target volume is larger than the source volume.

If the target volume is physically larger than the source volume, the additional space is not usable until ICKDSF is run with the REFVTOC option. TimeFinder automatically runs ICKDSF with the REFVTOC option when you specify REFVTOC(YES). The REFVTOC feature rebuilds the VTOC and VTOCIX to reflect the real device capacity.

Note: If a volume is online to another LPAR, it should be varied offline to all other LPARS before the snap with a REFVTOC(YES) is run.

For the REFVTOC processing to be performed, the target volume must be online to a host after the LINK processing completes. Otherwise, SnapVX will not be able to perform the REFVTOC processing automatically and messages ESNP922W and ESNP923I will be issued, prompting you to run the REFVTOC processing manually.

The REFVTOC parameter applies only to locally addressable volumes. REFVTOC is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters.

After the snapshot operation completes, the target volume is always immediately available for use.

The REFVTOC parameter has a matching site option, &REFVTOC.

REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]}])]
VOLUME (volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]}])
DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]}])}

The REMOTE parameter identifies a gatekeeper in the local storage system that allows access to devices in a remote storage system.

Note: You cannot use the LOCAL and REMOTE parameters in the same command.

The REMOTE parameter only works with full-volume sessions. REMOTE does not work with remote extent sessions.

One of the following must be present: UNIT, VOLUME, or DDNAME. UNIT and VOLUME can be specified together, or DDNAME may be used instead.

CONTROLLER

Specifies a remote storage system.

The purpose of specifying the CONTROLLER subparameter inside the REMOTE parameter is to verify the serial number of the storage system where the action is to take place.
You may specify a five-digit (xxxxx) or a 12-digit (xxxxxxxx-xxxxx) serial number. Or, you may specify a logical storage system name if you previously defined that name to ResourcePak Base.

If the logical storage system name is simple in format (single-word string, all upper case and no more than 64 characters), you can specify the storage system name without quotation marks.

If the logical storage system name is mixed case or contains spaces, you must enclose it in single quotation marks.

**DDNAME(ddname)**

Identifies the DD statement that refers to the gatekeeper.

**RAGROUP(nn.nn.nn.nn)**

Identifies the path through the remote network. This can consist of up to four (4) SRDF group numbers, separated by periods.

**UNIT(cuu)**

Specifies the unit address of the gatekeeper.

**VOLUME(volser)**

Specifies the volser of the gatekeeper.

**REPLACE({Yes|No})**

REPLACE specifies whether the source device is to overwrite data on an existing target device:

- **Yes** Specifies that an existing target device with user data be overwritten.
  
  **Note:** Using REPLACE(Y) speeds up the snap operation by not querying the VTOC for user dataset names.

- **No** (Default) Specifies that an existing target device with user data not be overwritten.

If the existing device contains user data and you do not specify this parameter, the snap operation terminates. A volume with no user data is defined as one with only a VTOC, a VTOC index, and a VVDS.

If you specify YES and the target volume is not empty, old data on that volume are overwritten. Catalog entries for any datasets existing on the target volume may become invalid.

The REPLACE parameter establishes the REPLACE value for all operations.

The REPLACE parameter only applies to locally addressable volumes. REPLACE is ignored if you specify it on actions with the SYMDV# or LOCAL or REMOTE parameters.
The REPLACE parameter is valid only with the TARGET parameter.

**Note:** For information about using REPLACE with the REUSE parameter, see the *TimeFinder/Clone Mainframe Snap Facility Product Guide*.

The REPLACE parameter has a matching site option, &REPLACE.

**SECure**({Yes|No})

Determines whether the activated snapshot(s) is secure (YES) or not (NO, default).

**Note:** “Secure snaps” on page 39 describes secure snapshots.

**CAUTION**

Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See Knowledgebase article 498316 for more information.

Specifying YES causes all of the snapshots being activated to be secure.

If SECure(YES) is specified, the snapshot Time to Live (TTL) must also be specified and be non-zero. Use the EXPIration parameter of the CONFIG or ACTIVATE command to set TTL.

The SECure parameter can be used on the ACTIVAE, CONFIG and GLOBAL commands.

The SECure parameter has a matching site option, &SECURE.

**SECURE_Query**({Yes|No})

When set to YES, only secure snapshots are displayed in the QUERY SNAPSHOT command output.

**Note:** “Secure snaps” on page 39 describes secure snapshots.

**SHOW_Tracks_to_be_defined**({Yes|No})

When set to Yes, the multiline query produced by the QUERY SNAPSHOT command shows the number tracks that still need to be defined for the target device.

The SHOW_TRACKS_TO_BE_DEFINED parameter has a matching site option, &SHOWDEF.

**SNAPSHOT_LIST**(filter1[,filter2,...,filterN])

The SNAPSHOT_LIST parameter is used to restrict the list of snapshots based on the snapshot type and status.

You can specify multiple filters separated by a comma. For example, SNAPSHOT_LIST(SECURE,LINKED) displays snapshots that are both secure and linked at the same time. Specifying filters that are opposite in their meaning (for example, ACTIVATED and NOT_ACTIVATED) is not allowed and results in error message EMCP035E.
Valid values for `filter` include:

- **ACTIVATED**: List activated snapshots (both softlinked and hardlinked).
  
- **ALL** *(Default)* List all snapshots on a device.
  Specifying ALL together with another filter within the same command is not allowed and results in error message EMCP016E.

- **COPY**: List snapshots with COPY mode set between the source and the target device.

- **DEFINED**: List snapshots with fully defined target devices.
  
  **Note**: “Target volume track definition” on page 37 discusses target track definition.

- **LINKED**: List links (target device is not X'FFFFFFFF').

- **NOCOPY**: List snapshots with NOCOPY mode set between the source and the target device.
  NCOPY is an alias of NOCOPY.

- **NOT_ACTIVATED**: List snapshots that have not been activated.
  NACTIVATED is an alias of NOT_ACTIVATED.

- **NOT_DEFINED**: List snapshots with undefined target devices.
  NDEFINED is an alias of NOT_DEFINED.
  
  **Note**: “Target volume track definition” on page 37 discusses target track definition.

- **NOT_LINKED**: List not linked snapshots (target device is X'FFFFFFFF' and there is no link with this snapshot).
  NLINKED is an alias of NOT_LINKED.

- **NOT_SECURE**: List snapshots that are not secure.
  
  **Note**: “Secure snaps” on page 39 discusses secure snapshots.

  NSECURE is an alias of NOT_SECURE.

- **NOT_ZDP**: List snapshots that are not managed by zDP.
  NZDP is an alias of NOT_ZDP.

- **SECURE**: List secure snapshots.
  
  **Note**: “Secure snaps” on page 39 discusses secure snapshots.

- **SNAPSHOT**: List snapshots (target device is X'FFFFFFFF').

- **ZDP**: List zDP-managed snapshots.

The `SNAPSHOT_LIST` parameter is also available as a site option, `&SNAPSHOT_LIST`.

**SOFTRestore**{(Yes|No)}

Determines whether to unlink the snapshot which has been linked (restored) back to the source device on which the snapshot was originally created:

- **Yes**: Unlinks a logical restore operation where a snapshot has the same source and target device.
- **No** *(Default)*: Unlinks a snapshot which has a different source and target device.

Since a source device can be both the linked target of a separate device, and the linked target of its own snapshot at the same time, specifying the SOFTRESTORE parameter clarifies which linked snapshot to unlink.
The SOFTRestore parameter has a matching site option, &SOFTRESTORE.

**SOUrce**

```plaintext
({
 VOLume(volser) |
 UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) |
 SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)}) |
 })
```

Specifies the source device or devices:

**VOLUME(volser)**

The volume serial number of a device that is online to the host attempting the snapshot.

**UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})**

The unit address of one or more devices that are online to the host attempting the snapshot.

You can specify a single unit address or a range of unit addresses. A range can be specified as follows:

You can specify a range of unit addresses as follows:

- Specify the lowest address in the range and the highest address in the range separated by a hyphen or a colon: `UNIT(lowcuu-highcuu)` or `UNIT(lowcuu:highcuu)`.

- Specify the starting address in the range and a count value (in parentheses) that indicates how many additional devices there are between that number and the highest address in the range: `UNIT(cuu(count))`. The count value includes the lowest and the highest units. For example, if you want to specify units between 10 and 13, you would enter 4 as the count.

**SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)})**

Identifies the PowerMax/VMAX device number in the source storage system for the snapshot operation.

You can specify a single device number or a range of device numbers. A range can be specified as follows:

- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a hyphen or a colon: `SYMDV#(lowsymdv#:highsymdv#)` or `SYMDV#(lowsymdv#:highsymdv#)`.

- Specify the starting device number in the range and a count value (in parentheses) that indicates how many additional devices there are between that number and the highest numbered device in the range: `SYMDV#(symdv#(count))`. The count value includes the lowest and the highest numbered device. For example, if you want to specify devices between 10 and 13, you would enter 4 as the count.

**Note:** For SnapVX commands, the total number of devices in the range cannot exceed 1024.
If you use SYMDV#, you must use it throughout the operation. That is, if you use SYMDV# for the source device, you must also use SYMDV# for the target device.

If you specify SYMDV#, the UNIT and VOLUME parameters are not allowed. However, you must use the LOCAL, REMOTE, or CONTROLLER parameters when you specify SYMDV#.

- LOCAL indicates that the device resides in a local storage system.
- REMOTE indicates that the device resides in a remote storage system accessible using the path that is specified with the RAGROUP subparameter.

SRDFA_CONSISTENT_RETRY({Yes | No | count})

Controls the retry attempts when SRDF/A is not consistent. The default value is 10 (retry attempts). Yes means retry indefinitely. No means do not retry at all. There is no minimum or maximum value for count.

When the suspend is attempted, it fails if any invalids exist on any R1 device in the group (not just R1 devices related to devices being copied). If it fails and retry is allowed, a wait occurs until the current cycle trips. Then the suspend is attempted again. The number of retries is a real count, not a time value. If multiple SRDF/A groups are involved, all of them switch to a new cycle before the suspend is retried.

SRDFA_CONSISTENT_RETRY is only used if ACTIVATE with CONSISTENT(YES) is specified. Otherwise it is ignored.

The SRDFA_CONSISTENT_RETRY parameter can be set as an ACTIVATE or GLOBAL parameter.

The SRDFA_CONSISTENT_RETRY parameter is also available as a site option, &SRDFA_RETRY.

Target

      ({{
        Volume (volser) |
        Unit ({{cuu | lowcuu-highcuu | lowcuu:highcuu | cuu (count) }} | 
        SYMDV# ({{symdv# | lowsmdv#-highsymdv# | lowsmdv#:highsymdv# | symdv#(count) }} | 
      })

Specifies the source device or devices.

VOLUME (volser)

The volume serial number of a device that is online to the host attempting the snapshot.

UNIT ({{cuu | lowcuu-highcuu | lowcuu:highcuu | cuu (count) }})

The unit address of one or more devices that are online to the host attempting the snapshot.
You can specify a single unit address or a range of unit addresses. A range can be specified as follows:

You can specify a range of unit addresses as follows:

- Specify the lowest address in the range and the highest address in the range separated by a hyphen or a colon: `UNIT(lowcuu-highcuu)` or `UNIT(lowcuu:highcuu)`.

- Specify the starting address in the range and a count value (in parentheses) that indicates how many additional devices there are between that number and the highest address in the range: `UNIT(cuu(count))`. The count value includes the lowest and the highest units. For example, if you want to specify units between 10 and 13, you would enter 4 as the count.

```
SYMDV#{{symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)}}
```

Identifies the PowerMax/VMAX device number in the source storage system for the snapshot operation.

You can specify a single device number or a range of device numbers. A range can be specified as follows:

- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a hyphen or a colon: `SYMDV#(lowsymdv#:highsymdv#)` or `SYMDV#(lowsymdv#:highsymdv#)`.

- Specify the starting device number in the range and a count value (in parentheses) that indicates how many additional devices there are between that number and the highest numbered device in the range: `SYMDV#(symdv#(count))`. The count value includes the lowest and the highest numbered device. For example, if you want to specify devices between 10 and 13, you would enter 4 as the count.

If you use `SYMDV#`, you must use it throughout the operation. That is, if you use `SYMDV#` for the source device, you must also use `SYMDV#` for the target device.

If you specify `SYMDV#`, the `UNIT` and `VOLUME` parameters are not allowed. However, you must use the `LOCAL`, `REMOTE`, or `CONTROLLER` parameters when you specify `SYMDV#`.

- `LOCAL` indicates that the device resides in a local storage system.

- `REMOTE` indicates that the device resides in a remote storage system accessible using the path that is specified with the `RAGROUP` subparameter.
TIMEOUT(nnn)

Determines the maximum time ECA is active during consistent snapshot operations.

**Note:** ECA (Enginuity Consistency Assist) is a feature of the Enginuity operating environment. ECA provides an enterprise solution for ensuring dependent write consistency. ECA requires that you have the TimeFinder/Consistency Group Licensed Feature Code (parameter CONSISTENT) installed. The Mainframe Enablers Installation and Customization Guide provides more information.

nnn

A value from zero (0) to 128 seconds. When this value is exceeded, ECA is released whether the consistent snap is formed or not. A message is issued indicating that the snap is not consistent.

When nnn is 0 or 128, the default timeout value is used (15 seconds).

The TIMEOUT parameter can be set on the ACTIVATE or GLOBAL command.

The TIMEOUT parameter has a matching site option, &TIMEOUT.

UNLINK_After_copy( {Yes|No})

Determines whether to unlink the target device after the snapshot has been fully copied to the target:

**YES**

Unlink the target device after the snapshot has been fully copied to the target.

If snapshots have been created off of the target device before the copy has been completed, and UNLINK_After_copy is set to YES, the storage system waits until the last snapshot has been deleted from the target device before it automatically unlinks the target device.

**NO** *(Default)*

Do not unlink the target device.

The UNLINK_After_copy parameter can be set on the LINK or GLOBAL command.

The UNLINK_After_copy parameter has a matching site option, &UNLNKAFT.

VARY_OFFline( {AUTO|NEVER})

Determines whether the target device must be offline before the request processing begins:

**AUTO** *(Default)*

Use VARY OFFLINE when appropriate. When VARY_OFFLINE is set to AUTO and the device is already in the required state, TimeFinder takes no action.

**NEVER**

Do not vary devices offline. When VARY_OFFLINE is set to NEVER, TimeFinder also takes no action, but leaves the device in the existing state.

The VARY_OFFLINE parameter only applies to locally addressable volumes. VARY_OFFLINE is ignored if you specify it on actions with the SYMDV# or LOCAL or REMOTE parameters.

The VARY_OFFLINE parameter has a matching site option, &VARYOFF.
VARY_ONline({AUTO|Yes|No})

Determines whether the target device must be online after the requested processing completes:

AUTO  (Default) Use VARY ONLINE when appropriate. When VARY_ONLINE is set to AUTO and the device is already in the required state, TimeFinder takes no action.

Yes  Use VARY ONLINE all the time whatever the current state. When VARY_ONLINE is set to YES, TimeFinder always varies the device online regardless of the device’s current state.

No  Do not vary devices online. When VARY_ONLINE is set to NO, TimeFinder takes no action, but leaves the device in the existing state.

The VARY_ONLINE parameter only applies to locally addressable volumes. VARY_ONLINE is ignored if you specify it on actions with the SYMDV# or LOCAL or REMOTE parameters.

The VARY_ONLINE parameter has a matching site option, &VARYON. &VARYON, however, only takes the values AUTO and NEVER. (NEVER has the same meaning as NO.)

VCLOSE({Yes|No})

The VCLOSE parameter determines whether a VCLOSE operator command is issued to the CATALOG address space for the target volume:

Yes  Issue the VCLOSE CAS modify command. VCLOSE (YES) can be safely specified all the time.

No  (Default) Do not issue the VCLOSE CAS modify command.

If a catalog resides on the target volume, the CATALOG address space should be notified or subsequent problems may occur.

VCLOSE is applied only to online volumes.

VCLOSE should be used whenever there is a catalog on a volume.

The VCLOSE parameter only applies to locally addressable volumes. VCLOSE is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters.

The VCLOSE parameter has a matching site option, &VCLOSE.

WAIT_FOR_Definition({Yes|No})

Determines whether the UNLINK or TERMINATE command will wait for all of the target tracks to be defined before the unlink occurs:

Yes  (Default) Wait for the target tracks to be fully defined before unlinking.

No  Do not wait for the target tracks to be fully defined before unlinking.

For the TERMINATE command, this parameter only applies when the AUTO_UNLink parameter is specified, and a linked target device is found.

Note: “Target volume track definition” on page 37 explains the track definition process.
The WAIT_FOR_DEFINITION parameter is automatically set to NO for a command when FREE(YES) is specified for the command.

Aliases for WAIT_FOR_DEFINITION include WAIT_FOR_D and WFD.

The WAIT_FOR_DEFINITION parameter has a matching site option, &WFDEF. This parameter can be set as an UNLINK or TERMINATE parameter.
**ACTIVATE**

Makes the point-in-time copy for a snapshot which has already been created.

Activated snapshots can be used to LINK point-in-time images to target devices. A *time-to-live* value may be associated with the snapshot by using the EXPIRATION parameter.

The ACTIVATE command for the new SnapVX commands may be used at the same time with traditional TimeFinder sessions. Consistency is maintained across multiple storage systems.

If the ACTIVATE command is not explicitly specified for a SnapVX snapshot, it is automatically added to the job. The one exception to this rule is for group processing.

**Syntax**

```plaintext
ACTIVATE
([optional_parameters])
```

**Where optional_parameters are:**

- `ACTIVATE_SUBTASK#(nnn)`
- `CONSISTENT(Yes|No)`
- `EXPIration(days)`
- `GROUP(grpname[,grpname,...])`
- `MESsages({DISplay|PROmpt|NONE|DETAIL})`
- `POSTSNAP({Yes|No})`
- `PRESNAP({Yes|No})`
- `SECure({Yes|No})`
- `SRDFA_CONSISTENT_RETRY({Yes|No|count})`
- `TIMEOUT(nnn)`

**Optional parameters**

- `ACTIVATE_SUBTASK#(nnn)`
  
  See “`ACTIVATE_SUBTASK#(nnn)`” on page 52.

- `CONSISTENT(Yes|No)`
  
  See “`CONSISTENT((Yes|No))`” on page 54.

- `EXPIration(days)`
  
  See “`EXPIration(days)`” on page 56.

- `GROUP(grpname[,grpname,...])`
  
  See “`GROUP(grpname[,grpname,...])`” on page 57.

- `MESsages({DISplay|PROmpt|NONE|DETAIL})`
  
  See “`MESsages({DISplay|PROmpt|NONE|DETAIL})`” on page 59.
POSTSNAP({Yes|No})

Indicates whether snapshot post processing should be automatically performed after the ACTIVATE command is executed or as part of the LINK command processing:

Yes Perform snapshot post processing automatically after the ACTIVATE command.
No (Default) Perform snapshot post processing as part of LINK processing.

Post processing involves making the target device ready to the channel and performing any label management. If requested, the target device is also varied online.

The POSTSNAP parameter only applies to regular input (after a //QCINPUT DD * JCL statement) LINK statement that references a GROUP, and is only valid if GROUP is also specified.

If both PRESNAP and POSTSNAP parameters are not specified, and a LINK operation is initiated that is not group processing, both PRESNAP and POSTSNAP are automatically set to YES.

PRESNAP({Yes|No})

Indicates whether snapshot preprocessing should be automatically performed, before the ACTIVATE command is executed or as part of the LINK command processing:

Yes Perform snapshot preprocessing automatically before the ACTIVATE command.
No (Default) Perform snapshot preprocessing as part of LINK processing.

Preprocessing involves:

1. Validating the request.
2. Taking the target device offline.
3. Making the target device not-ready to the channel.
4. Issuing the operating environment request to pair the two devices together.

The PRESNAP parameter only applies to regular input (after a //QCINPUT DD * JCL statement) LINK statement that references a GROUP, and is only valid if GROUP is also specified.

If both PRESNAP and POSTSNAP parameters are not specified, and a LINK operation is initiated that is not group processing, both PRESNAP and POSTSNAP are automatically set to YES.
SECure({Yes|No})

See “SECure((Yes|No))” on page 63.

⚠️ CAUTION

Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See Knowledgebase article 498316 for more information.

SRDFA_CONSISTENT_RETRY({Yes|No|count})

See “SRDFA_CONSISTENT_RETRY((Yes|No|count))” on page 66.

TIMEOUT(nnn)

See “TIMEOUT(nnn)” on page 68.

For the duration of this ACTIVATE command, TIMEOUT overrides any value set by the GLOBAL command TIMEOUT parameter or by the &TIMEOUT site option.
**CONFIG**

Specifies various snapshot settings.

**Syntax**

```
CONFiG
  (  
    TaRGet
      (  
        VOLume(volser)  |  
        UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)  |  
        SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(count)  |  
      )  
    )  
  )  

Where optional_parameters are:

[CONTROLLER({[xxxxxxx-]xxxxx|name}) ]
[EXPIration(days) ]
[GROUP(grpname,,grpname,..) ]
[LOCAL({  
    UNIT(cuu)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]  
    VOLUME(volser)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]  
    DDNAME(ddname)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]  
    CONTROLLER({[xxxxxxx-]xxxxx|name}) ]
  ]
[MODE({COPY|NOCOPY|NOCOPYRD}) ]
[NAME(snapshot_name) ]
[REMOTE (RAGROUP(nn.nn.nn.nn) 
  (UNIT(cuu)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]  
  VOLUME(volser)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]  
  DDNAME(ddname)  [CONTROLLER({[xxxxxxx-]xxxxx|name}) ]
  )]
[SECure(Yes) ]
[SNAPSHOTID(snapshot_id) ]
```

**Note:** Only one of the CONTROLLER, LOCAL, or REMOTE parameters can be present. They are mutually exclusive.

**Note:** In addition to the listed optional parameters, CONFIG also has legacy (non-SnapVX) parameters described in the TimeFinder/Clone Mainframe Snap Facility Product Guide.

---

1. The CONFIG parameters have no defaults.
**Required parameters**

TaRGet

\[
\{(VOLUME(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#:symdv#(count)}) | )\}
\]

See “TaRGet ({ VOLUME(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#:symdv#(count)}) | )” on page 66.

**Optional parameters**

CONTROLLER({ [xxxxxxx-]xxxxx | name })

You can use the CONTROLLER subparameter on LOCAL or REMOTE as a separate parameter. It allows you to verify that the storage system found using the gatekeeper (and RAGROUP if REMOTE) (where the request is to take place) is the storage system you want. When you use the separate CONTROLLER parameter, you do not include the LOCAL and REMOTE parameters.

**Note:** The CONTROLLER parameter is only needed and can only be used if you use the SYMDV# parameter.

You may specify a five-digit (xxxxx) or a 12 digit (xxxxxxx-xxxxx) serial number. Or, you may specify a logical storage system name if you previously defined that name to ResourcePak Base.

If the logical storage system name is simple in format (single-word string, all upper case and no more than 64 characters), you can specify the storage system name without quotation marks. If the logical storage system name is mixed case or contains spaces, you must enclose it in single quotation marks.

EXPIration(days)

See “EXPIration(days)” on page 56.

GROUP(grpname[, grpname,...])

See “GROUP(grpname,[grpname,...])” on page 57.
LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
CONTROLLER({[xxxxxxx-]xxxxx|name})})
See “LOCAL({ UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
CONTROLLER({[xxxxxxx-]xxxxx|name})})” on page 58.

MODE({COPY|NOCOPY|NOCOPYRD})
See “MODE({COPY|NOCOPY|NOCOPYRD})” on page 59.
For the duration of the current CONFIG command, MODE overrides any value set by the GLOBAL command MODE parameter.

NAME(snapshot_name)
The name of the snapshot to be configured. See “NAME(snapshot_name[%date[4|6|8]%[%time[4|6]%])” on page 60.
If the name contains hyphens, enclose it in single quotes.
The NAME parameter is ignored when specified with the following parameters:

- MODE(COPY)

REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})])|
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})])
See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu)
[CONTROLLER({[xxxxxxx-]xxxxx|name})])|
VOLUME(volser)
[CONTROLLER({[xxxxxxx-]xxxxx|name})])|
DDNAME(ddname)
[CONTROLLER({[xxxxxxx-]xxxxx|name})])}” on page 61.

SECure(Yes)
See “SECure(Yes|No)” on page 63.

⚠️ CAUTION ⚠️
Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See Knowledgebase article 498316 for more information.

Note: For the duration of the current CONFIG command, the SECure parameter overrides any value set by the GLOBAL command SECure parameter or the &SECURE site option.

SNAPSHOTID(snapshot_id)
The ID of the snapshot to be processed.
The SNAPSHOTID parameter is ignored when specified with the following parameters:

- MODE(COPY)
CREATE

Prepares a snapshot for a new point-in-time copy of a source device.

The CREATE command does not actually perform the copy. Once a CREATE command has been issued, then the ACTIVATE command is issued to perform the copy of the source device to the created snapshot (snapshot_name). The snapshot name/source device pair must be unique, even though different source devices can have the same snapshot name.

**Note:** See the PowerMax Family Product Guide or VMAX All Flash Product Guide for information about local replication compatibility.

The CREATE commands must be in a separate job step from the DEFINE GROUP.

The CREATE command is blocked if Dynamic Volume Expansion (DVE) is active on a requested device.

**Note:** The ResourcePak Base for z/OS Product Guide provides information about DVE.

**Syntax**

```
CREATE

(NAME(snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])

SOUrce

({

VOLume(volser)

UNIT({

lowcuu-highcuu|lowcuu:highcuu|cuu(count))

SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count))})

})

[optional_parameters]

)

Where optional_parameters are:

[GROUP(grpname[, grpname,...])]

[LOCAL({

UNIT(cuu) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

VOLUME(volser) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

DDNAME(ddname) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

[REMOTE (RAGROUP(nn.nn.nn.nn)

{UNIT(cuu) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

VOLUME (volser) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])]

DDNAME(ddname) [CONTROLLER([[xxxxx-xxxxx]-xxxxx|name]])])
```

Required parameters

NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])

See “NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])” on page 60.

Note: If the NAME(snapshot_name) parameter is specified in the GLOBAL command, it is not required in the CREATE command.

SOURCE

({
  VOLUME(volser)
  UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})
  SYMDV#{symdv#:lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)})
})

See “SOURCE ({ VOLUME(volser)| UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})| SYMDV#{symdv#:lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)}) })” on page 65.

Optional parameters

GROUP(grpname[, grpname,...])

See “GROUP(grpname[, grpname,...])” on page 57.

LOCAL(

{ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  CONTROLLER([xxxxxxx-]xxxxx|name))
})

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  CONTROLLER([xxxxxxx-]xxxxx|name)) )” on page 58.

REMOTE (RAGROUP(nn.nn.nn.nn)

{ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] }

See “REMOTE (RAGROUP(nn.nn.nn.nn) { UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
  DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] })” on page 61.

Example

CREATE (-)
  NAME (TESTSNAPSHOT) (-)
  SOURCE (SYMDV#(1D0)) (-)
  LOCAL (UNIT(6200) ) (-)
**FREE**

Clears the in-memory track pointers of a device.

After freeing the tracks that were linked to a device, the device will no longer be able to access the snapshot data that it was linked to.

If the target device of the FREE command is the last device or snapshot sharing these tracks, they will be returned to the free track list in the Storage Resource Pool, and this space will be freed up for future allocations.

**Note:** If you do not free tracks, then even after an unlink operation, the device will continue to have access to the snapshot data that it was linked to; these tracks will only be freed when no snapshot or device is sharing them.

Tracks that are allocated for snapshots and devices are only freed when no device or snapshot is sharing them. A snapshot unshares tracks only when it is terminated.

Multiple snapshots can share track allocations, so all snapshots sharing a track must be terminated (and no other source or target devices are sharing the track) before the track will be returned to its pool and the space freed up and ready for reallocation.

FREEing an SRDF device or a device with any session is blocked.

Thin Reclaim active on a device prevents it from being FREEd.

**Note:** To list sessions for a device, use the QUERY VOLUME command described in the *TimeFinder/Clone Mainframe Snap Facility Product Guide*. To verify the 04xx TRU session, use the TRU,DISPLAY,DEVICE command described in the *ResourcePak Base for z/OS Product Guide*.

**Syntax**

```
FREE
  (TARGet((UNIT(cuu)|SYMDV#(symdv#)))
   [optional_parameters]
  )
```

Where `optional_parameters` are:

- [CHECKBCVHoldstatus({Yes|No})]
- [GROUP(grpname[, grpname,...])]
- [LOCAL({UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name}] | VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name}] | DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name}] | CONTROLLER({[xxxxxxx-]xxxxx|name}] )})]
- [REMOTE (RAGROUP(nn.nn.nn.nn)
 {UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name}]]|...}]
```
VOLUME (volser) [CONTROLLER([xxxxxxx-]xxxxx|name)])
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)])
[STOP({Yes|No})]

Required parameters

TARGet({UNIT(cuu)|SYMDV#(symdv#)})

See “TaRGet ({ VOLume(volser)
UNIT([cuu]lowcuu-highcuu|lowcuu:highcuu|cuu(count))]| SYMDV#([symdv#]lowsymdv#:highsymdv#:symdv#(count))})” on page 66.

The target device must be offline, or CHECKONLINEPATHSTATUS(NO) must be specified before FREEing.

Optional parameters

CHECKBCVHoldstatus({Yes|No})

See “CHECKBCVHoldstatus({Yes|No})” on page 53.

GROUP(grpname[, grpname,...])

See “GROUP(grpname[, grpname,...])” on page 57.

LOCAL({
UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
CONTROLLER([xxxxxxx-]xxxxx|name))}

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
CONTROLLER([xxxxxxx-]xxxxx|name))}” on page 58.

REMOTE (RAGROUP(nn.nn.nn.nn)
(UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
VOLUME (volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)])}

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu)
[CONTROLLER([xxxxxxx-]xxxxx|name)]) |
VOLUME (volser)
[CONTROLLER([xxxxxxx-]xxxxx|name)]) |
DDNAME(ddname)
[CONTROLLER([xxxxxxx-]xxxxx|name)])}” on page 61.

STOP({Yes|No})

Kills the FREE background process in the storage system.

Example

```plaintext
FREE
    -
    TGT(SYMDV#(0090-093)) -
    LOCAL(UNIT(6200)) -
```

80 TimeFinder SnapVX and zDP 8.4 Product Guide
GLOBAL

Parameters specified on the GLOBAL command apply to all commands following it, unless you specifically override them through optional parameters specified with commands.

For example, if you specify REPLACE(YES) on the GLOBAL command, all commands following automatically have REPLACE(YES) as a default value.

All parameters to the GLOBAL command are optional.

Syntax

GLOBAL [optional_parameters]

Where optional_parameters are:

[ACTIVATE_SUBTASK#(nnn)]
[ALLOW_FBA_META({Yes|No})]
[ALLOW_LIGHTNING_DEVICE({Yes|No})]
[AUTO_UNLink({Yes|No})]
[BACKGROUND_COPY({Yes|No|NOCOPYRD|VSE})]
[CACHE_FULL_SYM({Yes|No})]
[CHECKBCV_Holdstatus({Yes|No})]
[CHECKONLINE_pathstatus({Yes|No|NEVER})]
[CHECK_SNAPSHOT_SIZE({Yes|No})]
[COMPACT_query({Yes|No})]
[CONDITION_VOLUME({ALL|LaBeL|DUMP})]
[CONSISTENT({Yes|No})]
[COPYVolid({Yes|No})]
[DEBUG({ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP})]
[EMUL_TYPE({ALL|HARDLINK|SNAPVX})]
[FREE({Yes|No})]
[GROUP_DATASET_name(’dataset_name’)]
[List([NO]STATEMENTS|[NO]HISTORY)]
[MAXRC(return_code_value)]
[MESsages({DISplay|PRoMpt|NONE|DETAIL})]
[MODE({COPY|NOCOPY|NOCOPYRD})]
[MULTI_LINE_query({Yes|No})]
[NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])]
[REFVTOC({Yes|No})]
[REPLace({Yes|No})]
[SECure({Yes|No})]
[SECURE_Query({Yes|No})]
[SHOW_Tracks_to_be_defined({Yes|No})]
[SNAPSHOT_LIST(filter1[,filter2,...,filterN])]
[SOFTlink({Yes|No})]
[SOFTRestore({Yes|No})]
[SRDFA_CONSISTENT_RETRY({Yes|No|count})]
[SRP_PERCENT(srp-percentage,{INFO|WARN|ERROR|SKIP})]
[TIMEOUT(nnn)]
[TYPRUN({RUN|NORUN|SCAN})]
[UNLINK_After_copy({Yes|No})]
[VARY_OFFline({AUTO|NEVER})]
[VARY_ONline({AUTO|Yes|No})]
[VCLOSE({Yes|No})]
[WAIT_FOR_DEFINITION({Yes|No})]
[ZDP({Yes|No})]

**Note:** This guide lists GLOBAL parameters pertaining to SnapVX. For other TimeFinder GLOBAL parameters and/or parameter values, see the *TimeFinder/Clone Mainframe Snap Facility Product Guide.*

**Optional parameters**

**ACTIVATE_SUBTASK#(nnn)**

See “ACTIVATE_SUBTASK#(nnn)” on page 52.

**ALLOW_FBA_META({Yes|No})**

Allows or prohibits execution of commands against FBA meta devices:

- **YES**  FBA meta devices are allowed.
- **NO**   (Default) FBA meta devices are not allowed.

ALLOWFBAMETA and ALLOFMETA are aliases for ALLOW_FBA_META.

The ALLOW_FBA_META parameter has a matching site option, &ALLOW_FBA_META.

**ALLOW_LIGHTNINGDEVICE({Yes|No})**

Allows or prohibits use of lightning devices as the source or the target of a command:

- **YES**  Lightning devices are allowed.
- **NO**   (Default) Lightning devices are not allowed.

The ALLOW_LIGHTNINGDEVICE parameter has a matching site option, &ALLOW_LIGHTNINGDEVICE.
AUTO_UNLink({Yes|No})

See “AUTO_UNLink({Yes|No})” on page 52.

BACKGROUNDCOPY ({Yes|No|NOCOPYRD|VSE})

The BACKGROUNDCOPY parameter specifies the background copy mode:

No Establishes a snap relationship where tracks are copied from the source to the target when tracks are updated on the source or target or read on the target. Additional updates to the same source track are not copied.

NOCOPYRD Specifies that the background copy occurs only when a track is changed on the source or target. This causes the original source track to be copied to the target. Additional updates to the same source track are not copied to the target. A read of the source or target track does not cause the track to be copied.

VSE This option allows you to run a background copy function when in VSE copy mode. This option is the same as MODE(VSE).

Yes (Default) Enables background copy.

You can specify the NOCOPYRD keyword as:

- NOBACKGROUNDCOPYONREAD
- NOBGCOPYONREAD
- NOCOPYONREAD
- NOCOPYREAD
- NOCOPYRD

The BACKGROUNDCOPY and MODE parameters serve the same purpose and cannot be specified at the same time. They are mutually exclusive.

The BACKGROUNDCOPY parameter has a matching site option, &BACKGRND.

CACHE_FULL_SYM({Yes|No})

This parameter improves job completion times for jobs with high device counts.

When set to YES, information for every device in the storage system is retrieved in one syscall with a range of all devices in the storage system, instead of issuing individual syscalls for each. This reduces the I/O latency of repeatedly querying the storage system.

Set CACHE_FULL_SYM(YES) for jobs with approximately 500 or more devices.

The CACHE_FULL_SYM parameter has a matching site option, &CACHESYM.

CHECKBCVHoldstatus({Yes|No})

See “CHECKBCVHoldstatus({Yes|No})” on page 53.

CHECKONLINEpathstatus({Yes|No|NEVER})

See “CHECKONLINEpathstatus({Yes|No|NEVER})” on page 53.

CHECK_SNAPSHOT_SIZE({Yes|No})

See “CHECK_SNAPSHOT_SIZE({Yes|No})” on page 53.

COMPACT_query({Yes|No})

See “COMPACT_query({Yes|No})” on page 53.
**CONDitionVOLume**({**ALL|LaBeL|DUMP**})

See “CONDitionVOLume({**ALL|LaBeL|DUMP**})” on page 54.

**CONSISTENT**({**Yes|No**})

See “CONSISTENT({**Yes|No**})” on page 54.

**COPYVolid**({**Yes|No**})

See “COPYVolid({**Yes|No**})” on page 55.

**DEBUG**({**ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP**})

The DEBUG parameter specifies the default debug option for all TimeFinder operations:

- **ALL**: Produce the TRACE and DEBUG information needed for most situations.
- **DUMP**: Produce normal TRACE and DEBUG output.
- **ERROR**: Record some TRACE and DEBUG output in memory. Put this information in the message log only if an error occurs.
  
  **Note:** DEBUG(ERROR) has a matching site option, &DEBUG_ERROR.

- **EXTRA**: Produce all possible TRACE and DEBUG information (more complete than **ALL**).
- **SDUMP**: When an abend occurs, an SDUMP (SVC DUMP) is automatically taken.
  
  **Note:** DEBUG(SDUMP) has a matching site option, &DEBUG_SDUMP.

- **TRACE**: Produce normal TRACE output.

**Note:** This parameter has no default value.

If you do not specify DEBUG, no debugging is performed.

**EMUL_TYPE**({**ALL|HARDLINK|SNAPVX**})

See “EMUL_TYPE({**ALL|HARDLINK|SNAPVX**})” on page 56.

**Global parameter effects**

This parameter sets a global value for the following command:

- **QUERY SNAPSHOT**

**FREE**({**Yes|No**})

See “FREE({**Yes|No**})” on page 57.

**GROUP_DATaset_name**(’**dataset_name**’)

The GROUP_DATaset_name parameter is valid only with the GLOBAL command and identifies the dataset used as the “working” group dataset that contains a group of SnapVX statements.

**Note:** “Defining groups of statements” on page 47 discusses groups of statements.

If this parameter omitted, TimeFinder uses a DDNAME of EMCGROUP as the “working” group dataset.
GROUP_DSName is an alias of GROUP_DATaset_name. The GROUP_DATaset_name parameter has a matching site option, &GROUP_DSNAME.

LIST({[NO]STAtements|[NO]HIStory})
See “LIST([NO]STAtements) [NO]HIStory)” on page 57.

MAXRC(return_code_value)

The MAXRC parameter specifies the maximum allowable return codes. If the return code value is exceeded by the code returned by a command, all commands following are bypassed.

Each message issued has a severity associated with it. Severities are associated with the last character of the message ID. For instance, if the last character is a 'I', the severity is 0. 'W' is 4 and 'E' is 8, 'S' is 16.

When a request (command) is completely processed, the highest severity for a message issued for that command is checked against the MAXRC setting to determine whether additional commands is processed.

return_code_value

The numeric value you want to use as the maximum allowable return code. The values you can use range from zero (0) through 16. By default, the MAXRC return_code_value is four (4).

Note: MAXRC does not apply until after the parsing phase is complete. If any error is encountered during the parsing phase, the run is always terminated.

Example

- Message ESNPxxxI - severity is 0. If MAXRC(4), then additional commands is executed (0 is not greater than 4).
- Message ESNPxxxW - severity is 4. If MAXRC(4), then additional commands is executed (4 is not greater than 4).
- Message ESNPxxxE - severity is 8. If MAXRC(4), then additional commands is NOT executed (8 is greater than 4).

MESSages({DISplay|PROmpt|NONE|DETAIL})
See “MESSages({DISplay|PROmpt|NONE|DETAIL})” on page 59.

MODE({COPY|NOCOPY|NOCOPYRD})
See “MODE({COPY|NOCOPY|NOCOPYRD})” on page 59.

MULTI_LINE_query({Yes|No})
See “MULTI_LINE_query({Yes|No})” on page 59.

NAME(snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])
See “NAME(snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])” on page 60.

REFVTOC({Yes|No})
See “REFVTOC({Yes|No})” on page 61.

REPLace({Yes|No})
See “REPLace({Yes|No})” on page 62.
SECure({Yes|No})

See “SECure((Yes|No))” on page 63.

⚠️ CAUTION
Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See Knowledgebase article 498316 for more information.

SECURE_Query({Yes|No})

See “SECURE_Query((Yes|No))” on page 63.

SHOW_Tracks_to_be_defined({Yes|No})

See “SHOW_Tracks_to_be_defined((Yes|No))” on page 63.

SNAPSHOT_LIST(filter1[,filter2,...,filterN])

See “SNAPSHOT_LIST(filter1[,filter2,...,filterN])” on page 63.

Global parameter effects
This parameter sets a global value for the following command:

- QUERY SNAPSHOT

SOFTlink({Yes|No})

Determines whether soft linking is used (YES) or not (NO, default):

- YES Creates a soft snapshot that can be linked and unlinked to multiple target devices.
- NO (Default) Creates a hard snapshot on the source and links it to a specific target device.

Note: “Softlinked and hardlinked snapshots” on page 36 provides information about soft and hard snapshots.

This parameter is required to allow using native SnapVX syscalls to create, activate and link a snapshot from the source device to the target all with just one traditional TF/Clone SNAP VOLUME command.

Note: The TimeFinder/Clone Mainframe Snap Facility Product Guide describes traditional TF/Clone commands.

When you specify SOFTlink(YES), a snapshot will be created on the source volume with the name specified in the NAME(snapshot_name) parameter.

Note that you cannot have duplicate snapshot names on a single source device. Issuing the SNAP VOLUME command with SOFTlink(YES) twice in a row without terminating the snapshot results in a duplicate snapshot name error. To prevent duplicate snapshot names, append the snapshot name with the %date% and %time% variables to generate a unique snapshot name.

Note: “NAME(snapshot_name[%date{%4|6|8}%][%time{%4|6}%])” on page 60 describes use of the date and time variables.
The SOFTlink parameter has a matching site option, &SOFTLINK.

SOFTRestore({Yes|No})

See “SOFTRestore({Yes|No})” on page 64.

SRDFA_CONSISTENT_RETRY({Yes|No|count})

See “SRDFA_CONSISTENT_RETRY({Yes|No|count})” on page 66.

SRP_PERCENT( srp-percentage, {INFO|WARN|ERROR|SKIP})

Determines whether to issue informational, warning, or error messages if the target device’s SRP used capacity exceeds the specified percentage during a LINK operation.

Note: “Monitoring SRP usage during linking” on page 38 discusses SRP usage monitoring.

The default setting is SRP_PERCENT(100,INFO), which means no messages are displayed.

SRP_PERCENT applies to the LINK command only and is ignored for other commands (such as CONFIG, CREATE SNAPSHOT, UNLINK, TERMINATE, QUERY).

Note: The SRP reports Total, FBA, and CKD capacity utilization values. Depending on the device type, only the CKD or FBA utilization is checked against the specified SRP_PERCENT percentage.

srp-percentage

The percentage of used SRP capacity that the SRP must be above for the messages to be displayed. Valid values are 0 through 100. The default value is 100.

This parameter has a corresponding site option, &SRPPERCT.

INFO|WARN|ERROR|SKIP

Sets the severity level of messages to be displayed: informational (return code 0), warning (return code 4), or error (return code 8).

Regardless of this setting, SRP summary information is always displayed unless the SKIP parameter is specified.

The INFO|WARN|ERROR keywords have a corresponding site option, &SRPMSGLVL.

The SKIP option skips SRP usage processing. If SKIP is specified, no SRP information is obtained or displayed.

Note: Specify the SKIP parameter to speed up job processing when SRP information is not required.

The SKIP keyword has a corresponding site option, &SRPMSGLVL.
TIMEOUT (nnn)

See “TIMEOUT(nnn)” on page 68.

TYPRUN ({RUN | NORUN | SCAN})

This parameter sets a global value for how commands are processed:

RUN  (Default) SnapVX performs full processing of all commands.

NORUN  SnapVX parses all commands and identifies the datasets that are going to be processed. No further processing is performed.

SCAN  SnapVX stops command processing after all commands have been parsed. In other words, SnapVX checks command syntax but does not perform any further processing.

UNLINK_After_copy ({Yes | No})

See “UNLINK_After_copy ({Yes|No})” on page 68.

VARY_OFFline ({AUTO | NEVER})

See “VARY_OFFline ({AUTO|NEVER})” on page 68.

VARY_ONline ({AUTO | Yes | No})

See “VARY_ONline ({AUTO|Yes|No})” on page 69.

VCLOSE ({Yes | No})

See “VCLOSE ({Yes|No})” on page 69.

WAIT_FOR_Definition ({Yes | No})

See “WAIT_FOR_Definition ({Yes|No})” on page 69.

ZDP ({Yes | No})

Determines whether SnapVX is able to perform actions on zDP snapshots:

Yes

SnapVX operations may run on zDP snapshots. SnapVX can perform actions that change the snapshot state (such as LINK, UNLINK, TERMINATE, CONFIG).

ZDP(YES) cannot be set for the CREATE and RENAME commands.

No

(Default) SnapVX operations may only be run against non-zDP snapshots. SnapVX operations run against zDP snapshots fail.
**LINK**

Relates a created and activated snapshot to a target device.

Once the link completes, the device can be accessed. The target device for the LINK command may be the original source device, which logically would simulate a restore operation, or any other compatible device with the same or larger number of cylinders.

Any previous tracks assigned to the target device are returned to the pool. If a snapshot is linked more than once to the same target device, only the differences are applied.

**Note:** When defining a group of statements using DEFINE GROUP, only the LINK and GLOBAL commands and the options associated with the LINK and GLOBAL commands can be used inside the DEFINE GROUP statement. The LINK statements are mandatory and GLOBAL statements are optional.

**Syntax**

```plaintext
LINK
(
NAME(snapshot_name)
SOUrce
({
VOLUME(volser) |
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(2)}) |
SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(2)}) |
})
TaRGet
({
VOLUME(volser) |
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(2)}) |
SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(2)}) |
})
[optional_parameters]
)
```

Where **optional_parameters** are:

- `AUTO_CREATE({Yes|No})`
- `CHECKONLINEpathstatus({Yes|No|NEVER})`
- `CHECK_SNAPSHOT_SIZE({Yes|No})`
- `CONDITIONVOLUME({ALL|LaBeL|DUMP})`
- `COPYVolid({Yes|No})`
- `GROUP(grpname[,grpname,...])`
- `LOCAL(({ UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name)})] |
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name)})] |
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name)})] |...})`
CONTROLLER(
[[xxxxxxx-]xxxxx|xname])
)

[MODE((COPY|NOCOPY|NOCOPYRD))]

[NEWVOLID(volser)]

[READY((Yes|No))]

[REFVTOC((Yes|No))]

[REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER(
[[xxxxxxx-]xxxxx|xname]])|}
VOLUME (volser) [CONTROLLER(
[[xxxxxxx-]xxxxx|xname]])|}
DDNAME(ddname) [CONTROLLER(
[[xxxxxxx-]xxxxx|xname]])])]

[REPLACE((Yes|No))]

[RESTORE_create((Yes|No))]

[RESTORE_CREate_NAME(snapshot_name)]

[SNAPSHOTID(snapshot_id)]

[UNLINK_After_copy((Yes|No))]

[VCLOSE((Yes|No))]

[VARY_OFFline({AUTO|NEVER})]

[VARY_ONline({AUTO|Yes|No})]

Required parameters

NAME(snapshot_name)

The name of the snapshot to be linked.

If the name contains hyphens, enclose it in single quotes.

SOURce

{(VOLUME(volser)|
UNIT((cuu|lowcuu:highcuu|lowcuu:highcuu|cuu(count)))|}
SYMDV#((symdv#|lowsyndv#:highsyndv#:lowsyndv#:highsyndv#:syndv#(count))])))

See “SOURce ((VOLUME(volser)|
UNIT((cuu|lowcuu:highcuu|lowcuu:highcuu|cuu(count)))|)
SYMDV#((symdv#|lowsyndv#:highsyndv#:lowsyndv#:highsyndv#:syndv#(count)))|)” on page 65.

TaRGet

{(VOLUME(volser)|
UNIT((cuu|lowcuu:highcuu|lowcuu:highcuu|cuu(count)))|}
SYMDV#((symdv#|lowsyndv#:highsyndv#:lowsyndv#:highsyndv#:syndv#(count)))))

See “TaRGet ((VOLUME(volser)|
UNIT((cuu|lowcuu:highcuu|lowcuu:highcuu|cuu(count)))|)
SYMDV#((symdv#|lowsyndv#:highsyndv#:lowsyndv#:highsyndv#:syndv#(count)))|)” on page 66.
Optional parameters

AUTO_CREATE({Yes|No})

Causes a snapshot to be created and activated if none currently exists matching the specified snapshot name.

The AUTO_CREATE parameter has a matching site option, &AUTOCREATE.

CHECKONLINEpathstatus({Yes|No|NEVER})

See “CHECKONLINEpathstatus({Yes|No|NEVER})” on page 53.

CHECK_SNAPSHOT_SIZE({Yes|No})

See “CHECK_SNAPSHOT_SIZE({Yes|No})” on page 53.

CONDITIONVOLUME({ALL|LaBeL|DUMP})

See “CONDITIONVOLUME({ALL|LaBeL|DUMP})” on page 54.

COPYVolid({Yes|No})

See “COPYVolid({Yes|No})” on page 55.

GROUP(grpname[, grpname,...])

See “GROUP(grpname[, grpname,...])” on page 57.

Note: The GROUP parameter is not allowed in LINK statements that are stored within a group. The GROUP parameter is allowed when the LINK statements occur in regular input, such as after a //QCINPUT DD * JCL statement.

LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxx| name})][
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxx| name})][
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxx| name})][
CONTROLLER({[xxxxxxx-]xxxx| name})]
})

See “LOCAL({ UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxx| name})][
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxx| name})][
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxx| name})][
CONTROLLER({[xxxxxxx-]xxxx| name})])” on page 58.

MODE({COPY|NOCOPY|NOCOPYRD})

See “MODE({COPY|NOCOPY|NOCOPYRD})” on page 59.

NEWVOLID(volser)

See “NEWVOLID(volser)” on page 60.

READY({Yes|No})

See “The NEWVOLID parameter only applies to locally addressable volumes. NEWVOLID is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters. READY({Yes|No})” on page 60.

REFVTOC({Yes|No})

See “REFVTOC({Yes|No})” on page 61.
For the duration of the current snapshot command, the value of REFVTTOC overrides any value set by the GLOBAL command REFVTTOC parameter or by the &REPLACE site option.

**REMOTE (RAGROUP(nn.nn.nn.nn) [UNIT(cuu) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])]] | VOLUME (volser) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])]] | DDNAME(ddname) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])])**

See “REMOTE (RAGROUP(nn.nn.nn.nn) [UNIT(cuu) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])]] | VOLUME (volser) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])]] | DDNAME(ddname) [CONTROLLER([[[xxxxxxx]-]xxxxx|name])])” on page 61.

**REPLACE({Yes|No})**

See “REPLACE({Yes|No})” on page 62.

**RESTore_create({Yes|No})**

The RESTore_create parameter, if set to YES, automatically creates a snapshot of the source device when linking a target device back to the source of its snapshot. This feature is designed to prevent unintentionally overwriting data on a source device by creating a point-in-time copy of the original source device.

**Note:** If the source does not match the target at the time the LINK command with the RESTore_create(YES) parameter is run, the RESTore_create(YES) setting is ignored.

You define the name of the snapshot to be created using the RESTore_CREATe_NAME(snapshot_name) parameter.

This parameter has a matching site option, &RESTORE_CREATE.

**RESTore_CREATe_NAME(snapshot_name)**

Specifies the name of snapshot to be created when issuing the LINK command with the RESTore_create(YES) parameter.

**SNAPSHOTID(snapshot_id)**

The ID of the snapshot to be processed.

The specified snapshot_id is verified against that of the snapshot ID SnapVX chooses for the link operation. If the snapshot IDs do not match, the command fails.

Specifying the SNAPSHOTID(snapshot_id) parameter protects against the condition where more than one snapshot on a single device share the same snapshot name. In normal operation this should not occur, but it can happen if snapshots are created through other products besides TimeFinder/Clone Mainframe Snap Facility that do not check for duplicate snapshot names. Snapshot IDs and snapshot names for all of the snapshots on a device can be found by issuing a QUERY SNAPSHOT command with the Multi_Line_Query(YES) parameter set.

**UNLINK_After_copy({Yes|No})**

See “UNLINK_After_copy({Yes|No})” on page 68.

**VARY_OFFline({AUTO|NEVER})**
See “VARY_OFFline({AUTO|NEVER})” on page 68.

VARY_ONline({AUTO|Yes|No})
See “VARY_ONline({AUTO|Yes|No})” on page 69.

VCLOSE({Yes|No})
See “VCLOSE({Yes|No})” on page 69.

Example

```
LINK {
    NAME (TESTSNAPSHOT) -
    SOURCE (UNIT(6250)) -
    TARGET (UNIT(6251)) -
}
```
QUERY FREE

The QUERY FREE command displays the number of tracks left to free up in SRP space and status of the FREE task (active or not active). It also shows a summary of allocated tracks for devices that are currently freeing.

Syntax

```plaintext
QUERY FREE

(VOLUME(volser)|UNIT(cuu)|
GROUP(grpname[,grpname,...])
[optional_parameters]
)
```

Where optional_parameters are:

```plaintext
[CCUU({cuu|low-high|low:high|ccuu(count)})]
[DEVice({symdv#|lowsymdv#-highsymdv#|
lowsymdv#:highsymdv#|symdv#(count)|ALL})]
[DISPLAY_CUU({Yes|No})]
[LOCAL({
UNIT(cuu) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
VOLUME(volser) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
DDNAME(ddname) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
CONTROLLER({'xxxxxxx-}xxxxx|name})
})]
[REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
VOLUME (volser) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
DDNAME(ddname) [CONTROLLER('{{xxxxxxx-}xxxxx|name}})|
CONTROLLER({'xxxxxxx-}xxxxx|name})
})]
```

Required parameters

```plaintext
GROUP(grpname[,grpname,...])

See “GROUP(grpname[,grpname,...])” on page 57.
```

VOLUME(volser)

Specifies the volser of the volume to be queried.

UNIT(cuu)

Specifies the unit address of the volume to be queried or the gatekeeper.

Optional parameters

```plaintext
CCUU({cuu|low-high|low:high|ccuu(count)})

Specifies the CUU of the volume to be queried.
```

DEVice({symdv#|lowsymdv#-highsymdv#|
lowsymdv#:highsymdv#|symdv#(count)|ALL})

See “DEVice({symdv#|lowsymdv#-highsymdv#|
lowsymdv#:highsymdv#|symdv#(count)|ALL})” on page 55.
DISPLAY_CUU({Yes|No})

See “DISPLAY_CUU({Yes|No})” on page 56.

LOCAL({
UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
CONTROLLER([xxxxxxx-]xxxxxx|name)])

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
CONTROLLER([xxxxxxx-]xxxxxx|name)])” on page 58.

The LOCAL parameter can be used together with the VOLume, UNIT, or DEVice parameters in the same command. You can combine the LOCAL parameter with VOLume or UNIT, and DEVice to further refine the selected devices.

REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxxx|name)])}

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxxx|name)]) |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxxx|name)])” on page 61.

The REMOTE parameter can be used together with the VOLume, UNIT, or DEVice parameters in the same command. You can combine the REMOTE parameter with VOLume or UNIT, and DEVice to further refine the selected devices.

Example

ESNP1AAI TOTAL NUMBER OF QUERY RESULTS: 8
ESNP1AAI
ESNP1AAI DEV# CUU VOLSER ALLOCATED TRKS FREE STATUS
ESNP1AAI
---------------------------------------------------------------
ESNP1AAI 00004333 (8D28) *8D28* 00000076 NOT ACTIVE
ESNP1AAI 00004334 (8D29) *8D29* 00000076 NOT ACTIVE
ESNP1AAI 00004335 (8D2A) *8D2A* 00000076 NOT ACTIVE
ESNP1AAI 00004336 (8D2B) *8D2B* 00000076 NOT ACTIVE
ESNP1AAI 00004337 (8D2C) *8D2C* 00000076 NOT ACTIVE
ESNP1AAI 00004338 (8D2D) *8D2D* 00015302 NOT ACTIVE
ESNP1AAI 00004339 (8D2E) *8D2E* 00000076 NOT ACTIVE
ESNP1AAI 0000433A (8D2F) *8D2F* 00000076 NOT ACTIVE
ESNP1AAI
---------------------------------------------------------------
ESNP1AAI
ESNP1AAI TOTAL TRACKS TO BE FREED: 0000000000000000
QUERY GLOBAL

The QUERY GLOBAL command displays both the site options table and any GLOBAL overrides that have been encountered in the input stream.

Syntax

```
QUERY GLOBAL
```

Example

```
E SNPW20I --- EMCSNAPO --- VER 8.4.0 --- SIZE 633 --- DATE/TIME 06/06/19 19.48 ---
E SNPW21I SITE SETTING GLOBAL OVERRIDE
E SNPW22I ACTIVATE SCF GATEKEEPER Y -N/A-
E SNPW22I ACTIVATE_SUBTASK# 03 03
E SNPW22I ADMINISTRATOR N N
E SNPW22I ALLOCATE_UNUSED_SPACE Y Y
E SNPW22I ALLOCATION_SEQUENCE D D
E SNPW22I ALLOCATION_UNITNAME SYSALLDA -N/A-
E SNPW22I ALLOW CANCEL LOCKED Y -N/A-
E SNPW22I ALLOW_FBA_META N N
E SNPW22I ALLOW_SYMDV# Y -N/A-
E SNPW22I ALLOW_LIGHTNING_DEVICE N N
E SNPW22I AUTOMATIC_ACTIVATE Y Y
E SNPW22I AUTO_BIND_TDEV N N
E SNPW22I AUTOMATIC_CLEANUP Y Y
E SNPW22I AUTOMATIC_CLEANUP_R2 Y Y
E SNPW22I AUTOMATIC_DEALLOC Y Y
...
```

Note: QUERY GLOBAL displays both SnapVX and legacy TF/Clone Mainframe Snap Facility site options. SnapVX options are described in “EMCSNAPO site options” on page 22. For legacy TF/Clone Mainframe Snap Facility options, see the TF/Clone Mainframe Snap Facility Product Guide.
### QUERY GROUP

The QUERY GROUP command allows you to query the contents of one or all groups. If you specify a group name, QUERY GROUP returns information about that group. If you do not specify a group name, QUERY GROUP returns information about all groups.

**Syntax**

```
QUERY GROUP grpname [(LIST([NO]STATEments) [NO]HISTORY))]
```

**Required parameters**

```
grpname
```

The name of the group. The name can contain as many as eight characters, with no embedded spaces. The characters you use must be valid for a PDS member name.

**Note:** You cannot reference a group that was defined or deleted in the current jobstep.

**Optional parameters**

```
LIST([NO]STATEments) [NO]HISTORY)
```

See “LIST([NO]STATEments) [NO]HISTORY)” on page 57.

For the duration of the current QUERY GROUP command, the value of LIST overrides any value set by the GLOBAL command LIST parameter.

**Example**

```
ESNPQ73I  GROUPNAME - STATUS - DESCRIPTION
ESNPR20I  SVX728GP - INITIAL -
ESNPR22I  HISTORY: RC DATE / TIME OLD STAT STATUS LPAR
ESNPR23I  DEFINE - 0000 2018-01-22 / 16:12:41 INITIAL -> INITIAL K143
ESNPR23I  CREATE - 0000 2018-01-22 / 16:13:09 INITIAL -> CREATED K143
ESNPR23I  ACTIVATE - 0000 2018-01-22 / 16:13:40 CREATED -> PRESNAP K143
ESNPR23I  ACTIVATE - 0000 2018-01-22 / 16:13:41 PRESNAP -> ACTIVATE K143
ESNPR23I  LINK - 0004 2018-01-22 / 16:14:55 ACTIVATE -> LINKED K143
ESNPR23I  UNLINK - 0000 2018-01-22 / 16:18:08 LINKED -> UNLINKED K143
ESNPR23I  TRMNT - 0000 2018-01-22 / 16:20:21 UNLINKED -> INITIAL K143
ESNPR23I  QRY FREE - 0000 2018-01-22 / 16:21:03 INITIAL -> INITIAL K143
ESNPR25I  STATEMENTS:
ESNPR26I  + *
ESNPR26I  + LINK( -
ESNPR26I  + NAME(SVX728_SNSHOT1) -
ESNPR26I  + SOU(UNIT(6930-6937)) -
ESNPR26I  + TARG(UNIT(6938-693F)) -
ESNPR26I  + END GROUP
```

**Note:** “SnapVX group status transitions” on page 37 explains group statuses that may appear in the QUERY GROUP output.
QUERY SNAPSHOT

Displays snapshots for the specified source device and their relationship to target devices.

To specify a range of devices, use the DEVICE option in conjunction with the LOCAL or REMOTE parameter.

**Note:** To view detailed information about the snapshot devices, run the QUERY VOLUME command described in the *TimeFinder/Clone Mainframe Snap Facility Product Guide*.

**Multiline query**

When MULTI_LINE_query(YES) is specified, properties of each snapshot are displayed, such as:

- Snapshot ID
- Track statistics
- Time since creation and Time to Live
- Whether the snapshot is secure

**Note:** See “Example 2: Multiline query” on page 103 for sample output and explanation of fields.

**Compact query**

When COMPACT_query(YES) is specified, status fields are displayed at the end of each line to provide a single-line summary of important snapshot information, including:

- Snapshot status and whether the snapshot has been activated
- Link type (softlink, hardlink)
- Snapshot copy mode (COPY or NOCOPY) and status (if every track on the source has been copied to the target at least one time)
- Whether the snapshot is secure and/or zDP-managed

**Note:** See “Example 3: Compact query” on page 105 for sample output and explanation of fields.
Syntax

QUERY_SNAPSHOT
(
{VOLUME(volser) | UNIT(cuu) | GROUP(grpname[, grpname, ...])
[optional_parameters]
)

Where optional_parameters are:
[CCUU({cuu|low-high|low:high|ccuu(count)})]
[COMPACT_QUERY({Yes|No})]
[DEVICE({symdv#:lowsymdv#:hysymdv# | lowsymdv#:hysymdv#:symdv#(count)|ALL})]
[DISPLAY_CUU({Yes|No})]
[EMUL_TYPE({ALL|HARDLINK|SNAPVX})]
[LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name})]|
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})]|
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})]|
CONTROLLER({[xxxxxxx-]xxxxx|name})
})]
[MULTI_LINE_QUERY({Yes|No})]
[NAME(snapshot_name)]
[REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name})]|
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})]|
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})]})]
[SECURE_QUERY({Yes|No})]
[SHOW_Tracks_to_be_defined({Yes|No})]
[SNAPSHOT_LIST(filter1[, filter2,..., filterN])]
[SORT({Yes|No})]

**Note:** You can use the LOCAL, REMOTE, and CONTROLLER parameters without a SYMDV# parameter being present.

Required parameters

GROUP(grpname[, grpname, ...])

See “GROUP(grpname[, grpname,...])” on page 57.

VOLUME(volser)

Specifies the volser of the volume to be queried.

UNIT(cuu)

Specifies the unit address of the volume to be queried or the gatekeeper.
Optional parameters

**CcUU**({*cuu*|*low-high*|*low:high*|*ccuu*(count)})

Specifies the CUU of the volume to be queried.

**COMPACT_query**({*Yes*|*No*})

See “COMPACT_query({*Yes*|*No*})” on page 53.

For the duration of the current QUERY SNAPSHOT command, the value of COMPACT_query overrides any value set by the GLOBAL command COMPACT_query parameter.

**DEVice**({*symdv#*|*lowsymdv#-highsymdv#*|*lowsymdv#:highsymdv#*|*symdv#*(count)*|*ALL*})

“DEVice({*symdv#*|*lowsymdv#-highsymdv#*|*lowsymdv#:highsymdv#*|*symdv#*(count)*|*ALL*})” on page 55.

**DISPLAY_CUU**({*Yes*|*No*})

See “DISPLAY_CUU({*Yes*|*No*})” on page 56.

**EMUL_TYPE**({*ALL*|*HARDLINK*|*SNAPVX*})

See “EMUL>Type({*ALL*|*HARDLINK*|*SNAPVX*})” on page 56.

For the duration of the current QUERY SNAPSHOT command, the value of EMUL_TYPE overrides any value set by the GLOBAL command EMUL_TYPE parameter or by the &EMUL_TYPE site option.

**LOCAL**({
  **UNIT**(*cuu*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
  **VOLUME**(*volser*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
  **DDNAME**(*ddname*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
  **CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*}) })

See “LOCAL({**UNIT**(*cuu*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
**VOLUME**(*volser*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
**DDNAME**(*ddname*) [**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*)}] |
**CONTROLLER**({[*xxxxxxx-]*xxxxx|*name*}) })” on page 58.

The LOCAL parameter can be used together with the VOLume, UNIT, or DEVice parameters in the same command. You can combine the LOCAL parameter with VOLume or UNIT, and DEVice to further refine the selected devices.

**MULTI_LINE_query**({*Yes*|*No*})

See “MULTI_LINE_query({*Yes*|*No*})” on page 59.

For the duration of the current QUERY SNAPSHOT command, the value of MULTI_LINE_query overrides any value set by the GLOBAL command MULTI_LINE_query parameter.

**NAME**(*snapshot_name*)

When NAME(*snapshot_name*) is specified, QUERY SNAPSHOT shows only the snapshot with the specified snapshot_name.

If the name contains hyphens, enclose it in single quotes.
REMOTE (RAGROUP(nn.nn.nn.nn)
   {UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})])|
   VOLUME (volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})])|
   DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})])}

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu)
   [CONTROLLER([[[xxxxxxx-]xxxxx|name]])] VOLUME (volser)
   [CONTROLLER([[[xxxxxxx-]xxxxx|name]])] DDNAME(ddname)
   [CONTROLLER([[[xxxxxxx-]xxxxx|name]])])” on page 61.

The REMOTE parameter can be used together with the VOLUME, UNIT, or DEVice parameters in the same command. You can combine the REMOTE parameter with VOLUME or UNIT, and DEVice to further refine the selected devices.

SECURE_Query({Yes|No})

See “SECURE_Query((Yes|No))” on page 63.

SHOW_Tracks_to_be_defined({Yes|No})

See “SHOW_Tracks_to_be_defined((Yes|No))” on page 63.

SNAPSHOT_LIST(filter1[,filter2,...,filterN])

See “SNAPSHOT_LIST(filter1,filter2,...,filterN)” on page 63.

For the duration of the current QUERY SNAPSHOT command, the value of SNAPSHOT_LIST overrides any value set by the GLOBAL command SNAPSHOT_LIST parameter or by the &SNAPSHOT_LIST site option.

SORT({Yes|No})

When set to YES, sorts snapshots by date. NO disables sorting by date.

Note: Sorting can dramatically increase the query time.

Example

Example 1: QUERY SNAPSHOT {
   DISPLAY_CUU(NO)
   UNIT(8D00) -
   DEV(436A-436B) -
   DISPLAY_CUU(NO) -
}

The output is similar to the following:

ESNP1AAI TOTAL NUMBER OF QUERY RESULTS: 00000002
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP1AAI
ESNP461I PROCESSING FOR STATEMENT #4 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0

Output fields

TOTAL NUMBER OF QUERY RESULTS

The total count of results returned in the query.
### Command Reference

<table>
<thead>
<tr>
<th>SRC</th>
<th>The PowerMax/VMAX device number of the source device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUU</td>
<td>The CUU of the source device.</td>
</tr>
<tr>
<td><strong>Note:</strong> With DISPLAY_CUU(NO), the value is displayed as N/A.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VOLSER</th>
<th>The volser of the source device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGT</td>
<td>The PowerMax/VMAX device number of the target device.</td>
</tr>
<tr>
<td>CUU</td>
<td>The CUU of the target device.</td>
</tr>
<tr>
<td><strong>Note:</strong> With DISPLAY_CUU(NO), the value is displayed as N/A.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VOLSER</th>
<th>The volser of the target device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>The snapshot name on the source device.</td>
</tr>
<tr>
<td><strong>For each device that has one or more scheduled snapshots on it, Automated_Snapshot is displayed.</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME STAMP</th>
<th>The approximate time when the snapshot was created.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STATUS</th>
<th>The current status of the snapshot:</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ CREATE HARD</td>
<td>(hardlinked snapshot)</td>
</tr>
<tr>
<td>✗ CREATE SOFT</td>
<td>(softlinked snapshot)</td>
</tr>
<tr>
<td>✗ EMUL RESNAP</td>
<td></td>
</tr>
<tr>
<td>✗ F-LINK</td>
<td>(failed LINK)</td>
</tr>
<tr>
<td>✗ F-SNAPSHOT</td>
<td>(failed snapshot)</td>
</tr>
<tr>
<td>✗ LINKED</td>
<td></td>
</tr>
<tr>
<td>✗ NONE</td>
<td></td>
</tr>
<tr>
<td>✗ RELINKED</td>
<td></td>
</tr>
<tr>
<td>✗ TERMINATED</td>
<td></td>
</tr>
<tr>
<td>✗ UNLINKED</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACT</th>
<th>Indicates whether the snapshot has been activated. Possible values: Y/N.</th>
</tr>
</thead>
</table>
Example 2: Multiline query

```
QUERY SNAPSHOT -
  (UNIT(cuu) -
    DEVICE(symdv#) -
    MULTI_LINE_QUERY(YES) -
    SHOW_TRACKS_TO_BE_DEFINED(YES) -
  )
```

The output is similar to the following:

```
TOTAL NUMBER OF QUERY RESULTS: 4390

<table>
<thead>
<tr>
<th>SRC</th>
<th>CUU</th>
<th>VOLSER</th>
<th>TGT</th>
<th>CUU</th>
<th>VOLSER</th>
<th>NAME</th>
<th>YYDDD/HH:MM:SS</th>
<th>STATUS</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000010E</td>
<td>N/A</td>
<td>FFFFFFFF</td>
<td>N/A</td>
<td>SAMPLE_SNAPSHOT_NAME</td>
<td>17110/13:13:40</td>
<td>CREATE SOFT</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIQUE TRACKS:</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRC MODIFIED TRACKS (UNIQUE+SHARED):</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGT MODIFIED TRACKS (UNIQUE+SHARED):</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKS TO BE COPIED:</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME TO LIVE DDDD/HH:MM:SS:</td>
<td>0000/00:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME SINCE CREATION DDDD/HH:MM:SS:</td>
<td>0004/20:09:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNAPSHOT ID:</td>
<td>ADAD0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURE SNAPSHOT:</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY MODE:</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKS TO BE DEFINED:</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0000010E</th>
<th>0000015E</th>
<th>SAMPLE_SNAPSHOT_NAME</th>
<th>17111/15:43:52</th>
<th>LINKED</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC MODIFIED TRACKS (UNIQUE+SHARED):</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGT MODIFIED TRACKS (UNIQUE+SHARED):</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKS TO BE COPIED:</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNAPSHOT ID:</td>
<td>ADAD0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURE SNAPSHOT:</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY MODE:</td>
<td>COPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKS TO BE DEFINED:</td>
<td>0000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output fields

The multiline query displays the following information for each snapshot:

**UNIQUE TRACKS**

The number of tracks that are only associated with the displayed snapshot. This is also the number of tracks that will be freed if the snapshot is terminated.

*Note: The unique track count is not updated immediately after a snapshot is terminated. It is updated periodically by a background task in the storage system.*

**SRC MODIFIED TRACKS**

The number of tracks that have changed on the snapshot source device since the activate was issued.

**TGT MODIFIED TRACKS**

The number of tracks that have changed on a snapshot linked target device since the LINK action.
<table>
<thead>
<tr>
<th>Command Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRACKS TO BE COPIED</strong></td>
</tr>
<tr>
<td><strong>TIME TO LIVE</strong></td>
</tr>
<tr>
<td><strong>TIME SINCE CREATION</strong></td>
</tr>
<tr>
<td><strong>SNAPSHOT ID</strong></td>
</tr>
<tr>
<td><strong>SECURE SNAPSHOT</strong></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td><strong>COPY MODE</strong></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td><strong>TRACKS TO BE DEFINED</strong></td>
</tr>
<tr>
<td><strong>TOTAL UNIQUE TRACKS IN REQUEST</strong></td>
</tr>
<tr>
<td><strong>TOTAL TRACKS TO BE COPIED IN REQUEST</strong></td>
</tr>
<tr>
<td><strong>TOTAL SRC MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST</strong></td>
</tr>
<tr>
<td><strong>TOTAL TGT MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST</strong></td>
</tr>
<tr>
<td><strong>TOTAL TRACKS TO BE DEFINED IN REQUEST</strong></td>
</tr>
</tbody>
</table>
Example 3: Compact query

```plaintext
QUERY SNAPSHOT -
  (UNIT(cuu) -
  DEVICE(symdv#) -
  COMPACT_query(YES) -
)
```

The output is similar to the following:

```
ESNP1AAI TOTAL NUMBER OF QUERY RESULTS: 16

... 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 
ESNP1AAI 

Output fields

The compact query adds status fields on the right of the display. Status column headings are stacked vertically so that each corresponds to a one-character value for each query result. STATUS FIELDS LEGEND above the table explains the status values.

Status fields have the following meaning:

**STA** (STATUS) Indicates the snapshot status:
- F—Failed snapshot
- H—Hardlink
- L—Linked to a target device
- S—Created softlinked snapshot
- U—Unlinked from a target device

**S/H** (Softlinked/Hardlinked) Indicates if the snapshot is softlinked or hardlinked.

**CPY** (COPYMODE) Indicates the copy mode that the source/target devices are in. Possible values are:
- C—COPY
- N—NOCOPY

**ZDP** (z/OS Data Protector) Indicates if the snapshot was created by zDP.
| **DEF** | (Defined) Indicates if the target device is fully defined or not. If not linked, this field shows a dash ('-'). |
| **ACT** | (Activated) Indicates if the snapshot has been activated. |
| **1PS** | (1 PASS COPY COMPLETE) Indicates if every track on the source has been copied to the target at least one time. For active snapshots, a dash ('-') is shown. |
| **SEC** | (Secure) Indicates if the source snapshot is secure or not. |
**RENAME**

Changes the name of a created snapshot.

**Syntax**

```
RENAME

(NAME(snapshot_name)
NEWNAME(new_snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])
SOURce

{(
VOLume(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
SYMDV#({symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)})|)

[optional_parameters]
)
```

Where optional parameters are:

```
[CHANGE_all_names({Yes|No})]
[LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxxxx-]xxxxx|name])}]|
VOLUME(volser) [CONTROLLER({[xxxxxxxxx-]xxxxx|name})}]|
DDNAME(ddname) [CONTROLLER({[xxxxxxxxx-]xxxxx|name})]|
CONTROLLER({[xxxxxxxxx-]xxxxx|name])]}]
[REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[xxxxxxxxx-]xxxxx|name})}]|
VOLUME (volser) [CONTROLLER({[xxxxxxxxx-]xxxxx|name})]|
DDNAME(ddname) [CONTROLLER({[xxxxxxxxx-]xxxxx|name})]})}
[SNAPSHOTID(snapshot_id)]
```

**Required parameters**

```
NAME(snapshot_name)

The name of the snapshot to be renamed.

If the name contains hyphens, enclose it in single quotes.

NEWNAME(new_snapshot_name[%date[{4|6|8}]%][%time[{4|6}]%])

The new 1-32 character name of the snapshot. new_snapshot_name allows upper and lower case alpha, numerics, hyphens (-) and underscores (_). If the name contains hyphens, enclose it in single quotes. Embedded spaces are not allowed.

To ensure a unique snapshot name per source device, append the new_snapshot_name with date and time variables:

- %date%—Substitutes the current date in MM_DD_YYYY format.
- %date4%—Substitutes the current date in MMDD format.
Command Reference

- `%date6%`—Substitutes the current date in MMDDYY format.
- `%date8%`—Substitutes the current date in MMDDYYYY format.
- `%time%`—Substitutes the current time in HH_MM_SS format.
- `%time4%`—Substitutes the current time in HHMM format.
- `%time6%`—Substitutes the current time in HHMMSS format.

SOURce

```plaintext
({
    VOLume(volser) | 
    UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | 
    SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(count)}) | 
})
```

See “SOURce ({ VOLume(volser)| UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})| SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(count)})| })” on page 65.

Optional parameters

CHANGE_all_names({Yes|No})

When set to YES, renames all sessions with a matching snapshot name across all devices on the specified storage system (the device you specify will act as a gatekeeper into the storage system).

LOCAL({
    UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    VOLume(volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    CONTROLLER({[[xxxxxxx-]xxxxx|name]})
})

See “LOCAL({ UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    VOLume(volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    CONTROLLER({[[xxxxxxx-]xxxxx|name]}) ” on page 58.

REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    VOLume(volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    CONTROLLER({[[xxxxxxx-]xxxxx|name]})})

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    VOLume(volser) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})] | 
    DDNAME(ddname) [CONTROLLER({[[xxxxxxx-]xxxxx|name]})]” on page 61.

SNAPSHOTID(snapshot_id)

The ID of the snapshot to be renamed.

Example

```plaintext
RENAME (NAME(TESTSNAPSHOT) - NEWNAME(TESTSNAPSHOT2) - SOURCE(SYMDV#(1D0)) - LOCAL(UNIT(6200)))
```
**TERMINATE**

Discards a created or activated snapshot.

For the TERMINATE command to function properly, no target devices can be linked to the snapshot, unless AUTO_UNLink is specified, in which case, actions will be taken to unlink all targets from the snapshot.

**Space considerations upon TERMINATE**

A background define process is started unconditionally and automatically by the operating environment upon a LINK operation. This operation updates the target's track pointer table to point directly to the track data instead of having to access the track using the snapshot. When an UNLINK operation occurs, the target continues to point to any tracks that have been defined or written to. So, when a snapshot is terminated, tracks that are still pointed to as a result of a previous link and background define operation to a target device, or written to on the target device, will not be freed, and will continue to consume space.

**Syntax**

```
TERMinate
(
NAME({snapshot_name|*})
SOUrce
({
VOLume(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#:highsymdv#|symdv#(count)})|
})
[optional_parameters]
)
```

Where `optional_parameters` are:

```
[AUTO_UNLink({Yes|No})]
[CHECKBCVHoldstatus({Yes|No})]
[FREE({Yes|No})]
[WAIT_FOR_Definition({Yes|No})]
```

```
[GROUP(grpname[,grpname,...])]
```

```
[LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxxx-]xxxxx|name})]|
VOLUME(volser) [CONTROLLER({[xxxxxxxx-]xxxxx|name})]|
DDNAME(ddname) [CONTROLLER({[xxxxxxxx-]xxxxx|name})]
})]
```

```
[REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[xxxxxxxx-]xxxxx|name})]|
VOLUME (volser) [CONTROLLER({[xxxxxxxx-]xxxxx|name})]|
DDNAME(ddname) [CONTROLLER({[xxxxxxxx-]xxxxx|name})])}
```

```
[SNAPSHOTID(snapshot_id)]
```
[TERminate_ALL({Yes|No})]

Required parameters

NAME({snapshot_name|*})

The name of the snapshot to be terminated. If the name contains hyphens, enclose it in single quotes.

Specifying * as the snapshot_name causes all snapshots on the specified source device to be terminated.

NAME(*) cannot be specified together with SNAPSHOTID(snapshot_id) or TERMinate_ALL(Yes).

SOUrce

({VOLUME(volser)|
  UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
  SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#::highsymdv#|symdv#(count)})|})

See “SOUrce ({VOLUME(volser)|
  UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
  SYMDV#({symdv#|lowsymdv#-highsymdv#|lowsymdv#::highsymdv#|symdv#(count)})|})” on page 65.

Optional parameters

[ AUTO_UNLink({Yes|No})
  [CHECKBCVHoldstatus({Yes|No})]
  [FREE({Yes|No})]
  [WAIT_FOR_Definition({Yes|No})]
]

See “AUTO_UNLink({Yes|No})” on page 52.

CHECKBCVHoldstatus({Yes|No})

See “CHECKBCVHoldstatus({Yes|No})” on page 53.

FREE({Yes|No})

See “FREE({Yes|No})” on page 57.

WAIT_FOR_Definition({Yes|No})

See “WAIT_FOR_Definition({Yes|No})” on page 69.

GROUP(grpname[, grpname, ...])

See “GROUP(grpname,grpname,...)” on page 57.
LOCAL({
    UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    CONTROLLER([xxxxxxx-]xxxxx|name))
})

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    CONTROLLER([xxxxxxx-]xxxxx|name))” on page 58.

REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) |
    CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)])
}

See “REMOTE (RAGROUP(nn.nn.nn.nn) (UNIT(cuu) |
    CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) |
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)])” on page 61.

SNAPSHOTID(snapshot_id)

The ID of the snapshot to be terminated.

SNAPSHOTID(snapshot_id) cannot be specified together with NAME(*).

TERMinate_ALL({Yes|No})

When set to Yes, terminates all snapshots that match the specified snapshot name, across all source devices on the entire storage system. The default value is No.

TERMinate_ALL(Yes) cannot be specified together with NAME(*).

This parameter has a matching site option, &TERM_ALL.

Example

TERM {
    NAME(TESTSNAPSHOT) -
    SOURCE(SYMDV#(1D0)) -
    LOCAL(UNIT(6200)) -
}
UNLINK

Breaks the relationship between a snapshot and a linked target device.

After an UNLINK, any copied tracks will remain on the device. The target device may be in one of the following states:

- If LINK with MODE(COPY) was specified, and the copy completed, the data will be whole.
- If LINK with MODE(COPY) was specified, and the copy was not completed, the device will be in an unpredictable state.
- If LINK with MODE(NOCOPY) was specified, the data on the target is retained.

The state of the data on the target device after an UNLINK is also dependent on whether the track definition process (explained in “Target volume track definition” on page 37) is complete on the target device. For example, if LINK with MODE(NOCOPY) was specified but the UNLINK was run without waiting for the track definition process to complete, the target device data would be in an unpredictable state.

Syntax

```
UNLINK
( TaRGet
  ({
    VOLume(volser)|
    UNIT({{cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}| |
      SYMDV#((symdv#|lowsymdv#:highsymdv#|lowsymdv#:highsymdv#|symdv#(count)| |
    }) |
  }) |
    [optional_parameters]
  )
Where optional_parameters are:
[CHECKBCVHoldstatus({Yes|No})]
[FREE({Yes|No})]
[GROUP(grpname[, grpname,...])]
[LOCAL({
  UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name}])| |
  VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx|name}])| |
  DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name}])| |
  CONTROLLER({[xxxxxxx-]xxxxx|name}}])
])
[NEWVOLID(volser)]
[READY({Yes|No})]
[REMOTE (RAGROUP(nn.nn.nn.nn)
  {UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx|name}])| |
  VOLUME (volser) [CONTROLLER({[xxxxxxx-]xxxxx|name}])| |
  DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name}})])
]```
Required parameters

TaRGet

{(VOLume(volser) | UNIT({cuu | lowcuu-highcuu | lowcuu:highcuu | cuu(count)}) | SYMDV#({symdv# | lowsymbv#:highsymv# | lowsymbv#:highsymv# | symdv#(count)}) )

See “TaRGet ({ VOLume(volser) | UNIT({cuu | lowcuu-highcuu | lowcuu:highcuu | cuu(count)}) | SYMDV#({symdv# | lowsymbv#:highsymv# | lowsymbv#:highsymv# | symdv#(count)}) )” on page 66.

Optional parameters

CHECKBCVHoldstatus({Yes|No})

See “CHECKBCVHoldstatus({Yes|No})” on page 53.

FREE({Yes|No})

See “FREE({Yes|No})” on page 57.

GROUP(grpname[, grpname, ...])

See “GROUP(grpname[, grpname,...])” on page 57.

LOCAL({
UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx | name})]]
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx | name})]]
DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx | name})]]
CONTROLLER({[xxxxxxx-]xxxxx | name})]

See “LOCAL({ UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx | name})]])
VOLUME(volser) [CONTROLLER({[xxxxxxx-]xxxxx | name})]) DDNAME(ddname)
[CONTROLLER({[xxxxxxx-]xxxxx | name})]]
CONTROLLER({[xxxxxxx-]xxxxx | name}) ]” on page 58.

NEWVOLID(volser)

See “NEWVOLID(volser)” on page 60.

READY({Yes|No})

See “The NEWVOLID parameter only applies to locally addressable volumes. NEWVOLID is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters. READY({Yes|No})” on page 60.

REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLLER({[xxxxxxx-]xxxxx | name})] |
Command Reference

VOLUME (volser) [CONTROLLER({[xxxxxxx-]xxxxx|name})] | DDNAME(ddname) [CONTROLLER({[xxxxxxx-]xxxxx|name})]

See “REMOTE (RAGROUP(nn.nn.nn.nn) (UNIT(cuu)
[CONTROLLER([xxxxxxx-]xxxxx|name)]))| VOLUME (volser)
[CONTROLLER([xxxxxxx-]xxxxx|name)])| DDNAME(ddname)
[CONTROLLER([xxxxxxx-]xxxxx|name)])” on page 61.

SOFTRestore({Yes|No})

See “SOFTRestore({Yes|No})” on page 64.

VARY_OFFline({AUTO|NEVER})

See “VARY_OFFline({AUTO|NEVER})” on page 68.

VARY_ONline({AUTO|Yes|No})

See “VARY_ONline({AUTO|Yes|No})” on page 69.

VCLOSE({Yes|No})

See “VCLOSE({Yes|No})” on page 69.

WAIT_FOR_Definition({Yes|No})

See “WAIT_FOR_Definition({Yes|No})” on page 69.

Example

UNLINK ( -
    TARGET( UNIT(6250) ) -
)
Grouping commands

Use grouping commands to group and store SnapVX statements in a PDS or PDS/E file. “Defining groups of statements” on page 47 provides detailed instructions on how to use this functionality.

The group commands are used in a life cycle order with DEFINE GROUP including GLOBAL and LINK commands, followed by an END GROUP command. Then the SnapVX CREATE command with the GROUP keyword is used to initialize the group. After the group is created, snapshots exist on the source devices, then they can be linked, unlinked, and terminated using corresponding GROUP keywords. When the group is no longer needed, the DELETE GROUP command removes the group.

Note: If the command spans multiple lines, add a dash (-) at the end of each command line except the last one.

DEFINE GROUP

Defines a group of LINK and GLOBAL statements which are then stored into a PDS or PDS/E file.

Requirements and restrictions
Consider the following requirements and restrictions when defining a group of statements using DEFINE GROUP:

◆ Only the LINK and GLOBAL commands and the options associated with the LINK and GLOBAL commands can be used inside the DEFINE GROUP statement.
◆ The LINK statements are mandatory and GLOBAL statements are optional within the DEFINE GROUP definition.
◆ Statements within the DEFINE GROUP definition cannot include a VOLUME(volser) parameter for a target.

Modifying previously defined groups
You cannot edit a previously defined group to change its contents. Instead, you must:
1. Delete the group using the DELETE GROUP command.
2. Redefine the group with DEFINE GROUP/REPLACE and a new set of commands.

Syntax

DEFINE GROUP grpname
[(optional_parameters)]

Where optional_parameters are:
[DESCRIPTION('descriptive text')]
[FORCE({Yes|No})]
[REPLACE({Yes|No})]
Command Reference

Required parameters

`grpname`

The name of the group.

The name can contain as many as eight characters, with no embedded spaces. The characters you use must be valid for a PDS member name.

**Note:** You cannot reference a group that was deleted in this job step.

Optional parameters

**DESCRIPTION ('descriptive text')**

A text string, of up to 64 characters, that describes the group. The description is stored in the group member and listed each time you query the group.

**FORCE({Yes|No})**

If an existing group has a status of PRESNAP or ACTIVATE, then FORCE(YES) is required to replace the group definition. This is intended to prevent a group from being redefined while in use, possibly impacting the relationship or status of existing devices in the group.

If the existing group has a status of INITIAL, POSTSNAP or FAILED, then the FORCE parameter is not required.

For the duration of the current DEFINE GROUP command, the value of FORCE overrides any value set by the GLOBAL command FORCE parameter or by the &FORCE site option.

**REPLACE({Yes|No})**

Determines whether an existing group may be replaced with a new group definition.

For the duration of the current DEFINE GROUP command, the value of REPLACE overrides any value set by the GLOBAL command REPLACE parameter or by the &REPLACE site option. See “REPLACE({Yes|No})” on page 62.
DELETE GROUP

Deletes an existing group (defined with DEFINE GROUP).
The deletion removes the definition from the PDS in which it is stored.

Syntax

DELETE GRouP grpname
([optional_parameters])

Where optional_parameters are:

FORCE({Yes|No})

Required parameters

grpname

The name of the group. The name can contain as many as eight characters, with no embedded spaces. The characters you use must be valid for a PDS member name.

Note: You cannot reference a group that was defined or already deleted in this jobstep.

Optional parameters

FORCE({Yes|No})

If an existing group has a status of PRESNAP or ACTIVATE, then FORCE(YES) is required to replace the group definition. This is intended to prevent a group from being redefined while in use. A redefinition of a group in use could impact the relationship or status of existing devices in the group. If a group has the status of INITIAL, POSTSNAP, or FAILED, then you would not have to use FORCE.

For the duration of the current DELETE GROUP command, the value of FORCE overrides any value set by the global FORCE parameter or by the &FORCE site option.

END GROUP

Completes the definition of a group.

You must enter an END GROUP after you finish entering the statements that define the group. You may follow END GROUP with other commands, including commands that refer to the group or commands that define other groups.

Syntax

END  GROUP
Part 2 of this guide describes how to use Data Protector for z Systems (zDP).

Chapter 6, “Introduction,” provides an overview of zDP capabilities.

Chapter 7, “Features and Concepts,” explains zDP features and concepts.

Chapter 8, “zDP Definition Utility,” discusses how to run zDP using the runtime environment.

Chapter 9, “zDP ISPF Interface,” shows how to run zDP using the ISPF interface.

Chapter 10, “SMF Records,” describes zDP SMF records.
This chapter covers the following topics:

- Overview .......................................................... 122
- Implementation ............................................... 122
- Requirements and limitations ................................ 123
- Security considerations ...................................... 124
Overview

Data Protector for z Systems (zDP) delivers the capability to recover from logical data corruption with minimal data loss. zDP provides multiple, frequent, consistent point-in-time copies of data in an automated fashion. From the point-in-time copies, an application level recovery can be conducted, or the environment can be restored to a point before the logical corruption.

By providing easy access to multiple different point-in-time copies of data (with a granularity of minutes), precise remediation of logical data corruption can be performed using application-based recovery procedures at the dataset level. zDP results in minimal data loss compared to the previous method of restoring data from daily or weekly backups. This is important for non-DBMS data, which does not have the granular recovery options provided by log files and image copies associated with database management systems.

zDP enables you to create and manage multiple point-in-time snapshots of volumes. A snapshot is a pointer-based, point-in-time image of a single volume. These point-in-time copies are created using the SnapVX feature of PowerMaxOS 5978 and HYPERMAX OS 5977. SnapVX is a space-efficient method for making volume-level snapshots of thin devices and consuming additional storage capacity only when updates are made to the source volume.

SnapVX provides targetless snapshot capabilities. There is no need to copy each snapshot to a target volume as SnapVX separates the capturing of a point-in-time copy from its usage. Capturing a point-in-time copy does not require a target volume. Using a point-in-time copy from a host requires linking the snapshot to a target volume. One snapshot of each volume across a group of volumes is termed a snapset. You can make multiple snapshots of each source volume.

You can link, unlink, relink snapshots to the same set of target volumes thus having the flexibility in the point-in-time copy that is presented on a single set of target devices. These snapshots share allocations to the same track image whenever possible while ensuring they each continue to represent a unique point-in-time image of the source volume. Despite the space efficiency that is achieved through shared allocation to unchanged data, additional capacity is required to preserve the pre-update images of changed tracks captured by each point-in-time snapshot.

Implementation

zDP implementation is a two-stage process—the planning phase and the implementation phase.

- The planning phase is done with your Dell EMC representative. The Dell EMC representative has access to tools to help size the capacity that is needed for zDP in case you are a storage system user. This process involves use of ChangeTracker (a component of ResourcePak Base) and an additional sizing tool available to Dell EMC representatives.

- The implementation phase uses the following methods:

  - An ISPF interface that allows you to define and manage the zDP runtime environment. Chapter 9, “zDP ISPF Interface”, describes the zDP ISPF interface.
Introduction

- A batch interface that allows you to submit jobs to define and manage zDP. Chapter 8, “zDP Definition Utility”, describes the zDP Definition Utility and its configuration statements.

- A zDP runtime environment that executes under SCF to create snapsets. The ResourcePak Base for z/OS Product Guide describes the following zDP commands that can be issued from SCF:
  - ZDP,CREATE
  - ZDP,ECACLEAR
  - ZDP,MODIFY,SMF
  - ZDP,PAUSE
  - ZDP,RELEASEDEVICELOCK
  - ZDP,RESUME
  - ZDP,START
  - ZDP,STOP

Before starting a zDP process, use the zDP Definition Utility to define and manage the environment. After defining the environment, you can start and stop zDP runtime tasks by using Modify commands that are issued to SCF.

**Note:** The zDP control blocks are allocated in Extended Common Storage (XCSA) and are not persistent after an IPL.

Requirements and limitations

zDP requirements and limitations are as follows:

- zDP must be APF-authorized and access to output datasets should be limited to authorized personnel.

- A source volume may not be in two active Versioned Data Groups (VDG).

- For remote requests, the final destination storage system must be running PowerMaxOS 5978 or HYPERMAX OS 5977, while other storage systems in the hoplist may be Enginuity 5876.

- All devices must be in the same SRDF mode to allow for consistency.

- Devices in adaptive copy mode can be added to a zDP configuration, but zDP does not run with ADCOPY devices when consistency is enabled.

- Soft-fenced devices, dummy devices, guest operating system devices, and BCV devices are not supported.
Security considerations

The EMCSAFI Security Interface feature provides additional security checks for environments where multiple groups of users are using different devices in a single storage system. All zDP commands are secured with SAF XFACILIT.

**Note:** The *Mainframe Enablers Installation and Customization Guide* summarizes the resource validation requests for zDP features and functions with XFACILIT.
CHAPTER 7
Features and Concepts

This chapter covers the following topics:

- Key concepts ................................................................. 126
- Secure snaps ................................................................. 127
- Persistent snapsets ......................................................... 128
- Saved snapsets ............................................................. 128
- Copy-once devices ......................................................... 129
- On-demand snapsets .................................................... 130
- Dynamic VDG update .................................................... 131
Key concepts

The following terms are used to describe zDP functionality:

- **Versioned Data Group (VDG)**
- **Snapset**
- **Snapshot**
- **Target set (TGT)**

Versioned Data Group (VDG)

A logical group that associates volumes along with settings for creating multiple point-in-time images.

A VDG can have up to 1024 snapsets associated with it, on condition that all source devices in the VDG reside on storage systems running PowerMaxOS 5978.0300 and later. For lower levels of the operating environment, the maximum number of snapsets is 256.

Snapset

A named point-in-time consistent image of all the source volumes in a VDG.

A snapshot can be nonconsistent if specified; however, by default a consistent snapshot is created.

Snapshot name

The snapshot name is a string of 32 characters:

VDGnamexxxxxxxxxxYYDDHHMMtcccccrr

Where:

- **VDGnamexxxxxxxxx** identifies the VDG for which the snapshot is created. If the VDG name is fewer than 15 characters, it is padded with periods, for example:
  
  VDG_UYF1........153281429C00008

- **YYDDHHMMtcccccrr** is the snapshot ID, consisting of the following:
  
  - **YYDD** is the date (Julian—Year, Month Day).
  - **HH** is hours (0-23), **MM** is minutes (0-59).
  - **t** indicates the snapshot type:
    
    - **C** = Cyclical Create
    - **I** = Secure
    - **S** = Saved
    - **U** = On-Demand
  
  - **ccccc** is the zDP managed cycle number.
  
  - **rr** is reserved for future use.
Snapshot

A pointer-based, point-in-time image of a single volume. One volume can have up to 256 snapshots created on it.

A PowerMax/VMAX feature, Enginuity Consistency Assist (ECA), is used to create dependent-write consistency on the source volumes as part of the snapshot creation process.

Target set (TGT)

A group of devices which have been defined to zDP as the required LINK targets to a snapset.

Target sets go through a validation process to ensure that they are eligible candidates to be linked to.

Secure snaps

zDP supports secure snapsets.

Note: Secure snapsets are created with the Secure Snaps feature of SnapVX, as described in “Secure snaps” on page 39.

CAUTION

Secure snapsets may only be terminated after they expire or by customer-authorized Dell EMC Support. See “Secure snaps” on page 39 for more information.

zDP supports snapsets initially created as secure and also lets you make an existing snapset secure.

zDP support of secure snapsets requires PowerMaxOS 5978 or HYPERMAX OS 5977.1028 and later. If all systems in a VDG are not running PowerMaxOS 5978 or HYPERMAX OS 5977.1028 or later, a warning message is issued and the Secure setting is ignored.

To create secure snapsets, use the CYCLE_TIME[,SECURE] or SAVED_SNAPSETS[,SECURE] options of the DEFINE VDG and MODIFY VDG OPTIONS commands. To convert an existing snapset to secure, issue the SECURITY VDG command. To reset the Secure attribute, specify SECURITY,NO for the CYCLE_TIME/SAVED_SNAPSETS parameter on the MODIFY VDG OPTIONS command.

Secure snapsets are indicated with “-I” in the QUERY VDG SNAPSET command output.

- A secure snapshot is created with “-I” in its snapshot ID and state, for example:
  
  EIP0039I VDG1...........170971440I00001 ACT-I 05/07/2017 14:40:15
An existing snapset is not renamed upon conversion to secure; however, it is displayed with "-I" due to the secure attribute:

EIP0039I  VDG1........171311014C00002  ACT-I  05/11/2017 10:14:39

To view the current secure settings for a VDG, issue the QUERY VDG STATUS command and look for the SECURE values:

...EIP0021I  Cycle_Time(5,0,SECURE,1),Cycle_Overflow(IMMED),Consistent(YES),Time out(15,CONT)
...EIP0021I  Max_Snapsets(256),Saved_Snapsets(1,2,SECURE),Persistent_Copy_Limit(32)
...

Note: The zDP ISPF interface also provides controls to view and manage secure snapsets.

Persistent snapsets

A persistent snapset cannot be automatically terminated when the MAX_SNAPSETS limit is reached.

To mark a snapset persistent or remove the Persistent attribute, use the PERSISTENT command.

You can set the maximum number of persistent snapsets to be created using the PERSISTENT_COPY_LIMIT parameter on the DEFINE VDG or MODIFY VDG OPTIONS command.

Saved snapsets

A saved snapset is a snapset that is marked with a Saved attribute and is retained for the user-defined number of days (the retention period).

During snapset creation, zDP determines if a snapset should be marked as saved by comparing the current time to the time of the last saved snapset. If the user-defined number of days required between saved snapsets is reached, the snapset is marked as saved and its expiration time is set according to the retention period.

To create a saved snapset, use the SAVED_SNAPSETS parameter on the DEFINE VDG or MODIFY VDG OPTIONS command.

Saved snapsets are indicated with ‘S’ for the snapset type in the snapset name, for example:

VDG3........17131101480002
Copy-once devices

If a device is specified as a copy-once device, it is included only in the first snapshot created after the start of a VDG.

If a device is dynamically added to the VDG as a copy-once device, zDP creates a saved snapshot for all devices in the VDG, including the newly added copy-once device. If all devices in a VDG are defined as copy-once devices, the VDG stops after the first cycle.

The Copy Once feature can be used, for example, when you want to include the system volumes in a snapshot for recovery purposes, but the data on a page volume is not useful or needed for recovery. Page, dump, and sysres volumes are good examples to use this feature.

**Note:** A copy-once snapshot is created as a saved snapshot, with no expiration time. The Copy Once attribute takes precedence over the Secure attribute (a secure snapshot must have an expiration time).

The zDP Definition Utility allows you to define copy-once devices, view copy-once devices and snapshots, and control inclusion of copy-once devices in LINK or RESTORE operations.

**Note:** The zDP ISPF interface also provides controls to view and manage copy-once devices.

Defining copy-once devices

To set the copy-once attribute for the devices, specify the COPY_ONCE keyword on the MODIFY VDG,ADD statement.

Viewing copy-once devices and snapshots

In the QUERY VDG,DEVICE report, copy-once devices are indicated with '/O' following the device number:

```
EIP0027I CCUU DEVICE TYPE SIZE SRP ID RDF INFO/MODE
EIP0027I ____ ______ ____ ________ ______ ______________
EIP0028I 3150 000170/O CKD 32760 0001 ---
```

To view the last snapshot that includes a copy-once device, specify the COPY_ONCE option on the QUERY VDG,DEVICE command. The snapshot is displayed in the COPY_ONCE SSET field:

```
EIP0027I CCUU DEVICE TYPE SIZE SRP ID RDF INFO/MODE COPY_ONCE SSET
EIP0027I ____ ______ ____ ________ ______ ______________ ______________
EIP0028I 3150 000170/O CKD 32760 0001 --- 171360952800001
```

To view a list of snapshots that contain copy-once devices defined in the current VDG, issue the QUERY VDG,SNAPSET command with the COPY_ONCE option. The copy-once filter is indicated with ‘(COPY_ONCE)’ in the query heading:

```
EIP0035I Snapset Query for VDG EMCVDGC1 (COPY_ONCE)
```
Features and Concepts

Linking and restoring copy-once devices

To include copy-once devices for a LINK or RESTORE of a snapset that does not contain the copy-once devices, specify COPY_ONCE(INCLUDE) on the LINK or RESTORE command.

To manually link or restore the copy-once devices from a specific snapset, specify COPY_ONCE(ONLY) on the LINK or RESTORE command.

Unlinking copy-once devices

An UNLINK for a linked snapset unlinks all devices (including copy-once devices). However, because a target set is not employed for RESTORE, an UNLINK for a restored snapset unlinks only the devices in the specific snapset. A separate UNLINK command for the snapset containing the copy-once devices must be issued.

On-demand snapsets

This feature lets you create a snapset immediately for the specified VDG. The snapset is created when the VDG waits for the start of the next cycle.

You create an on-demand snapset with the ZDP CREATE command issued through SCF or using the Z=Create line command on the zDP VDG Monitor ISPF panel.

Note: The ResourcePak Base for z/OS Product Guide describes the ZDP CREATE command.

An on-demand snapset is indicated with "U" (user-created) for the snapset type in the snapset name, for example:

TEST_VDG.......181841157U00002
Dynamic VDG update

This feature lets you dynamically update the definition of an active VDG without VDG restart.

If a VDG device definition is modified for an active VDG, a dynamic change flag is set for the VGD at the end of zDP Definition Utility (EIPINIT) processing. When the Dynamic VDG Update feature is enabled, zDP checks for a dynamic change at the end of each cycle. If the dynamic change flag is set, the local VDG control blocks are dynamically rebuilt before the start of the next cycle. In addition to device changes, any change to VDG runtime options are also dynamic.

You can enable or disable the Dynamic VDG Update feature using the DYNAMIC_CHANGE parameter on the DEFINE VDG or MODIFY VDG OPTIONS command. Alternatively, you can select the Dynamic Change option in the Specify Versioned Data Group Parameters panel.

To view the status of the Dynamic VDG Update feature, issue the QUERY VDG STATUS command and look for the Dynamic_Change value:

```
EIP0021I Terminate_Policy(OLDEST), Dynamic_Change(NO)
```

The status is also shown in the zDP VDG Configuration Information panel and in the zDP VDG Monitor report.

The follows the process that is
CHAPTER 8
zDP Definition Utility

This chapter covers the following topics:
◆ Overview ............................................................................................................ 134
◆ Running zDP Definition Utility ............................................................................ 138
◆ zDP Definition Utility statements ....................................................................... 139
Overview

The zDP Definition Utility (EIPINIT) allows you to define and manage the zDP runtime environment. The zDP Definition Utility can be run in batch or called from the ISPF interface as described in Chapter 9, “zDP ISPF Interface.”

Summary of operations

Table 13 lists operations for managing VDGs.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define VDG</td>
<td>DEFINE VDG command</td>
</tr>
<tr>
<td>Add/remove devices in VDG</td>
<td>MODIFY VDG ADD</td>
</tr>
<tr>
<td>Bypass sort of VDG devices by PowerMax/VMAX device number</td>
<td>DEFINE VDG command, NOSORT parameter MODIFY VDG OPTIONS command, NOSORT parameter</td>
</tr>
<tr>
<td>Enable or disable dynamic VDG updates</td>
<td>• DEFINE VDG command, DYNAMIC_CHANGE parameter • MODIFY VDG OPTIONS command, DYNAMIC_CHANGE parameter</td>
</tr>
<tr>
<td>Delete VDG</td>
<td>DELETE VDG command</td>
</tr>
</tbody>
</table>

Table 14 lists operations for managing target sets.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define target set</td>
<td>DEFINE TGT command</td>
</tr>
<tr>
<td>Add/remove devices in target set</td>
<td>MODIFY TGT ADD</td>
</tr>
<tr>
<td>Bypass sort of target devices by PowerMax/VMAX device number during linking</td>
<td>DEFINE TGT command, NOSORT parameter</td>
</tr>
<tr>
<td>Unlink target set</td>
<td>UNLINK TGT command</td>
</tr>
<tr>
<td>Free target device tracks from SRP</td>
<td>UNLINK TGT command, FREE parameter</td>
</tr>
<tr>
<td>Stop freeing target device tracks from SRP</td>
<td>STOP_FREE TGT command UNLINK TGT command, STOP_FREE parameter</td>
</tr>
<tr>
<td>Wait for target track definition before unlinking</td>
<td>UNLINK TGT command, WAIT FOR DEFINITION parameter</td>
</tr>
<tr>
<td>Delete TGT</td>
<td>DELETE TGT command</td>
</tr>
</tbody>
</table>

Table 15 lists operations for managing snapsets.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link snapshot volumes to target set</td>
<td>LINK VDG command</td>
</tr>
<tr>
<td>Set background copy mode</td>
<td>LINK VDG command, MODE parameter</td>
</tr>
<tr>
<td>Set/reset “persistent” attribute for snapshot</td>
<td>PERSISTENT command, SET or RESET parameter</td>
</tr>
</tbody>
</table>
Table 15  Managing snapsets

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Create secure snapsets                 | - SECURE VDG command  
|                                        | - DEFINE VDG command, CYCLE_TIME[,SECURE] parameter  
|                                        | - DEFINE VDG command, SAVED_SNAPSET[,SECURE] parameter  
|                                        | - MODIFY VDG OPTIONS command, CYCLE_TIME[,SECURE] parameter  
|                                        | - MODIFY VDG OPTIONS command, SAVED_SNAPSET[,SECURE] parameter         |
| Restore all source volumes in snapshot | RESTORE VDG command                                                     |
| Unlink restored snapshot from VDG      | UNLINK VDG command                                                      |
| Wait for target track definition before unlinking | UNLINK VDG command, WAIT_FOR_DEFINITION parameter |
| Terminate snapsets                     | TERMINATE VDG command                                                   |
| Simulate snapshot termination by date/time | GLOBAL command, SIMulate_TERMinate parameter                              |

Table 16 lists operations for setting VDG snapshot-related settings.

Table 16  Setting VDG snapshot-related options

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Determine whether consistency for all snapshots in a snapshot is required | DEFINE VDG command, CONSistent parameter  
|                                                                          | MODIFY VDG OPTIONS command, CONSistent parameter                       |
| Set timeout interval and action for consistency operations               | DEFINE VDG command, TIMEOUT parameter  
|                                                                          | MODIFY VDG OPTIONS command, TIMEOUT parameter                           |
| Specify how often and for how many cycles to create snapshots            | DEFINE VDG, CYCLE_TIME parameter  
|                                                                          | MODIFY VDG OPTIONS, CYCLE_TIME parameter                                |
| Determine when to create the next snapshot if the cycle time is exceeded | DEFINE VDG command, CYCLE_OVERFLOW parameter  
|                                                                          | MODIFY VDG OPTIONS command, CYCLE_OVERFLOW parameter                    |
| Set the snapshot limit                                                   | DEFINE VDG command, MAX_SNAPSET parameter  
|                                                                          | MODIFY VDG OPTIONS command, MAX_SNAPSET parameter                      |
| Determine action if the snapshot limit is reached                         | DEFINE VDG command, TERMINate_POLICY parameter  
|                                                                          | MODIFY VDG OPTIONS command, TERMINate_POLICY parameter                 |
| Set the maximum number of persistent snapshots to be created             | DEFINE VDG command, PERSISTENT_COPY_LIMIT parameter  
|                                                                          | MODIFY VDG OPTIONS command, PERSISTENT_COPY_LIMIT parameter             |
| Create saved snapshots                                                   | DEFINE VDG command, SAVED_SNAPSETS parameter  
|                                                                          | MODIFY VDG OPTIONS command, SAVED_SNAPSETS parameter                   |
| Create secure saved snapshots                                           | DEFINE VDG command, SAVED_SNAPSETS parameter with the SECURE option  
|                                                                          | MODIFY VDG OPTIONS command, SAVED_SNAPSETS parameter with the SECURE option |
**Table 17** lists operations for setting SRP usage and RDP cache utilization parameters.

### Table 17  Setting SRP usage and RDP cache utilization parameters

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define the maximum percentage of the SRP that can be allocated for SnapVX</td>
<td>DEFINE VDG command, SRP_SNAP% parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, SRP_SNAP% parameter</td>
</tr>
<tr>
<td>Define the threshold of SRP usage to initiate automatic termination of snapsets</td>
<td>DEFINE VDG command, SRP_TERM% parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, SRP_TERM% parameter</td>
</tr>
<tr>
<td>Define the threshold to issue a warning message when the total SRP usage exceeds this value</td>
<td>DEFINE VDG command, SRP_WARN% parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, SRP_WARN% parameter</td>
</tr>
<tr>
<td>Monitor RDP cache utilization</td>
<td>DEFINE VDG command, RDP_CACHE_UTIL% parameter</td>
</tr>
</tbody>
</table>

**Table 18** lists operations for managing copy-once devices.

### Table 18  Managing copy-once devices

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define copy-once devices</td>
<td>MODIFY VDG ADD</td>
</tr>
<tr>
<td>List snapsets with copy-once devices</td>
<td>QUERY VDG command, SNAPSET parameter, COPY_ONCE option</td>
</tr>
<tr>
<td>View last snapshot with copy-once device</td>
<td>QUERY VDG command, DEVICE parameter, COPY_ONCE option</td>
</tr>
<tr>
<td>Include copy-once devices for a LINK of snapshot that does not contain the copy-once devices</td>
<td>LINK VDG command, COPY_ONCE(INCLUDE) parameter</td>
</tr>
<tr>
<td>Link snapshot’s copy-once devices manually</td>
<td>LINK VDG command, COPY_ONCE(ONLY) parameter</td>
</tr>
<tr>
<td>Include copy-once devices for a RESTORE of snapshot that does not contain the copy-once devices</td>
<td>RESTORE VDG command, COPY_ONCE(INCLUDE) parameter</td>
</tr>
<tr>
<td>Restore snapshot’s copy-once devices manually</td>
<td>RESTORE VDG command, COPY_ONCE(ONLY) parameter</td>
</tr>
</tbody>
</table>

**Table 19** lists operations for monitoring zDP status.

### Table 19  Monitoring zDP status

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display VDG status</td>
<td>QUERY VDG command, STATUS parameter</td>
</tr>
<tr>
<td>Display VDG devices</td>
<td>QUERY VDG command, DEVICE parameter</td>
</tr>
<tr>
<td>Display VDG snapshots</td>
<td>QUERY VDG command, SNAPSET parameter</td>
</tr>
<tr>
<td>Display target set status</td>
<td>QUERY TGT command, STATUS parameter</td>
</tr>
<tr>
<td>Display target set devices</td>
<td>QUERY TGT command, DEVICE parameter</td>
</tr>
<tr>
<td>Display FREE task status</td>
<td>QUERY FREE command</td>
</tr>
<tr>
<td></td>
<td>QUERY TGT command, FREE parameter</td>
</tr>
</tbody>
</table>
Table 20 lists operations to control zDP messaging and SMF records.

**Table 20  zDP messaging and SMF records**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set whether to display (extended) status messages</td>
<td>DEFINE VDG command, DEBUG parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, DEBUG parameter</td>
</tr>
<tr>
<td>Set whether to issue WTOR for TERMINATE VDG, Date command</td>
<td>GLOBAL command, WTOR_TERMINATE parameter</td>
</tr>
<tr>
<td></td>
<td>TERMINATE VDG command, WTOR parameter</td>
</tr>
<tr>
<td>Where to route zDP messages</td>
<td>DEFINE VDG command, LOG_OPT parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, LOG_OPT parameter</td>
</tr>
<tr>
<td>Set the maximum allowable return code</td>
<td>GLOBAL command, MAX_RC parameter</td>
</tr>
<tr>
<td>Enable SMF recording</td>
<td>DEFINE VDG command, SMF parameter</td>
</tr>
</tbody>
</table>

Table 21 lists operations for setting zDP runtime task options.

**Table 21  Setting zDP runtime task options**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the maximum number of allowed VDG definitions</td>
<td>GLOBAL command, MAX_VDG parameter</td>
</tr>
<tr>
<td>Set the maximum number of allowed target set definitions</td>
<td>GLOBAL command, MAX_TGT parameter</td>
</tr>
<tr>
<td>Set the maximum allowable return code</td>
<td>DEFINE VDG command, MAX_RUNTIME_RC parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, MAX_RUNTIME_RC parameter</td>
</tr>
</tbody>
</table>
Running zDP Definition Utility

You run the zDP Definition Utility as a batch job. The JCL is as follows:

```
//JOBCARD
//*
//ZDPDEF EXEC PGM=EIPINIT,REGION=0M
//STEPLIB DD DISP=SHR, DSN=ds_prefix.LINKLIB
//SYSPRINT DD SYSOUT=*, DCB=BLKSIZE=121
//ZDPRPT DD SYSOUT=*, DCB=BLKSIZE=121
//ZDPERR DD SYSOUT=*, DCB=BLKSIZE=121
//*ZDPSNAP DD SYSOUT=* 
//*EMCQCAPI DD SYSOUT=* 
//*ESNAZDPL DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SCF$nnnn DD DUMMY 
//*
//SYSIN DD *
<zDP Definition Utility statements>
```

Where:

- `ds_prefix` is the product dataset name prefix that you specified during installation of Mainframe Enablers as described in the Mainframe Enablers Installation and Customization Guide.
- The commented statements (ZDPSNAP, EMCQCAPI, and ESNAZDPL) are intended for diagnostic purposes and should be used under the direction of Dell EMC Technical Support.
- If any datasets are specified for the SYSPRINT, ZDPRPT, and ZDPERR statements, they must have BLKSIZE=LRECL=121.
- `nnnn` identifies the ResourcePak Base task that the job runs against.
- zDP Definition Utility statements are listed in “zDP Definition Utility statements” on page 139.
**zDP Definition Utility statements**

**Syntax conventions**

The commands follow these syntax conventions:

- Conventions that are listed in “Conventions used in this document” on page 10
- Keywords appear in uppercase (for example, `CREATE`). They must be spelled exactly as shown.
- For easy reference, keywords can be supplemented by lowercase letters to form a meaningful word (for example, `SOUrce`). When typing a command, use only CAPITALIZED characters of any keyword.
- Aside from the characters that are listed in “Conventions used in this document” on page 10, you must type all other characters that are shown in the syntax statements.

**Common parameters**

`ALLOWNonEmpty`

Allows a delete of a nonempty zDP definition.

`ALLOWNonEmpty` requires `MAXRC(nn)` set to 4 or higher.

```
(CCUU(gk, start_ccuu[-end_ccuu], start_ccuu[-end_ccuu]...) | 
SYMDEV(gk, start_symdv#[-end_symdv#], start_symdv#[-end_symdv#]...) | 
RDFGroup(gk, srdfgrp) | 
SCFG(gnsgrp))
```

Allows you to specify devices to which the command applies.

Devices can be specified as follows. Only one type is supported per MODIFY statement.

```
CCUU(gk, start_ccuu[-end_ccuu], start_ccuu[-end_ccuu]...) 
```

Specifies a z/OS device number or range of device numbers.

```
SYMDEV(gk, start_symdv#[-end_symdv#], start_symdv#[-end_symdv#]...) 
```

Specifies a PowerMax/VMAX device number or range of device numbers.

```
RDFGroup(gk, srdfgrp) 
```

Specifies an SRDF group.

`srdfgrp` is a one- or two-digit hexadecimal value representing a single SRDF group.

Device addition or removal by SRDF group is not dynamic. If devices are later added to the SRDF group, they are not automatically included into or excluded from the VDG or target set definition. Another MODIFY TGT ADD|REMOVE or MODIFY VDG ADD|REMOVE statement must be issued.
**SCFG(gnsgrp)**

Specifies the name of a Group Named Services (GNS) definition, which can contain PowerMax/VMAX device numbers or z/OS device numbers. A GNS definition can span storage systems and can contain remote devices.

**Note:** BCV and R2 devices cannot be configured in a target set. BCV devices cannot be configured in a VDG.

**CONSistent({Yes|No})**

Defines whether consistency for all snapshots in a snapset is required. The default is YES. Consistency is managed using Enginuity Consistent Assist (ECA) or SRDF/A Suspend or Resume as appropriate, to hold write I/O on the source volumes during SnapVX ACTIVATE command processing.

**Note:** Because only one method can be used to guarantee consistency across all volumes in a snapset, a VDG cannot contain both SRDF/A and non-SRDF/A volumes simultaneously.

When CONSistent(YES) is specified, a consistency check is performed before each zDP cycle to ensure that active R2s do not have R1 invalid tracks.

If any R2s have R1 invalid tracks, message EIP0242W is issued. When the STATUSE debug option is enabled, the message is issued for each R2 with R1 invalid tracks. Otherwise, the message is issued only for the first discovered device with R1 invalid tracks. After all devices have been checked, this process continues every 30 s until all R1 invalid tracks are resolved. A STOP command can be issued to interrupt this process and stop the VDG.

**Note:** With CONSistent(YES), Consistency Exempt (CEXMPT) devices do not cause the VDG to stop unless the VDG is running with MAXRC(0).

**COPY_ONCE({INCLUDE|ONLY})**

Determines the action for copy-once devices:

**INCLUDE** Includes copy-once devices for a LINK of a snapset that does not contain the copy-once devices.

**ONLY** Allows you to manually link the copy-once devices from a specific snapset.

**Note:** “Copy-once devices” on page 129 describes copy-once devices.

By default, the LINK VDG and RESTORE VDG commands operate only on the devices in the snapset (neither of COPY_ONCE values is set).

**CYCLE_OVERFLOW({IMMED|NEXT})**

Defines the action if the creation of a snapset cannot be completed within the cycle time. IMMED (default) starts the creation of the next snapset immediately, NEXT waits for the next interval to expire (CYCLE_TIME minutes).

**CYCLE_TIME(mmmn[,count][,SECURE,ddd[,skip]])**

Specifies how often and for how many cycles to create snapshots.
`mddd` defines the time interval to create snapsets (in minutes). Valid values are 5 through 9999. The default value is 60.

`count` defines the number of cycles. zDP terminates when `count` is reached. Valid values are 0 through 9999. A value of 0 is unlimited and is the default.

`SECURE, ddd[, skip]`

Creates secure snapsets.

**Note:** “Secure snaps” on page 127 describes zDP secure snapsets.

`ddd` defines the retention period in days (1-400). The default value is 1.

`skip` defines the skip value (0-1024). When specifying a `skip` value, the first created snapshot is always secure. A skip value of “1” results in secure snapshots 1, 3, 5, and so on. Specifying “2” results in secure snapshots 1, 4, 7, and so on.

The default value is not to create secure snapshots. If the skip value is 0, all snapshots will be secure.

`DEBUG([STATUS][, STATUSE])`

Specifies debug options for the zDP runtime task. `STATUS` displays status messages. `STATUSE` displays extended status messages.

`DYNAMIC_CHANGE({Yes|No})`

Enables (Yes) or disables (No, default) the Dynamic VDG Update capability for the current VDG.

**Note:** “Dynamic VDG update” on page 131 describes the Dynamic VDG Update feature.

`gk`

The z/OS device number (CUU) of the gatekeeper device.

`LOG_OPT({SCF|SYSOUT(ddname)})`

Defines the message log option. By default all zDP messages are issued to the SCF log. Specifying `SYSOUT(ddname)` routes the messages to the `ddname` (it must be allocated to SCF).

`MAX_RC(nn)`

Defines the maximum allowable return code for the zDP Definition Utility. The default value is 0. Valid values are 0 through 99.

**Note:** Do not confuse `MAX_RC` with the `MAXRC` keyword used as an alias of the MAX_RUNTIME_RC parameter.

`MAX_RUNTIME_RC(nn)`

Defines the maximum allowable return code for the zDP runtime task. The default is 0, which treats any warning as a terminating error. Valid values are 0 through 99.
MAXRC (no underline) is an alias of MAX_RUNTIME_RC.

**Note:** Do not confuse MAXRC with the MAX_RC parameter which is spelled with an underline.

**MAX_SNAPSETS (nnnn)**
Defines the snapshot limit. When this limit is reached, snapshots are terminated based upon the termination policy.

The maximum MAX_SNAPSETS value is 1024. The default value is 256.

A MAX_SNAPSETS value exceeding 256 requires that all source devices in the VDG reside on storage systems running PowerMaxOS 5978.0300 and higher. For lower levels of the operating environment, the maximum number of snapshots is 256.

You must redefine the VDG to change the MAX_SNAPSET parameter from a value less than or equal to 256 to a value exceeding 256.

Dynamic change from a MAX_SNAPSET value less than or equal to 256 to a value exceeding 256 is not allowed.

**NOSORT**
The NOSORT option bypasses sorting of VDG or target set devices by PowerMax/VMAX device numbers. When NOSORT is specified, the devices are displayed and processed in the order in which the devices were added to the VDG or the target set.

If NOSORT is specified for both the VDG and the target set, linking follows the order of statements that define the source and target groups. For example, devices that are specified on source VDG statement 1, are linked to devices that are specified on target set statement 1, and so on.

**Note:** Redefine the VDG or target set if you want to cancel NOSORT and apply device sorting again.

**PERSISTENT_COPY_LIMIT (nnnn)**
Defines maximum number of persistent snapshots that can be created. This includes all saved and persistent snapshots. When the limit is reached, a persistent snapshot can be terminated or a PERSISTENT RESET command (described in “PERSISTENT” on page 160) can be issued to a snapshot that has persistent set before any additional persistent snapshots can be created.

Valid values are 0 through 1024. The default setting is 32.

**RDP_CACHE_UTIL% (ww, cc)**
Enables checking of Replication Data Pointer (RDP) cache utilization during each cycle against the following thresholds:

\[
\text{When this threshold is exceeded, a warning message is issued and processing continues. The default value is 60%}
\]
When this threshold is exceeded, an error message is issued and processing stops. The default value is 95%.

**Note:** Terminating a snapshot does not necessarily affect RDP cache utilization.

**RMT(srdfgrp)**

RMT applies to all devices in the MODIFY TGT ADD|REMOVE or MODIFY VDG ADD|REMOVE statement (except CUU and SCFG), defining the path to the target storage system.

This parameter specifies the SRDF group through which to perform a remote operation. `srdfgrp` is a one or two-digit hexadecimal value, a single SRDF group or for multihop configurations, a list of up to four 1 or 2-digit SRDF groups separated by periods.

REMOTE is an alias of RMT.

**SAVED_SNAPSETS(ddd, nnnn[, SECURE[, ,skip]])**

Creates saved snapsets.

**Note:** “Saved snapsets” on page 128 discusses saved snapsets.

**ddd**

Specifies the number of days required between snapshot creation. The default value is 0, which indicates that no saved snapshots will be created. Valid values are 0-999.

**nnnn**

Specifies the retention period in days. When `nnnn` is exceeded, the oldest saved snapshot is rolled off. The default value is 1. Valid values are 1-9999.

`nnnn` must be equal to or greater than `ddd`.

If the specified `nnn` value is less than `ddd`, it will be set equal to `ddd`, because specifying a retention period lower than the interval (`ddd`) will result in a timeframe with no saved snapshots.

For example:

- `(1,14)` creates a snapshot every day and maintains a 14 day history. On the 15th day, the first snapshot is removed.
- `(7,4)` is automatically adjusted to `(7,7)`.

**SECURE[, ,skip]**

Creates secure saved snapsets.

`skip` defines the skip value (0-1024). When specifying a `skip` value, the first snapshot created is always secure. A `skip` value of “1” results in secure snapshots 1, 3, 5, and so on. Specifying “2” results in secure snapshots 1, 4, 7, and so on.

The default behavior is not to create secure snapshots. If the `skip` value is 0, all saved snapshots will be secure.
SMF({[Yes,]|smf_id[,VOLUME(INITIAL)][,TRACKS({Yes|No})]|No})

Enables or disables zDP SMF recording for a VDG.

When this parameter is not specified, no SMF recording is performed (default).

This setting can be altered dynamically using the ZDP,MODIFY,SMF command of SCF, as described in the ResourcePak Base for z/OS Product Guide.

SMF(Yes|No)

Enables (Yes) or disables (No, default) SMF recording.

Note: When YES is specified, smf_id is required. YES is optional when smf_id is specified.

smf_id

Specifies an SMF record ID. Valid values are 128 through 255.

VOLUME(INITIAL)

Includes VDG devices on the startup record (CCUU and PowerMax/VMAX device numbers).

TRACKS({Yes|No})

Determines whether to include (Yes) or not (No, default) the total counts of changed and unique tracks for each device of each storage system in the VDG.

SNAPSET(snapset_id)

Specifies the snapset ID.

A snapshot ID is the portion of the snapset name after the VDG name. For example, for snapshot "VDG_UYF1.......153281429C00008", the vdg_name is "VDG_UYF1" and the snapset_id is "153281429C00008".

srdfgrp

A one or two-digit hexadecimal value representing a single SRDF group.

SRP_SNAP%(nnn)

Defines the maximum percentage of the SRP that can be allocated for SnapVX usage. When this value is reached, automatic termination of snapsets will be initiated based upon the termination policy. The default is 50%.

SRP_TERM%(nnn)

Defines the threshold of SRP usage to initiate automatic termination of snapshots, based upon the termination policy. The default is 100%.

SRP_WARN%(nnn)

Defines the threshold to issue a Warning message when the total Storage Resource Pool (SRP) usage exceeds this value. The default is 80%.

TERMinate_POLICY({OLDEST|STOP})

Defines the action to take when the MAX_SNAPSETS( nn nn) or either the SRP_SNAP% or the SRP_TERM% value is reached.
- **OLDEST** (default) terminates the oldest eligible snapshot.
- **STOP** terminates the VDG if the MAX\_RETURN\_RC value set for the VDG is less than 4.

### tgtst\_name

Specifies the target set.

**tgtst\_name** is case-sensitive, consisting of 1-15 alphanumeric characters. Special characters are allowed. Enclose the name in single quotes if it contains a dash.

### TIMEOUT\((nnn[,\{CONTinue|STOP\}])\)

*nnn* defines the timeout interval for consistency operations in seconds. Valid values are 1 through 60. The default value is 15.

The **TIMEOUT** parameter controls both the ECA timeout and the ACTIVATE retry timeout. If the SnapVX ACTIVATE commands do not complete within this interval, consistency cannot be guaranteed.

**CONTinue** (default) issues a warning message and continues zDP processing. **STOP** terminates in the event of a timeout.

### vdg\_name

Specifies the VDG name.

**vdg\_name** is case-sensitive, consisting of 1-15 alphanumeric characters. Special characters dash (-) and underscore (_) are allowed. Enclose the name in single quotes if it contains a hyphen. For example: 'MYVDG-P3'

### WAIT\_FOR\_Definition\({\{Yes|No\}}\)

Determines whether the UNLINK TGT or UNLINK VDG command will wait for all of the target tracks to be defined before the unlink occurs:

- **Yes** Wait for the target tracks to be fully defined before unlinking.
- **No** (Default) Do not wait for the target tracks to be fully defined before unlinking.

---

**Note:** “Target volume track definition” on page 37 explains the track definition process.

The **WAIT\_FOR\_DEFINITION** parameter is automatically set to NO for a command when FREE(YES) is specified for the command.

Aliases for **WAIT\_FOR\_DEFINITION** include **WAIT\_FOR\_D** and **WFD**.
DEFINE TGT

The DEFINE TGT statement creates and defines a new target set.

Note: “Target set (TGT)” on page 127 discusses target sets.

Syntax

DEFINE TGT tgtst_name[,NOSORT]

Note: TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

Required parameters

tgtst_name

See “tgtst_name” on page 145.

Optional parameters

NOSORT

See “NOSORT” on page 142.

Example

DEFINE TGT EMCTGT_QTR
DEFINE VDG

The DEFINE VDG statement creates and defines a new VDG.

Note: “Versioned Data Group (VDG)” on page 126 discusses VDGs.

Syntax

DEFINE VDG vdg_name
   [,,CONSistent({Yes|No})]
   [,,CYCLE_OVERFLOW({IMMED|NEXT})]
   [,,CYCLE_TIME(mmmn[,count][,SECURE,dd[,skip]])]
   [,,DEBUG([STATUS][,STATUSE])]
   [,,DYNAMIC_CHANGE({Yes|No})]
   [,,LOG_OPT({SCF|SYSOUT(ddname)})]
   [,,NOSORT]
   [,,MAX_RUNTIME_RC(nn)]
   [,,MAX_SNAPSETS(nnnn)]
   [,,PERSISTENT_COPY_LIMIT(nnnn)]
   [,,RDP_CACHE_UTIL%(ww,cc)]
   [,,SAVED_SNAPSETS(ddd,nnn[,SECURE[,skip]])]
   [,,SMF([[Yes,]smf_id[,VOLUME(INITIAL)][,TRACKS([Yes|No])|No]])]
   [,,SRP_SNAP%(nnn)]
   [,,SRP_TERM%(nnn)]
   [,,SRP_WARN%(nnn)]
   [,,TERMinate_POLICY({OLDEST|STOP})]
   [,,TIMEOUT(nn[,{CONTinue|STOP}])]

Note: VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

Required parameters

vdg_name
   See “vdg_name” on page 145.

Optional parameters

CONSistent({Yes|No})
   See “CONSistent((Yes|No))” on page 140.

CYCLE_OVERFLOW({IMMED|NEXT})
   See “CYCLE_OVERFLOW({IMMED|NEXT})” on page 140.
**CYCLE_TIME**([,,count][,SECURE,,ddd[,skip]])

See “CYCLE_TIME(mm,mm[,count][,SECURE,ddd[,skip]])” on page 140.

**DEBUG**([STATUS][,STATUSE])

See “DEBUG([STATUS][,STATUSE])” on page 141.

**DYNAMIC_CHANGE**([Yes|No])

See “DYNAMIC_CHANGE([Yes|No])” on page 141.

**LOG_OPT**([SCF|SYSOUT(ddname)])

See “LOG_OPT([SCF|SYSOUT(ddname)])” on page 141.

**MAX_RUNTIME_RC**(nn)

See “MAX_RUNTIME_RC(nn)” on page 141.

**MAX_SNAPSETS**(nnnn)

See “MAX_SNAPSETS(nnnn)” on page 142.

**NOSORT**

See “NOSORT” on page 142.

**PERSISTENT_COPY_LIMIT**(nnnn)

See “PERSISTENT_COPY_LIMIT(nnnn)” on page 142.

**RDP_CACHE_UTIL%**(ww,cc)

See “RDP_CACHE_UTIL%(ww,cc)” on page 142.

**SAVED_SNAPSETS**(ddd,nnnn[,SECURE[,skip]])

See “SAVED_SNAPSETS(ddd,nnnn[,SECURE[,skip]])” on page 143.

**SMF**([Yes,]smf_id[,VOLume(INITIAL)][,TRACKS([Yes|No])]|No)

See “SMF([Yes,]smf_id[,VOLume(INITIAL)][,TRACKS([Yes|No])]|No)” on page 144.

**SRP_SNAP%**(nnn)

See “SRP_SNAP%(nnn)” on page 144.

**SRP_TERM%**(nnn)

See “SRP_TERM%(nnn)” on page 144.

**SRP_WARN%**(nnn)

See “SRP_WARN%(nnn)” on page 144.

**TERMinate_POLICY**([OLDEST|STOP])

See “TERMinate_POLICY([OLDEST|STOP])” on page 144.

**TIMEOUT**(nnn[,{CONTinue|STOP}])

See “TIMEOUT(nnn,[{CONTinue|STOP}])” on page 145.
Example

DEFINE VDG EMCVDG1,
   CYCLE_TIME(10,128), CYCLE_OVERFLOW(NEXT),
   CONSISTENT(YES), TIMEOUT(30),
   TERM_POLICY(OLDEST),
   SRP_WARN%(75),
   SRP_TERM%(85),
   SRP_SNAP%(50),
   SAVED_SNAPSETS(5,5),
   MAX_SNAPSETS(025),
   PERSISTENT_COPY_LIMIT(005),
   LOG_OPT(SYSOUT(ZDPOUT5)),
   MAXRC(4)
DELETE TGT

The DELETE TGT statement deletes a specified target set. The target set cannot be in use (linked).

Syntax

DELETE TGT tgtst_name [,ALLOWNonEmpty]

**Note:** TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

Required parameters

- **tgtst_name**
  
  See “tgtst_name” on page 145.

Optional parameters

- **ALLOWNonEmpty**
  
  See “ALLOWNonEmpty” on page 139.

Example

DELETE TGT EMCTGT1,ALLOWNonEMPTY
DELETE VDG

The DELETE VDG statement deletes the specified VDG definition. The VDG must be inactive.

Syntax

```
DELETE VDG vdg_name [,ALLOWNonEmpty]
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

Required parameters

```
vdg_name
```

See “vdg_name” on page 145.

Optional parameters

```
ALLOWNonEmpty
```

See “ALLOWNonEmpty” on page 139.

Example

```
DELETE VDG EMCVDG1,ALLOWNE
```
GLOBAL

The GLOBAL statement specifies zDP Definition Utility execution options for each LPAR.

Only one GLOBAL statement is allowed. The GLOBAL statement should be the first statement in the input file.

Syntax

GLOBAL

[,MAX_RC(nn)]
[,MAX_VDG(nnnn)]
[,MAX_TGT(nnnn)]
[,SIMulate_TERMINate({Yes|No})]
[,WTOR_TERMINate({Yes|No})]

Optional parameters

MAX_RC(nn)

See “MAX_RC(nn)” on page 141.

MAX_TGT(nnnn)

Defines the maximum number of allowed target set definitions. The default value is 32. Valid values are 1 through 1024.

MAX_VDG(nnnn)

Defines the maximum number of allowed VDG definitions. The default value is 32. Valid values are 1 through 1024.

SIMulate_TERMINate({Yes|No})

When set to YES, simulates execution of the TERMINATE VDG,Date command and produces a report of all the snapsets eligible for termination by date/time. The default value is NO.

WTOR_TERMINate({Yes|No})

Determines whether a WTOR is issued to allow or deny the Terminate by date/time range action requested with the Date parameter of the TERMINATE VDG command.

- **YES**—*(Default)* Issue one WTOR per job step; where:
  - Replying CONTinue allows all TERMINATE VDG commands to proceed.
  - Replying CANcel ends the jobstep without processing any additional commands.
- **NO**—Do not issue a WTOR.

**Note:** Use the WTOR parameter of the TERMINATE VDG command to adjust the behavior for an individual TERMINATE VDG command.

Example

GLOBAL MAX_RC(4), MAX_TGT(1024), MAX_VDG(256)
LINK VDG

This statement links the volumes in the specified snapset to the specified target set.

Syntax

```
LINK VDG vdg_name, SNAPSET(snapset_id) [TO] TGT tgtst_name
[,ALLOW_LARGER_TARGET]
[,COPY_ONCE({INCLUDE|ONLY})]
[,MODE({COPY|NOCOPY})]
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: VDG(vdg_name) or TGT(tgtst_name).

Required parameters

**SNAPSET(snapset_id)**

See “SNAPSET(snapset_id)” on page 144.

**[TO] TGT tgtst_name**

Specifies the target set to which to link the volumes in the snapset.

**Note:** ‘TO TGT’ is an alias of ‘TGT’.

Enclose the name in single quotes if it contains a dash.

**vdg_name**

See “vdg_name” on page 145.

Optional parameters

**ALLOW_LARGER_TARGET**

Enables linking when a target set contains device(s) that have larger capacity than the VDG’s device(s).

ALLOW_LARGER_TARGET can be specified immediately after the VDG name or after the TGT name.

The aliases for ALLOW_LARGER_TARGET are LARGER_TARGET or LRGTGT.

**COPY_ONCE({INCLUDE|ONLY})**

See “COPY_ONCE({INCLUDE|ONLY})” on page 140.

Using COPY_ONCE(INCLUDE) on the LINK VDG command requires MAXRC=4 or higher.
zDP Definition Utility

MODE({COPY | NOCOPY})

Specifies when the background copy from source to target occurs:

COPY The source to target background copy should begin immediately after the snap is issued.

NOCOPY The background copy task is not initiated. As a result of the LINK operation, new track allocations for the target device will only occur when tracks on the target device are modified.

Example

LINK VDG EMCVDGR,SNAPSET(153621136C00018) TGT EMCTGTR
**MODIFY TGT ADD|REMOVE**

The MODIFY TGT ADD|REMOVE statement adds devices to or removes devices from the specified target set. You can specify multiple MODIFY statements.

**Syntax**

```plaintext
MODIFY TGT  tgtst_name,
ADD|REMOVE
{,CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|
,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|
,RDFGroup(gk,srdfgrp)|
,SCFG(gnsgrp)} [,RMT(srdfgrp)]
```

**Note:** TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

**Required parameters**

**ADD|REMOVE**

Specifies whether to add or remove devices.

**Note:** BCV and R2 devices cannot be configured in a target set.

```plaintext
CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|
,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|
,RDFGroup(gk,srdfgrp)|
,SCFG(gnsgrp)
```

See “CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|
,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|
,RDFGroup(gk,srdfgrp)|,SCFG(gnsgrp)” on page 139.

**tgtst_name**

See “tgtst_name” on page 145.

**Optional parameters**

**RMT(srdfgrp)**

See “RMT(srdfgrp)” on page 143.

**Examples**

```plaintext
MODIFY TGT EMCTGTR,ADD,
 SYMDEV(3008,0001A0-0001CF),
 REMOTE(70)
MODIFY TGT EMCTGT1,ADD,
 CCUU(3008,3270-32AF)
MODIFY TGT EMCTGT1,ADD,
 SCFG(EMCTGTGNS1)
MODIFY TGT EMCTGT1,REMOVE,
 CCUU(3008,3280-328F)
MODIFY TGT EMCTGT2,REMOVE,
 SYMDEV(3408,0260-026F)
MODIFY TGT EMCTGTR1,REMOVE,
 SYMDEV(3408,0001B0-0001B7),
 REMOTE(70.E0)
```
MODIFY VDG ADD|REMOVE

The MODIFY VDG ADD|REMOVE statement adds devices to or removes devices from the specified VDG. You can specify multiple MODIFY statements for the same VDG.

A device change will not affect an active VDG. The VDG must be stopped and restarted to recognize the changed configuration.

Syntax

```
MODIFY VDG  vdg_name,
ADD|REMOVE
{,CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|,RDFGroup(gk,srdfgrp)|,SCFG(gnsgrp)} [,RMT(srdfgrp)]
[,COPY_ONCE]
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

**Note:** COPY_ONCE cannot be used with MODIFY VDG, REMOVE and SCFG(gnsgrp).

Required parameters

ADD | REMOVE

Specifies whether to add or remove devices.

**Note:** BCV devices cannot be configured in a VDG.

```
CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|,RDFGroup(gk,srdfgrp)|,SCFG(gnsgrp)
```

See "CCUU(gk,start_ccuu[-end_ccuu][,start_ccuu[-end_ccuu]...])|,SYMDEV(gk,start_symdv#[-end_symdv#][,start_symdv#[-end_symdv#]...])|,RDFGroup(gk,srdfgrp)|,SCFG(gnsgrp)" on page 139.

vdg_name

See "vdg_name" on page 145.

Optional parameters

COPY_ONCE

Sets the copy-once attribute for devices included by the MODIFY VDG,ADD statement. If the device does not exist in the VDG, it will be added with the COPY_ONCE attribute. If it already exists, it will be replaced with the COPY_ONCE attribute.

**Note:** “Copy-once devices” on page 129 describes copy-once devices.
COPY_ONCE cannot be used with MODIFY VDG, REMOVE.

COPY_ONCE cannot be used with SCFG(gnsgrp).

RMT(srdfgrp)

See “RMT(srdfgrp)” on page 143.

Examples

MODIFY VDG EMCVDGR, ADD,
   SYMDEV(3008, 000170-00019F),
   REMOTE(70)

MODIFY VDG EMCVDG1, ADD,
   CCUU(3008, 3230-326F)

MODIFY VDG EMCVDG1, ADD,
   SCFG(EMCGNS1)

MODIFY VDG EMCVDG1, REMOVE,
   CCUU(3008, (3240-324F))

MODIFY VDG EMCVDG2, REMOVE,
   SYMDEV(3408, (0250-025F))

MODIFY VDG EMCVDGR1, REMOVE,
   SYMDEV(3408, 000180-000187),
   REMOTE(70.E0)

Following is an example of a remotely controlled VDG (SRDF group CA):

DEFINE VDG MYVDG,
   CONS(YES),
   CYCLE_TIME(10, 0),
   CYCLE_OVERFLOW(IMMED),
   TIMEOUT(10, CONT)

MODIFY VDG MYVDG, ADD,
   SYMDEV(AA00, 0000F0-0000F7),
   RMT(CA)
MODIFY VDG OPTIONS

The MODIFY VDG OPTIONS statement alters options for an existing VDG. For an active zDP task, any changed options will not affect the active VDG. You must stop and restart the VDG to implement the changes.

Syntax

```plaintext
MODIFY VDG vdg_name
  ,OPTIONS(option1[,option2]...[,option_n])
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

Possible values for `option` are:

- CONSistent(Yes|No)
- CYCLE_OVERFLOW(IME|NEXT)
- CYCLE_TIME(mmm,count[,SECURE,(NO|ddd[,skip])])
- DEBUG([STATUS][,STATUSE])
- DYNAMIC_CHANGE(Yes|No)
- LOG_OPT({SCF|SYSOUT(ddname)})
- MAX_RUNTIME_RC(nn)
- MAX_SNAPSETS(nn)
- NOSORT
- PERSISTENT_COPY_LIMIT(nn)
- RDP_CACHE_UTIL%(ww,cc)
- SAVED_SNAPSETS(ddd,nnn[,SECURE,(NO|skip)])
- SMF([Yes,]smf_id[,TRACKS([Yes|No])]|No))
- SRP_SNAP%(nnn)
- SRP_TERM%(nnn)
- SRP_WARN%(nnn)
- TERMINATE_POLICY(OLDEST|STOP))
- TIMEOUT(nn[,CONTINUE|STOP])

**Required parameters**

- `vdg_name`

  See “`vdg_name`” on page 145.

**Optional parameters**

- CONSistent(Yes|No)

  See “CONSistent(Yes|No)” on page 140.
CYCLE_OVERFLOW({IMMED|NEXT})

See “CYCLE_OVERFLOW({IMMED|NEXT})” on page 140.

CYCLE_TIME(mmmm[,count][,SECURE,NO|ddd[,skip]])

See “CYCLE_TIME(mmmm[,count][,SECURE,ddd[,skip]])” on page 140.

SECURE,NO resets the Secure setting.

DEBUG([STATUS][,STATUSE])

See “DEBUG([STATUS][,STATUSE])” on page 141.

DYNAMIC_CHANGE({Yes|No})

See “DYNAMIC_CHANGE({Yes|No})” on page 141.

LOG_OPT({SCF|SYSOUT(ddname)})

See “LOG_OPT({SCF|SYSOUT(ddname)})” on page 141.

MAX_RUNTIME_RC(nn)

See “MAX_RUNTIME_RC(nn)” on page 141.

MAX_SNAPSETS(nnnn)

See “MAX_SNAPSETS(nnnn)” on page 142.

NOSORT

See “NOSORT” on page 142.

PERSISTENT_COPY_LIMIT(nnnn)

See “PERSISTENT_COPY_LIMIT(nnnn)” on page 142.

RDP_CACHE_UTIL%(ww,cc)

See “RDP_CACHE_UTIL%(ww,cc)” on page 142.

SAVED_SNAPSETS(ddd,nnnn[,SECURE,[NO|skip]])

See “SAVED_SNAPSETS(ddd,nnnn[,SECURE,[skip]])” on page 143.

SECURE,NO resets the Secure setting.

SMF({[Yes,]smf_id[,TRACKS({Yes|No})]|No})

See “SMF({[Yes,]smf_id[,VOLUME(INITIAL)][,TRACKS({Yes|No})]|No})” on page 144.

Note: The VOLUME(INITIAL) parameter is not available on the MODIFY VDG OPTIONS command.

SRP_SNAP%(nnn)

See “SRP_SNAP%(nnn)” on page 144.

SRP_TERM%(nnn)

See “SRP_TERM%(nnn)” on page 144.

SRP_WARN%(nnn)

See “SRP_WARN%(nnn)” on page 144.

TERMinate_POLICY({OLDEST|STOP})
See “TERMinate_POLICY((OLDEST|STOP))” on page 144.

TIMEOUT(nnn[,{CONTinue|STOP}])

See “TIMEOUT(nnn[,{CONTinue|STOP}])” on page 145.

Example

```
MODIFY VDG MYVDG0, OPTIONS(MAX_SNAPSETS(25))
```

**PERSISTENT**

This statement sets or resets the “persistent” attribute for an existing snapset. Assigning the persistent attribute makes the snapset ineligible from automatic termination.

*Note:* “Persistent snapshots” on page 128 discusses persistent snapshots.

**Syntax**

```
PERSISTENT {SET|RESET}, VDG vdg_name, SNAPSET(snapset_id)
```

*Note:* VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

**Required parameters**

- **SET|RESET**
  
  Determines whether to set or reset the “persistent” attribute.

- **SNAPSET(snapset_id)**
  
  See “SNAPSET(snapset_id)” on page 144.

- **vdg_name**
  
  See “vdg_name” on page 145.

**Examples**

```
PERSISTENT SET, VDG EMCVDG1, SNAPSET(152761254C00011)
PERSISTENT RESET, VDG EMCVDG_TST, SNAPSET(153621211C00002)
```
QUERY FREE

This statement displays the number of tracks left to free up in SRP space and status of the FREE task (active or not active).

Syntax

```
QUERY FREE TGT {tgtst_name|*}
QUERY TGT {tgtst_name|*}, FREE
```

Note: TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

Required parameters

```
tgtst_name|*
```

See “tgtst_name” on page 145.

An asterisk (*) specifies to generate a report on all defined target sets.

Example

```
EMCP001I QUERY TGT TS_NBA1_N24B_A,FREE
EIP0029I TGT TS_NBA1_N24B_A is Not Linked
EIP0132I Free Query for TGT TS_NBA1_N24B_A

EIP0023I SYMM 0001970-00774, Microcode level 5978_0070, Type VMAX450F
EIP0024I Gatekeeper A028, Device Count: 16

EIP0131I CCUU DEVICE ALLOCATED TRKS FREE STATUS
EIP0131I ______ ___________ ___________ ___________

EIP0028I CC30 0003D0 0000001E NOT ACTIVE
EIP0028I CC31 0003D1 00012FB2 NOT ACTIVE
EIP0028I CC32 0003D2 00012FB2 NOT ACTIVE
EIP0028I CC33 0003D3 00012FB2 NOT ACTIVE
EIP0028I CC34 0003D4 00012FB2 NOT ACTIVE
EIP0028I CC35 0003D5 00012FB2 NOT ACTIVE
EIP0028I CC36 0003D6 00012FB2 NOT ACTIVE
EIP0028I CC37 0003D7 00012FB2 NOT ACTIVE
EIP0028I CC38 0003D8 00012FB2 NOT ACTIVE
EIP0028I CC39 0003D9 00012FB2 NOT ACTIVE
EIP0028I CC3A 0003DA 00012FB2 NOT ACTIVE
EIP0028I CC3B 0003DB 00012FB2 NOT ACTIVE
EIP0028I CC3C 0003DC 00012FB2 NOT ACTIVE
EIP0028I CC3D 0003DD 00012FB2 NOT ACTIVE
EIP0028I CC3E 0003DE 00012FB2 NOT ACTIVE
EIP0028I CC3F 0003DF 00012FB2 NOT ACTIVE
EIP0034I QUERY command completed
```

Output fields

- **CCUU**: The z/OS device number or "----" if not addressed.
- **DEVICE**: The PowerMax/VMAX device number.
- **ALLOCATED TRKS**: The number of allocated tracks.
- **FREE STATUS**: The status of the FREE process.
**QUERY TGT**

This statement displays target set information.

**Syntax**

```
QUERY TGT {tgtst_name|*} [,STATUS] [,DEVICE]
```

**Note:** TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

**Required parameters**

```
tgtst_name|*
```

See “tgtst_name” on page 145.

An asterisk (*) specifies to generate a report on all defined target sets.

**Optional parameters**

```
DEVICE
```

Generates a report of the devices comprising the target set definition.

```
STATUS
```

*(Default)* Displays target set status.

**Examples**

**Example 1:** The following examples show QUERY TGT, STATUS output, unlinked and linked:

```
EMCP001I QUERY TGT EMCTGTR,STATUS
EIP0029I TGT EMCTGTR is Not Linked
EIP0108I Options: NOSORT

EMCP001I QUERY TGT EMCTGT1,STATUS
EIP0029I TGT EMCTGT1 is Linked, SNAPSET EMCVDG1........160051020C00003
EIP0034I QUERY command completed
```
Example 2: 
**DEVICE, all devices not linked**

The following example shows QUERY TGT, DEVICE output when all devices are not linked:

EMCP001I QUERY TGT EMCTGTCC1,DEVICE

EIP0030I Device Query for TGT EMCTGTCC1

EIP0023I SYMM 0001234-05678, Microcode level 5977_0772, Type VMAX200K
EIP0024I Gatekeeper 3008, Device Count: 4, Remote(71)

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>000280</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: 70,71 /S</td>
</tr>
<tr>
<td>----</td>
<td>000281</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: 70,71 /S</td>
</tr>
<tr>
<td>----</td>
<td>000282</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: 70,71 /S</td>
</tr>
<tr>
<td>----</td>
<td>000283</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: 70,71 /S</td>
</tr>
</tbody>
</table>

EIP0023I SYMM 0001234-08765, Microcode level 5977_0772, Type VMAX200K
EIP0024I Gatekeeper 3008, Device Count: 4, Remote(EA)

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>000280</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: EB,EA /S</td>
</tr>
<tr>
<td>----</td>
<td>000281</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: EB,EA /S</td>
</tr>
<tr>
<td>----</td>
<td>000282</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: EB,EA /S</td>
</tr>
<tr>
<td>----</td>
<td>000283</td>
<td>CKD</td>
<td>10017</td>
<td>0001</td>
<td>R2: EB,EA /S</td>
</tr>
</tbody>
</table>

EIP0034I QUERY command completed

Example 3: 
**DEVICE, some devices linked**

The following example shows QUERY TGT, DEVICE output that includes both linked and unlinked devices:

EIP0030I Device Query for TGT TL_QRTGTDEV

EIP0023I SYMM 0001977-00074, Microcode level 5978_0144, Type VMAX950F
EIP0024I Gatekeeper 826E, Device Count: 6

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
<th>SRCDEV/CCUU</th>
</tr>
</thead>
<tbody>
<tr>
<td>826E</td>
<td>0003BE</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003BC/826C</td>
</tr>
<tr>
<td>826F</td>
<td>0003BF</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003BD/826D</td>
</tr>
<tr>
<td>8270</td>
<td>0003C0</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003CC/826E</td>
</tr>
<tr>
<td>8271</td>
<td>0003C1</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003CD/826E</td>
</tr>
<tr>
<td>8272</td>
<td>0003C2</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003CE/826E</td>
</tr>
<tr>
<td>8273</td>
<td>0003C3</td>
<td>CKD</td>
<td>1113</td>
<td>0102</td>
<td>---</td>
<td>0003CF/826E</td>
</tr>
</tbody>
</table>

Output fields

See “Output fields” on page 166.

**SRCDEV/CCUU**

The source PowerMax/VMAX device number and CCUU for linked devices.

**Note:** The 000000/---- value in the SRCDEV/CCUU column is shown for devices that are not linked.
zDP Definition Utility

QUERY VDG

This statement displays VDG information.

Syntax

QUERY VDG {vdg_name|*}
[,DEVice[,COPY_ONCE]]
[,SNAPSET[,DETAIL][,COPY_ONCE]]
[,start_dev[-end_dev]]
[,STATUS]
[,SYMM(gk)]

Note: The SNAPSET keyword may only be followed by DETAIL, COPY_ONCE, STATUS, SYMM(gk), or start_dev[-end_dev].

Note: VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

Required parameters

vdg_name|*

See “vdg_name” on page 145.

An asterisk (*) specifies to generate a report on all defined VDGs.

Optional parameters

DEVice[,COPY_ONCE]

Generates a report of the devices comprising the VDG definition.

COPY_ONCE

Filters the display to include only copy-one devices. For each copy-once device, displays the last snapset containing the copy-once device.

The COPY_ONCE parameter can be specified only once per command. If both DEVice and SNAPSET keywords are specified on the same command, the COPY_ONCE parameter applies to both DEVice and SNAPSET queries.

SNAPSET[,DETAIL][,COPY_ONCE]

Displays the snapsets created for the VDG.

Note: The SNAPSET keyword may only be followed by DETAIL, COPY_ONCE, STATUS, SYMM(gk), or start_dev[-end_dev].

COPY_ONCE

Displays only snapsets that contain copy-once devices.

The COPY_ONCE parameter can be specified only once per command. If both DEVice and SNAPSET keywords are specified on the same command, the COPY_ONCE parameter applies to both DEVice and SNAPSET queries.
DETAIL
Displays each device in the snapset.

\[\text{start\_dev}[-\text{end\_dev}]\]
Narrows the query to the specified range of devices.

STATUS
Displays VDG status, cycle and properties.

\[\text{SYMM}\(gk)\]
Narrows the query to the storage system specified with the gatekeeper CCUU.

Examples

**Example 1:**
**STATUS**
The following example shows QUERY VDG,STATUS output:

```
EMCP001I QUERY VDG EMC_SECURE,STATUS
EIP0020I VDG EMC_SECURE is Inactive, Cycle 0
EIP0020I Cycle_Time(10,0,SECURE,1),Cycle_Overflow(IMMED),Consistent(YES),Timeout(15,CONT)
EIP0020I Terminate_Policy(OLDEST),Dynamic_Change(NO)
EIP0020I SRP_Warn%(80),SRP_Snap%(50),SRP_Term%(100),RDP_Cache_Util%(60,95)
EIP0020I Max_Snapsets(256),Saved_Snapsets(0,1),Persistent_Copy_Limit(32)
EIP0021I Log_Opt(SCF)
EIP0021I SMF(NO)
EIP0021I MAXRC(4)
```

**Note:** The QUERY VDG,STATUS command lists properties set for the VDG. For explanation of each property, see the description of DEFINE VDG command parameters.

**Example 2:**
**DEVICE**
The following example shows QUERY VDG,DEVICE output for local devices.

```
EMCP001I QUERY VDG EMCVDG1,DEVICE
EIP0022I Device Query for VDG EMCVDG1
EIP0023I SYMM 0001234-05678, Microcode level 5977_0772, Type VMAX200K
EIP0024I Gatekeeper 3008, Device Count: 32
EIP0025I SRP ID/Name: 0001/SRP_1, Reserved Capacity: 10%
EIP0026I Total Capacity: 506M, Total Allocated: 4789K, Snap Allocated: 0
```

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF</th>
<th>INFO/MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3150</td>
<td>000170</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3151</td>
<td>000171</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3152</td>
<td>000172</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3153</td>
<td>000173</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3154</td>
<td>000174</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3155</td>
<td>000175</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3156</td>
<td>000176</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3157</td>
<td>000177</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3158</td>
<td>000178</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3159</td>
<td>000179</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>315A</td>
<td>00017A</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>315B</td>
<td>00017B</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td>---</td>
<td></td>
</tr>
</tbody>
</table>
The following example shows QUERY VDG,DEVICE output for remote devices.

**EMCP001I** QUER Y VDG EMCVDGCC1,DE VICE

**EIP002I** Device Query for VDG EMCVDGCC1

**EIP0022I** SYMM 0001234-05678, Microcode level 5977_0772, Type VMAX200K
**EIP0024I** Gatekeeper 3008, Device Count: 4, Remote(71)
**EIP0025I** SRP ID/Name: 0001/SRP_1, Reserved Capacity: 10%
**EIP0026I** Total Capacity: 506M, Total Allocated: 3557K, Snap Allocated: 22347

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td></td>
</tr>
</tbody>
</table>

**EIP0023I** SYMM 0001234-08765, Microcode level 5977_0772, Type VMAX200K
**EIP0024I** Gatekeeper 3008, Device Count: 4, Remote(EA)
**EIP0025I** SRP ID/Name: 0001/SRP_1, Reserved Capacity: 10%
**EIP0026I** Total Capacity: 506M, Total Allocated: 3698K, Snap Allocated: 22347

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td></td>
</tr>
</tbody>
</table>

The following example illustrates the QUERY VDG,DEVICE output with the COPY_ONCE option specified:

<table>
<thead>
<tr>
<th>CCUU</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
<th>SRP ID</th>
<th>RDF INFO/MODE</th>
<th>COPY_ONCE</th>
<th>SSET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3150</td>
<td>CKD</td>
<td>32760</td>
<td>0001</td>
<td></td>
<td>171360952800001</td>
<td></td>
</tr>
</tbody>
</table>

**Output fields**

**SRP ID/Name**
Lists the storage resource pool ID and name.

**Reserved Capacity**
Indicates the maximum percentage of the SRP that is allocated for host writes.

**Total Capacity**
Lists the total track capacity of the SRP.

**Total Allocated**
Lists the CKD and FBA tracks allocated to the SRP.

**Snap Allocated**
Lists the SnapVX-specific CKD and FBA tracks allocated to the SRP.

**CCUU**
The z/OS device number or "---" if not addressed.

**DEVICE**
Lists the PowerMax/VMAX device number of the device. "/O" indicates a copy-once device.
TYPE Lists the device type (CKD or FBA).
SIZE Lists the device size (in tracks).
SRP ID Lists the ID of the SRP to which the device belongs.
RDF INFO/MODE Lists the SRDF device type (R1, R2, R1, R21 or R22), SRDF groups, and SRDF mode for the device (S for synchronous, A for asynchronous (SRDF/A), and ADC for Adaptive Copy). For an active R2, the R2 group, R1 group /status is shown.
COPY_ONCE SSET Indicates the last saved snapset containing the copy-once device.

Example 3: SNAPSET

The following example shows QUERY VDG,SNAPSET output:

EMCP001I QUERY VDG EMCVDGC1,SNAPSET

Output fields

SRP ID/Name Lists the storage resource pool ID and name.
Reserved Capacity Indicates the maximum percentage of the SRP that is allocated for host writes.
Total Capacity Lists the total track capacity of the SRP.
Total Allocated Lists the CKD and FBA tracks allocated to the SRP.
Snap Allocated Lists the SnapVX-specific CKD and FBA tracks allocated to the SRP.
SNAPSET_NAME Specifies the snapshot name, with the VDG name followed by the snapshot ID.

Note: “Snapshot name” on page 126 explains the snapshot name string.
**STATE** Indicates the snapshot states as follows:

- CRE: Created
- ACT: Activated
- LNK: Linked
- RST: Restored

A STATE suffix value indicates the following:

- –S: Saved snapshot
- –P: Persistent snapshot
- –I: Secure snapshot

**CREATE DATE** Indicates the snapshot creation date.

**CREATE TIME** Indicates the snapshot creation time.

**SOURCE TRACKS CHANGED** Lists the number of changed tracks for the source volume.

**SOURCE TRACKS UNIQUE** Lists the number of tracks (for the source volume) that have only one snapshot pointing to them.

**EXPIRATION DATE** Indicates the date the snapshot will expire.

**EXPIRATION TIME** Indicates the time the snapshot will expire.

**Example 4:**

The following example shows the detailed part of the report produced with QUERY VDG,SNAPSET,DETAIL:

```
  EIP0036I SNAPSET_NAME STATE DATE TIME CHANGED UNIQUE SRCDEV/CCUU TGTDEV/CCUU
  ------------------------------------ ----- ------- ------- ------ ------- ------ ------- -------
  EIP0036I VDG_UYF1.......161371429800006 ACT-S 05/16/2016 14:29:19 0 0 000140/----
  EIP0036I ACT-S 05/16/2016 14:29:19 0 0 000141/----
  EIP0036I ACT-S 05/16/2016 14:29:19 0 0 000142/----
  EIP0036I ACT-S 05/16/2016 14:29:19 0 0 000143/----
  EIP0036I VDG_UYF1.......161381523800001 ACT-S 05/17/2016 15:23:10 0 0 000141/----
  EIP0036I ACT-S 05/17/2016 15:23:10 0 0 000142/----
  EIP0036I ACT-S 05/17/2016 15:23:10 0 0 000143/----
  EIP0036I ACT-S 05/17/2016 15:23:10 0 0 000144/----
  EIP0036I VDG_UYF1.......1613815444C00008 ACT 05/17/2016 15:44:10 0 0 000141/----
  EIP0036I ACT 05/17/2016 15:44:10 0 0 000142/----
  EIP0036I ACT 05/17/2016 15:44:10 0 0 000143/----
  EIP0036I ACT 05/17/2016 15:44:10 0 0 000144/----
  EIP0036I VDG_UYF1.......161381547C00009 ACT 05/17/2016 15:47:11 0 0 000141/----
  EIP0036I ACT 05/17/2016 15:47:11 0 0 000142/----
  EIP0036I ACT 05/17/2016 15:47:11 0 0 000143/----
  EIP0036I ACT 05/17/2016 15:47:11 0 0 000144/----
  EIP0036I VDG_UYF1.......1613815500C00010 ACT 05/17/2016 15:50:11 0 0 000141/----
  EIP0036I ACT 05/17/2016 15:50:11 0 0 000142/----
  EIP0036I ACT 05/17/2016 15:50:11 0 0 000143/----
  EIP0036I ACT 05/17/2016 15:50:11 0 0 000144/----
```

**Output fields**

The fields in this example have the same meaning as in the QUERY VDG,SNAPSET,DETAIL report. The two additional fields are as follows:

**SRCDEV/CCUU** The PowerMax/VMAX device number/CCUU of the source device.
TGTDEV/CCUU

The PowerMax/VMAX device number/CCUU of the target device.

**Note:** Target devices can be displayed for linked sources only. If a source device is not linked, the TGTDEV/CCUU column is blank.

### RESTORE VDG

The RESTORE VDG statement restores all of the source volumes in the specified snapset.

**Note:** A RESTORE is not allowed for active VDG source devices, online devices, and active R2 devices.

#### Syntax

RESTORE VDG  *vdg_name*, SNAPSET(*snapset_id*)

[*, COPY_ONCE({INCLUDE | ONLY})]*

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(*vdg_name*).

#### Required parameters

**SNAPSET(*snapset_id*)**

See “SNAPSET(*snapset_id*)” on page 144.

**vdg_name**

See “vdg_name” on page 145.

#### Optional parameters

**COPY_ONCE({INCLUDE | ONLY})**

See “COPY_ONCE({INCLUDE | ONLY})” on page 140.

#### Example

RESTORE VDG EMCVDG_PROD, SNAPSET(153621026C00004)
**SECURE VDG**

Converts an existing snapset to secure.

**Note:** “Secure snaps” on page 127 describes zDP secure snapsets.

**Syntax**

SECURE VDG  vdg_name,SNAPSET(snapset_id)
[,,EXPiration(ddd)[,,EXTEND]]

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

**Required parameters**

SNAPSET(snapset_id)
See “SNAPSET(snapset_id)” on page 144.

vdg_name
See “vdg_name” on page 145.

**Optional parameters**

EXPiration(ddd)[,,EXTEND]

The expiration time is specified as the number of days. The default value is 1 (one day).

When the expiration time expires, the operating environment automatically terminates the snapset (the VDG does not need to be active).

If the snapset is already secure, the action is bypassed with message EIP0081W.

EXTEND

Extends the current expiration time by the specified number of days.

**Example**

SECURE VDG  EMCVDG_PROD,SNAPSET(153621026C00004),EXPIration(30),EXTEND
STOP_FREE TGT

This statement terminates the process of freeing backend allocations held by a device.

Syntax

STOP_FREE TGT tgtst_name

Note: TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

Required parameters

tgtst_name

See “tgtst_name” on page 145.

Examples

STOP_FREE TGT EMCTGT_RMT
TERMINATE VDG

The TERMINATE VDG statement terminates the specified snapset in a VDG.

**Note:** Terminating a snapset does not necessarily affect RDP cache utilization.

### Syntax

```
TERMINATE VDG  vdg_name,
SNAPSET({(snapset_id|*INV*)})
TERMINATE VDG  vdg_name,
Date(startdate-enddate) [,WTOR({Yes|No})]
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

### Required parameters

```
Date(startdate-enddate) [,WTOR({Yes|No})]
```

Terminates snapsets within the specified period.

The `startdate` and `enddate` are specified in the following format: `yydddhhmm`.

**Note:** Saved and persistent snapshots are excluded from processing.

```
WTOR({Yes|No})
```

Determines whether a WTOR is issued to allow or deny the Terminate action.

- **Yes**—*(Default)* Issue one WTOR per jobstep; where:
  - Replying CONTinue allows all TERMINATE VDG commands to proceed.
  - Replying CANcel ends the jobstep without processing any additional commands.
- **No**—Do not issue a WTOR.

**Note:** Use the WTOR_TERMINate parameter of the GLOBAL command to set the WTOR behavior globally.

```
SNAPSET({(snapset_id|*INV*)})
```

Terminates the specified snapset or all invalid snapsets.

- ***INV***
  Terminates invalid snapsets.
  An invalid snapset is any snapset that is not in an Activated state.

`snapset_id`

Specifies the snapset ID. See “SNAPSET(snapset_id)” on page 144.

`vdg_name`

See “vdg_name” on page 145.

### Example

```
TERMINATE VDG  EMCVDG1,SNAPSET(153621045C00003)
```
**UNLINK TGT**

This statement removes the links from a target set.

The **UNLINK TGT** command is for a linked target set with the snapset showing a state of LNK.

**Syntax**

```
UNLINK TGT tgtst_name
{[,FREE(Yes|No)] | [,STOP_FREE(Yes|No)]}
[,WAIT_FOR_Definition(Yes|No)]
```

**Note:** TGT is an alias for TARGET_SET. You can also use optional parenthesis as follows: TGT(tgtst_name).

**Required parameters**

- `tgtst_name`
  - See “tgtst_name” on page 145.

**Optional parameters**

- `FREE(Yes|No)`
  - When FREE is set to YES, the tracks associated with the target devices are freed from the storage resource pool. The default value is NO.

  FREE processing will fail if a replication session exists on a target device. No backend space will be freed if any device in the request has a replication session.

  The most likely cause of a FREE failure would be a Thin Reclaim (TRU) SDDF session. The *ResourcePak Base for z/OS Product Guide* describes TRU and the commands required to display TRU status and stop or start TRU. If multiple SCF tasks are active, each one should be checked.

  FREE processing is initiated after UNLINK. If FREE processing fails, you can resubmit the same UNLINK command with FREE(YES) after the replication sessions on all of the target devices are removed. The “unlinked” state will be recognized and only FREE processing will be performed.

- `STOP_FREE(Yes|No)`
  - Terminates FREE processing.

- `WAIT_FOR_Definition(Yes|No)`
  - See “WAIT_FOR_Definition(Yes|No)” on page 145.

**Examples**

```
UNLINK TGT EMCTGT_RMT
```
UNLINK VDG

This statement unlinks a restored snapset (with the snapset state of RST) from a VDG.

Syntax

UNLINK VDG vdg_name,SNAPSET(snapset_id)
[,WAIT_FOR_Definition({Yes|No})]

Note: VDG is an alias for VERSIONED_DATA_GROUP. You can also use optional parenthesis as follows: VDG(vdg_name).

Required parameters

SNAPSET(snapset_id)
  See “SNAPSET(snapset_id)” on page 144.

vdg_name
  See “vdg_name” on page 145.

WAIT_FOR_Definition({Yes|No})
  See “WAIT_FOR_Definition({Yes|No})” on page 145.

Example

UNLINK VDG EMCVDGR,SNAPSET(153621026C00004)
CHAPTER 9
zDP ISPF Interface

This chapter covers the following topics:

◆ Introduction ........................................................................................................ 176
◆ Logging in to zDP ................................................................................................ 176
◆ Logging out of zDP ............................................................................................. 176
◆ Accessing zDP ISPF tools .................................................................................. 177
◆ Setting session options and GLOBAL parameters ............................................. 178
◆ Accessing zDP ISPF help .................................................................................... 178
◆ Monitoring and managing VDGs ....................................................................... 179
◆ Managing VDG members .................................................................................... 184
◆ Managing target sets ........................................................................................ 185
◆ Managing target set members .......................................................................... 190
◆ Managing snapsets ............................................................................................ 191
Introduction

The zDP ISPF interface uses standard IBM ISPF conventions. zDP ISPF panels provide a visual user interface to the zDP runtime environment. It allows you to manage, monitor, and report on various aspects of zDP.

Logging in to zDP

1. Modify EIPCLIST in the SAMPLIB to specify the following variables:
   - SCF:SUFFIX
     Certain functions within the zDP ISPF environment require connectivity to the SCF environment. The specified suffix is appended to SCF$ to provide this connection and must match the ddname in use within the SCF environment. For example, the suffix would be 'nnnn' if the SCF started task JCL has a ddname entry as follows:
     
     //SCF$nnnn DD DUMMY
   - SCF:JOBNAME
     Specify the SCF jobname that coincides with the SCF suffix.
   - UNIT
     Specify the unit for dataset allocations.
   - VOLUME
     Specify an optional volser to be used for dataset allocations.
   - EIP:LOAD_LIBRARY
     Specify the library containing the zDP runtime modules. Enter the PDS name in standard TSO format. This library must be authorized.

2. Log into a TSO account that is authorized to run zDP.

3. Enter the following command:

   TSO EIPCLIST

   The zDP Tool List panel is displayed.

Logging out of zDP

To log out of the zDP ISPF interface:

1. Press PF4 to return to the zDP Tool List panel.

2. Press PF3 to exit.
Accessing zDP ISPF tools

After you complete the steps described in “Logging in to zDP” on page 176, the zDP Tool List panel is displayed:

![Figure 1 zDP Tool List panel](image)

The zDP Tool List panel provides the following capabilities:

- Monitoring and managing VDGs (M Monitor)
- Managing VDG members (CV Configure)
- Managing target sets (DT Display)
- Managing target set members (CT Configure)
- Managing snapsets (1 Query)
- Setting session options and GLOBAL parameters (S Set Options)
- Exit the zDP ISPF interface

Before using zDP ISPF tools, you have to set the session control options. Follow the instructions provided in “Setting session options and GLOBAL parameters” on page 178.
Setting session options and GLOBAL parameters

In the zDP Tool List panel, choose S Set Options to open the Specify the Session parameters panel:

```
--- Specify the Session parameters ---

Command ===> .

SCF Suffix ........ LSV8 SCF Jobname ........ ESCF83VV
Product Load Library .... 'SCLM.ICDP.DEVVSU.V.LINKLIB'
VDG Member Library .... 'ABCDEF1.ICDP.VDG1'
TGT Member Library .... 'ABCDEF1.ICDP.TGT1'
Console Message Timeout .. 2
Unit .............. SYSALLDA and Volume .......... ______
Debug Mode ............... N (Y/N)

GLOBAL Options: MAX_R .. 00
MAX_TGT .. 32
MAX_VDG .. 32
SIM_TERM .. Y

PF1: Help Enter: Save Session PF3: Exit
```

Figure 2  Specify the Session parameters panel

The Specify the Session parameters panel lets you set zDP session control and GLOBAL parameters. These parameters must be specified before using other zDP functions. The saved options are stored in the ISPF user profile.

**Note:** For detailed information about each option, press PF1 to access help.

Accessing zDP ISPF help

zDP ISPF help panels provide detailed instructions on using the zDP ISPF panels.

Help information includes the panel purpose, description of input fields, user actions, and other useful details.

To access zDP ISPF help from any of zDP ISPF panels:

- Press PF1 to access help.
- Press PF8 to scroll through each Help panel.
- Press PF3 to return to the original panel.
Monitoring and managing VDGs

In the zDP Tool List panel, choose M Monitor to open the zDP VDG Monitor panel:

```
command=> Refresh=> 07:31:22 11/28/18
primary Cmds: CV = VDG  CT = TGT  U = SRP  Use REF = Refresh
line Cmds: S = Sel  C = Cnfg  M = Modify  D = Delete  Q = Query  R = Rept
X = Start  F = Stop  U = Relock  G = Resume  H = Pause  F = SMF
L = PCLEAR  Z = Clear
Max Snp  Snap Start  Next Snap
VDG Name  Status  Cnt  Cnt  Date  Time  Date  Time

AKR_VDG1  NOT ACTIVE  256  72
AKR_VDG2  ACTIVE    1024  120  11/28/18  07:10  11/28/18  08:10
ASDASD  NOT ACTIVE  256  0
```

Figure 3 zDP VDG Monitor panel

The zDP VDG Monitor panel allows you to monitor and manage runtime VDGs.

The contents of this panel refreshes automatically when one of the following zDP actions completes successfully: DELETE, DEFINE, PAUSE, STOP, START, RESUME, UNLINK, LINK. You can manually refresh the panel using the REF primary command.

Use this panel to:

- **Define VDG (CV primary command)**
- **Define target set (CT primary command)**
- **View SRP usage (U primary command)**
- **View VDG configuration parameters (S line command)**
- **View target set storage details (C line command)**
- **Modify VDG member (M line command)**
- **Delete inactive VDG from the runtime system (D line command)**
- **Query VDGs (Q line command) or view an abbreviated VDG report (R line command)**
- **Start VDG processing (X line command*) and stop it (P line command*)**
- **Pause VDG processing (H line command*) and resume (G line command*)**
- **Create an on-demand snapshot for an active VDG (Z line command*)**
- **Set SMF parameters (F line command*)**
- **Release zDP device locks for the VDG (U line command*)**
- **Clear ECA for an inactive VDG (L line command*)**

Commands marked with * in the list can also be executed through SCF, as described in the ResourcePak Base for z/OS Product Guide.

**Note:** For detailed information about each field and option, press PF1 to access help.
Enable automatic refresh

To automatically refresh the data shown in the zDP VDG Monitor panel, specify the refresh interval in the Refresh field. Valid values are 3 through 300 seconds.

Once automatic refresh is enabled, the panel is locked for input. Issuing an Attention request will interrupt the automatic refresh process and unlock the panel.

Define VDG

To define a new VDG, choose either of the following:

- In the zDP VDG Monitor panel, enter the CV primary command.
- In the zDP Target Set Display panel, enter the CV primary command.
- In the zDP Manage VDG Members panel, enter the CR primary command.

The Specify Versioned Data Group Parameters panel appears:

```
Specify Versioned Data Group Parameters

Command=> Scroll=> CSR

VDG Name . . ____________
Consistent . Yes _ or No _ Cycle Time . . ___ , ___ Secure ___ , ___
Cycle Overflow Immed _ or Next _ Debug . . Status _
Log Opt DDname ________ Max Snapsets . . ___
MaxRC . . . . ___ SRP Warn . . ___ SRP Term ___ SRP Snap ___
Persistent Limit . . ___
RDP Cache Util ___ , ___
Saved Snapsets ___ , ___ Secure ___
Timeout . . . ___ Cont _ Stop _ Term Policy . . Oldest _ Stop _
SMF rec number ___ Dynamic Change _
TRACKS . . . __ Nosort . . . __
VOLUME(INITIAL) _

Primary Commands: EDit SAve PRomote
```

Figure 4 Specify Versioned Data Group Parameters panel

The Specify Versioned Data Group Parameters panel lets you define a VDG and save it to the configuration dataset. After you set VDG operation parameters and press Enter, you are prompted to specify the VDG devices (see “Define VDG devices” on page 181).

Note: For detailed information about each option, press PF1 to access help.
Define VDG devices

The Specify VDG Device Parameters panel lets you define and modify the VDG devices:

```
| Specify VDG Device Parameters Enter required field |
|-------------------|----------------------------------|
| GK Device ____ Remote . . . . . . |
| GNS Group | |
| RDF Group CCUU ____ RA ____ |
| Symdev _ or CCUU _ Copy Once _ |
| Device ______ ______ ______ ______ ______ ______ ______ |
| ______ ______ ______ ______ ______ ______ ______ |
| ______ ______ ______ ______ ______ ______ ______ |
| ______ ______ ______ ______ ______ ______ ______ |
| ______ ______ ______ ______ ______ ______ ______ |
| Range ______ - ______ ______ - ______ ______ - ______ |
| ______ - ______ ______ - ______ ______ - ______ |
| ______ - ______ ______ - ______ ______ - ______ |
| ______ - ______ ______ - ______ ______ - ______ |
| ______ - ______ ______ - ______ ______ - ______ |
| Enter to Save or PF3 to Exit |
```

**Figure 5** Specify VDG Device Parameters panel

Complete the fields in the Specify VDG Device Parameters panel and press Enter to save. To create multiple device statements, press Enter multiple times.

**Note:** For detailed information about each option, press PF1 to access help.

View SRP usage

In the zDP VDG Monitor panel, enter the U primary command to open the zDP SRP Monitor panel:

```
| zDP SRP Monitor ------------------ Row 1 to 4 of 4 |
|-------------------|----------------------------------|
| Command=> |
| Refresh=> 05:38:37 11/30/18 |
| Ser | Rsv CKD CKD CKD FBA FBA CKD CKD CKD FBA FBA |
| Num | ID Cap Cap Aloc Aloc Cap Aloc Aloc Cap Aloc Aloc |
| Tot | Tot Snap Tot Tot Snap Tot Tot Snap Tot Tot Snap |
| Tot | Tot Tot Tot |
| Ser | Rsv CKD CKD CKD FBA FBA CKD CKD CKD FBA FBA |
| Num | ID Cap Cap Aloc Aloc Cap Aloc Aloc Cap Aloc Aloc |
| 00075 0102 10% 689M 37M 49K 0 0 0 39T 2T 3G 0 0 0 |
| 00074 0102 10% 689M 3M 299K 0 0 0 39T 193G 17G 0 0 |
| 00825 0001 10% 186M 5M 76K 19M 0 0 11T 256G 4G 1T 224G 0 |
| 00823 0001 10% 197M 27M 123K 0 0 0 11T 2T 7G 0 0 0 |
```

**Figure 6** zDP SRP Monitor panel

The zDP SRP Monitor panel shows the status of the storage resource pools (SRP) on all storage systems in the zDP runtime configuration.

**Note:** For detailed information about each field, press PF1 to access help.
Enable automatic refresh

To automatically refresh the data shown in the zDP SRP Monitor panel, specify the refresh interval in the Refresh field. Valid values are 120 to 1200 seconds.

Once automatic refresh is enabled, the panel is locked for input. Issuing an Attention request will interrupt the automatic refresh process and unlock the panel.

View VDG configuration parameters

In the zDP VDG Monitor panel, enter the S line command to open the zDP VDG Configuration Information panel:

```
+---------------- zDP VDG EDP_UBA1 Configuration Information -----------------+
| Command...: |
| Cycle Time: 3.0 Secure,No Cycle OverFlow: Immed |
| Cons: Yes Persistent Limit: 33 Timeout: 3,Cont |
| Max Snapsets: 1024 Term Policy: Oldest SMF Records: No |
| Saved Snapsets: 1,7 SRP Warn: 5 SRP Term: 15 SRP Snap: 10 |
| RDP Cache Util: 60,95 Dynamic Change: No |
| MAXRC: 0 Debug: STATUSE |
| SYSOUT DD Name: ICDPLOG Nosort: No |
| PF1: Help PF3: Exit |
+-----------------------------------------------------------------------------+
```

*Figure 7* zDP VDG Configuration Information panel

The zDP VDG Configuration Information panel shows the current settings for the selected VDG.

**Note:** For detailed information about each option, press PF1 to access help.

Modify VDG member

In the zDP VDG Monitor panel, enter the M line command to open the Modify Group Parameters panel. A panel similar to the one shown in *Figure 15 on page 188* appears, where you can edit, save, or promote the VDG configuration member.

**Note:** If you override an existing PDS member, the DEFINE VDG statement and the most recently specified MODIFY statements will be saved.
Query VDGs

In the zDP VDG Monitor panel, enter the Q line command to open the QUERY VDG options panel:

```
Row 1 to 3 of 3
Specify additional options for QUERY VDG command
Command=>
DEVICE S (S=Select)
STATUS S (S=Select)
COPY ONCE __________ (S=Select)
************************ Bottom of data ****************************
```

Figure 8 QUERY VDG options panel when querying VDGs

Set the query options as required and press Enter.

See “DEVice[,COPY_ONCE]” on page 164 and “SNAPSET[,DETAIL][,COPY_ONCE]” on page 164 for information about these settings.

Set SMF parameters

In the zDP VDG Monitor panel, enter the F line command to open the SMF parameters panel:

```
SMF parameters

Please update SMF parameters for VDG EDP_UBAUDA
SMF rec number 201
TRACKS .... S
```

Figure 9 SMF parameters panel

Set the SMF parameters as required and press Enter.

Note: For detailed information about each option, press PF1 to access help. See “SMF({[Yes,]smf_id[,VOLUME(INITIAL)][,TRACKS([Yes|No])]})” on page 144 for additional information about these settings.
Managing VDG members

In the zDP Tool List panel, choose CV Configure to open the Manage VDG Members panel:

```
Figure 10  Manage VDG Members panel
```

The Manage VDG Members panel allows you to view and manage VDG configuration members.

**Note:** For detailed information about each option, press PF1 to access help.
Managing target sets

In the zDP Tool List panel, choose DT Display to open the zDP Target Set Display panel:

![zDP Target Set Display panel](image)

Figure 11  zDP Target Set Display panel

The zDP Target Set Display panel allows you to view, link, and unlink target sets.

The contents of this panel refreshes automatically when one of the following zDP actions completes successfully: DELETE, DEFINE, PAUSE, STOP, START, RESUME, UNLINK, LINK. You can manually refresh the panel using the REF primary command.

Use this panel to:
- Define VDG (CV primary command)
- Define target set (CT primary command)
- View SRP usage (U primary command)
- View target set storage details (C line command)
- Modify target set member (M line command)
- Delete non-linked target set (D line command)
- Query target sets (Q line command) or view an abbreviated target set report (R line command)
- View the number of tracks left to free up in the SRP and whether the FREE task is active or not (O line command)
- Link target set (L line command)
- Unlink target set (U line command)
Define target set

To define a new target set, choose either of the following:

- In the zDP VDG Monitor panel, enter the CT primary command.
- In the zDP Target Set Display panel, enter the CT primary command.
- In the zDP Manage TGT Members panel, enter the CR primary command.

The Specify Target Set Parameters panel appears:

![Specify Target Set Parameters panel]

The Specify Target Set Parameters panel lets you define or modify the target set (TGT) name and devices.

To create multiple device statements, press Enter multiple times.

**Note:** For detailed information about each option, press PF1 to access help.
View target set storage details

In the zDP VDG Monitor panel or the zDP Target Set Display panel, enter the C line command to open the Configuration panel:

```
+---------------- TGT83 Configuration ----------------+ Row 1 to 1 of 1
Command==>  
<p>| Serial | GK | Eng | Dev | Remote |</p>
<table>
<thead>
<tr>
<th>Number</th>
<th>CCUU</th>
<th>Rel</th>
<th>Count</th>
<th>Hop List</th>
</tr>
</thead>
<tbody>
<tr>
<td>00197000825</td>
<td>6200</td>
<td>59780360</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Bottom of data **********************************</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 13 Configuration panel

The Configuration panel shows the storage system details for the selected target set.

Note: For detailed information about each option, press PF1 to access help.

View target set devices

In the Configuration panel, enter the S line command to open the Device Information panel:

```
+-------------------------- Device Information --------------------------+ Row 1 to 17 of 30
Command==>  
<table>
<thead>
<tr>
<th>CCUU</th>
<th>SDEV</th>
<th>CCUU</th>
<th>SDEV</th>
<th>CCUU</th>
<th>SDEV</th>
<th>CCUU</th>
<th>SDEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>----- 000014 ----- 000015 ----- 000016 ----- 000017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000018 ----- 000019 ----- 00001A ----- 00001B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 00001C ----- 00001D ----- 00001E ----- 00001F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000020 ----- 000021 ----- 000022 ----- 000023</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000024 ----- 000025 ----- 000026 ----- 000027</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000028 ----- 000029 ----- 00002A ----- 00002B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 00002C ----- 00002D ----- 00002E ----- 00002F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6200 000030 ----- 000031 ----- 000032 ----- 000033</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000034 ----- 000035 ----- 000036 ----- 000037</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000038 ----- 000039 ----- 00003A ----- 00003B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 00003C ----- 00003D ----- 00003E ----- 00003F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000040 ----- 000041 ----- 000042 ----- 000043</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000044 ----- 000045 ----- 000046 ----- 000047</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000048 ----- 000049 ----- 00004A ----- 00004B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 00004C ----- 00004D ----- 00004E ----- 00004F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000050 ----- 000051 ----- 000052 ----- 000053</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----- 000054 ----- 000055 ----- 000056 ----- 000057</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 14 Device Information panel

The Device Information panel lists the CCUUs and PowerMax/VMAX device numbers of devices defined for the selected target set on the selected storage system.

Note: For detailed information about each option, press PF1 to access help.
Modify target set member

In the zDP Target Set Display panel, enter the M line command to open the Modify Group Parameters panel:

```
  . Modify Group Parameters  Row 1 to 3 of 3 .
  . Command=>                Scroll=> CSR   .
  . *********************** Primary Commands: EDit SAve PRomote *********************** .
  . MODIFY TGT VBTGT1,ADD,   .
  .     SYMDEV(6200,         .
  .        000130)          .
  . *********************** Bottom of data ************************************************
```

**Figure 15** Modify Group Parameters panel for a target set

The Modify Group Parameters panel lets you edit, save, and promote the configuration member.

**Note:** For detailed information about each option, press PF1 to access help.

Query target sets

In the zDP Target Set Display panel, enter the Q line command to open the QUERY TGT options panel:

```
+----------------------------------------------------+
| Row 1 to 2 of 2                                      |
| Specify additional options                          |
| for QUERY TGT command                                |
| Command=>                                           |
| STATUS     S  (S=Select)                            |
| DEVICE     S  (S=Select)                            |
| ***************** Bottom of data *****************    |
+----------------------------------------------------+
```

**Figure 16** QUERY TGT options panel

Set the query options as required and press Enter.

**Note:** See “DEVice” on page 162 and “STATUS” on page 162 for information about these settings.
Link target set

In the zDP Target Set Display panel, enter the L line command to open the Specify Link Snapset Parameters panel:

```
Specify Link Snapset Parameters
Command=> Scroll=> CSR
VDG Name _______________ and SnapSet Name _______________
```

**Figure 17** Specify Link Snapset Parameters panel

Specify the VDG and the snapset to link directly to the selected target set. Press Enter to continue.

Unlink target set

In the zDP Target Set Display panel, enter the U line command to open the UNLINK options panel:

```
+----------------------------------------------------+
| Row 1 to 1 of 1 |
| Specify additional options |
| for UNLINK command |
| Command=> | |
| FREE N (Y=YES N=NO) |
| ******************** Bottom of data ******************** |
```

**Figure 18** UNLINK options panel

Set the FREE option as required and press Enter to continue.

**Note:** “FREE((Yes|No))” on page 173 describes the FREE option.
Managing target set members

In the zDP Tool List panel, choose CT Configure to open the Manage TGT Members panel:

![Figure 19 Manage TGT Members panel](image)

The Manage TGT Members panel allows you to view and manage target set configuration members.

**Note:** For detailed information about each option, press PF1 to access help.
Managing snapsets

In the zDP Tool List panel, choose 1 Query to open the zDP Snapset Controller Display panel:

Figure 20  zDP Snapset Controller Display panel

The zDP Snapset Controller Display panel shows storage system details per VDG. It lets you view and manage snapsets for the defined VDGs.

The contents of this panel refreshes automatically when one of the following zDP actions completes successfully: DELETE, DEFINE, PAUSE, STOP, START, RESUME, UNLINK, LINK. You can manually refresh the panel using the REF primary command.

Use this panel to:

◆ View VDG snapshots (S line command) and perform various snapshot actions on subsequent panels
◆ Query snapshots (Q line command)

Note: For detailed information about each field and option, press PF1 to access help.
View VDG snapshots

1. In the zDP Snapset Controller Display panel, enter the S line command to open the zDP VDG SnapSet Device panel:

```
1. ------------------- zDP VDG SDM_VDG3 SnapSet Device -- Row 1 to 1 of 1 .
   Command===>
   .
   . CCUU SDEV CCUU SDEV CCUU SDEV CCUU SDEV .
   . -------------------------------------------------------------------------- .
   . _ 0827C 0003CC _ 0827D 0003CD _ ----- ------ _ ----- ------ .
   . ******************************************************************************** Bottom of data ******************************************************************************** .
```

**Figure 21** zDP VDG SnapSet Device panel

The zDP VDG SnapSet Device panel lists storage system devices for the selected VDG.

*Note:* For detailed information about each field, press PF1 to access help.

2. In the zDP VDG SnapSet Device panel, select the devices and press Enter to open the zDP Snapset Display panel:

```
2. SDM_VDG3 zDP Snapset Display Row 1 to 69 of 209 .
   Command===>
   .
   . Primary Cmds: TR = Terminate Range  TI = Terminate Invalid .
   . Line Cmds: P = Set Persistence  L = Link  R = Restore .
   . E = Reset Persistence  U = Unlink VDG  T = Terminate .
   . S = Select  I = Make Secure .
   . Dev Num Date Time Chg Uni Chg Uni State Date Time .
   . ******************************************************************************** Bottom of data ******************************************************************************** .
```

**Figure 22** zDP Snapset Display panel

The zDP Snapset Display panel shows snapsets and their status.

*Note:* For detailed information about each option, press PF1 to access help.
Use this panel to:

- **Link target set** (L line command)
- Make snapshot persistent (P line command) or not (E line command)
- Make snapshot secure (I line command)
- **Restore to source** (R line command)
- Unlink target set (U line command)
- Terminate invalid snapshots (TI primary command)
- Terminate a snapshot for all volumes in the VDG (T line command)
- Terminate range of snapshots (TR primary command)

**Link target set**

1. In the zDP Snapset Display panel, select one or more snapshots and enter the L line command to open the Select Target Set to Link panel:

```
+---------------- Select Target Set to Link ----------------+
| Row1 to 5 of 5 |
| Command==> |
| Line Cmds: L = Link |
| TGT Name Serial GK Dev Remote |
| Number CCUU Count Hop List |
| AKR_TGT1 000197700075 AA00 5 |
| EDP_UBA_TGT1 000197000825 6203 4 |
| EDP_UDA_TGT1 000197700075 AA02 4 |
| EGJUBA_TGT 000197000825 6200 1 |
| UDB1_SDM_01 000197700074 8200 2 |
| ******************** Bottom of data ********************* |
```

*Figure 23  Select Target Set to Link panel*

The Select Target Set to Link panel lets you link a snapshot to the selected target set. Only unlinked target sets are displayed.

2. In the Select Target Set to Link panel, select a target set and enter the L line command to link it to the snapshot. The LINK options panel appears:

```
+---------------- Specify additional options for LINK command ----------------+
| Row 1 to 3 of 3 |
| Command==> |
| MAX_RC 04 |
| COPY ONCE (I=Include O=Only) |
| MODE COPY (COPY/NOCOPY) |
| ******************** Bottom of data ********************* |
```

*Figure 24  LINK options panel*
3. In the LINK options panel, specify the LINK options and press Enter.

For information about each option, see:

- “MAX_RC(nn)” on page 141 for MAX_RC
- “COPY_ONCE({INCLUDE|ONLY})” on page 140 for COPY ONCE
- “MODE({COPY|NOCOPY})” on page 154 for MODE

Make snapset secure

In the zDP Snapset Display panel, select one or more snapsets and enter the I line command to open the Make a Snapset Secure panel:

--- Make a Snapset Secure ---

Specify an expiration value and confirm the action.

Expiration: ___
VDG SDM_VDG3

Are you sure you want to do this?
N (Y=Yes N=No)

--- Extend expiration time for Secure Snapset ---

Specify an expiration value and confirm the action.

Expiration: ___
Extend: S (S=Select)
VDG SDM_VDG3

Are you sure you want to do this?
N (Y=Yes N=No)

Figure 25 Make a Snapset Secure panel

In the Make a Snapset Secure panel, specify the number of day after which the snapset or snapsets are to expire and confirm the action.

Note: For detailed information about each field, press PF1 to access help.

If the snapset you selected in the zDP Snapset Display panel is already secure, the Extend expiration time for Secure Snapset panel appears, prompting you to extend the snapset expiration period:

--- Extend expiration time for Secure Snapset ---

Specify an expiration value and confirm the action.

Expiration: ___
Extend: S (S=Select)
VDG SDM_VDG3

Are you sure you want to do this?
N (Y=Yes N=No)

--- Extend expiration time for Secure Snapset ---

Figure 26 Extend expiration time for Secure Snapset panel

Note: For detailed information about each option, press PF1 to access help.

If the Extend option not selected, the action is bypassed with message EIP0081W.
Restore to source

When you restore a snapshot back to its original source device, data on the source will appear as it was at the point in time the restored snapshot was taken.

In the zDP Snapset Display panel, enter the R line command to open the RESTORE options panel:

```
+----------------------------------------------------+
| Specify additional options                          |
| for RESTORE command                                 |
| COPY ONCE    (I=Include O=Only)                    |
+----------------------------------------------------+
```

Figure 27  RESTORE options panel

Set the COPY ONCE option as required and press Enter.

**Note:** See “COPY_ONCE((INCLUDE|ONLY))” on page 140 for information about the COPY ONCE option values.

Terminate range of snapsets

In the zDP Snapset Display panel, select the snapsets and enter the TR primary command to open the Terminate a range of snapsets panel:

```
+------- Terminate a range of snapsets -------+
| From date 10 / 22 / 18  05 : 42            |
| To date  10 / 22 / 18  07 : 42             |
| Format   mm / dd / yy  hh : mm             |
| Simulate Y (Y=Yes/N=No)                    |
+---------------------------------------------+
```

Figure 28  Terminate a range of snapsets panel

The Terminate a range of snapsets panel lets you specify two dates to terminate the snapsets within the specified period. Creation dates of the selected snapsets are used as the start and end dates for the range of snapsets.
Query snapsets

In the zDP Snapset Controller Display panel, enter the Q line command to open the QUERY VDG options panel:

```
+----------------------------------------------------+
| Row 1 to 2 of 2                                       |
| Specify additional options                           |
| for QUERY VDG command                                |
| Command=>                                               |
| DETAIL ________ (S=Select)                            |
| COPY ONCE ________ (S=Select)                          |
| ***************** Bottom of data *****************       |
+----------------------------------------------------+
```

Figure 29  QUERY VDG options panel when querying snapsets

Set the QUERY VDG options as required and press Enter.

Note: “SNAPSET[,DETAIL][,COPY_ONCE]” on page 164 provides information about the DETAIL and COPY_ONCE options.
CHAPTER 10
SMF Records

This chapter covers the following topics:

- Overview .......................................................... 198
- SMF record format .............................................. 198
Overview

zDP can optionally generate an SMF record for each cycle, as well as at VDG startup and shutdown.

**Note:** By default, no SMF recording is performed.

To set up zDP SMF recording, use the following controls:

- The DEFINE VDG,SMF command of the zDP Definition Utility described in “DEFINE VDG” on page 147.
- The 'SMF rec number' and 'TRACKS' options in the zDP ISPF Specify Versioned Data Group Parameters panel; F = SMF command in the zDP VDG Monitor panel
- The ZDP,MODIFY,SMF command of Symmetrix Control Facility (SCF) described in the ResourcePak Base for z/OS Product Guide.

**Note:** The SCF ZDP,MODIFY,SMF command requires the VDG to be active. The SMF parameters can be changed for an inactive VDG with the zDP Definition Utility.

When SMF recording is enabled, message EIP0250I is issued during initialization of the VDG. This message is also issued at the start of a cycle on behalf of any SMF parameter change using a ZDP,MODIFY, SMF command. If SMF recording is dynamically disabled, message EIP0251I is issued at the start of the next cycle.

To view SMF recording status with the zDP Definition Utility, issue the QUERY VDG,STATUS command and look for the string starting with 'SMF'.

SMF record format

zDP SMF records have a standard SMF header and multiple data sections in a self-describing format. The zDP SMF record contains a section representing the VDG, and one section for each storage system configured in the VDG.

SMF record header

```
**
* STANDARD SMF RECORD HEADER
**
EIPSMFD_LEN DS H RECORD LENGTH
EIPSMFD_SEG DS H SEGMENT DESCRIPTOR (ALWAYS 0)
EIPSMFD_FLG DS XL1 SYSTEM INDICATOR
EIPSMFD_RTY DS XL1 SMF RECORD NUMBER
EIPSMFD_TME DS XL4 TIME IN 100THS OF A SECOND
EIPSMFD_DTE DS XL4 DATE IN PACK DECIMAL - 0CYYDDDF
EIPSMFD_SID DS XL4 SYSTEM ID
**
* SUBTYPE IDENTIFIER
**
EIPSMFD_SSI DS XL4 SUBSYSTEM ID
EIPSMFD_STY DS XL2 SUBTYPE (USES X'00'-'X'03')
EIPSMFD_STY_START EQU 1 . Start record
EIPSMFD_STY_CYCLE EQU 2 . Cycle record
EIPSMFD_STY_STOP EQU 3 . Stop record
EIPSMFD_HDR_LEN DS H 1-byte gas
EIPSMFD_HDR_LEN DS H 2-byte gas
```

EIPSMFD_HDR_LEN zzzDS zH zzzzzzzLength of the SMF Record header
Section header

The SMF header is immediately followed by a header describing the number of data sections.

**
* Section Header - Version, section count
**

EIPSMFD_DATA_HDR DS 0H Data Header
EIPSMFD_HDR_LENGTH DS H Length of this header
EIPSMFD_VERSION DS XL2 Record Version
EIPSMFD_VERSION# EQU 1 Current version
EIPSMFD_SECTION_CNT DS H Number of Section Descriptors
EIPSMFD_SECTION# EQU 3 Currently 3

EIPSMFD_F1_SEG# DS XL1 Segment number

EIPSMFD_SEG# DS XL1 Flag byte 1
EIPSMFD_F1_SEGLAST EQU X'80' . Last Segment record
EIPSMFD_F1_SEG EQU X'02' . Segment record
EIPSMFD_F1_DEVS EQU X'01' . Device list included

EIPSMFD_SECTIONS DS 0F First section descriptor
EIPSMFD_HDR_DSLN EQU *-EIPSMFD_DATA_HDR Header length
EIPSMFD_DSLN EQU *-EIPSMFD Total length to this point

Section descriptors

The section header is immediately followed by a variable number of section descriptors whose format is described by the following DSECT.

**
* DSECT for the section descriptors - OFF/LEN/CNT fields
*
* Currently, there are three: BASE, SYMM, DEV
**

EIPSMFDS DSECT , Offsets are from EIPSMFD
EIPSMFDS_LEN DS F Length of this structure
EIPSMFDS_OFF DS F Offset of section
EIPSMFDS_COUNT DS F Number of entries in section
EIPSMFDS_ID DS F Section ID
EIPSMFDS_ID_BASE EQU 1 . BASE section
EIPSMFDS_ID_SYMM EQU 2 . SYMM section
EIPSMFDS_ID_DEV EQU 3 . DEV section
EIPSMFDS_DSLN EQU *-EIPSMFDS Section descriptor length

There are potentially three types of sections:

- Base section—contains general information describing the zDP environment.
- Symm section—contains entries for a variable number of storage systems.
- Device section (optional)—contains device information for each zDP managed device on the associated storage system.

Records always contain the Base and Symm data elements.

If the device data is not being recorded, then data for all managed storage systems are contained in one SMF record and the section descriptor for the Dev section has offset and number of entries set to zero.
If device data is available, then each record contains a Symm section for one storage system and the Device section contains the data for all devices under its control.

**Note:** If the total number of devices on a storage system exceeds EIPSMFD_SMF_SYMDEV##_MAX, then the number of device entries per SMF record is limited to that number and multiple records are written for the same storage system until all device data has been recorded.

### Base section

The Base section is defined by the following DSECT.

```plaintext
**
* Basic Section - Corresponds to EIPEXIT header
**
EIPSMFD_SMF_BASE DSECT , Basic Section
EIPSMFD_SMF_BASELEN DS F Length of this entry
EIPSMFD_SMF_SYSNAME DS CL8 System Name
EIPSMFD_SMF_CYCTM DS F Cycle time (minutes)
EIPSMFD_SMF_CYC# DS F Cycle number
EIPSMFD_SMF_SSET DS CL32 Snapset Name
EIPSMFD_SMF_VDGNAME DS CL15 VDG Name

.* Time fields are in SMF time/date format:
.* - Time (binary), in hundreths of a second since midnight
.* - Date (packed), in the form 0cyydddF
EIPSMFD_SMF_TMSTART DS XL8 Cycle Start time
EIPSMFD_SMF_TMCREATE DS XL8 Cycle Create time
EIPSMFD_SMF_TMEND DS XL8 Cycle End time
EIPSMFD_SMF_TMNEXT DS XL8 Next Cycle Start time

.*
EIPSMFD_SMF_SYM# DS F Count of Symmetrix controllers
EIPSMFD_SMF_TOTDEV# DS F Total VDG Device count

.*
EIPSMFD_SMF_SSET# DS XL2 Snapset count
EIPSMFD_SMF_RC DS XL2 Return Code
EIPSMFD_SMF_RSCN DS XL2 Reason Code
EIPSMFD_SMF_RDGROUP DS XL4 Extended Return Code

.*
EIPSMFD_SMF_VDGFLGS DS XL16 VDG Flags (see EIPVDG_FLAGS)

.*
EIPSMFD_SMF_BASE_LN EQU *-EIPSMFD_SMF_BASE
```

### Symm section

Each entry in the Symm section is described by the following DSECT.

```plaintext
**
* Symmetrix Controller Section - one controller
**
EIPSMFD_SMF_SYM DSECT , Symmetrix Controller
EIPSMFD_SMF_SYMMLEN DS F Section length
EIPSMFD_SMF_SYMMLENSRP DS F - Length including SRP(s)
EIPSMFD_SMF_SYMMLENDEVS DS F - Length including SRP(s) + DEVs
EIPSMFD_SMF_SYMGRK DS F GateKeeper CCUU
EIPSMFD_SMF_SYMSER DS CL12 Serial number
EIPSMFD_SMF_SYMMXLC DS XL4 Microcode level
EIPSMFD_SMF_SYMRDFGRP DS XL16 RDF Group list (x'FF' for local)
EIPSMFD_SMF_SYMUCB@ DS A GateKeeper UCB address
EIPSMFD_SMF_SYMDEV DS CL15 VDG Name
EIPSMFD_SMF_SYMDEV# DS F Device Count - total for Symm
```
Within the SYMM section, each SYMM element is immediately followed by elements describing the Symmetrix Resource Pools on that storage system that are in use within the snapset. The number of SRP elements is EIPSMFD_SMF_SRP# within the owning storage system.

**
* SRP Entry - One per SRP per controller
**

EIPSMFD_SMF_SRPLN EQU *-EIPSMFD_SMF_SRPLN Section length
EIPSMFD_SMF_SRPID DS XL4 SRP Id
EIPSMFD_SMF_SRPNM DS CL32 SRP Name
EIPSMFD_SMF_SRPRSV DS X SRP Reserved Capacity %
EIPSMFD_SMF_SRPTOT DS XL8 SRP Total Tracks
EIPSMFD_SMF_SRPAKC DS XL8 SRP Allocated Tracks
EIPSMFD_SMF_SRPSNP DS XL8 SRP Snap Tracks

Device section

The Device section is defined by the following DSECT.

**
* Device Section, Devices for one controller
* - optional, Device count in EIPSMFD_SMF_SYMDEV#_DEV
**

EIPSMFD_SMF_DEV DSECT , Devices
EIPSMFD_SMF_DEVCCUU DS F CCUU
EIPSMFD_SMF_DEVSYMDS DS F Symm device number
EIPSMFD_SMF_DEVLN1 EQU *-EIPSMFD_SMF_DEV (Short length)
EIPSMFD_SMF_DEVTOTCHG DS F Source changed track count
EIPSMFD_SMF_DEVTUKUNI DS F Source unique track count
EIPSMFD_SMF_DEVNEXT EQU * Next entry, etc. (if any)
EIPSMFD_SMF_DEVLN EQU *-EIPSMFD_SMF_DEV