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 Getting Started

TimeFinder SnapVX and TimeFinder/Clone Mainframe Snap Facility ..... 20
Post-installation ............................................................................... 20
Software interoperability considerations ......................................... 20
SIBBATCH ...................................................................................... 20
SRS .............................................................................................. 20
FlashCopy ...................................................................................... 20
Starting ResourcePak Base ............................................................. 21
Setting up security .......................................................................... 21
Running TimeFinder/Clone Mainframe Snap Facility ..................... 22

Chapter 3 Configuration

TimeFinder SnapVX configuration layers ........................................... 24
Configuration Layer 1: EMCSNAPO site options ......................... 24
Configuration Layer 2: GLOBAL command parameters .............. 24
Configuration Layer 3: Parameters on other commands ............... 25
Editing the EMCSNAPO macro ...................................................... 25
EMCSNAPO site options .................................................................. 26
Summary ........................................................................................ 26
ACTIVATE_SUBTASK# .............................................................. 27
ALLOW_FBA_META ................................................................. 27
ALLOWZDP ................................................................................. 28
AUTOCREATE ............................................................................... 28
AUTOUNLINK ............................................................................... 28
BACKGRND ............................................................................... 28
CACHESYM ............................................................................... 28
CHKLTGT .................................................................................. 28
CHKONLIN ............................................................................... 28
CONDVOL .................................................................................. 29
CONSIST ................................................................................... 29
COPYVOL .................................................................................. 29
DEBUG_ERROR .......................................................................... 29
DEBUG_SDUMP .......................................................................... 29
EMUL_TYPE ............................................................................... 29
FREEUNLK ............................................................................... 29
GROUP_DSNAME ................................................................. 30
MESSAGE .................................................................................. 30
REFVTOC ................................................................................. 30
## Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELLKTGT</td>
<td>30</td>
</tr>
<tr>
<td>REPLACE</td>
<td>30</td>
</tr>
<tr>
<td>RESTORE_CREATE</td>
<td>30</td>
</tr>
<tr>
<td>SECURE</td>
<td>31</td>
</tr>
<tr>
<td>SECURE_QUERY</td>
<td>31</td>
</tr>
<tr>
<td>SETLKTGT</td>
<td>31</td>
</tr>
<tr>
<td>SNAPSHOT_LIST</td>
<td>31</td>
</tr>
<tr>
<td>SNAPSHOT_NAME</td>
<td>31</td>
</tr>
<tr>
<td>SOFTLINK</td>
<td>31</td>
</tr>
<tr>
<td>SOFTRESTORE</td>
<td>31</td>
</tr>
<tr>
<td>SRDFA_RETRY</td>
<td>32</td>
</tr>
<tr>
<td>SRPPERCT</td>
<td>32</td>
</tr>
<tr>
<td>SRPMSGLEVEL</td>
<td>32</td>
</tr>
<tr>
<td>TERM_ALL</td>
<td>32</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>32</td>
</tr>
<tr>
<td>VARYOFF</td>
<td>32</td>
</tr>
<tr>
<td>VARYON</td>
<td>32</td>
</tr>
<tr>
<td>VCLOSE</td>
<td>33</td>
</tr>
<tr>
<td>UNLINKAFT</td>
<td>33</td>
</tr>
<tr>
<td>WFDEF</td>
<td>33</td>
</tr>
</tbody>
</table>

### Chapter 4 Operations

- Softlinked and hardlinked snapshots ........................................................ 36
- Snapshot life cycle .................................................................................. 36
- SnapVX group status transitions ........................................................... 37
- Target volume track definition ............................................................... 37
- Link target holds ................................................................................... 38
- Monitoring SRP usage during linking ....................................................... 39
- Secure snaps ............................................................................................ 40
- Summary of operations ............................................................................ 42
  - Managing snapshots ........................................................................... 42
  - Viewing snapshots .............................................................................. 42
  - Viewing groups .................................................................................. 43
  - Fine-tuning snapshot processing ........................................................ 43
  - Linking/unlinking and copying ........................................................... 43
  - Setting up consistency .................................................................... 44
  - Managing link target holds ................................................................. 45
  - Controlling device status ................................................................ 45
  - Conditioning target volume ............................................................... 46
- Miscellaneous SnapVX settings ............................................................. 46
- Example of operations ........................................................................... 47
- Defining groups of statements ............................................................... 48
  - Specifying the group dataset ............................................................... 48
  - Group dataset allocation example ...................................................... 49
- Defining groups .................................................................................... 49
  - Referencing other groups from within a group ................................ 50
  - Using groups as arguments to SnapVX commands ............................ 51

### Chapter 5 Command Reference

- Conventions ............................................................................................. 54
- SnapVX commands .................................................................................. 54
  - Common parameters .......................................................................... 54
  - ACTIVATE .......................................................................................... 72

---

4 TimeFinder SnapVX and zDP 8.3 Product Guide
## Configurations

- **CONFIG** .................................................................................................................. 75
- **CREATE** .................................................................................................................. 78
- **FREE** ...................................................................................................................... 80
- **GLOBAL** ................................................................................................................ 82
- **LINK** ....................................................................................................................... 90
- **QUERY FREE** ......................................................................................................... 94
- **QUERY GLOBAL** .................................................................................................... 96
- **QUERY GROUP** ....................................................................................................... 97
- **QUERY SNAPSHOT** ............................................................................................... 98
- **RENAME** ................................................................................................................ 106
- **TERMINATE** .......................................................................................................... 108
- **UNLINK** ................................................................................................................ 111

## Grouping Commands

- **DEFINE GROUP** .................................................................................................... 114
- **DELETE GROUP** ..................................................................................................... 116
- **END GROUP** ......................................................................................................... 116

## Overview

- **Chapter 6 Introduction**
  - Overview .................................................................................................................. 120
  - Terminology .......................................................................................................... 120
  - Implementation ..................................................................................................... 121
  - Requirements and limitations .............................................................................. 122
  - Security considerations ......................................................................................... 122

- **Chapter 7 zDP Definition Utility**
  - Overview .................................................................................................................. 124
    - Summary of operations .......................................................................................... 124
    - Secure snaps ........................................................................................................ 128
    - Copy-once devices ............................................................................................... 130
  - Running zDP definition utility ............................................................................. 132
  - zDP definition utility statements ........................................................................ 133
  - Syntax conventions ................................................................................................ 133
  - DEFINE TGT ........................................................................................................... 133
  - DEFINE VDG .......................................................................................................... 134
  - DELETE TGT ............................................................................................................ 139
  - DELETE VDG .......................................................................................................... 139
  - GLOBAL .................................................................................................................. 140
  - LINK VDG ............................................................................................................... 141
  - MODIFY TGT ADD|REMOVE .................................................................................. 143
  - MODIFY VDG ADD|REMOVE .................................................................................. 145
  - MODIFY VDG OPTIONS ......................................................................................... 148
  - PERSISTENT ............................................................................................................. 150
  - QUERY FREE ........................................................................................................... 151
  - QUERY TGT ............................................................................................................. 152
  - QUERY VDG ............................................................................................................. 154
  - RESTORE VDG ....................................................................................................... 159
  - SECURE VDG .......................................................................................................... 160
  - STOP_FREE TGT ..................................................................................................... 161
  - TERMINATE VDG ................................................................................................... 162
  - UNLINK TGT ............................................................................................................ 164
  - UNLINK VDG ........................................................................................................... 165

---

TimeFinder SnapVX and zDP 8.3 Product Guide
Chapter 8  zDP ISPF Interface

Introduction ................................................................. 168
Logging in to zDP .......................................................... 168
Logging out of zDP ......................................................... 169
zDP functions .............................................................. 169
  zDP help ....................................................................... 170
  Session Control options .............................................. 171
  VDG monitor ............................................................. 171
  VDG configure .......................................................... 173
  TGT display .............................................................. 173
  TGT configure .......................................................... 174
Snapset functions ......................................................... 175

Chapter 9  SMF Records

Overview ................................................................. 178
SMF record format ....................................................... 178
  SMF record header .................................................... 178
  Section header ......................................................... 179
  Section descriptors ................................................. 179
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EMCSNAPO site options</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>Allowable SnapVX commands per snapshot state</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>SnapVX group actions and statuses</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>Managing snapshots</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>Viewing snapshots</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Viewing groups</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Fine-tuning snapshot processing</td>
<td>43</td>
</tr>
<tr>
<td>8</td>
<td>Linking/unlinking and copying</td>
<td>43</td>
</tr>
<tr>
<td>9</td>
<td>Setting up consistency</td>
<td>44</td>
</tr>
<tr>
<td>10</td>
<td>Controlling device status</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>Controlling device status</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>Conditioning target volume</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>Miscellaneous SnapVX settings</td>
<td>46</td>
</tr>
<tr>
<td>14</td>
<td>Managing VDGs</td>
<td>124</td>
</tr>
<tr>
<td>15</td>
<td>Managing target sets</td>
<td>124</td>
</tr>
<tr>
<td>16</td>
<td>Managing snapsets</td>
<td>124</td>
</tr>
<tr>
<td>17</td>
<td>Setting VDG snapset-related options</td>
<td>125</td>
</tr>
<tr>
<td>18</td>
<td>Setting SRP usage and RDP cache utilization parameters</td>
<td>125</td>
</tr>
<tr>
<td>19</td>
<td>Managing copy-once devices</td>
<td>126</td>
</tr>
<tr>
<td>20</td>
<td>Monitoring zDP status</td>
<td>126</td>
</tr>
<tr>
<td>21</td>
<td>zDP messaging and SMF records</td>
<td>127</td>
</tr>
<tr>
<td>22</td>
<td>Setting zDP runtime task options</td>
<td>127</td>
</tr>
</tbody>
</table>
PREFACE

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.
- *VMAX3 Family Product Guide*—Documents the features and functions of the VMAXX 100K, 200K, and 400K storage systems.
- *VMAX Family Product Guide*—Documents the features and functions of the VMAX 10K, 20K, and 40K storage systems.
- *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 conventions for special notices:

- **CAUTION**, used with the safety alert symbol, indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

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

- **IMPORTANT**

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

Typographical conventions

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

- **Normal**: Used in running (nonprocedural) text for:
  - Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus
  - Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities
  - URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, and notifications

- **Bold**: Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages

  Used in procedures for:
  - Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus
  - What the user specifically selects, clicks, presses, or types
Where to get help

Dell EMC support, product, and licensing information can be obtained on the Dell EMC Online Support site as described next.

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

Product information

For documentation, release notes, software updates, or for information about Dell EMC products, licensing, and service, go to the Dell EMC Online Support site (registration required) at:

support.EMC.com

Technical support

Dell EMC offers a variety of support options.

Support by Product—Dell EMC offers consolidated, product-specific information at:

support.EMC.com/products

The Support by Product web pages offer quick links to Documentation, White Papers, Advisories (such as frequently used Knowledgebase articles), and Downloads, as well as more dynamic content, such as presentations, discussion, relevant Customer Support Forum entries, and a link to Dell EMC Live Chat.

Dell EMC Live Chat—Open a Chat or instant message session with a Dell EMC Support Engineer.
eLicensing support

To activate your entitlements and obtain your license files, visit the Service Center on support.EMC.com, 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's 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 2, “Getting Started,” describes SnapVX post-installation activities.

Chapter 3, “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
- VMAX3 Family Product Guide
- VMAX Family 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.

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: The PowerMax Family Product Guide, VMAX All Flash Product Guide and VMAX3 Family Product Guide provide additional information about SnapVX.
Introduction

**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
Getting Started

This chapter covers the following topics:

◆ TimeFinder SnapVX and TimeFinder/Clone Mainframe Snap Facility ...................... 20
◆ Post-installation .................................................................................................. 20
◆ Software interoperability considerations .......................................................... 20
◆ Starting ResourcePak Base ................................................................................ 21
◆ Setting up security .............................................................................................. 21
◆ Running TimeFinder/Clone Mainframe Snap Facility ......................................... 22
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.
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 48 discusses groups of statements.
CHAPTER 3
Configuration

This chapter covers the following topics:

◆ TimeFinder SnapVX configuration layers .............................................................. 24
◆ Editing the EMCSNAPO macro ............................................................................ 25
◆ EMCSNAPO site options ..................................................................................... 26
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 26 and “Editing the EMCSNAPO macro” on page 25 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
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>
<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>27</td>
<td>ACTIVATE_SUBTASK#</td>
<td>3</td>
<td>0-255</td>
<td>ACTIVATE_SUBTASK#</td>
</tr>
<tr>
<td>27</td>
<td>ALLOW_FBA_META</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>ALLOWZDP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>AUTOCREATE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>AUTOUNLINK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>BACKGRND</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>CACHESYM</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>CHKLTGT</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>28</td>
<td>CHKONLIN</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>CONDVOL</td>
<td>ALL</td>
<td>ALL</td>
<td>DUMP</td>
</tr>
<tr>
<td>29</td>
<td>CONSIST</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>COPYVOL</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>DEBUG_ERROR</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>DEBUG_SDUMP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>29</td>
<td>EMUL_TYPE</td>
<td>ALL</td>
<td>ALL</td>
<td>HARDLINK</td>
</tr>
<tr>
<td>29</td>
<td>FREEUNLK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>30</td>
<td>GROUP_DSNAME</td>
<td>n/a</td>
<td>dataset_name</td>
<td>GROUP_DSNAME</td>
</tr>
<tr>
<td>30</td>
<td>MESSAGE</td>
<td>NONE</td>
<td>DISPLAY</td>
<td>PROMPT</td>
</tr>
<tr>
<td>30</td>
<td>REFVTOC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>30</td>
<td>RELLKLTGT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>30</td>
<td>REPLACE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>30</td>
<td>RESTORE_CREATE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
Table 1 EMCSNAPO site options  (page 2 of 2)

<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>31</td>
<td>SECURE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>31</td>
<td>SECURE_QUERY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>31</td>
<td>SETLNKGT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>31</td>
<td>SNAPSHOT_LIST</td>
<td>ALL</td>
<td>snapshot_filter</td>
<td>SNAPSHOT_LIST</td>
</tr>
<tr>
<td>31</td>
<td>SNAPSHOT_NAME</td>
<td>n/a</td>
<td>snapshot_name</td>
<td>NAME</td>
</tr>
<tr>
<td>31</td>
<td>SOFTLINK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>31</td>
<td>SOFTRESTORE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>32</td>
<td>SRDFA_RETRY</td>
<td>10</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>32</td>
<td>SRPPERCENT</td>
<td>100</td>
<td>0-100</td>
<td>SRP_PERCENTAGE</td>
</tr>
<tr>
<td>32</td>
<td>SRPMSGLEVEL</td>
<td>INFO</td>
<td>INFO</td>
<td>WARN</td>
</tr>
<tr>
<td>32</td>
<td>TERM_ALL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>32</td>
<td>TIMEOUT</td>
<td>0</td>
<td>number of seconds</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>32</td>
<td>VARYOFF</td>
<td>AUTO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>32</td>
<td>VARYON</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
<tr>
<td>33</td>
<td>VCLOSE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>33</td>
<td>UNLNKAPT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>33</td>
<td>WFDEF</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

**ACTIVATE_SUBTASK#**

See “ACTIVATE_SUBTASK#(nnn)” on page 54.

**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 83.

**Syntax**

**ALLOW_FBA_META** = **YES | NO**
Configuration

ALLOWZDP

See “ZDP(Yes|No)” on page 89.

Syntax

ALLOWZDP=YES | NO

AUTOCREATE

See “AUTO_CREATE(Yes|No)” on page 91.

Syntax

AUTOCREATE=YES | NO

AUTO_UNLINK

See “AUTO_UNLink(Yes|No)” on page 55.

Syntax

AUTO_UNLINK=YES | NO

BACKGRND

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

Syntax

BACKGRND=YES | NO | NOCOPYRD | VSE

CACHESYM

See “CACHE_FULL_SYM(Yes|No)” on page 84.

Syntax

CACHESYM=YES | NO

CHKLTGT

See “Check_Link_Target_Hold(Yes|No)” on page 55.

Syntax

CHKLTGT=YES | NO | NEVER

CHKONLIN

See “CHECKONLINEpathstatus(Yes|No|NEVER)” on page 55.

Syntax

CHKONLIN=YES | NO | NEVER
**CONDVOL**

See “CONDitionVOLUMe(ALL|LaBeL|DUMP)” on page 56.

Syntax

```
CONDVOL=ALL | DUMP | LABEL
```

**CONSIST**

See “CONSISTENT(Yes|No)” on page 56.

Syntax

```
CONSIST=YES | NO
```

**COPYVOL**

See “COPYVolid(Yes|No)” on page 57.

Syntax

```
COPYVOL=YES | NO
```

**DEBUG_ERROR**

See “DEBUG(ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP)” on page 85, the ERROR option.

Syntax

```
DEBUG_ERROR=YES | NO
```

**DEBUG_SDUMP**

See “DEBUG(ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP)” on page 85, the SDUMP option.

Syntax

```
DEBUG_SDUMP=YES | NO
```

**EMUL_TYPE**

See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 58.

Syntax

```
EMUL_TYPE=ALL | HARDLINK | SNAPVX
```

**FREEUNLK**

See “FREE(Yes|No)” on page 58.

Syntax

```
FREEUNLK=YES | NO
```
GROUP_DSNAME
See “GROUP_DATaset_name(‘dataset_name’)” on page 85.
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 60.

Syntax
MESSAGE=DISPLAY | PROMPT | NONE | DETAIL

REFVTOC
See “REFVTOC(Yes|No)” on page 62.

Syntax
REFVTOC=YES | NO

RELLKTGT
See “Release_Link_target_hold(Yes|No)” on page 62.

Syntax
RELLKTGT=YES | NO

REPLACE
See “REPLace(Yes|No)” on page 64.

Syntax
REPLACE=YES | NO

RESTORE_CREATE
See “REStore_create(Yes|No)” on page 93.

Syntax
RESTORE_CREATE=YES | NO
**SECURE**

See “SECure(Yes|No)” on page 64.

Syntax

```
SECURE=YES | NO
```

**SECURE_QUERY**

See “SECURE_Query(Yes|No)” on page 65.

Syntax

```
SECURE_QUERY=YES | NO
```

**SETLKTGT**

See “Set_Link_TarGeT_Hold(Yes|No)” on page 65.

Syntax

```
SETLKTGT=YES | NO
```

**SNAPSHOT_LIST**

See “SNAPSHOT_LIST(snapshot_filter)” on page 65.

Syntax

```
SNAPSHOT_LIST=snapshot_filter
```

**SNAPSHOT_NAME**

See “NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])” on page 61.

Syntax

```
SNAPSHOT_NAME=snapshot_name
```

**SOFTLINK**

See “SOFTlink(Yes|No)” on page 87.

Syntax

```
SOFTLINK=YES | NO
```

**SOFTRESTORE**

See “SOFTRestore(Yes|No)” on page 66.

Syntax

```
SOFTRESTORE=YES | NO
```
Configuration

**SRDFA_RETRY**

See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 68.

**Syntax**

SRDFA_RETRY=YES | NO | nn

*Note: The default value is 10.*

**SRPPERCT**

See “SRP_PERCENT(srp-percentage,{{INFO|WARN|ERROR}|SKIP})” on page 88.

**Syntax**

SRPPERCT=srp-percentage

*Note: The default value is 100.*

**SRPMSGLVL**

See “SRP_PERCENT(srp-percentage,{{INFO|WARN|ERROR}|SKIP})” on page 88.

**Syntax**

SRPMSGLVL={INFO | WARN | ERROR | SKIP}

**TERM_ALL**

See “TERminate_ALL(Yes|No)” on page 110.

**Syntax**

TERM_ALL=YES | NO

**TIMEOUT**

See “TIMEOUT(nnn)” on page 69.

**Syntax**

TIMEOUT=number_of_seconds

*Note: The default value is 0.*

**VARYOFF**

See “VARY_Offline(AUTO|NEVER)” on page 70.

**Syntax**

VARYOFF=Auto | NEVER

**VARYON**

See “VARY_Online(AUTO|Yes|No)” on page 70.
Syntax

VARYON= \texttt{AUTO} \vert \texttt{NEVER}

\textbf{VCLOSE}

See “\texttt{VCLOSE(Yes|No)}” on page 71.

Syntax

\texttt{VCLOSE=} \texttt{YES} \vert \texttt{NO}

\textbf{UNLNKAFT}

See “\texttt{UNLINK\_After\_copy(Yes|No)}” on page 93.

Syntax

\texttt{UNLNKAFT=} \texttt{YES} \vert \texttt{NO}

\textbf{WFDEF}

See “\texttt{WAIT\_FOR\_Definition(Yes|No)}” on page 71.

Syntax

\texttt{WFDEF=} \texttt{YES} \vert \texttt{NO}
CHAPTER 4
Operations

This chapter covers the following topics:

◆ Softlinked and hardlinked snapshots ................................................................. 36
◆ Snapshot life cycle ............................................................................................ 36
◆ Secure snaps .................................................................................................... 40
◆ Summary of operations .................................................................................... 42
◆ Example of operations ..................................................................................... 47
◆ Defining groups of statements ......................................................................... 48
Operations

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 specific source and target device.

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

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>
<thead>
<tr>
<th>SnapVX commands</th>
<th>Snapshot states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NONE</td>
</tr>
<tr>
<td>CREATE</td>
<td>×</td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>×</td>
</tr>
<tr>
<td>RENAME</td>
<td>×</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>

**Table 2** Allowable SnapVX commands per snapshot state
SnapVX group status transitions

Table 3 shows SnapVX group status transitions.

Note: “Defining groups of statements” on page 48 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.
Link target holds

You can set a hold on a link target device to protect from being overlaid or unlinked until the hold is released.

SnapVX sets, checks, and releases holds on link targets as follows:

- The LINK command sets a hold on the target device being linked if the LINK/GLOBAL SET_LINK_TARGET_HOLD parameter or the &SETLKTGT site option is set to Yes.
- The UNLINK, FREE, TERMINATE AUTO_UNLINK(YES) commands check for holds before unlinking if the command-specific or GLOBAL CHECK_LINK_TARGET_HOLD parameter or the &CHKLTGT site option is set to YES. When a hold exists, unlinking/freeing fails with a corresponding error message.
- The UNLINK, FREE, TERMINATE AUTO_UNLINK(YES) commands release the existing holds on linked target devices if the UNLINK/GLOBAL RELEASE_LINK_TARGET_HOLD parameter or the &RELLKTGT site option is set to YES.

In addition, you can use the CONFIG command with the SET_LINK_TARGET_HOLD(YES) or RELEASE_LINK_TARGET_HOLD(YES) parameters to set or release link target holds manually.

To see if a hold is set on the link target, issue the QUERY SNAPSHOT command with the MULTI_LINE_query(YES) parameter. If a hold exists, the TARGET HOLD field has the value of YES:

```
TARGET HOLD: YES
```

**Note:** N/A in the TARGET HOLD field indicates an unlinked snapshot. If the hold query failed for any reason, question marks (??) are displayed.

In addition, you can issue the QUERY SNAPSHOT command with the COMPACT_query(YES) parameter and check the LTH column in the output.

You can also view holds with the REC,QUERYDEVICELOCK command of SCF (Symmetrix Control Facility), as described in the *ResourcePak Base for z/OS Product Guide*. 
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:FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
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 traditional snapshot can be converted to a secure snapshot, but a secure snapshot may not be converted to a traditional snapshot. All SnapVX operations and rules for traditional 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
Secure snaps

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 traditional 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>
<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>
<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>Include status summary for each snapshot</td>
<td>• GLOBAL command, COMPACT_query parameter</td>
</tr>
</tbody>
</table>
### Viewing groups

Table 4 lists operations for 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</td>
<td></td>
</tr>
<tr>
<td>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>
<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>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable/disable softlinking</td>
<td>● GLOBAL command, SOFTlink parameter</td>
</tr>
<tr>
<td></td>
<td>● &amp;SOFTLINK site option</td>
</tr>
<tr>
<td>Enable/disable soft restore</td>
<td>● GLOBAL command, SOFTRestore parameter</td>
</tr>
<tr>
<td></td>
<td>● UNLINK command, SOFTRestore parameter</td>
</tr>
<tr>
<td></td>
<td>● &amp;SOFTRESTORE site option</td>
</tr>
<tr>
<td>Select background copy mode</td>
<td>● CONFIG command, MODE parameter</td>
</tr>
<tr>
<td></td>
<td>● GLOBAL command, MODE parameter</td>
</tr>
<tr>
<td></td>
<td>● LINK command, MODE parameter</td>
</tr>
<tr>
<td>Link and overwrite data on existing target device</td>
<td>● GLOBAL command, REPLACE parameter</td>
</tr>
<tr>
<td></td>
<td>● LINK command, REPLACE parameter</td>
</tr>
<tr>
<td></td>
<td>● &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>
## 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>
<tbody>
<tr>
<td>Use ECA</td>
<td><strong>ACTIVATE</strong> command, CONSISTENT parameter</td>
</tr>
<tr>
<td></td>
<td><strong>GLOBAL</strong> command, CONSISTENT parameter</td>
</tr>
<tr>
<td></td>
<td>&amp;CONSIST site option</td>
</tr>
<tr>
<td>Set up consistency when using ECA</td>
<td><strong>ACTIVATE</strong> command, SRDFA_CONSISTENT_RETRY parameter</td>
</tr>
<tr>
<td></td>
<td><strong>GLOBAL</strong> command, SRDFA_CONSISTENT_RETRY parameter &amp;SRDFA_RETRY site option</td>
</tr>
<tr>
<td>Set ECA timeout</td>
<td><strong>ACTIVATE</strong> command, TIMEOUT parameter</td>
</tr>
<tr>
<td></td>
<td><strong>GLOBAL</strong> command, TIMEOUT parameter</td>
</tr>
<tr>
<td></td>
<td>&amp;TIMEOUT site option</td>
</tr>
</tbody>
</table>
Managing link target holds

Table 11 lists operations for managing link target holds.

Table 10  Controlling device status

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set hold on link target during linking</td>
<td>● GLOBAL command, Set_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● CONFIG command, Set_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● LINK command, Set_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● &amp;SETLKTGT site option</td>
</tr>
<tr>
<td>Check if a hold exists on a target</td>
<td>● GLOBAL command, Check_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● UNLINK command, Check_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● TERMINATE command with AUTO_UNLINK option, Check_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● FREE command, Check_Link_TarGeT_Hold parameter</td>
</tr>
<tr>
<td></td>
<td>● &amp;CHKLTGT site option</td>
</tr>
<tr>
<td>Release link target hold</td>
<td>● GLOBAL command, Release_Link_target_hold parameter</td>
</tr>
<tr>
<td></td>
<td>● CONFIG command, Release_Link_target_hold parameter</td>
</tr>
<tr>
<td></td>
<td>● UNLINK command, Release_Link_target_hold parameter</td>
</tr>
<tr>
<td></td>
<td>● TERMINATE command, AUTO_UNLINK option, Release_Link_target_hold parameter</td>
</tr>
<tr>
<td></td>
<td>● FREE command, Release_Link_target_hold parameter</td>
</tr>
<tr>
<td></td>
<td>● &amp;RELLKTGT site option</td>
</tr>
</tbody>
</table>

Controlling device status

Table 11 lists operations for controlling device status.

Table 11  Controlling device status

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

**Conditioning target volume**

*Table 12* 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 13* lists operations to make miscellaneous SnapVX settings.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
</table>
| Set up subtasking | • *ACTIVATE* command, *ACTIVATE_SUBTASK#* parameter  
• *GLOBAL* command, *ACTIVATE_SUBTASK#* parameter  
• &ACTIVATE_SUBTASK# site option |
| Allow or prohibit command execution against FBA meta devices | • *GLOBAL* command, *ALLOW_FBA_META* parameter  
• &ALLOW_FBA_META site option |
| Improve job completion times for jobs with >500 devices | • *GLOBAL* command, *CACHE_FULL_SYM* parameter  
• CACHESYM site option |
| Set up messaging | • *ACTIVATE* command, *MESSages* parameter  
• *GLOBAL* command, *MESSages* parameter  
• &MESSAGE site options  
• *GLOBAL* command, *MAXRC* parameter |
| Turn on/off debugging | • *GLOBAL* command, *DEBUG* parameter  
• &DEBUG_SDUMP site option  
• &DEBUG_ERROR site option |
| View global settings | QUERY *GLOBAL* command |
| View site options | QUERY *GROUP* command |
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, as described in “GROUP(grpname[,grpname,...])” on page 72.
- Use the DD statement //EMCGROUP DD to point to the group dataset.
- Define GROUP_DSNAME in the site options table EMCSNAPO, as described in “GROUP_DSNAME” on page 30.

Note: You can concatenate multiple datasets together with EMCGROUP.

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.

```
//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
```
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:

◆ Conventions ......................................................................................................... 54
◆ SnapVX commands .............................................................................................. 54
◆ Grouping commands ........................................................................................... 114
Conventions

- Keywords appear in uppercase (for example, CREATE). They must be spelled exactly as shown.
- For easy reference, command keywords are supplemented by lowercase letters to form a meaningful word (for example, SOurce). When typing a command, use only CAPITALIZED characters of any keyword.
- Variables appear in lowercase and italics (for example, volser). They represent user-supplied names or values in the syntax.
- Square brackets [ ] indicate an optional entry (for example, cuu[-cuu]).
- The vertical bar | indicates alternative argument values (for example, SOurce|UNIT).
- Curly brackets {} are used to group a series of alternative values that can be used with a single keyword, for example: (SOurce|UNIT|Volume).
- Aside from the square and curly brackets and the vertical bar characters, you must type all other characters that are shown in the syntax statements.
- Default values are indicated by an underline. For example, if the parameter has the following option, (Yes|No), the underlined No indicates the default value.
- 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 sysecalls 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.

**Check_Link_Target_Hold(Yes|No)**

When set to Yes, SnapVX checks for existing link target holds before further processing. The default value is Yes.

Note: “Link target holds” on page 38 discusses link target holds.

Aliases of CHECK_LINK_TARGET_HOLD include CLTH and CHK_LINK_TGT_HOLD.

The CHECK_LINK_TARGET_HOLD parameter has a matching site option, &CHKLTGT.

**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.

**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 nnnn 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({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(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(dev#)

- 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(lowdev#-highdev#)

  - Specify the lowest device in the range and the highest device in the range separated by a colon:

    DEVICE(lowdev#:highdev#)

  - 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.
DEVICE(dev#(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.

DISPLAY_CUU(Yes | 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.

EMUL_TYPE(ALL | HARDLINK | 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.

EXPIration(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 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-]xxxxx|name)] |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
CONTROLLER([xxxxxxx-]xxxxx|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-]xxxxx|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 EQCA921I. 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.
- **PROMpnt** 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.

**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.

READY (Yes | No)
Determine 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)
Determine 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.

Release_Link_target_hold (Yes | No)
When set to Yes, SnapVX releases existing holds on link targets upon unlinking. The default value is No.

Note: “Link target holds” on page 38 discusses link target holds.

RELEASE_LINK_TARGET_HOLD and SET_LINK_TARGET_HOLD are mutually exclusive.
Aliases of RELEASE_LINK_TARGET_HOLD include RELEASE_LINK_TGT_HOLD, REL_LINK_TARGET_HOLD, RLTTH, RELEASE_LINK, REL_LINK.

The RELEASE_LINK_TARGET_HOLD parameter has a matching site option, &RELLKTGT.

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 (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.

**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.

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

**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 40 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 40 describes secure snapshots.

Set_Link_TarGeT_Hold(Yes | No)
When set to Yes, the LINK or CONFIG command sets a hold on the target device. The default value is No.

Note: “Link target holds” on page 38 discusses link target holds.

SET_LINK_TARGET_HOLD and RELEASE_LINK_TARGET_HOLD are mutually exclusive.

Aliases of SET_LINK_TARGET_HOLD include SLTH and SET_LINK_TGT_HOLD.

The SET_LINK_TARGET_HOLD parameter has a matching site option, &SETLKTGT.

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

Valid values for snapshot_filter include:

<table>
<thead>
<tr>
<th>snapshot_filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVATED</td>
<td>List activated snapshots (both softlinked and hardlinked).</td>
</tr>
<tr>
<td>ALL</td>
<td>(Default) List all links and snapshots.</td>
</tr>
<tr>
<td>COPY</td>
<td>List snapshots with COPY mode set between the source and the target device.</td>
</tr>
<tr>
<td>DEFINED</td>
<td>List snapshots with fully defined target devices.</td>
</tr>
<tr>
<td>LINKED</td>
<td>List links (target device is not X'FFFFFFFF').</td>
</tr>
<tr>
<td>NOCOPY</td>
<td>List snapshots with NOCOPY mode set between the source and the target device. NOCOPY is an alias of NOCOPY.</td>
</tr>
<tr>
<td>NOT_ACTIVATED</td>
<td>List snapshots that have not been activated. NACTIVATED is an alias of NOT_ACTIVATED.</td>
</tr>
<tr>
<td>NOT_DEFINED</td>
<td>List snapshots with undefined target devices. NDEFINED is an alias of NOT_DEFINED.</td>
</tr>
<tr>
<td>NOT_LINKED</td>
<td>List not linked snapshots (target device is X'FFFFFFFF' and there is no link with this snapshot). NLINKED is an alias of NOT_LINKED.</td>
</tr>
</tbody>
</table>

Note: “Target volume track definition” on page 37 discusses target track definition.
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 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({VOLume(volser)|UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count))}|SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#:dev#(count)})})

Specifies the source device.

You must specify an input source device by using the INDDname or SOUrce parameters. Either may be used, but not both.

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 also specify a range of unit addresses. You can write a range in three ways:

- Specify the lowest address in the range and the highest address in the range separated by a dash:
  
  UNIT(lowcuu-highcuu)
- Specify the lowest address in the range and the highest address in the range separated by a colon:

```
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. 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.)

```
UNIT(cuu(count))
```

**SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})**

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

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.

When SYMDV# is used, the LOCAL or REMOTE parameter must be specified.

- LOCAL—This is a device in a local storage system.
- REMOTE—This is a device in a remote storage system if the REMOTE parameter and RAGROUP subparameter is used to identify the storage system.

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

You can specify a single PowerMax/VMAX device number:

```
SYMDV#(dev#)
```

You can also specify a range of device numbers.

You can specify a device range in three ways:

- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a dash:

```
SYMDV#(lowdev#:highdev#)
```

- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a colon:

```
SYMDV#(lowdev#:highdev#)
```

- 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. 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.)

```
SYMDV#(dev#(count))
```
SRDFA_CONSISTENT_RETRY (Yes | No | nn)

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.

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#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})))

Specifies a target device.

You must specify a target device by using the OUTDname or TARget parameters. Either may be used, but not both.

When specifying a range of devices, the total number of devices in the range cannot exceed 1024.

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 also specify a range of unit addresses. You can write a range in three ways:

- Specify the lowest address in the range and the highest address in the range separated by a dash:
  
  UNIT(lowcuu-highcuu)

- Specify the lowest address in the range and the highest address in the range separated by a colon:
  
  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. 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.)
  
  UNIT(cuu(count))
SYMDV#({(dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count))})

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

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

When SYMDV# is used, the LOCAL or REMOTE parameter must be specified.
- LOCAL—This is a device in a local storage system.
- REMOTE—This is a device in a remote storage system if the REMOTE parameter and RAGROUP subparameter is used to identify the storage system.

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

You can specify a single PowerMax/VMAX device number:

SYMDV#(dev#)

You can also specify a range of device numbers.

You can specify a device range in three ways:
- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a dash:

SYMDV#(lowdev#-highdev#)

- Specify the lowest numbered device in the range and the highest numbered device in the range separated by a colon:

SYMDV#(lowdev#:highdev#)

- 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. 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.)

SYMDV#(dev#(count))

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 TF/Consistency Group Licensed Feature Code (parameter CONSISTENT) installed. The *Mainframe Enablers Installation and Customization Guide* provides more information.
**nnn is a value from zero (0) to 127 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. The default nnn value is 0 (zero), meaning that the duration of the ECA assist is determined by the time needed to initiate a consistent snap.**

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

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)]
[SIRDFA_CONSISTENT_RETRY(Yes|No|nn)]
[TIMEOUT(nnn)]

Optional parameters

ACTIVATE_SUBTASK#(nnn)

See “ACTIVATE_SUBTASK#(nnn)” on page 54.

CONSISTENT(Yes|No)

See “CONSISTENT(Yes|No)” on page 56.

EXPIration(days)

See “EXPIration(days)” on page 58.

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

See “GROUP(grpname[,grpname,...])” on page 59.

MESsages(DISplay|PROmpt|NONE|DETAIL)

See “MESsages(DISplay|PROmpt|NONE|DETAIL)” on page 60.
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 64.

⚠️ 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 | nn)

See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 68.

TIMEOUT (nnn)

See “TIMEOUT(nnn)” on page 69.

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

(NAME(snapshot_name)
TaRGet({VOLume(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count))| |
SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})}))
[optional_parameters]
)
```

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)])
  [Release_Link_target_hold(Yes|No)]
- [MODE(COPY|NOCOPY|NOCOPYRD)]
- [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)]
- [Set_Link_Target_Hold(Yes|No)]
- [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.
Command Reference

Required parameters

NAME(snapshot_name)

The name of the snapshot to be configured.

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

TaRGet({VOLume(volser) |
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | 
SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})

See “TaRGet({VOLUME(volser) |
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) |
SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})” on page 68.

For the CONFIG command, the total number of devices in the range cannot exceed 256.

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 58.

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

See “GROUP(grpname[,grpname,...])” on page 59.
LOCAL({
UNIT(cuu) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
VOLUME(volser) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
DDNAME(ddname) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
CONTROLLER([xxxxxxxx--]xxxxx|name))
)

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
VOLUME(volser) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
DDNAME(ddname) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
CONTROLLER([xxxxxxxx--]xxxxx|name))” on page 59.

MODE(COPY|NOCOPY|NOCOPYRD)

See “MODE(COPY|NOCOPY|NOCOPYRD)” on page 61.

For the duration of the current CONFIG command, MODE overrides any value set by the GLOBAL command MODE parameter.

Release_Link_target_hold(Yes|No)

See “Release_Link_target_hold(Yes|No)” on page 62.

For the duration of the current CONFIG command, Release_Link_target_hold overrides any value set by the GLOBAL command Release_Link_target_hold parameter or the &RELLKTGT site option.

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

See “REMOTE (RAGROUP(nn.nn.nn.nn) {}
UNIT(cuu) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
VOLUME(volser) [CONTROLLER([xxxxxxxx--]xxxxx|name)] |
DDNAME(ddname) [CONTROLLER([xxxxxxxx--]xxxxx|name)])” on page 63.

SECure(Yes)

See “SECure(Yes|No)” on page 64.

⚠️ 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.

Set_Link_TarGeT_Hold(Yes|No)

See “Set_Link_TarGeT_Hold(Yes|No)” on page 65.

For the duration of the current CONFIG command, Set_Link_TarGeT_Hold overrides any value set by the GLOBAL command Set_Link_TarGeT_Hold parameter or the &SETLKTGT site option.

SNAPSHOTID(snapshot_id)

The ID of the snapshot to be processed.
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, VMAX All Flash Product Guide or VMAX3 Family 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({cuu|lowcuu-highcuu|lowcuu:higheu|cuu(count)}|
SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)}))
[optional_parameters]
  )

Where optional_parameters are:

  [GROUP(grpname[, grpname,...])]
  [LOCAL({
    UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxx|name)]|
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]|
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxx|name)]|
    CONTROLLER([xxxxxxx-]xxxxx|name)
  })]
  [REMOTE (RAGROUP(nn.nn.nn.nn)
    (UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxx|name)]|
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]|
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxx|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 61.

**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#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count))}))
```

See “SOUrce({VOLume(volser) |
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)) |
SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count))}))” on page 66.

**Optional parameters**

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

See “GROUP(grpname[, grpname,...])” on page 59.

```
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 59.

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

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) |
[VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]})” on page 63.

**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 FREEed.

**Syntax**

FREE

(  
TARGET(UNIT(cuu) | SYMDV#(dev#))
[optional_parameters]
)

Where **optional_parameters** are:

[Check_Link_TarGeT_Hold(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) )})

[Release_Link_target_hold(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)])  
[STOP(Yes|No)]


Required parameters

**TARGet (UNIT (cuu) | SYMDV# (dev#))**

See “TARGet({VOLUME(volser)|UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count))})” on page 68.

The target device must be offline before FREEing.

Optional parameters

**Check_Link_TarGeT_Hold(Yes|No)**

See “Check_Link_TarGeT_Hold(Yes|No)” on page 55.

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

See “GROUP(grpname[, grpname,...])” on page 59.

**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 59.

**Release_Link_target_hold(Yes|No)**

See “Release_Link_target_hold(Yes|No)” on page 62.

**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 63.

**STOP(Yes|No)**

Kills the FREE background process in the storage system.

Example

```plaintext
FREE {
  TGT(SYMDV#(0090-093)) -
  LOCAL(UNIT(6200)) -
}
```
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)]
[AUTO_UNLink(Yes|No)]
[BACKGROUND_COPY(Yes|No|NOCOPYRD|VSE)]
[CACHE_FULL_SYM(Yes|No)]
[Check_Link_Target_Hold(Yes|No)]
[CHECKONLINEpathstatus(Yes|No|NEVER)]
[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)]
[Release_Link_target_hold(Yes|No)]
[SECure(Yes|No)]
[SECURE_Query(Yes|No)]
[Set_Link_TarGeT_Hold(Yes|No)]
[SNAPSHOT_LIST(snapshot_filter)]
[SOFTlink(Yes|No)]
[SOFTRestore(Yes|No)]
[SRDFA_CONSISTENT_RETRY(Yes|No|nn)]
[SRP_PERCENT(srp-percentage,{{INFO|WARN|ERROR}|SKIP})]
[TIMEOUT(nn)]
[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 54.

**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 ALLFMETA are aliases for ALLOW_FBA_META.

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

**AUTO_UNLink(Yes|No)**

See “AUTO_UNLink(Yes|No)” on page 55.

**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 or defaulted 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.

**Check_Link_TarGeT_Hold(Yes | No)**

See “Check_Link_TarGeT_Hold(Yes|No)” on page 55.

**CHECKONLINEpathstatus(Yes | No | NEVER)**

See “CHECKONLINEpathstatus(Yes|No|NEVER)” on page 55.

**COMPACT_query(Yes | No)**

See “COMPACT_query(Yes|No)” on page 55.

**CONDitionVOLUME(ALL | LaBeL | DUMP)**

See “CONDitionVOLUME(ALL|LaBeL|DUMP)” on page 56.

**CONSISTENT(Yes | No)**

See “CONSISTENT(Yes|No)” on page 56.

**COPYVolid(Yes | No)**

See “COPYVolid(Yes|No)” on page 57.
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.

You can control DEBUG and TRACE default actions on specific TimeFinder commands through the DEBUG(ON|OFF) and TRACE(ON|OFF) parameters.

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

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

See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 58.

**Global parameter effects**

This parameter sets a global value for the following command:

- QUERY SNAPSHOT

**FREE (Yes | No)**

See “FREE(Yes|No)” on page 58.

**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 48 discusses groups of statements.

If this parameter omitted, TimeFinder uses a DDNAME of EMCGROUP as the “working” group dataset.

**GROUP_DSNName** is an alias of GROUP_DATASET_name. The GROUP_DATASET_name parameter has a matching site option, &GROUP_DSNNAME.

**LIST ([NO]STATEMENTS | [NO]HISTORY)**

See “LIST([NO]STATEMENTS|[NO]HISTORY)” on page 59.
MAXRC\(\text{return\_code\_value}\)

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, 'E' is 8 and '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 are processed.

\(\text{return\_code\_value}\)

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 \(\text{return\_code\_value}\) 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\(\text{DISplay|PROmpt|NONE|DETAIL}\)

See “MESSages\(\text{DISplay|PROmpt|NONE|DETAIL}\)” on page 60.

MODE\(\text{COPY|NOCOPY|NOCOPYRD}\)

See “MODE\(\text{COPY|NOCOPY|NOCOPYRD}\)” on page 61.

MULTI\_LINE\_query\(\text{Yes|No}\)

See “MULTI\_LINE\_query\(\text{Yes|No}\)” on page 61.

NAME\(\text{snapshot\_name[%date[4|6|8]%][%time[4|6]%]}\)

See “NAME\(\text{snapshot\_name[%date[4|6|8]%][%time[4|6]%]}\)” on page 61.

REFVTOC\(\text{Yes|No}\)

See “REFVTOC\(\text{Yes|No}\)” on page 62.

Release\_Link\_target\_hold\(\text{Yes|No}\)

See “Release\_Link\_target\_hold\(\text{Yes|No}\)” on page 62.

REPLace\(\text{Yes|No}\)

See “REPLace\(\text{Yes|No}\)” on page 64.

SECure\(\text{Yes|No}\)

See “SECure\(\text{Yes|No}\)” on page 64.
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 65.

Set_Link_TarGeT_Hold(Yes | No)
See “Set_Link_TarGeT_Hold(Yes|No)” on page 65.

SNAPSHOT_LIST(snapshot_filter)
See “SNAPSHOT_LIST(snapshot_filter)” on page 65.

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 61 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 66.

SRDFA_CONSISTENT_RETRY(Yes|No|nn)
See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 68.

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 39 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
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.

This parameter has a corresponding site option, &SRPMSGLVL.

SKIP

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.

This parameter has a corresponding site option, &SRPMSGLVL.

TIMEOUT(nnn)
See “TIMEOUT(nnn)” on page 69.
UNLINK_After_copy(Yes|No)
See “UNLINK_After_copy(Yes|No)” on page 70.

VARY_OFFline(AUTO|NEVER)
See “VARY_OFFline(AUTO|NEVER)” on page 70.

VARY_ONline(AUTO|Yes|No)
See “VARY_ONline(AUTO|Yes|No)” on page 70.

VCLOSE(Yes|No)
See “VCLOSE(Yes|No)” on page 71.

WAIT_FOR_Definition(Yes|No)
See “WAIT_FOR_Definition(Yes|No)” on page 71.

ZDP(Yes|No)
Any snapshot that is created through the zDP product will not be actionable through the snap by default. To bypass this restriction, set the ZDP(YES) parameter which allows actions that change the snapshot state (such as LINK, UNLINK, TERMINATE, CONFIG).

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

```
LINK

(NAME(snapshot_name)
 SOURCE((VOLUME(volser)|UNIT(({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|SYMDV#(({dev#|lowdev#:highdev#|lowdev#|highdev#}|dev#(count))}))
 TARGET((VOLUME(volser)|UNIT(({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|SYMDV#(({dev#|lowdev#:highdev#|lowdev#|highdev#}|dev#(count))}))
 [optional_parameters]
 )

Where optional_parameters are:

[AUTO_CREATE(Yes|No)]
[CHECKONLINE(pathstatus(Yes|No|NEVER))]
[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|name)]
 })]
[MODE(COPY|NOCOPY|NOCOPYRD)]
[NEWVOLID(volser)]
[READY(Yes|No)]
```
SnapVX commands

[REFVTOC(Yes|No)]
[REMOTE (RAGROUP(nn.nn.nn.nn)
(UNIT(cuu) [CONTROLLER([xxxxxxxx-]xxxxx|name)]|
VOLUME (volser) [CONTROLLER([xxxxxxxx-]xxxxx|name)]|
DDNAME (ddname) [CONTROLLER([xxxxxxxx-]xxxxx|name)])]
[REPLACE(Yes|No)]
[RESTORE_CREATE(Yes|No)]
[RESTORE_CREAtE_NAME(snapshot_name)]
[SET_LINK_TarGeT_Hold(Yes|No)]
[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#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})})
   See “SOURce({VOLUMe(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})})” on page 66.

TaRGet({VOLUMe(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})})
   See “TaRGet({VOLUMe(volser)|
UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|
SYMDV#({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)})})” on page 68.

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 55.
CONDitionVOLUME(ALL|LaBeL|DUMP)

See “CONDitionVOLUME(ALL|LaBeL|DUMP)” on page 56.

COPYVolid(Yes|No)

See “COPYVolid(Yes|No)” on page 57.

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

See “GROUP(grpname[, grpname, ...])” on page 59.

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-]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 59.

MODE(COPY|NOCOPY|NOCOPYRD)

See “MODE(COPY|NOCOPY|NOCOPYRD)” on page 61.

NEWVOLID(volser)

Creates a new volume ID using the specified volser name.

READY(Yes|No)

See “READY(Yes|No)” on page 62.

REFVTOC(Yes|No)

See “REFVTOC(Yes|No)” on page 62.

For the duration of the current snapshot command, the value of REFWPT TOC overrides any value set by the GLOBAL command REFWPT TOC 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 63.

REPLace(Yes|No)

See “REPLace(Yes|No)” on page 64.
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.

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.

Set_Link_Target_Hold(Yes|No)

See “Set_Link_Target_Hold(Yes|No)” on page 65.

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 70.

VARY_OFFline(AUTO|NEVER)

See “VARY_OFFline(AUTO|NEVER)” on page 70.

VARY_ONline(AUTO|Yes|No)

See “VARY_ONline(AUTO|Yes|No)” on page 70.

VCLOSE(Yes|No)

See “VCLOSE(Yes|No)” on page 71.

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

QUERY FREE
(
VOLume(volser) | UNIT(cuu)
[optional_parameters]
)

Where optional_parameters are:

[CcUU(cuu|low-high|low:high|ccuu(count))]
[DEVice({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)|ALL})]
[DISPLAY_CUU(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)])]

Required parameters

VOLume(volser) | UNIT(cuu)

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.

DEVice({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)|ALL})

See “DEVice({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)|ALL})” on page 57.

DISPLAY_CUU(Yes|No)
See “DISPLAY_CUU(Yes|No)” on page 58.
GROUP(grpname[, grpname,...])

See “GROUP(grpname[, grpname,...])” on page 59.

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)]” on page 59.

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-]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 63.

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

...  
ESNPW20I --- EMCSNAPO --- VER 8.3.0 --- SIZE 633 --- DATE/TIME 06/06/17 19.48 ---  
ESNW21I SITE SETTING GLOBAL OVERRIDE  
ESNW22I ACTIVATE SCF GATEKEEPER Y -N/A-  
ESNW22I ACTIVATE_SUBTASK# 03 03  
ESNW22I ADMINISTRATOR N N  
ESNW22I ALLOCATE_UNUSED_SPACE Y Y  
ESNW22I ALLOCATION_SEQUENCE D D  
ESNW22I ALLOCATION_UNITNAME SYSALLDA -N/A-  
ESNW22I ALLOW CANCEL LOCKED Y -N/A-  
ESNW22I ALLOW_FBA_META N N  
ESNW22I ALLOW_SYMDV# Y -N/A-  
ESNW22I ALLOW_LIGHTNING_DEVICE N N  
ESNW22I AUTOMATIC_Activate Y Y  
ESNW22I AUTO_BIND_TDEV N N  
ESNW22I AUTOMATIC_CLEANUP Y Y  
ESNW22I AUTOMATIC_CLEANUP_R2 Y Y  
ESNW22I 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 26. 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 59.

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
ESNPR23I FREE - 0004 2018-01-22 / 16:21:38 INITIAL -> INITIAL K143
ESNPR25I STATEMENTS:
ESNPR26I + *
ESNPR26I + LINK ( -
ESNPR26I + NAME(SVX728_SNAPSHOT1) -
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

“Example 2: Multiline query” on page 102 illustrates the multiline query output.

**Note:** See “Example 2: Multiline query” on page 102 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)
- Link target information (fully defined or not, target hold set or not)
- 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 104 for sample output and explanation of fields.
Syntax

QUERY_SNAPSHOT

(   
VOLUME(volser) | UNIT(cuu)
[optional_parameters]
)

Where **optional_parameters** are:

[CcUU(cuu|low-high|low:high|ccuu(count))]

[COMPACT_query(Yes|No)]

[DEVICE({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)|ALL})]

[DISPLAY_CUU(Yes|No)]

[EMUL_TYPE(ALL|HARDLINK|SNAPVX)]

[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)]
})]

[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)]

[SNAPSHOT_LIST(snapshot_filter)]

[SORT(Yes|No)]

**Note:** You can use the LOCAL, REMOTE, and CONTROLLER parameters without a SYMDV# parameter being present.

**Required parameters**

VOLUME(volser) | UNIT(cuu)

VOLUME(volser) specifies the volser of the volume to be queried.

UNIT(cuu) specifies the unit address of the volume to be queried. When followed by the DEVICE parameter, UNIT(cuu) specifies a 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 55.

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({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)|ALL})
See “DEVice({dev#|lowdev#-highdev#|lowdev#:highdev#|dev#(count)|ALL})” on
page 57.

DISPLAY_CUU(Yes|No)
See “DISPLAY_CUU(Yes|No)” on page 58.

EMUL_TYPE(ALL|HARDLINK|SNAPVX)
See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 58.

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.

GROUP(grpname[, grpname,...])
See “GROUP(grpname[, grpname,...])” on page 59.

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 59.

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 61.

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 63.

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.

SORT (Yes | No)
When set to YES, sorts snapshots by date. NO disables sorting by date.

Note: Sorting can dramatically increase the query time.

SECURE_Query (Yes | No)
See “SECURE_Query(Yes|No)” on page 65.

SNAPSHOT_LIST(snapshot_filter)
See “SNAPSHOT_LIST(snapshot_filter)” on page 65.

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.

Example

Example 1: QUERY_SNAPSHOT (DISPLAY_CUU(NO)
UNIT(8D00) -
DEV(436A-436B) -
DISPLAY_CUU(NO) -
)

The output is similar to the following:

TOTAL NUMBER OF QUERY RESULTS: 00000002

<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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

SRC
The PowerMax/VMAX device number of the source device.
CUU  The CUU of the source device.

**Note:** With DISPLAY_CUU(NO), the value is displayed as N/A.

VOLSER  The volser of the source device.

TGT  The PowerMax/VMAX device number of the target device.

CUU  The CUU of the target device.

**Note:** With DISPLAY_CUU(NO), the value is displayed as N/A.

VOLSER  The volser of the target device.

NAME  The snapshot name on the source device.

TIME STAMP  The approximate time when the snapshot was created.

STATUS  The current status of the snapshot.

- SOF indicates a softlinked snapshot.
- HAR indicates a hardlinked snapshot.

ACT  Indicates whether the snapshot has been activated. Possible values: Y/N.

**Example 2:**

```sql
QUERY SNAPSHOT -
  (  
    UNIT(cuu) -
    DEVICE(dev#) -
    MULTI_LINE_QUERY(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 TGT</th>
<th>CUU</th>
<th>VOLSER NAME</th>
<th>TIME STAMP</th>
<th>YYDDDD/HH:MM:SS</th>
<th>STATUS</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>000010E(N/A )</td>
<td>FFFFFFFF(N/A )</td>
<td>BJ</td>
<td>17110/13:13:40</td>
<td>CREATE SOF</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000010E(N/A )</td>
<td>0000015E(N/A )</td>
<td>BJ</td>
<td>17111/15:43:52</td>
<td>LINKED</td>
<td>Y</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.

- **TRACKS TO BE COPIED**
  - The number of tracks that remain to be copied to the linked target device.

- **TIME TO LIVE**
  - The number of half seconds until a snapshot will be automatically terminated. For expired snapshots, ‘EXPIRED’ is displayed.

- **TIME SINCE CREATION**
  - The time that has elapsed since snapshot creation.

- **SNAPSHOT ID**
  - The ID given to a snapshot at creation time. This ID is only unique for each source device, and are reused after snapshots are terminated.

- **SECURE SNAPSHOT**
  - Indicates whether the snapshot is secure or not.

  **Note:** “Secure snaps” on page 40 describes secure snapshots.

- **COPY MODE**
  - Indicates the current copy mode (COPY or NOCOPY). For unlinked snapshots, N/A is displayed.

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

- **TARGET HOLD**
  - Indicates whether a hold exists (YES) or not (NO) on the link target.

  N/A is displayed for snapshots that do not have a linked target. If there was an error obtaining the lock information, his field shows ???.

  **Note:** “Link target holds” on page 38 discusses link target holds.

- **TOTAL UNIQUE TRACKS IN REQUEST**
  - The number of tracks that are associated with all snapshots in the current query.
TOTAL TRACKS TO BE COPIED IN REQUEST

The number of tracks that remain to be copied to the linked target devices for all snapshots in the current query.

TOTAL SRC MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST

The number of tracks that have been modified on all snapshot source devices in the current query since the ACTIVATE was issued.

TOTAL TGT MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST

The number of tracks that have been modified on all snapshot linked target devices since the LINK actions.

Example 3: Compact query

```plaintext
QUERY SNAPSHOT
 (UNIT(cuu)
  DEVICE(dev#)
  COMPACT_query(YES)
)
```

The output is similar to the following:

```
ESNP1AAI TOTAL NUMBER OF QUERY RESULTS: 16
ESNP1AAI ****************************************STATUS FIELDS LEGEND*****************************************
ESNP1AAI * STA=SNAPSHOT STATUS | S/H=SOFT/HARD LINK | LTH=LINK TARGET HOLD | CPY=COPYMODE | ZDP=ZDP MANAGED *
ESNP1AAI * DEF=TARGET DEFINED | ACT=ACTIVATED | 1PS=1 PASS COMPLETE | SEC=SECURE | *
ESNP1AAI *****************************************************************************************************
ESNP1AAI ESNP1AAI ESNP1AAI ESNP1AAI ESNP1AAI
ESNP1AAI ESNP1AAI ESNP1AAI ESNP1AAI ESNP1AAI
ESNP1AAI SRC CUU VOLSER TGT CUU VOLSER NAME YYDDD/HH:MM:SS A H H Y P F T S C
ESNP1AAI 000000F3(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F3(N/A ) 000000FB(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F4(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F4(N/A ) 000000FC(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F5(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F5(N/A ) 000000FD(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F6(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F6(N/A ) 000000FE(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F7(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F7(N/A ) 00000100(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F8(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F8(N/A ) 00000101(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N
ESNP1AAI 000000F9(N/A ) FFFFFFFF(N/A ) SNVX_S1 18017/12:42:53 S S - - N - Y - N
ESNP1AAI 000000F9(N/A ) 00000102(N/A ) SNVX_S1 18017/12:42:57 L S Y N N Y - N

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.

**LTH** (Link Target Hold) Indicates if the link target hold is set for the target device. If there is no target device, the field shows a dash (‘-’).

**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.

**IPS** (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#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})}))

[optional_parameters]
)

Where optional_parameters are:

[CHANGE_all_names(Yes|No)]

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

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.
   - %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({VOLUME(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})
```

See “SOURce({VOLUME(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})” on page 66.

**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 59.

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

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu) | VOLUME(volser) | DDNAME(ddname) | CONTROLLER([xxxxxxx-]xxxxx|name})”) on page 63.

**SNAPSHOTID(snapshot_id)**

The ID of the snapshot to be renamed.

**Example**

```
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#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count))})})
[optional_parameters]
)

Where optional_parameters are:

[   
AUTO_UNLink(Yes|No)   
[Check_Link_Target_Hold(Yes|No)]
[FREE(Yes|No)]
[Release_Link_target_hold(Yes|No)]
[WAIT_FOR_Definition(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)])})
]

[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 SNAPSHOTTID(snapshot_id) or TERMinate_ALL(Yes).

SOURce({VOLume(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})

See “SOURce({VOLume(volser)|UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)})|SYMDV#({dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count)})})” on page 66.

Optional parameters

[AUTO_UNLink(Yes|No) | Check_Link_TarGeT_Hold(Yes|No) | FREE(Yes|No) | Release_Link_target_hold(Yes|No) | WAIT_FOR_Definition(Yes|No)]

See “AUTO_UNLink(Yes|No)” on page 55.

Check_Link_TarGeT_Hold(Yes|No)

See “Check_Link_TarGeT_Hold(Yes|No)” on page 55.

FREE(Yes|No)

See “FREE(Yes|No)” on page 58.

Release_Link_target_hold(Yes|No)

See “Release_Link_target_hold(Yes|No)” on page 62.

WAIT_FOR_Definition(Yes|No)

See “WAIT_FOR_Definition(Yes|No)” on page 71.

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

See “GROUP(grpname[,grpname,...])” on page 59.
LOCAL({
UNIT (cuu) [CONTROLLER([xxxxxxxx-]xxxxx|name)] |
VOLUME (volser) [CONTROLLER([xxxxxxxx-]xxxxx|name)] |
DDNAME (ddname) [CONTROLLER([xxxxxxxx-]xxxxx|name)] |
CONTROLLER([xxxxxxxx-]xxxxx|name)
})

See “LOCAL({ UNIT(cuu) [CONTROLLER([xxxxxxxx-]xxxxx|name)] |
VOLUME(volser) [CONTROLLER([xxxxxxxx-]xxxxx|name)] | DDNAME(ddname) |
[CONTROLLER([xxxxxxxx-]xxxxx|name]) CONTROLER([xxxxxxxx-]xxxxx|name) })” on page 59.

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

See “REMOTE (RAGROUP(nn.nn.nn.nn) {UNIT(cuu)
[CONTROLLER([xxxxxxxx-]xxxxx|name]) VOLUME (volser) |
[CONTROLLER([xxxxxxxx-]xxxxx|name]) DDNAME(ddname) |
[CONTROLLER([xxxxxxxx-]xxxxx|name])” on page 63.

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. Depending on the copy mode, and the timing of the UNLINK command, 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.

Syntax

UNLINK

( TaRGet({VOLume(volser) | UNIT({cuu|lowcuu-highcuu|lowcuu:highcuu|cuu(count)}) | SYMDV#({dev#|lowdev#:highdev#|lowdev#|highdev#|dev#(count)})}) [optional_parameters] )

Where optional_parameters are:

[Check_Link_TarGeT_Hold(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)]
[Release_Link_target_hold(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)])]
[SOFTRestore(Yes|No)]
[VARY_Offline(AUTO|NEVER)]
[VARY_Online(AUTO|Yes|No)]
[VCLOSE(Yes|No)]
[WAIT_FOR_Definition(Yes|No)]
Required parameters

TaRGet((VOLUMe(volser) | UNIT((cuu|lowcuu-highcuu|lowcuu-highcuu|cuu(count))) | SYMDV#((dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count))))))

See “TaRGet((VOLUMe(volser) | UNIT((cuu|lowcuu-highcuu|lowcuu-highcuu|cuu(count))) | SYMDV#((dev#|lowdev#:highdev#|lowdev#:highdev#|dev#(count))))))” on page 68.

Optional parameters

Check_Link_TarGeT_Hold(Yes|No)

See “Check_Link_TarGeT_Hold(Yes|No)” on page 55.

FREE(Yes|No)

See “FREE(Yes|No)” on page 58.

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

See “GROUP(grpname[, grpname,...])” on page 59.

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 59.

NEWVOLID(volser)

Creates a new volume ID using the specified volser name.

READY(Yes|No)

See “READY(Yes|No)” on page 62.

Release_Link_target_hold(Yes|No)

See “Release_Link_target_hold(Yes|No)” on page 62.

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 63.

SOFTRestore(Yes|No)

See “SOFTRestore(Yes|No)” on page 66.
VARY_OFFline(AUTO | NEVER)
   See "VARY_OFFline(AUTO|NEVER)" on page 70.

VARY_ONline(AUTO | Yes | No)
   See "VARY_ONline(AUTO|Yes|No)" on page 70.

VCLOSE(Yes | No)
   See "VCLOSE(Yes|No)" on page 71.

WAIT_FOR_Definition(Yes | No)
   See "WAIT_FOR_Definition(Yes|No)" on page 71.

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 48 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 \n[(optional\_parameters)]
```

Where **optional\_parameters** are:

```
[DESCRIPTION(\{\ descriptive\ text\})]
[FORCE(Yes|No)]
[REPLACE(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 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 64.
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,” describes zDP usage concepts.

Chapter 7, “zDP Definition Utility,” describes how to run zDP using the run-time environment.

Chapter 8, “zDP ISPF Interface,” describes how to run zDP using the ISPF interface.

Chapter 9, “SMF Records,” describes zDP SMF records.
CHAPTER 6
Introduction

This chapter covers the following topics:

◆ Overview............................................................................................................ 120
◆ Implementation................................................................................................... 121
◆ Requirements and limitations ............................................................................. 122
◆ Security considerations...................................................................................... 122
Overview

Data Protector for z Systems (zDP) delivers the capability to recover from logical data corruption with minimal data loss. zDP achieves this by providing multiple, frequent, consistent point-in-time copies of data in an automated fashion from which an application level recovery can be conducted, or the environment restored to a point prior to 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 (up to 256) of each source volume.

You can link, unlink, relink snapshots to the same set of target volumes thus having the ability to have 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 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.

Terminology

The following terms are used to describe zDP functionality.

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 256 snapsets associated with it.

Snapshot

A named point-in-time consistent image of all the source volumes in a VDG.

A snapshot can be non-consistent if specified; however, CONSistent(YES) is the default.
Introduction

Snapset capture

The continuous automated process of creating snapsets for a VDG.

Snapshot

A pointer-based, point-in-time image of a single volume. One volume can have up to 256 snapshots created on it.

Enginuity Consistency Assist (ECA)

A PowerMax/VMAX feature 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.

Implementation

zDP implementation is a two-stage process—the planning phase and the implementation phase.

◆ The planning phase is done in conjunction with your Dell EMC representative who has access to tools that can help size the capacity needed for zDP if you are currently a storage system user. This process involves the use of the ChangeTracker component of ResourcePak Base and an additional tool available to Dell EMC representatives to aid in this sizing process.

◆ The implementation phase utilizes the following methods:

  ■ An ISPF interface that allows you to define and manage the zDP run-time environment. Chapter 8, “zDP ISPF Interface”, describes the zDP ISPF interface.

  ■ A batch interface that allows you to submit jobs to define and manage zDP. Chapter 7, “zDP Definition Utility”, describes the zDP definition utility and its configuration statements.

  ■ A zDP run-time 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 to manage zDP processes:

      - ZDP,ECACLEAR
      - ZDP,MODIFY,SMF
      - ZDP,PAUSE
      - ZDP,RELEASEDEVICELOCK
      - ZDP,RESUME
      - ZDP,START
      - ZDP,STOP
Introduction

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 run-time tasks by using Modify commands 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 VDGs.
◆ 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.

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 controller. 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
zDP Definition Utility

This chapter covers the following topics:

- Overview............................................................................................................. 124
- Running zDP definition utility............................................................................. 132
- zDP definition utility statements........................................................................ 133
zDP Definition Utility

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 8, “zDP ISPF Interface.”

Summary of operations

Table 14 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>Delete VDG</td>
<td>DELETE VDG command</td>
</tr>
</tbody>
</table>

Table 15 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>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, STOP_FREE parameter</td>
</tr>
<tr>
<td>Delete TGT</td>
<td>DELETE TGT command</td>
</tr>
</tbody>
</table>

Table 16 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>
<tr>
<td>Create secure snapsets</td>
<td>• SECURE VDG command</td>
</tr>
<tr>
<td></td>
<td>• DEFINE VDG command, CYCLE_TIME[,SECURE] parameter</td>
</tr>
<tr>
<td></td>
<td>• DEFINE VDG command, SAVED_SNAPSET[,SECURE] parameter</td>
</tr>
<tr>
<td></td>
<td>• MODIFY VDG OPTIONS command, CYCLE_TIME[,SECURE] parameter</td>
</tr>
<tr>
<td></td>
<td>• MODIFY VDG OPTIONS command, SAVED_SNAPSET[,SECURE] parameter</td>
</tr>
<tr>
<td>Restore all source volumes in snapshot</td>
<td>RESTORE VDG command</td>
</tr>
</tbody>
</table>
### Table 16 Managing snapsets

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unlink restored snapshot from VDG</td>
<td>UNLINK VDG command</td>
</tr>
<tr>
<td>Terminate snapshots</td>
<td>TERMINATE VDG command</td>
</tr>
<tr>
<td>Simulate snapshot termination by date/time</td>
<td>GLOBAL command, SIMulate_TERMinate parameter</td>
</tr>
</tbody>
</table>

### Table 17 Setting VDG snapshot-related options

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine whether consistency for all snapshots in a snapshot is required</td>
<td>DEFINE VDG command, CONSistent parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, CONSistent parameter</td>
</tr>
<tr>
<td>Set timeout interval and action for consistency operations</td>
<td>DEFINE VDG command, TIMEOUT parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, TIMEOUT parameter</td>
</tr>
<tr>
<td>Specify how often and for how many cycles to create snapshots</td>
<td>DEFINE VDG, CYCLE_TIME parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS, CYCLE_TIME parameter</td>
</tr>
<tr>
<td>Determine when to create the next snapshot if the cycle time is exceeded</td>
<td>DEFINE VDG command, CYCLE_OVERFLOW parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, CYCLE_OVERFLOW parameter</td>
</tr>
<tr>
<td>Set the snapshot limit</td>
<td>DEFINE VDG command, MAX_SNAPSET parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, MAX_SNAPSET parameter</td>
</tr>
<tr>
<td>Determine action if the snapshot limit is reached</td>
<td>DEFINE VDG command, TERMINate_POLICY parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, TERMINate_POLICY parameter</td>
</tr>
<tr>
<td>Set the maximum number of persistent snapshots to be created</td>
<td>DEFINE VDG command, PERSISTENT_COPY_LIMIT parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, PERSISTENT_COPY_LIMIT parameter</td>
</tr>
<tr>
<td>Create saved snapshots</td>
<td>DEFINE VDG command, SAVED_SNAPSETS parameter</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, SAVED_SNAPSETS parameter</td>
</tr>
<tr>
<td>Create secure saved snapshots</td>
<td>DEFINE VDG command, SAVED_SNAPSETS parameter with the SECURE option</td>
</tr>
<tr>
<td></td>
<td>MODIFY VDG OPTIONS command, SAVED_SNAPSETS parameter with the SECURE option</td>
</tr>
</tbody>
</table>

### Table 18 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>
</tbody>
</table>

Table 17 lists operations for setting VDG snapshot-related settings.

Table 18 lists operations for setting SRP usage and RDP cache utilization parameters.
Table 18 Setting SRP usage and RDP cache utilization parameters

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<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 19 lists operations for managing copy-once devices.

Table 19 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 20 lists operations for monitoring zDP status.

Table 20 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 snapsets</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 21 lists operations to control zDP messaging and SMF records.

### Table 21  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</td>
<td>DEFINE VDG command, DEBUG parameter</td>
</tr>
<tr>
<td>messages</td>
<td>MODIFY VDG OPTIONS command, DEBUG parameter</td>
</tr>
<tr>
<td>Set whether to issue WTOR for TERMINATE</td>
<td>GLOBAL command, WTOR_TERMINATE parameter</td>
</tr>
<tr>
<td>VDG, Date command</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>Set the maximum allowable return code</td>
<td>MODIFY VDG OPTIONS command, LOG_OPT parameter</td>
</tr>
<tr>
<td>Enable SMF recording</td>
<td>DEFINE VDG command, SMF parameter</td>
</tr>
</tbody>
</table>

Table 22 lists operations for setting zDP definition utility options.

### Table 22  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</td>
<td>GLOBAL command, MAX_VDG parameter</td>
</tr>
<tr>
<td>definitions</td>
<td></td>
</tr>
<tr>
<td>Set the maximum number of allowed target set</td>
<td>GLOBAL command, MAX_TGT parameter</td>
</tr>
<tr>
<td>definitions</td>
<td></td>
</tr>
<tr>
<td>Set the maximum allowable return code</td>
<td>GLOBAL command, MAXRC parameter</td>
</tr>
</tbody>
</table>
Secure snaps

zDP supports secure snapsets.

---

**Note:** Secure snapshots are created with the Secure Snaps feature of SnapVX, as described in “Secure snaps” on page 40.

---

**CAUTION**

Secure snapshots may only be terminated after they expire or by customer-authorized Dell EMC Support. See “Secure snaps” on page 40 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.

For secure snapshots, the PERSISTENT_COPY_LIMIT at the time the VDG is started is the baseline. All secure snapsets created while the VDG is active are included in the limit check. However, the conversion of an existing snapset to secure will not be recognized until the VDG is restarted.

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 SECURE VDG command. To reset the Secure attribute, specify SECURE,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 snapset ID and state, for example:
  
  EIP0039I VDG1...........170971440I00001 ACT-I 05/07/2017 14:40:15

- An existing snapshot 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.
Copy-once devices

Copy-once devices are included only in the first snapshot created after the start of a VDG. If all devices in a VDG are defined as copy-once devices, the VDG will stop 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 make use of 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/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   Snapshot Query for VDG EMCVDGCl (COPY_ONCE)
```
**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 needs to be issued.
Running zDP definition utility

You normally run the zDP definition utility as a batch job. The JCL is as follows:

```plaintext
//JOBCARD
//*
//ZDPDEF EXEC PGM=EIPINIT,REGION=0M
//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:

- **YOUR.zDP** is the product dataset name prefix 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.

- **nnnn** identifies the ResourcePak Base task that the job runs against.

- zDP definition utility statements are listed in "zDP definition utility statements" on page 133.
zDP definition utility statements

Syntax conventions

The commands follow these syntax conventions:

- Keywords appear in uppercase (for example, **ALL**). They must be spelled exactly as shown.
- For easy reference, command keywords are supplemented by lowercase letters to form a meaningful word (for example, **CONTinue**). When typing a command, use only **CAPITALIZED** characters of any keyword.
- Variables appear in lowercase and italics (for example, **vdg_name**). They represent user-supplied names or values in the syntax.
- Square brackets [ ] indicate an optional entry (for example, 

  ```
  [,MAXRC(nn)]
  ```

- The vertical bar | indicates alternative argument values (for example, **OLDEST|STOP**).
- Curly brackets {} are used to group a series of alternative values that can be used with a single keyword, for example:

  ```
  [ {IMMED|NEXT} ]
  ```

- Aside from the square and curly brackets and the vertical bar characters, you must type all other characters that are shown in the syntax statements.
- Default values are indicated by an underline. For example, if the parameter has the following option, (Yes | **No**), the underlined **No** indicates the default value.

**DEFINE TGT**

The DEFINE TGT statement creates and defines a new target set.

**Syntax**

```
DEFINE TGT  tgt_set_name
```

**Note:** TGT is an alias for TARGET_SET.

**Required parameters**

```
tgt_set_name
```

Specifies the target set. **tgt_set_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.

You can also use optional parenthesis as follows: TGT(**tgt_set_name**)  

**Example**

```
DEFINE TGT EMCTGT_QTR
```
**DEFINE VDG**

The **DEFINE VDG** statement creates and defines a new VDG.

**Syntax**

```plaintext
DEFINE VDG  vdg_name
[ ,CONSistent(Yes|No) ]
[ ,CYCLE_OVERFLOW(IMMED|NEXT) ]
[ ,CYCLE_TIME(mmnn[,count][,SECURE,dd[,skip]]) ]
[ ,DEBUG[STATUS][,STATUSE] ]
[ ,LOG_OPT({SCF|SYSOUT(ddname)}) ]
[ ,MAXRC(nn) ]
[ ,MAX_SNAPSETS(nnnn) ]
[ ,PERSISTENT_COPY_LIMIT(nnnn) ]
[ ,RDP_CACHE_UTIL%(ww,cc) ]
[ ,SAVED_SNAPSETS(ddd,nnnn[,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(nnn[,CONTinue|STOP]) ]
```

---

**Note:** VDG is an alias for VERSIONED_DATA_GROUP.

**Required parameters**

**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 dash. For example: `VDG'MYVDG-P3'`

You can also use optional parenthesis as follows: `VDG(vdg_name)`
Optional parameters

CONSistent(Yes | No)

Defines whether consistency for all of the snapshots in a snapset is required. The default is YES. Consistency is managed using Enginuity Consistent Assist (ECA) or SRDF/A Suspend/Resume as appropriate, to hold write I/Os on the source volumes for the duration of the SnapVX ACTIVATE commands.

**Note:** Because only one method can be used to guarantee consistency across all volumes in a snapset, a mix of SRDF/A and non-SRDF/A volumes is not allowed in a VDG.

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 device discovered with R1 invalid tracks. After all devices have been checked, this process continues every 30 seconds until all of the 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).

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, and NEXT waits for the next interval to expire (CYCLE_TIME minutes).

CYCLE_TIME(mmm[,count][,SECURE,ddd[,skip]])

Specifies how often and for how many cycles to create snapsets.

*mmm* defines the time interval to create snapsets (in minutes). The range is 5 to 9999 and the default is 60.

*count* defines the number of cycles. zDP will terminate when count is reached. A value of 0 is unlimited and is the default.

*SECURE, ddd[,skip]*

Creates secure snapsets.

**Note:** “Secure snaps” on page 128 describes zDP secure snapsets.

*ddd* defines the retention period in days (1-400). The default value is 1.

*skip* defines the skip value (0-256). When specifying a skip value, the first snapset created will always be secure. A skip value of “1” will result in secure snapsets 1, 3, 5, and so on. Specifying “2” will result in secure snapsets 1, 4, 7, and so on.

The default value is not to create secure snapsets.
DEBUG([STATUS], [STATUSE])

Specifies debug options for the zDP run-time task. STATUS displays status messages and STATUSE displays extended status messages.

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).

MAXRC(nn)

Defines the maximum allowable return code for the zDP run-time task. The default is 0, which will treat any Warning as a terminating error.

MAX_SNAPSETS(nn)

Defines the snapset limit. When this limit is reached, snapsets are terminated based upon the termination policy. The maximum value is 256 and this is also the default setting.

PERSISTENT_COPY_LIMIT(nn)

Defines maximum number of persistent snapsets that can be created. This includes all saved, persistent, secure, linked and restored snapsets. When the limit is reached, a persistent snapshot can be terminated or a PERSISTENT RESET command (described in “PERSISTENT” on page 150) can be issued to a snapshot that has persistent set before any additional persistent snapshots can be created.

The default setting is 32.

RDP_CACHE_UTIL%(ww, cc)

Enables checking of RDP (Replication Data Pointer) cache utilization during each cycle against the following thresholds:

ww

When this threshold is exceeded, a warning message is issued and processing continues. The default value is 60%.

cc

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.

SAVED_SNAPSETS(dd, nnnn[, SECURE[, skip]])

Creates saved snapshots.

A saved snapshot is a snapshot that is marked with a SAVED attribute and is retained for the number of days specified in the retention period (nnnn). As part of snapshot creation logic, a check is made to determine if a snapshot should be marked as saved by comparing the current time to the time of the last saved snapshot. If the number of days required between saved snapshots (dd) is reached, the snapshot is marked as saved and its expiration time is set according to the retention period (nnnn).
Specifies the days between snapset creation. The default value is 0, which indicates that no saved snapsets will be created. Valid values are 0-999.

Specifies the retention period in days. When nnnn is exceeded, the oldest saved snapset 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 snapsets.

For example:
- (1,14) creates a snapset every day and maintains a 14 day history. On the 15th day, the first snapset is removed.
- (7,4) is automatically adjusted to (7,7).

Creates secure saved snapsets.

skip defines the skip value (0-256). When specifying a skip value, the first snapset created is always secure. A skip value of “1” results in secure snapsets 1, 3, 5, and so on. Specifying “2” results in secure snapsets 1, 4, 7, and so on. The default behavior is not to create secure snapsets.

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.

Enables SMF recording.

Specifies an SMF record ID.

Includes VDG devices on the startup record (CCUU and PowerMax/VMAX device numbers).

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.

(Defaut) Disables SMF recording.
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 snapsets, 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(nnnn) or either the SRP_SNAP% or the SRP_TERM% value is reached.

- OLDEST (default) terminates the oldest eligible snapset.
- STOP terminates the VDG.

TIMEOUT(nnn[, CONTinue | STOP])

nnn defines the timeout interval for consistency operations in seconds. 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.

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 tgt_set_name [,ALLOWNONEMPTY]

Note: TGT is an alias for TARGET_SET.

Required parameters

tgt_set_name

Specifies the target set. tgt_set_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.

You can also use optional parenthesis as follows: TGT(tgt_set_name)

Optional parameters

ALLOWNonEmpty

When specified in conjunction with MAX_RC(nn) set to 4 or higher, this parameter allows a delete of a non-empty definition.

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.

Required parameters

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 dash. For example: VDG'MYVDG-P3'

You can also use optional parenthesis as follows: VDG(vdg_name)

Optional parameters

ALLOWNonEmpty

When specified in conjunction with MAX_RC(nn) set to 4 or higher, this parameter allows a delete of a non-empty definition.

Example

DELETE VDG EMCVDG1,ALLOWNE
GLOBAL

The GLOBAL statement specifies zDP definition utility execution options for each LPAR.

Syntax

GLOBAL
[ ,MAX_RC (nn) ]
[ ,MAX_VDG (nnn) ]
[ ,MAX_TGT (nnn) ]
[ ,SIMulate_TERMinate(Yes|No) ]
[ ,WTOR_TERMinate(Yes|No) ]

Optional parameters

MAX_RC (nn)
Defines the maximum allowable return code for the zDP definition utility. The default value is 0.

MAX_TGT (nnn)
Defines the maximum number of allowed target set definitions. The default value is 32.

MAX_VDG (nnn)
Defines the maximum number of allowed VDG definitions. The default value is 32.

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, [ALLOW_LARGER_TARGET, ]
SNAPSET(snapset_id)
[TO] TGT tgt_set_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.

**Required parameters**

- `SNAPSET(snapset_id)`
  - Specifies the snapset. This is the portion of the snapset name after the VDG name.
  - For example, for snapset "VDG_UYF1......153281429C00008", the `vdg_name` is "VDG_UYF1" and the `snapset_id` is "153281429C00008".

- `[TO] TGT tgt_set_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`
  - 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 dash. For example: `VDG'MYVDG-P3'`
  - You can also use optional parenthesis as follows: VDG(vdg_name)

**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.
zDP Definition Utility

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.

Note: Using COPY_ONCE(INCLUDE) on the LINK VDG command requires MAXRC=4 or higher.

ONLY  Allows you to manually link the copy-once devices from a specific snapset.

Note: “Copy-once devices” on page 130 describes copy-once devices.

By default, the LINK command operates only on the devices in the snapset (neither of COPY_ONCE values is set).

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

```
MODIFY TGT tgt_set_name,
ADD|REMOVE
{{,CCUU(gk_ccuu, start_ccuu[-end_ccuu][, start_ccuu[-end_ccuu]...])|
,SYMDEV(gk_ccuu, start_dev[-end_dev][, start_dev[-end_dev]...])|
,RDFGroup(gk_ccuu, srdf_group)|
,SCFG(gns_group)}|
[,RMT(srdf_group)]}
```

**Note:** TGT is an alias for TARGET_SET.

Required parameters

ADD|REMOVE

- Specifies whether to add or remove devices.
- Devices can be specified as follows. Only one type is supported per MODIFY statement.
  - CCUU(gk_ccuu, start_ccuu[-end_ccuu][, start_ccuu[-end_ccuu]...])
    - Specifies a z/OS device number or range of device numbers.
  - SYMDEV(gk_ccuu, start_dev[-end_dev][, start_dev[-end_dev]...])
    - Specifies a PowerMax/VMAX device number or range of device numbers.
  - RDFGroup(gk_ccuu, srdf_group)
    - Specifies an SRDF group.
    - `srdf_group` is a one or two-digit hexadecimal value representing a single SRDF group.
    - Device addition/removal by SRDF group is not dynamic. If devices are subsequently added to the SRDF group, they will not automatically be included into/excluded from the target set definition. Another MODIFY TGT ADD|REMOVE statement would need to be issued.
  - SCFG(gns_group)
    - 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.

tgt_set_name

- Specifies the target set. `tgt_set_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.
Optional parameters

RMT(srdf_group)

RMT applies to all of the devices in the MODIFY statement (except SCFG), defining the path to the target storage system.

RMT srdf_group specifies the SRDF group through which to perform a remote operation. srdf_group 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.

Examples

MODIFY TGT EMCTGTR,ADD,
       SYMDEV(3008,0001A0-0001CF),
       REMOTE(70)

MODIFY TGT EMCTGT1,ADD,
       CCUU(3008,3270-32AF)

MODIFY TGT EMCTGT1,ADD,
       SCFG(EMCTGTONS1)

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_ccuu,start_ccuu[-end_ccuu],start_ccuu[-end_ccuu]...)}
,SYNDEV(gk_ccuu,start_dev[-end_dev],start_dev[-end_dev]...)}
,RDFGroup(gk_ccuu,srdf_group)}
,SCFG(gns_group)}
[,COPY_ONCE]
[,REMOTE|RMT(srdf_group)]
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP.

**Note:** COPY_ONCE cannot be used with MODIFY VDG, REMOVE and SCFG(gns_group).

Required parameters

ADD|REMOVE

Specifies whether to add or remove devices.

Devices can be specified as follows. Only one type is supported per MODIFY statement.

`CCUU(gk_ccuu,start_ccuu[-end_ccuu],start_ccuu[-end_ccuu]...)`

Specifies a z/OS device number or range of device numbers. The system running zDP must have channel access to the storage systems to use CCUU.

`SYNDEV(gk_ccuu,start_dev[-end_dev],start_dev[-end_dev]...)`

Specifies a PowerMax/VMAX device number or range of device numbers.

`RDFGroup(gk_ccuu,srdf_group)`

Specifies an SRDF group.

`srdf_group` is a one or two-digit hexadecimal value representing a single SRDF group.

Device addition/removal by SRDF group is not dynamic. If devices are subsequently added to the SRDF group, they will not automatically be included into/excluded from the VDG definition. Another MODIFY VDG ADD|REMOVE statement would need to be issued.
SCFG(gns_group)

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 devices cannot be configured in a VDG.

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 dash, for example: VDG'MYVDG-P3'.

You can also use optional parenthesis as follows: VDG(vdg_name)

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 130 describes copy-once devices.

COPY_ONCE cannot be used with MODIFY VDG,REMOVE.

COPY_ONCE cannot be used with SCFG(gns_group).

**REMOTE|RMT(srdf_group)**

REMOTE applies to all of the devices in the MODIFY statement (except SCFG), defining the path to the target storage system.

RMT *srdf_group* specifies the SRDF group through which to perform a remote operation. *srdf_group* 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.

Examples

```plaintext
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 EMVDG1, 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

MODIFY VDG  vdg_name
,OPTIONS(option1[,option2]...[,option_n])

Note: VDG is an alias for VERSIONED_DATA_GROUP.

Possible values for option are:

CONSistent(Yes|No)
CYCLE_OVERFLOW(IMMED|NEXT)
CYCLE_TIME(mmmm[,count][,SECURE,(NO|ddd[,skip])])
DEBUG([STATUS][,STATUSE])
LOG_OPT(SCF|SYSOUT(ddname))
MAXRC(nn)
MAX_SNAPSETS(nnnn)
PERSISTENT_COPY_LIMIT(nnnn)
SAVED_SNAPSETS(ddd,nnnn[,SECURE,[NO|skip]])
SMF(Yes[,smf_record_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

  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 dash, for example: VDG'MYVDG-P3'.

  You can also use optional parenthesis as follows: VDG(vdg_name)

Optional parameters

CONSistent(Yes|No)

  See “CONSistent(Yes|No)” on page 135.
CYCLE_OVERFLOW(IMMED|NEXT)
   See “CYCLE_OVERFLOW(IMMED|NEXT)” on page 135.

CYCLE_TIME(mm, count[, SECURE, {NO|ddd[, skip]}])
   See “CYCLE_TIME(mm[,count][,SECURE,ddd[.skip]])” on page 135.
   SECURE, NO resets the Secure setting.

DEBUG([STATUS[, STATUSE]])
   See “DEBUG([STATUS[,STATUSE]])” on page 136.

LOG_OPT(SCF|SYSOUT(ddname))
   See “LOG_OPT((SCF|SYSOUT(ddname)))” on page 136.

MAXRC(nn)
   See “MAXRC(nn)” on page 136.

MAX_SNAPSETS(nnnn)
   See “MAX_SNAPSETS(nnnn)” on page 136.

PERSISTENT_COPY_LIMIT(nnnn)
   See “PERSISTENT_COPY_LIMIT(nnnn)” on page 136.

RDP_CACHE_UTIL%(ww, cc)
   See “RDP_CACHE_UTIL%(ww,cc)” on page 136.

SAVED_SNAPSETS(ddd, nnnn[, SECURE[, {NO|skip}]]])
   See “SAVED_SNAPSETS(ddd,nnnn[,SECURE[,skip]])” on page 136.
   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 137.
   Note: The VOLUME(INITIAL) parameter is not available on the MODIFY VDG OPTIONS command.

SRP_SNAP%(nnn)
   See “SRP_SNAP%(nnn)” on page 138.

SRP_TERM%(nnn)
   See “SRP_TERM%(nnn)” on page 138.

SRP_WARN%(nnn)
   See “SRP_WARN%(nnn)” on page 138.

TERMinate_POLICY(OLDEST|STOP)
   See “TERMinate_POLICY(OLDEST|STOP)” on page 138.

TIMEOUT(nn[, CONTinue|STOP])
   See “TIMEOUT(nn[,CONTinue|STOP])” on page 138.

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.

**Syntax**

```
PERSISTENT SET|RESET, VDG  vdg_name, SNAPSHOT(snapset_id)
```

**Note:** VDG is an alias for VERSIONED_DATA_GROUP.

**Required parameters**

- **SET | RESET**
  Determines whether to set or reset the “persistent” attribute.

- **SNAPSHOT(snapset_id)**
  Specifies the snapset. This is the portion of the snapset name after the VDG name.

  For example, for snapset "VDG_UYF1......153281429C00008", the `vdg_name` is "VDG_UYF1" and the `snapset_id` is "153281429C00008".

- **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 dash, for example: `VDG'MYVDG-P3'`

  You can also use optional parenthesis as follows: `VDG(vdg_name)`

**Examples**

```
PERSISTENT SET, VDG EMCVG1, SNAPSHOT(152761254C00011)
PERSISTENT RESET, VDG EMCVG_TST, SNAPSHOT(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 {tgt_set_name|*}
QUERY TGT {tgt_set_name|*}, FREE

Note: TGT is an alias for TARGET_SET.

Required parameters

tgt_set_name | *

Specifies the target set. tgt_set_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.

You can also use optional parenthesis as follows: TGT(tgt_set_name)

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 0001DA 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
QUERY TGT

This statement displays target set information.

Syntax

```
QUERY TGT {tgt_set_name|}* [,STATUS] [,DEVice]
```

**Note:** TGT is an alias for TARGET_SET.

Required parameters

```
tgt_set_name|*
```

Specifies the target set. `tgt_set_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.

You can also use optional parenthesis as follows: `TGT(tgt_set_name)`

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

EMCP001J QUERY TGT EMCTGT1,STATUS
EIP0029I  TGT EMCTGT1 is Linked, SNAPSET EMCVDG1.........160051020C00003
EIP0034I QUERY command completed
```
Example 2: 
**DEVICE, not linked**

The following example shows QUERY TGT, DEVICE output for devices that are not linked:

**Example 2:**
**DEVICE, 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)

---

**Example 3:**
**DEVICE, linked**

The following example shows QUERY TGT, DEVICE output for devices that are linked:

**Example 3:**
**DEVICE, linked**

EMCP001I QUERY TGT EMCTGT1,DEVICE

EIP0030I Device Query for TGT EMCTGT1

EIP0023I SYMM 0001234-05678, Microcode level 5977_0772, Type VMAX200K
EIP0024I Gatekeeper 3008, Device Count: 32

---
zDP Definition Utility

**QUERY VDG**

This statement displays VDG information.

**Syntax**

```plaintext
QUERY VDG {vdg_name|*} [,DEVice[,COPY_ONCE]] [,SNAPSET[,DETAIL][,COPY_ONCE]] [,start_dev[-end_dev]] [,STATUS] [,SYMM(gatekeeper)]
```

---

**Note:** The SNAPSET keyword may only be followed by DETAIL, COPY_ONCE, SYMM(gatekeeper), or start_dev[-end_dev].

---

**Note:** VDG is an alias for VERSIONED_DATA_GROUP.

**Required parameters**

- `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 dash. For example: VDG'MYVDG-P3'

  You can also use optional parenthesis as follows: VDG(vdg_name)

  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.

- `SNAPSET[,DETAIL][,COPY_ONCE]`  
  Displays the snapsets created for the VDG.

**Note:** The SNAPSET keyword may only be followed by DETAIL, or SYMM(gatekeeper), or start_dev[-end_dev].

- `COPY_ONCE`  
  Displays only snapsets that contain copy-once devices.

- `DETAIL`  
  Displays each device in the snapset.
start_dev[-end_dev]

Narrows the query to the specified range of devices.

STATUS

Displays VDG status, cycle and properties.

SYMM(gatekeeper)

Narrows the query to the storage system specified with the gatekeeper.

Examples

Example 1: STATUS

The following example shows QUERY VDG,STATUS output:

```plaintext
EMCP001I QUERY VDG VDG_TEST,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)
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)
EIP0020I   Log_Opt(SCF)
EIP0020I   SMF(NO)
EIP0020I   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.

```plaintext
EMCP001I QUERY VDG EMCVDG1,DEVICE
EIP0023I Device Query for VDG EMCVDG1
EIP0024I     SYMM 0001234-05678, Microcode level 5977_0772, Type VMAX200K
EIP0024I     Gatekeeper 3008, Device Count: 32
EIP0024I     SRP ID/Name: 0001/SRP_1, Reserved Capacity: 10%
EIP0026I     Total Capacity: 506M, Total Allocated: 4789K, Snap Allocated: 0
```

```plaintext
  CCUU  DEVICE  TYPE  SIZE  SRP ID  RDF INFO/MODE
EIP0027I  ____  ______  ____  ________  ______   ______________
EIP0027I  3150  000170/O  CKD  32760  0001    ---
EIP0027I  3151  000171  CKD  32760  0001    ---
EIP0027I  3152  000172  CKD  32760  0001    ---
EIP0027I  3153  000173  CKD  32760  0001    ---
EIP0027I  3154  000174  CKD  32760  0001    ---
EIP0027I  3155  000175  CKD  32760  0001    ---
EIP0027I  3156  000176  CKD  32760  0001    ---
EIP0027I  3157  000177  CKD  32760  0001    ---
EIP0027I  3158  000178  CKD  32760  0001    ---
EIP0027I  3159  000179  CKD  32760  0001    ---
EIP0027I  315A  00017A  CKD  32760  0001    ---
EIP0027I  315B  00017B  CKD  32760  0001    ---
EIP0027I  315C  00017C  CKD  32760  0001    ---
EIP0027I  315D  00017D  CKD  32760  0001    ---
EIP0027I  315E  00017E  CKD  32760  0001    ---
EIP0027I  315F  00017F  CKD  32760  0001    ---
```
The following example shows QUERY VDG,DEVICE output for remote devices.

```
EMCP001I QUERY VDG EMCVGCCC1,DEVICE
EIP0022I  Device Query for VDG EMCVGCCC1
EIP0023I  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
```

```
EIP0027I  CCUU DEVICE TYPE SIZE SRP ID RDF INFO/MODE
EIP0027I  ____ ______ ____ ________ ______ ______________
EIP0028I  ---- 000270 CKD  10017  0001 R2:  70,71 /S
EIP0028I  ---- 000271 CKD  10017  0001 R2:  70,71 /S
EIP0028I  ---- 000272 CKD  10017  0001 R2:  70,71 /S
EIP0028I  ---- 000273 CKD  10017  0001 R2:  70,71 /S
```

The following example illustrates the QUERY VDG,DEVICE output with the COPY_ONCE option specified:

```
EIP0027I  CCUU DEVICE TYPE SIZE SRP ID RDF INFO/MODE COPY_ONCE SSET
EIP0027I  ____ ______ ____ ________ ______ ______________ ______________
EIP0028I  3150 000170/O CKD  32760  0001 ---          171360952800001
```

**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**
  - Lists the CCUU of the device.

- **DEVICE**
  - Lists the PowerMax/VMAX device number of the device. ‘/O’ indicates a copy-once device.

- **TYPE**
  - Lists the device type.

- **SIZE**
  - Lists the device size.

- **SRP ID**
  - Lists the ID of the SRP to which the device belongs.

- **RDF INFO/MODE**
  - Lists the SRDF device type, SRDF groups, and SRDF mode for the device.

- **COPY_ONCE SSET**
  - Indicates the last saved snapshot containing the copy-once device.
Example 3: SNAPSET

The following example shows QUERY VDG, SNAPSET output:

EMCP001I QUERY VDG EMCDGCl, SNAPSET

EIP0035I Snapset Query for VDG EMCDGCl

EIP0023I SYMM 0001900-01234, Microcode level 5977_0811, Type VMAX200K
EIP0024I Gatekeeper D008, Device Count: 48, Snapset Count: 12
EIP0025I SRP ID/Name: 0001/SRP_1, Reserved Capacity: 10%
EIP0026I Total Capacity: 506M, Total Allocated: 20M, Snap Allocated: 10M

EIP0036I

<table>
<thead>
<tr>
<th>SNAPSET_NAME</th>
<th>CREATE</th>
<th>STATE</th>
<th>EXPIRATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYVDGC1........161231257S00012</td>
<td>ACT-S</td>
<td>05/02/2016 12:57:25</td>
<td>4598 44296 05/07/2016 12:57:25</td>
</tr>
<tr>
<td>MYVDGC1........161231307C00013</td>
<td>ACT</td>
<td>05/02/2016 13:07:25</td>
<td>4518 44186</td>
</tr>
<tr>
<td>MYVDGC1........161231317C00014</td>
<td>ACT</td>
<td>05/02/2016 13:17:25</td>
<td>4598 44296</td>
</tr>
<tr>
<td>MYVDGC1........161231327C00015</td>
<td>ACT-P</td>
<td>05/02/2016 13:27:25</td>
<td>4390 43042</td>
</tr>
<tr>
<td>MYVDGC1........161231337C00016</td>
<td>ACT</td>
<td>05/02/2016 13:37:24</td>
<td>4837 45184</td>
</tr>
<tr>
<td>MYVDGC1........161231347C00017</td>
<td>ACT-P</td>
<td>05/02/2016 13:47:25</td>
<td>3092 31296</td>
</tr>
<tr>
<td>MYVDGC1........161231400C00018</td>
<td>ACT</td>
<td>05/02/2016 14:00:32</td>
<td>4245 42245</td>
</tr>
<tr>
<td>MYVDGC1........161231410C00019</td>
<td>ACT</td>
<td>05/02/2016 14:10:31</td>
<td>4159 40305</td>
</tr>
<tr>
<td>MYVDGC1........161231420C00020</td>
<td>ACT</td>
<td>05/02/2016 14:20:32</td>
<td>3092 31296</td>
</tr>
<tr>
<td>MYVDGC1........161231430C00021</td>
<td>ACT</td>
<td>05/02/2016 14:30:31</td>
<td>3145 30164</td>
</tr>
<tr>
<td>MYVDGC1........161231440C00022</td>
<td>ACT</td>
<td>05/02/2016 14:40:31</td>
<td>2097 20429</td>
</tr>
<tr>
<td>MYVDGC1........161231450C00023</td>
<td>ACT</td>
<td>05/02/2016 14:50:31</td>
<td>4592 45296</td>
</tr>
<tr>
<td>MYVDGC1........161231460C00024</td>
<td>ACT</td>
<td>05/02/2016 15:00:32</td>
<td>4245 42245</td>
</tr>
</tbody>
</table>

EIP0034I QUERY command completed

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 snapset name, with the VDG name followed by the snapset ID.

The snapset name is a string of 32 characters:

VDGnamexxxxxxxxYYDDDHMMttccccrr

Where:

- **YYDDDHMM** is the timestamp for the creation of the snapshot.
  - **YYDDD** is the date (Julian: Year, Month Day).
  - **HH** is hours (0-23), **MM** is minutes (0-59).
- **t** indicates the snapshot type: **P** = Persistent, **S** = Saved, **C** = Cyclical Create, **I** = Secure.
- **cccccc** is the zDP managed cycle number.
- **rr** is reserved for future use.

If the VDG name is less than 15 characters, it is padded with periods.
**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|TIME** Indicates the snapshot creation date and time.

**SOURCE TRACKS**
- **CHANGED** Lists the number of changed tracks.
- **UNIQUE** Lists the number of tracks that have only one snapshot pointing to them.

**EXPIRATION DATE|TIME** Indicates the date and time the snapshot will expire.

**Note:** The expiration date is generated when a saved snapshot is created.

**Example 4:** The following example shows the detailed part of the report produced with QUERY VDG,SNAPSET,DETAIL:

<table>
<thead>
<tr>
<th>EIP0036I</th>
<th>SNAPSET_NAME</th>
<th>STATE</th>
<th>CREATE DATE</th>
<th>TIME</th>
<th>CHANGED</th>
<th>UNIQUE</th>
<th>SRCDEV/CCUU</th>
<th>TGTDEV/CCUU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIP0038I</td>
<td>VDG_UYF1........161371429800006</td>
<td>ACT-S</td>
<td>05/16/2016</td>
<td>14:29:19</td>
<td>0</td>
<td>0</td>
<td>000140/----</td>
<td>000141/----</td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT-S</td>
<td>05/16/2016</td>
<td>14:29:19</td>
<td>0</td>
<td>0</td>
<td>000142/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT-S</td>
<td>05/16/2016</td>
<td>14:29:19</td>
<td>0</td>
<td>0</td>
<td>000143/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>VDG_UYF1........161381523800001</td>
<td>ACT-S</td>
<td>05/17/2016</td>
<td>15:23:10</td>
<td>0</td>
<td>0</td>
<td>000140/----</td>
<td>000141/----</td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT-S</td>
<td>05/17/2016</td>
<td>15:23:10</td>
<td>0</td>
<td>0</td>
<td>000142/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT-S</td>
<td>05/17/2016</td>
<td>15:23:10</td>
<td>0</td>
<td>0</td>
<td>000143/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>VDG_UYF1........161381544C00008</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:44:10</td>
<td>0</td>
<td>0</td>
<td>000140/----</td>
<td>000141/----</td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:44:10</td>
<td>0</td>
<td>0</td>
<td>000142/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:44:10</td>
<td>0</td>
<td>0</td>
<td>000143/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>VDG_UYF1........161381547C00009</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:47:11</td>
<td>0</td>
<td>0</td>
<td>000140/----</td>
<td>000141/----</td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:47:11</td>
<td>0</td>
<td>0</td>
<td>000142/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:47:11</td>
<td>0</td>
<td>0</td>
<td>000143/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>VDG_UYF1........161381550C00010</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:50:11</td>
<td>0</td>
<td>0</td>
<td>000140/----</td>
<td>000141/----</td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:50:11</td>
<td>0</td>
<td>0</td>
<td>000142/----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP0038I</td>
<td>ACT</td>
<td>05/17/2016</td>
<td>15:50:11</td>
<td>0</td>
<td>0</td>
<td>000143/----</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Output fields**

The fields in this example have the same meaning as in the QUERY VDG,SNAPSET 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**

```plaintext
RESTORE VDG  vdg_name,SNAPSET(snapset_id)
 [,COPY_ONCE(INCLUDE|ONLY)]
```

**Required parameters**

```plaintext
SNAPSET(snapset_id)
```

Specifies the snapset ID. This is the portion of the snapset name after the VDG name.

For example, for snapset "VDG_UYF1......153281429C00004", the vdg_name is "VDG_UYF1" and the snapset_id is "153281429C00004".

**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 dash. For example: VDG'LVVDG-P3'

You can also use optional parenthesis as follows: VDG(vdg_name)

**Optional parameters**

```plaintext
COPY_ONCE(INCLUDE|ONLY)
```

Determines the action for copy-once devices:

**INCLUDE**

Includes copy-once devices for a RESTORE of a snapset that does not contain the copy-once devices.

**ONLY**

Allows you to manually restore the copy-once devices from a specific snapset.

**Note:** “Copy-once devices” on page 130 describes copy-once devices.

By default, the RESTORE command operates only on the devices in the snapset (neither of COPY_ONCE values is set).

**Example**

```plaintext
RESTORE VDG EMCVDG_PROD,SNAPSET(153621026C00004)
```
SECURE VDG

Converts an existing snapset to secure.

**Note:** “Secure snaps” on page 128 describes zDP secure snapsets.

**Syntax**

```
SECURE VDG  vdg_name,SNAPSET(snapset_id)
[,,EXPIration(ddd) [,EXTEND]]
```

**Required parameters**

```
SNAPSET(snapset_id)
```

Specifies the snapshot ID. This is the portion of the snapshot name after the VDG name.

For example, for snapshot “VDG_UYF1.......153281429C00004”, the `vdg_name` is “VDG_UYF1” and the `snapset_id` is “153281429C00004”.

```
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 dash. For example: `VDG'LVVDG-P3'`

You can also use optional parenthesis as follows: VDG(`vdg_name`)

**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 snapshot (the VDG does not need to be active).

If the snapshot 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 tgt_set_name
```

**Note:** TGT is an alias for TARGET_SET.

Required parameters

```
tgt_set_name
```

Specifies the target set. `tgt_set_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.

You can also use optional parenthesis as follows: TGT(`tgt_set_name`)

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.

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. This is the portion of the snapset name after the VDG name.

For example, for snapset "VDG_UYF1.......153281429C00008", the vdg_name is "VDG_UYF1" and the snapset_id is "153281429C00008".
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 dash. For example:
VDG'LVVDG-P3'

You can also use optional parenthesis as follows: VDG(vdg_name)

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 tgt_set_name [,FREE(Yes|No)]
UNLINK TGT tgt_set_name [,STOP_FREE(Yes|No)]

Note: TGT is an alias for TARGET_SET.

Required parameters

tgt_set_name

Specifies the target set. tgt_set_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.

You can also use optional parenthesis as follows: TGT(tgt_set_name)

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.

Examples

UNLINK TGT EMCTGT_RMT
UNLINK VDG

This statement unlinks a restored snapset (with the snapset state of RES) from a VDG.

Syntax

UNLINK VDG vdg_name, SNAPSET(snapset_id)

Note: VDG is an alias for VERSIONED_DATA_GROUP.

Required parameters

SNAPSET(snapset_id)

Specifies the snapset. This is the portion of the snapset name after the VDG name.

For example, for snapset "VDG_UYF1.......153281429C00004", the vdg_name is "VDG_UYF1" and the snapset_id is "153281429C00004".

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 dash. For example: VDG'LVVDG-P3'

You can also use optional parenthesis as follows: VDG(vdg_name)

Example

UNLINK VDG EMCVDGR, SNAPSET(153621026C00004)
zDP Definition Utility
CHAPTER 8
zDP ISPF Interface

This chapter covers the following topics:

◆ Introduction ....................................................................................................... 168
◆ Logging in to zDP ............................................................................................... 168
◆ Logging out of zDP ............................................................................................ 169
◆ zDP functions .................................................................................................... 169
Introduction

The zDP ISPF interface uses standard IBM ISPF conventions. The interface provides a visual user interface to the zDP run-time 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_RACF_START
     Specify the resource name for the START command.
   - EIP_RACF_STOP
     Specify the resource name for the STOP command.
   - EIP_RACF_RELDLOCK
     Specify the resource name for the RELDLOCK command.
   - EIP_RACF_PAUSE
     Specify the resource name for the PAUSE command.
   - EIP_RACF_RESUME
     Specify the resource name for the RESUME command.
   - EIP_LOAD_LIBRARY
     Specify the library containing the zDP runtime modules. Enter the PDS name in standard TSO format. This library must be authorized.
   - EIP_RACF_SMF
     Specify the resource name for the ZDP,MODIFY,SMF command.
   - EIP_RACF_TERM_RANGE
     Specify the resource name for the Terminate Range command.
Logging out of zDP

To log out of the zDP ISPF interface:
1. Press PF4 to return to the main panel.
2. Press PF3 to exit.

zDP functions

The zDP Tool List panel is shown in Figure 1.

```
Enter a command option ===>

   VDG Functions           TGT Functions
      M Monitor             DT Display
      CV Configure          CT Configure

   Snap Set Functions
      1 Query

   Session Control Options
      S Set Options

PF1: Help       PF3: Exit
```

Figure 1 zDP Tool List panel

IMPORTANT
The following sections provide high-level overviews of the zDP panel functions. For more information about individual panel operations, press PF1 to access the corresponding help panel.
**zDP help**

The zDP Main Help panels provide detailed information about zDP panel usage and operation.

Command=>

F7=PrvPage  F8=NxtPage  F10=PreTopic  F11=NxtTopic  Enter=Next

More: +

From the main zDP Tools panel you can access various functions as listed below. Help for these functions is explained here.

**VDG Functions**

**M Monitor** - This option will invoke the VDG monitor. From this option you can Start, Stop and Display various aspects of any Run Time defined VDG.

**CV Configure** - From this function you can manage Versioned Data Groups that reside in the VDG Member Library.

**TGT Functions**

**DT Display** - This function will display any Run Time defined Target Sets. From this option you can Display, Link, Unlink, Report etc on various aspects of any Run Time defined TGT.

**Figure 2 zDP Main Help panel**

Help information includes the panel purpose, description of input fields, user actions, and other useful details.

- Press PF1 to access help.
- Press PF8 to scroll through each Help panel.
- Press PF3 to return to the original panel.
Session Control options

This function allows you to set zDP session control parameters. From the zDP Tool List panel, enter S (Set Options). The following panel displays:

```
+-----------------------------------------------+-----------------------------------------------+
| SCF Suffix . . . . . . RCG | SCF Jobname . . . . . . SCFRCG                  |
| Product Load Library . . . 'RFBNNE1.ZDP.LINKLIB'       |
| VDG Member Library . . . 'RFBNNE1.ICDP.VDG'           |
| TGT Member Library . . . 'RFBNNE1.ICDP.TGT'           |
| Console Message Timeout .. 1                          |
| Unit . . . . . . . . . SYSALLDA and Volume . . . . . .   |
| RACF Start Process Resource Name . . EMC.ADMIN.CMD.ZDP.START |
| RACF Stop Process Resource Name . . EMC.ADMIN.CMD.ZDP.STOP     |
| RACF Reldlock Process Resource Name . . EMC.ADMIN.CMD.ZDP.RELDLOCK |
| RACF Pause Process Resource Name . . EMC.ADMIN.CMD.ZDP.PAUSE  |
| RACF Resume Process Resource Name . . EMC.ADMIN.CMD.ZDP.RESUME|
| RACF SMF Process Resource Name . . EMC.ADMIN.CMD.ZDP.MODIFY  |
| RACF Term Range Process Resource Name . . EMC.ADMIN.CMD.ZDP.TERMINATE.RANGE |
| RACF ECA Clear Process Resource Name . . EMC.ADMIN.CMD.ZDP.ECACLEAR |
| RACF Secure SS Process Resource Name . . EMC.ADMIN.CMD.ZDP.SECURE |
| Debug Mode . . . . . . . . . . . . . . N (Y/N)            |
| PF1: Help Enter: Save Session PF3: Exit               |
```

Figure 3 Specify the Session parameters panel

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 panel usage, press PF1 to access help.

VDG monitor

This function allows you to monitor and control the runtime VDGs. From the zDP Tool List panel, enter M (Monitor). The following panel displays:

```
+-----------------------------------------------+-----------------------------------------------+
| Command=>                                     Refresh=> 07:20:25 01/11/17                  |
| Primary Cmds: VDG CT TGT U SRP Use            |
| Line Cmds: S Sel C Cnfg X Start P Stop Q Rept |
| M Modify D Delete U Reldlock H Pause G Resum F SMF L ECACLEAR |
| Max Snp Snap Start Next Snap                  |
| VDG Name Status Cnt Cnt Date Time Date Time   |
```

```
_ EGJ_UZB NOT ACTIVE 256 0
_ RDG_UYG1_PR PAUSED 256 0
_ VAS_VDGCKO NOT ACTIVE 10 0
```

Figure 4 zDP VDG Monitor panel
Use this function to:

◆ Start and stop VDGs.
◆ Automatically update per a refresh interval.
◆ Modify and save a VDG to a repository PDS.

Note: If you override an existing PDS member, the DEFINE VDG statement and the most recently specified MODIFY statements will be saved.

◆ Query status and generate reports.
◆ Delete a VDG from the runtime system.

Note: For detailed information about panel usage, press PF1 to access help.

The following commands available in the zDP VDG Monitor panel can also be executed through SCF:

◆ X = Start
◆ P = Stop
◆ U = Rellock
◆ H = Pause
◆ G = Resum
◆ F = SMF
◆ L = ECACLEAR

See the ResourcePak Base for z/OS Product Guide for information about these commands.

N/A status

zDP VDG Monitor shows ‘N/A’ in the Status field for VDGs that are not available in the ISPF interface because of an API error. Such VDGs require user attention and actions. See N/A status details provided in the PF1 help to repair the VDG.

IMPORTANT

If there is at least one VDG with N/A status, all zDP ISPF commands and options are blocked.
VDG configure

This function allows you to create and manage VDGs. From the zDP Tool List panel, enter CV (Configure). The following panel displays:

![Manage VDG Members panel](image)

Use this function to:

- Create a VDG member.
- Copy, delete, edit, promote, rename, and display a member.
- Specify volumes using Group Name Services (GNS). You can specify volumes to only be copied once or be copied on demand during the capture cycles.
- Define the interval at which snapshots will be repeatedly taken for all volumes in the VDG group.
- Define the maximum number of versions for a VDG and options when that value is reached.
- Allow consistent snapset generation.
- Allow logging to a separate log.
- Generate LOGREC and SMF records.
- Account for an auto-swap situation.
- Allow a warning for excess pool utilization.

**Note:** For detailed information about panel usage, press PF1 to access help.

TGT display

This function allows you to display, link, and unlink target sets. From the zDP Tool List panel, enter DT (Display). The following panel displays:

![zDP Target Set Display](image)
Use this function to:

- Configure devices within the target set and the VDG.
- Query status and generate reports.
- Allow link and unlink commands to be executed.

**Note:** When linking a target set from this panel (as opposed to the Snapset functions), specify the VDG and snapset name to be used.

- Modify and save a TGT to a repository PDS.
- Query and terminate the process of freeing backend allocations held by a device.
- Access the zDP SRP Monitor.
- Delete a TGT from the runtime system.

**Note:** For detailed information about panel usage, press **PF1** to access help.

### TGT configure

This function allows you to create and manage a TGT member. Once the TGT is created, you can save it to the configuration dataset and/or promote it to the run-time environment. From the zDP Tool List panel, enter **CT** (Configure). The following panel displays:

<table>
<thead>
<tr>
<th>Manage TGT Members</th>
<th>Scroll=&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command=&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>Primary Cmnds:</strong></td>
<td></td>
</tr>
<tr>
<td>C = Create</td>
<td></td>
</tr>
<tr>
<td>C = Copy, D = Delete, E = Edit, P = Promote, R = Rename</td>
<td></td>
</tr>
<tr>
<td>S = Display</td>
<td></td>
</tr>
<tr>
<td>Name Created Changed Size Mod ID</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7** Manage TGT Members panel

Use this function to:

- Create a TGT member.
- Copy, delete, edit, promote and display a TGT member.

**Note:** For detailed information about panel usage, press **PF1** to access help.
Snapset functions

This function allows you to query status, terminate, and link snapsets associated with the selected VDG. From the zDP Tool List panel, enter 1 (Query). The following panel displays:

--- zDP Snapset Controller Display ---
Command==> Line Cmds: S = Sel  Q = Query

Serial     GK    Dev     Snap   RDP  Remote
VDG Name    Number   CCUU  Count  Count  Util Hop List
--------------------------------------------------------
_ RDG_SSM1  000196701130 6234  2   33  1%  
_ RDG_TERM_ALL  000196701130 6234  2   0  1%  
_ UYE1_1DEV_TEST1 000196701130 6234  1   12  1%

--- zDP VDG TESTVDG ---
Command==> Snapset Device --
CCUU  SDEV     CCUU  SDEV     CCUU  SDEV     CCUU  SDEV
_ 0AA1A 00003A _ 0AA1B 00003B _ ----- ------ _ ----- ------ _ ----- ------

--- zDP Snapset Display ---
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
Source Cycle     Creation     Src Trk     Src Byte     Expiration
Dev       Num     Date     Time     Chg     Uni     Chg     Uni     State     Date     Time

---

Figure 8 zDP SnapSet Controller Display panel

From the zDP SnapSet Controller Display panel, you can query a VDG to generate a report or select a VDG for processing. When you choose S (select), the following panel displays the storage system device list for the selected VDG:

--- zDP VDG TESTVDG ---
Command==> Snapset Device --
CCUU  SDEV     CCUU  SDEV     CCUU  SDEV     CCUU  SDEV
_ 0AA1A 00003A _ 0AA1B 00003B _ ----- ------ _ ----- ------ _ ----- ------

--- zDP Snapset Display ---
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
Source Cycle     Creation     Src Trk     Src Byte     Expiration
Dev       Num     Date     Time     Chg     Uni     Chg     Uni     State     Date     Time

---

Figure 9 SnapSet Device panel

Select one or more devices from the SnapSet Device panel. The following panel displays:

--- zDP Snapset Display ---
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
Source Cycle     Creation     Src Trk     Src Byte     Expiration
Dev       Num     Date     Time     Chg     Uni     Chg     Uni     State     Date     Time

---

Figure 10 zDP SnapSet Display panel
From this panel you can link a snapset to a TGT, unlink a snapset from a TGT, restore it to a VDG, or terminate a snapset. You can also set or reset snapset persistence.

**Note:** For detailed information about panel usage, press **PF1** to access help.
CHAPTER 9
SMF Records

This chapter covers the following topics:

◆ Overview............................................................................................................. 178
◆ SMF record format ............................................................................................. 178
SMF Records

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 134.
- 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 XL1 1-byte gas
DS H 2-byte gas
EIPSMFD_HDR_LEN EQUS *-EIPSMFD LENGTH OF 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_FLG1 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_SEG# DS XL1 Segment number
*
.
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 Total 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.

```assembly
**
* 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
ORG EIPSMFD_SMF_SSET
EIPSMFD_SMF_VDGNAME    DS   CL15    VDG Name
ORG
.* 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_SYM#_DEV   DS  F       Device Count - total for Symm
EIPSMFD_SMF_SYMDEV#_DEV DS  F       EIPSMFD_SMF_DEV device Count
```

### Symm section

Each entry in the Symm section is described by the following DSECT.

```assembly
**
* Symmetrix Controller Section - one controller
**
EIPSMFD_SMF_SYMM      DSECT ,      Symmetrix Controller
EIPSMFD_SMF_SYMMLEN    DS   F       Section length
EIPSMFD_SMF_SYMMLENNSRP DS   F       - Length including SRP(s)
EIPSMFD_SMF_SYMMLENDEV DS   F       - Length including SRP(s) + DEVs
EIPSMFD_SMF_SYMK       DS   F       GateKeeper CCUU
EIPSMFD_SMF_SYMSER     DS   CL12    Serial number
EIPSMFD_SMF_SYMMCL     DS   XL4     Microcode level
EIPSMFD_SMF_SYMRFGRP   DS   XL16    RDF Group list (x'FF' for local)
EIPSMFD_SMF_SYMUCB@    DS   A       GateKeeper UCB address
EIPSMFD_SMF_SYMDEV#    DS   F       Device Count - total for Symm
EIPSMFD_SMF_SYMDEV#_DEV DS  F       EIPSMFD_SMF_DEV device Count
```
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_SRPLEN DS F 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_SRPALC DS XL8 SRP Allocated Tracks
EIPSMFD_SMF_SRPSNP DS XL8 SRP Snap Tracks
EIPSMFD_SMF_SRP#L EQU *-EIPSMFD_SMF_SRPLEN .SRP entry Length

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_DEVSYMD DS F Symm device number
EIPSMFD_SMF_DEVLN1 EQU *-EIPSMFD_SMF_DEV (Short length)
EIPSMFD_SMF_DEVTKCHG DS F Source changed track count
EIPSMFD_SMF_DEVTKUNI DS F Source unique track count
EIPSMFD_SMF_DEVNEXT EQU * Next entry, etc. (if any)
EIPSMFD_SMF_DEVLN EQU *-EIPSMFD_SMF_DEV