## Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVATE_SUBTASK#</td>
<td>54</td>
</tr>
<tr>
<td>ADMIN</td>
<td>55</td>
</tr>
<tr>
<td>ALLOFAIL</td>
<td>55</td>
</tr>
<tr>
<td>ALLOSEQ</td>
<td>55</td>
</tr>
<tr>
<td>ALLUNIT</td>
<td>55</td>
</tr>
<tr>
<td>ALLOW_CANCEL_LOCKED</td>
<td>55</td>
</tr>
<tr>
<td>ALLOW_FB_META</td>
<td>56</td>
</tr>
<tr>
<td>ALLOW_SYMDEV#</td>
<td>56</td>
</tr>
<tr>
<td>ALUNUSED</td>
<td>56</td>
</tr>
<tr>
<td>AUTOACTIVATE</td>
<td>56</td>
</tr>
<tr>
<td>AUTO_BIND_TDEV</td>
<td>56</td>
</tr>
<tr>
<td>AUTOCLN</td>
<td>57</td>
</tr>
<tr>
<td>AUTODEAL</td>
<td>57</td>
</tr>
<tr>
<td>AUTORLSE</td>
<td>57</td>
</tr>
<tr>
<td>AUTO_UNBIND_TDEV</td>
<td>57</td>
</tr>
<tr>
<td>AUTOXPND</td>
<td>57</td>
</tr>
<tr>
<td>BACKGRND</td>
<td>57</td>
</tr>
<tr>
<td>BCVONLY</td>
<td>57</td>
</tr>
<tr>
<td>CACHESYM</td>
<td>58</td>
</tr>
<tr>
<td>CATALOG</td>
<td>58</td>
</tr>
<tr>
<td>CHECKBCV</td>
<td>58</td>
</tr>
<tr>
<td>CHKONLIN</td>
<td>58</td>
</tr>
<tr>
<td>CLEAN_R2</td>
<td>58</td>
</tr>
<tr>
<td>CLEANDIFF</td>
<td>58</td>
</tr>
<tr>
<td>CMPLT</td>
<td>58</td>
</tr>
<tr>
<td>CMPLTMSG</td>
<td>59</td>
</tr>
<tr>
<td>COLLAPSE</td>
<td>59</td>
</tr>
<tr>
<td>COMPACT_MISMATCH</td>
<td>59</td>
</tr>
<tr>
<td>CONDVOL</td>
<td>59</td>
</tr>
<tr>
<td>CONGROUP</td>
<td>59</td>
</tr>
<tr>
<td>CONGROUP_LDMF</td>
<td>60</td>
</tr>
<tr>
<td>CONSALL</td>
<td>61</td>
</tr>
<tr>
<td>CONSIST</td>
<td>61</td>
</tr>
<tr>
<td>CONSVOL</td>
<td>61</td>
</tr>
<tr>
<td>COPYCYL</td>
<td>61</td>
</tr>
<tr>
<td>COPYFAIL</td>
<td>62</td>
</tr>
<tr>
<td>COPYVOL</td>
<td>62</td>
</tr>
<tr>
<td>CSMSDATA</td>
<td>62</td>
</tr>
<tr>
<td>CSMSMGMT</td>
<td>62</td>
</tr>
<tr>
<td>CSMSSTOR</td>
<td>62</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>62</td>
</tr>
<tr>
<td>DATAMOVR</td>
<td>62</td>
</tr>
<tr>
<td>DS1DSCHA</td>
<td>63</td>
</tr>
<tr>
<td>DEALLOC</td>
<td>63</td>
</tr>
<tr>
<td>DEBUG_ERROR</td>
<td>63</td>
</tr>
<tr>
<td>DEBUG_SDUMP</td>
<td>63</td>
</tr>
<tr>
<td>DFDSS_ADMIN</td>
<td>63</td>
</tr>
<tr>
<td>DFDSS_CC</td>
<td>64</td>
</tr>
<tr>
<td>DFDSS_OP</td>
<td>64</td>
</tr>
<tr>
<td>DIFF</td>
<td>64</td>
</tr>
<tr>
<td>DIFFDSN</td>
<td>64</td>
</tr>
<tr>
<td>DMIDCAMS</td>
<td>64</td>
</tr>
<tr>
<td>EATTR</td>
<td>64</td>
</tr>
<tr>
<td>EATTR</td>
<td>64</td>
</tr>
<tr>
<td>EMCALLOC_TRACE</td>
<td>65</td>
</tr>
<tr>
<td>EMCCOPY</td>
<td>65</td>
</tr>
</tbody>
</table>
VALIDATE........................................................................................... 89
VALFIRST ........................................................................................... 89
VALLAST ........................................................................................... 90
VALRANGE_LOCAL ........................................................................... 90
VALRANGE_REMOTE ........................................................................ 90
VALSMS ............................................................................................. 90
VARYOFF ........................................................................................... 91
VARYON ............................................................................................. 91
VCLOSE ............................................................................................. 91
VDEV_REUSE .................................................................................... 91
VDEVWAIT ......................................................................................... 92
VERIFY_OPEN_SOURCE ................................................................... 92
VERIFY ............................................................................................... 92
VSAMENQ .......................................................................................... 92
VSAMFAIL .......................................................................................... 92
VTOCIX .............................................................................................. 92
WAIT_OFFLINE_LIMIT ....................................................................... 92
WAIT_ONLINE_LIMIT ........................................................................ 93
WAIT_PRECOPY ................................................................................ 93
WAIT .................................................................................................. 93
WDEFINE ............................................................................................ 93
XTNTBNDRY ...................................................................................... 93

Chapter 4 Operations and Examples
Specifying devices ................................................................................... 96
Defining a group of statements ................................................................ 97
  Specifying the group dataset .............................................................. 97
  Group dataset allocation example ....................................................... 98
  Defining the groups ........................................................................... 98
  Referencing other groups from within a group .................................... 99
Using groups as arguments to TimeFinder commands .......................... 100
Performing a SNAP VOLUME copy ......................................................... 101
  Options to consider ........................................................................... 101
Important points ............................................................................... 102
  Thick and thin device support ........................................................... 103
  Extended address volumes ............................................................... 104
  Diskless SRDF devices ...................................................................... 104
  Offline volume support ..................................................................... 104
  Full-device resnap operations ............................................................ 104
  Incremental clone restore ............................................................... 105
  Multidevice operations ..................................................................... 106
  SNAP/FlashCopy coexistence ............................................................ 106
  R21 device recognition .................................................................... 107
  Concurrent R2 (R22) device recognition ........................................... 107
  Security considerations .................................................................... 107
  SNAP VOLUME with the COPYVOLID(NO) parameter ..................... 108
  SNAP VOLUME with the COPYV(N) and CONDVOL(ALL) parameters .. 108
GCM support .................................................................................... 109
Examples .......................................................................................... 109
Performing a SNAP VOLUME using virtual devices ............................ 111
  SNAP VDEVICE example .................................................................. 112
Performing Cascaded clone operations ............................................... 114
  Influencing SMS volume selection .................................................. 117
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTENT_ALLOCATION</td>
<td>117</td>
</tr>
<tr>
<td>Internal selection</td>
<td>117</td>
</tr>
<tr>
<td>Set and query volume preferencing</td>
<td>118</td>
</tr>
<tr>
<td>Selecting volume processing by phases</td>
<td>119</td>
</tr>
<tr>
<td>Presnap processing</td>
<td>119</td>
</tr>
<tr>
<td>Activate processing</td>
<td>120</td>
</tr>
<tr>
<td>Postnap processing</td>
<td>120</td>
</tr>
<tr>
<td>Phase processing and group status</td>
<td>120</td>
</tr>
<tr>
<td>Examples</td>
<td>121</td>
</tr>
<tr>
<td>Performing a SNAP DATASET copy</td>
<td>122</td>
</tr>
<tr>
<td>Supported dataset types</td>
<td>122</td>
</tr>
<tr>
<td>Unsupported dataset types</td>
<td>123</td>
</tr>
<tr>
<td>SNAP DATASET options and operations</td>
<td>123</td>
</tr>
<tr>
<td>Snaps from offline or cloned volumes</td>
<td>136</td>
</tr>
<tr>
<td>Snapping to GSPACE datasets</td>
<td>138</td>
</tr>
<tr>
<td>Cascaded operations</td>
<td>138</td>
</tr>
<tr>
<td>Security considerations</td>
<td>138</td>
</tr>
<tr>
<td>Examples</td>
<td>138</td>
</tr>
<tr>
<td>Performing a Parallel Snap</td>
<td>142</td>
</tr>
<tr>
<td>Parallel Snap software requirements</td>
<td>142</td>
</tr>
<tr>
<td>Parallel Snap operations</td>
<td>142</td>
</tr>
<tr>
<td>Invoking Parallel Snap</td>
<td>143</td>
</tr>
<tr>
<td>Performing queries</td>
<td>145</td>
</tr>
<tr>
<td>QUERY GROUP display example</td>
<td>145</td>
</tr>
<tr>
<td>Remote QUERY VOLUME example</td>
<td>146</td>
</tr>
<tr>
<td>Cleaning up volumes</td>
<td>146</td>
</tr>
<tr>
<td>Remote CLEANUP example</td>
<td>146</td>
</tr>
<tr>
<td>Using SRDF/A R2 Wait for Precopy</td>
<td>147</td>
</tr>
<tr>
<td>Viewing GCM status</td>
<td>148</td>
</tr>
</tbody>
</table>

## Chapter 5

### Command Reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventions</td>
<td>150</td>
</tr>
<tr>
<td>Syntax conventions</td>
<td>150</td>
</tr>
<tr>
<td>Traditional TimeFinder commands</td>
<td>151</td>
</tr>
<tr>
<td>Customer task guide for traditional TF commands</td>
<td>151</td>
</tr>
<tr>
<td>GPM commands</td>
<td>152</td>
</tr>
<tr>
<td>Common parameters</td>
<td>153</td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>214</td>
</tr>
<tr>
<td>CLEANUP [EXTENT TRACK ON]</td>
<td>217</td>
</tr>
<tr>
<td>CONFIG (TF/Clone)</td>
<td>217</td>
</tr>
<tr>
<td>DEFINE SOURCE_VOLUME_LIST (TF/Clone)</td>
<td>220</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>220</td>
</tr>
<tr>
<td>QUERY DATASET (TF/Clone)</td>
<td>226</td>
</tr>
<tr>
<td>QUERY GLOBAL</td>
<td>259</td>
</tr>
<tr>
<td>QUERY GROUP</td>
<td>260</td>
</tr>
<tr>
<td>QUERY VDEVICE (TF/Snap)</td>
<td>260</td>
</tr>
<tr>
<td>QUERY VOLUME</td>
<td>263</td>
</tr>
<tr>
<td>RESTORE VOLUME (TF/Snap)</td>
<td>263</td>
</tr>
<tr>
<td>SNAP DATASET (TF/Clone)</td>
<td>273</td>
</tr>
<tr>
<td>SNAP VOLUME</td>
<td>280</td>
</tr>
<tr>
<td>STOP SNAP TO DATASET (TF/Clone)</td>
<td>300</td>
</tr>
<tr>
<td>STOP SNAP TO VOLUME</td>
<td>314</td>
</tr>
<tr>
<td>Grouping commands (TF/Snap)</td>
<td>315</td>
</tr>
<tr>
<td>DEFINE GROUP</td>
<td>320</td>
</tr>
</tbody>
</table>
# FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parallel clone</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>Clone restore virtual snaps (CRVS)</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>VP Snap Restore to Target (VRRT)</td>
<td>31</td>
</tr>
<tr>
<td>4</td>
<td>Incremental refresh/resnap of clone</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>SNAP VOLUME using virtual devices</td>
<td>111</td>
</tr>
<tr>
<td>6</td>
<td>Cascaded clone</td>
<td>114</td>
</tr>
<tr>
<td>7</td>
<td>Cascaded clone to cascaded clone emulation</td>
<td>115</td>
</tr>
<tr>
<td>8</td>
<td>Cascaded clone emulation to cascaded clone</td>
<td>115</td>
</tr>
<tr>
<td>9</td>
<td>Cascaded clone emulation to cascaded clone emulation</td>
<td>115</td>
</tr>
<tr>
<td>10</td>
<td>Parallel Snap operation</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>DOIO error code format</td>
<td>325</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Protection session limits</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>EMCSNAPO site options</td>
<td>47</td>
</tr>
<tr>
<td>3</td>
<td>Phase processing and group status</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>Summary of multivolume SNAP DATASET scenarios</td>
<td>127</td>
</tr>
<tr>
<td>5</td>
<td>Snapping non-VSAM datasets</td>
<td>131</td>
</tr>
<tr>
<td>6</td>
<td>Snapping VSAM datasets</td>
<td>131</td>
</tr>
<tr>
<td>7</td>
<td>VSAM dataset share options and the VSAMENQMODE</td>
<td>134</td>
</tr>
<tr>
<td>8</td>
<td>DFDSS masking rules</td>
<td>135</td>
</tr>
<tr>
<td>9</td>
<td>Customer task guide for traditional TF commands</td>
<td>151</td>
</tr>
<tr>
<td>10</td>
<td>REPLACE and REUSE effects</td>
<td>193</td>
</tr>
<tr>
<td>11</td>
<td>Command actions when SOFTlink(YES) is specified</td>
<td>197</td>
</tr>
<tr>
<td>12</td>
<td>Global parameters</td>
<td>226</td>
</tr>
<tr>
<td>13</td>
<td>Effect of GLOBAL DEBUG</td>
<td>238</td>
</tr>
<tr>
<td>14</td>
<td>Abend codes</td>
<td>325</td>
</tr>
<tr>
<td>15</td>
<td>PowerMax/VMAX interface error codes</td>
<td>326</td>
</tr>
<tr>
<td>16</td>
<td>EXTENTS error codes</td>
<td>328</td>
</tr>
<tr>
<td>17</td>
<td>EMCDSSU parameters</td>
<td>331</td>
</tr>
<tr>
<td>18</td>
<td>SNSMFRC - STANDARD SMF RECORD HEADER</td>
<td>336</td>
</tr>
<tr>
<td>19</td>
<td>SNSMFCMN - COMMON SEGMENT PORTION</td>
<td>337</td>
</tr>
<tr>
<td>20</td>
<td>SNSMFID - IDENTIFICATION SEGMENT</td>
<td>337</td>
</tr>
<tr>
<td>21</td>
<td>SNSMFREQ - REQUEST INFORMATION</td>
<td>337</td>
</tr>
<tr>
<td>22</td>
<td>SNSMFOPT - OPTIONS INFORMATION</td>
<td>339</td>
</tr>
<tr>
<td>23</td>
<td>SNSMFSRC - SOURCE DATASET INFORMATION</td>
<td>349</td>
</tr>
<tr>
<td>24</td>
<td>SNSMFTGT - TARGET DATASET INFORMATION</td>
<td>350</td>
</tr>
<tr>
<td>25</td>
<td>SNSMFGTK - GATEKEEPER INFORMATION</td>
<td>350</td>
</tr>
<tr>
<td>26</td>
<td>SNSMFSXL - SOURCE EXTENT LIST</td>
<td>351</td>
</tr>
<tr>
<td>27</td>
<td>SNSMFTXL - TARGET EXTENT LIST</td>
<td>351</td>
</tr>
<tr>
<td>28</td>
<td>SNSMFCXL - COPY EXTENT LIST</td>
<td>352</td>
</tr>
<tr>
<td>29</td>
<td>SMF Record sub-sections and TimeFinder actions</td>
<td>354</td>
</tr>
<tr>
<td>30</td>
<td>SMF Record sub-sections and TimeFinder actions</td>
<td>354</td>
</tr>
<tr>
<td>31</td>
<td>SMF Record sub-sections and TimeFinder actions</td>
<td>366</td>
</tr>
<tr>
<td>32</td>
<td>REXX TF command parameters</td>
<td>369</td>
</tr>
<tr>
<td>33</td>
<td>ESNAACTV arguments</td>
<td>371</td>
</tr>
<tr>
<td>34</td>
<td>ESNANEWD arguments</td>
<td>372</td>
</tr>
<tr>
<td>35</td>
<td>ESNASCRA arguments</td>
<td>373</td>
</tr>
<tr>
<td>36</td>
<td>SMS Class Name Validation arguments</td>
<td>374</td>
</tr>
<tr>
<td>37</td>
<td>TDEV Pool Name Validation arguments</td>
<td>375</td>
</tr>
<tr>
<td>38</td>
<td>VDEV Pool Name Validation arguments</td>
<td>375</td>
</tr>
<tr>
<td>39</td>
<td>VARY Device Online/Offline arguments</td>
<td>376</td>
</tr>
<tr>
<td>40</td>
<td>Write SMF arguments</td>
<td>377</td>
</tr>
</tbody>
</table>
Tables
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 the following Dell EMC products:
- TimeFinder/Clone for z/OS
- TimeFinder/Snap for z/OS
- TimeFinder/Consistency Group

Coverage

This document describes TimeFinder/Clone Mainframe Snap Facility for z/OS when used in the following operating environments:
- PowerMaxOS 5978
- HYPERMAX OS 5977
- Enginuity 5876
- Enginuity 5773

Note: See prior versions of the TimeFinder/Clone Mainframe Snap Facility Product Guide for information pertaining to other Enginuity levels.

Audience

This guide is intended for the host system administrator, system programmer, or operator who is evaluating, planning for, managing, or using Dell EMC TimeFinder/Clone Mainframe Snap Facility.

1. Enginuity 5773 is not supported with Mainframe Enablers 8.3. For Mainframe Enablers 8.0, 8.1, and 8.2, Enginuity 5773 is not supported in SRDF configurations that include a storage system running PowerMaxOS 5978 or HYPERMAX OS 5977.
Related documentation

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


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 VMAX3 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 type style conventions in this document:

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

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

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

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

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

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

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

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

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

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

Where to get help

Product information

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

Technical support

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

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

eLicensing support

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

For help with missing or incorrect entitlements after activation (that is, expected functionality remains unavailable because it is not licensed), contact your Dell EMC Account Representative or Authorized Reseller.
For help with any errors applying license files through Solutions Enabler, contact Dell EMC Customer Support.

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

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

Your comments

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

VMAXContentFeedback@emc.com
CHAPTER 1
Introduction

This chapter covers the following topics:

- Introduction to TimeFinder ................................................................. 20
- TimeFinder/Clone .................................................................................. 22
- TimeFinder/Snap ................................................................................... 28
- TimeFinder/Consistency Group .............................................................. 34
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 the TimeFinder/Clone Mainframe Snap Facility, which is a TimeFinder product that supports the z/OS mainframe environment as one of the components of the Dell EMC Mainframe Enablers.

**Note:** For a comprehensive description of TimeFinder SnapVX, see the *TimeFinder SnapVX and zDP 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 your 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 Mainframe Enablers kit, you install the software for all the components.

---

\(^1\) zDP requires TimeFinder SnapVX but is a separately licensed product.
**TimeFinder/Clone Mainframe Snap Facility**

TimeFinder/Clone Mainframe Snap Facility is the software foundation for four functional products:

- TimeFinder SnapVX

  **Note:** The *TimeFinder SnapVX and zDP Product Guide* describes TimeFinder SnapVX.

- TimeFinder/Clone

- TimeFinder/Snap

- TimeFinder/Consistency Group¹

TimeFinder/Clone Mainframe Snap Facility consists of common code and specific code for each of the functional products.

**IMPORTANT**

In the documentation, any information that specifically applies to one of these functional products is labeled for that product. Any information that applies to all of the functional products and the common code uses the term “TimeFinder” for TimeFinder/Clone Mainframe Snap Facility.

**Licensing**

See the following documents for information about licensing:

- *Mainframe Enablers Installation and Customization Guide*

  **Note:** Since Mainframe Enablers 8.3 do not support Enginuity 5773, for information about licenses required under Enginuity 5773 see the *Mainframe Enablers Installation and Customization Guide* for Mainframe Enablers 8.0, 8.1, or 8.2.

- *PowerMax Family Product Guide*

- *VMAX All Flash Product Guide*

- *VMAX3 Family Product Guide*

- *VMAX Family Product Guide*

---

¹. TimeFinder/Consistency Group should not be confused with Consistency Groups for z/OS. They are separate products. “TimeFinder/Consistency Group” on page 34 describes TF/Consistency Group. The *Consistency Groups for z/OS Product Guide* describes Consistency Groups for z/OS.
Introduction

TimeFinder/Clone

TimeFinder/Clone for z/OS (TF/Clone) produces point-in-time copies of full volumes or of individual datasets. TF/Clone operations involve full volumes or datasets where the amount of data at the source is the same as the amount of data at the target.

**Note:** All TF/Clone commands and syntax are supported with PowerMaxOS 5978 and HYPERMAX OS 5977.

In addition to providing real-time, non-disruptive backup and restore, TF/Clone can compress the cycle time for such processes as:

- Application testing
- Software development
- Loading or updating a data warehouse

TF/Clone also provides significant configuration flexibility because clone copies do not require storage system mirror positions. The clone copies can have any configuration except VDEV (virtual device); that is, they can have any form of RAID protection. The clone target can also be configured as a standard device or as a Business Continuance Volume (BCV).

The source devices can have any configuration except VDEV.

You can use TF/Clone to perform the following operations:

- Dataset snap
- Full-volume snap
- Remote full-volume snap
- Simultaneous (parallel) clone
- VP snaps
- Clone restore virtual snaps (CRVS)

Dataset snap

A dataset snap copies replicas of individual datasets to target datasets. Dataset snap copies the contents of the source dataset to a new or existing target dataset.

**Note:** “SNAP DATASET (TF/Clone)” on page 280 lists the types of datasets that TF/Clone can snap.

As long as the source and target reside on the same storage system, a dataset snap uses the operating environment to copy the dataset. This allows you to initiate the request, and before the copy process is finished, start using the target.

Requests for snaps between two different storage systems can invoke an external datamover to perform the copy. In this case, the target dataset is available when the external copy is completed.
**Full-volume snap**

A full-volume snap captures a complete replica of the source volume on the target volume in the local storage system.

TF/Clone requests that span storage system devices can invoke an external datamover to accomplish the request. This allows volume placement to be flexible without requiring changes to snap jobs. Parameters you place on the command specify how the storage system performs the request.

**Remote full-volume snap**

A remote, full-volume snap captures a complete replica of the source volume on a target volume in a remote storage system that is connected to the local storage system by SRDF links. You initiate remote snap commands in a local z/OS system. A local channel-attached storage system passes the commands on to the remote storage system for execution.

Remote snap requires all storage systems involved be running a supported level of the operating environment. This includes each storage system in the SRDF link.

You identify the volumes to be copied by specifying the PowerMax/VMAX device number (SYMDV#) rather than by specifying the host channel ID (CCUU or UNIT) or the volume label (VOLSER). All volumes must reside in the same storage system and have the same emulation and device geometry (CKD or FBA, 3380 or 3390, and so forth).

A gatekeeper device is required to send commands and provide access to a remote storage system. The gatekeeper device in a local channel-attached storage system must be identified by using the REMOTE parameter (UNIT, VOLUME, or DDNAME subparameters).

Additionally, the “path” from the local channel-attached storage system to the remote storage system must be provided using the REMOTE parameter (RAGROUP subparameter). The path consists of the SRDF group identifiers associated with the SRDF links to the remote storage system. The commands to perform the snap operation are sent down this path to the remote storage system and then executed on the remote storage system.

When there is the possibility of multiple, remote storage systems, it is recommended to also use the REMOTE parameter (CONTROLLER subparameter) to verify that the remote storage system found is the storage system you want.

The target volume cannot be relabeled as part of the snap process.

You can use the CONTROLLER parameter in place of the REMOTE parameter if the remote storage system is only one or two hops away. The CONTROLLER parameter automatically chooses a local device and the SRDF group to be used to reach the remote storage system.
Simultaneous (parallel) clone

PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876 with both synchronous SRDF and TimeFinder/Clone support simultaneous copies of source (R1) volumes and target (R2) to separate volumes in the respective storage systems. This avoids transmission of the clone tracks across the SRDF links. This feature is called parallel clone and is analogous to IBM's remote pair FlashCopy.

SRDF/S creates a synchronous remote image of the production R1 volumes on the R2 volumes.

When conditions are met, a dual clone session is established between the source and target R2 devices, avoiding the secondary SRDF/S transmission of a copied dataset from the target R1 to the corresponding R2 device.

Figure 1 Parallel clone

With Enginuity 5773, SRDF/S mirroring is used to create remote mirrors of the production R1 volumes and their local replicas.

PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876 use TF/Clone remote command support to simultaneously create TF/Clone local replicas on both sides of the SRDF links and thus reduce bandwidth consumption.

Note: This feature supports full-device and extent-level TF/Clone operations in SRDF/S configurations. It can be used in Concurrent SRDF solutions on the SRDF/S branch of the concurrent configuration. Parallel clone operations in SRDF/A configurations are not supported.
Parallel clone ensures that disaster restartability is intact at all times and is allowed on the SNAP DATASET and the SNAP VOLUME statements, and can also be set as a site option or Global parameter using the PARALLEL_CLONE option.

When enabling this feature, the ACTIVATE CONSISTENT(YES) parameter is also required, and if omitted, the simultaneous TF/Clone operation is still performed and the ESNPF37I message is issued.

When using Parallel clone, the following requirements apply:

- PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876 is required on both sides of the SRDF/S link.
- The R2 source and target clone volumes are located in the same storage system.
- The R2 source and target devices cannot be larger than the R1 devices. Currently, TimeFinder blocks this operation.

Parallel clone does not support:

- Cascaded SRDF devices
- SRDF/Star environments
- Configurations where R1 devices are under PowerMaxOS 5978 or HYPERMAX OS 5977 and R2 devices are under Enginuity 5876 or earlier
- Configurations where R1 devices are under Enginuity 5876 or earlier and R2 devices are under PowerMaxOS 5978 or HYPERMAX OS 5977
- FlashCopy
- Virtual Provisioning
- PPRC mode volumes

The following SRDF operations are blocked on Parallel clone devices:

- DELETE and HDELETE
- SWAP and HSWAP
- MOVEPAIR and HMOVEPAIR

**Note:** The PARALLEL_CLONE parameter should not be confused with the Global PARALLEL parameter for multitasking or the Parallel Snap solution.

### VP snaps

A VP snap leverages TF/Clone technology to create space-efficient snaps for thin devices by allowing multiple sessions to share allocations within a thin pool. VP Snap provides the efficiency of Snap technology with improved cache utilization and simplified pool management. With VP Snap, tracks can be stored in the same thin pool as the source, or in another pool of your choice.
VP Snap sessions copy data from the source device to the target device only if triggered by a host I/O. Read I/Os to protected tracks on the target device do not result in data being copied.

**Note:** VP snaps only apply to thin devices with the MODE(VSE) parameter specified for the copy.

**Clone restore virtual snaps (CRVS)**

TF/Clone supports an incremental restore to a source that has active or VP snaps. CRVS only supports a restore to its standard, which is also a virtual/VP Snap source, and does not support a restore to third device.

The following example shows a relationship exists between A and B and another relationship exists between A and C, where C is a VDEV. With CRVS, you now have ability to perform a restore or incremental restore from B to A without terminating the relationship between A and C.

**CRVS requirements/limitations**

The following requirements and limitations apply to CRVS actions:

- Requires Enginuity 5876 on VMAX 40K only.
- CRVS does not support a restore to a third device, such as an additional virtual/VP Snap standard.
- A Resnap of the virtual/VP Snap session is not allowed when CRVS is in progress.
- All other restrictions related to “restore in progress” also apply when CRVS is in progress.
- CRVS is not supported for Clone Emulation.
- Incremental resnap of a clone during a persisting TimeFinder cascaded VP Snap session.
Introduction

- Incremental resnap during a cascaded virtual snap session
- A restore from a TF Cascaded Virtual Snap to a fully copied clone target.

**Additional TF/Clone capabilities**

- Differential snap operations require only the changed data be copied on subsequent snaps.
- Compatibility with STK Snapshot Copy and IBM Snap products including reuse of its SIBBATCH syntax.
- Compatibility with TimeFinder Utility for z/OS, which conditions the catalog by relabeling and recataloging entries and thereby avoids the issues associated with duplicate volume serial numbers in the mainframe environment.
- Compatibility with mainframe security mechanisms such as RACF.
- Integration with many mainframe-specific Independent Software Vendors (ISVs) and their products.

**Working with clone copies**

TF/Clone takes a point-in-time copy of data at the dataset or volume level.

After you issue a TF/Clone command, the TF/Clone process initiates data movement from the source dataset to the target dataset, which is dynamically allocated on a standard or a BCV volume. You can take multiple copies of a dataset. Copies are immediately available for read and write access while the copying process completes as a background task.

Compared to traditional data copying, TF/Clone minimizes the downtime for applications by reducing the time required for copying the data being used by the applications. Replication of the data occurs within the storage system and requires minimum resources from the host.
Introduction

TimeFinder/Snap

TF/Snap produces pointer-based replicas where the preimages of changed data are written, along with changes to the snap device.

Note: All TF/Snap commands and syntax are supported with PowerMaxOS 5978 and HYPERMAX OS 5977.

TF/Snap is a space-saving, snapshot-copy product available in the enterprise storage arena. Because TF/Snap does not actually create a full copy of the source data, its copies take only a fraction of the space a full-volume snap would.

TF/Snap is ideal when you require fast, temporary, parallel access to production data. However, keep in mind that if the source data is destroyed (for example, because of multiple disk failures), the snap will also be lost and cannot be used for the restore.

Therefore, you should not rely on TF/Snap as the sole means of local replica protection if you need absolute availability nor use it with unprotected source volumes as TF/Snap relies on the availability of the source data.

Virtual devices

Virtual devices (VDEVs) are space-efficient copies that consist of tables and pointers to capture a point in time. Virtual devices can be CKD or FBA. Virtual devices are host accessible and do not consume physical storage. However, because they are host addressable, virtual devices do consume PowerMax/VMAX device numbers and host addresses.

TimeFinder supports two virtual device methodologies. The original and default method uses a single session for each virtual device in a relationship with a standard device. This allows a maximum of 8 virtual devices, or 16 sessions, per standard device (four datasets, four full volumes, and 8 virtual volumes).

The multivirtual method (implemented using MULTI_VIRTUAL parameter) uses a single session on the standard device, that allows up to 128 virtual devices to “share” the single device session. Each virtual device is monitored by an independent session that is not associated with the source device.

The MULTI_VIRTUAL parameter is available with Enginuity 5773 and 5876.

With Enginuity 5876, the multivirtual method is the only method used, so whether the MULTI_VIRTUAL parameter is set to NO or to YES, or whether it is used at all, the system always allows 128 virtual device sessions.

With PowerMaxOS 5978 and HYPERMAX OS 5977, the MULTI_VIRTUAL parameter is not supported. Whether set to YES or NO, the system allows up to 32 virtual device sessions for a single standard device. To overcome this limitation, use the SOFTlink parameter.

VSE FBA

Virtual Space Efficient(VSE) FBA is available allowing THIN FBA devices to use shared allocations. The shared allocation makes it a VSE device.

VSE (thin devices using shared allocation) may have up to 32 sessions that do not use the standard 16 clone session positions. This means that you may have a total of 48 sessions present (32 VSE plus the 16 original) at one time. Only 32 sessions may be VSE sessions.
Snap pool devices

Snap pool devices (SNAPDEVs) are log devices. Unlike VDEVs, they are not host accessible; but, do consume physical storage. Snap pool devices are gathered in snap device pools.

Snap device pools (SNAPPOOLs) are named groups of snap pool devices that provide a pool of physical space used to store pre-update images of tracks changed on the source device or new writes to the virtual devices. As specific virtual device sessions are terminated, the space associated with them is returned to free space in the snap device pools, while the actual snap pool devices are saved.

Note: To perform virtual-device snaps and use SNAPPOOLs, you must configure storage systems with virtual and snap pool devices to use TF/Snap.

Basic Snap operations

Upon creating a session, a virtual snap is activated to capture the point-in-time image. As writes arrive at the source volume, the existing tracks or pre-update images are moved to the snap device pool you specify, to preserve the point in time of the snap.

Subsequently, activated virtual devices for the same source go through the same steps of creating pointers to capture the point-in-time image. As long as a virtual device remains active, its responsibility is to point to tracks, on the source or snap pool device.

The consumption of storage in the snap device pool (SNAPPOOL) is determined by the new data change rate on a track-by-track basis. This means that the same track updated several times takes only the space for one preimage. This uses far less space in the snap device pool than the same number of updates affecting completely different tracks.

You can create multiple snap device pools to isolate workloads. This alleviates contention for snap pool device space among several sessions and lessens the possibility of a single session consuming all the space.

SNAPPOOL management

The devices for SNAPPOOL device pools come from a special pool called the DEFAULT_POOL. The DEFAULT_POOL contains snap pool devices that have not been assigned to any named pool, but are available for use.

ResourcePak Base provides a set of General Pool Management (GPM) commands that can be executed online or in batch mode. Devices in pool storage are a predefined set of devices that provide a pool of physical space.

Multiple SNAPPOOLs can be created to isolate workloads. This alleviates contention for device space among several users and lessens the possibility of a single pool consuming all the available space.

Devices can also be removed from a SNAPPOOL, but the devices must first be "drained" and disabled. Drained devices will become inactive within their pool and can then be moved out of the pool to be available for other pools.

The ResourcePak Base for z/OS Product Guide describes the GPM commands and provides a complete description of creating pools and managing the pooling process.
Virtual restore operations

Virtual restore allows you to restore a virtual device. There are two types of virtual restore:

- From the virtual device to the original source of the snap.

  The virtual device session being restored of course might exist, but no other sessions are allowed, as identified previously. The virtual snap device referenced in the restore is removed when the restore occurs and can be used for other purposes.

- From a virtual device to a different standard device (STD).

  The virtual snap device referenced in the restore is removed when the restore occurs. The virtual snap device can be used for other purposes.

Virtual restores have the following limitations:

- For a restore from a VDEV to a clone (or clone emulation) source device, the clone session must be active (SPLIT) with NO tracks to be copied.

- The target of the restore cannot be an active target of any SNAP VOLUME or SNAP DATASET.

- The target of the restore cannot be a clone-emulation BCV as long as the clone-emulation relationship exists.

- With PowerMaxOS 5978 and HYPERMAX OS 5977, the target of the restore can only be the original source of the snap.

These states can be determined using the QUERY VOLUME command, which reports on the type of sessions currently active for the device.
VP Snap restore to copied clone target (VRTT)

TimeFinder provides ability to do a VP Snap (VSE) Restore to a copied Clone Target (VRTT), when the standard (source) of the VP Snap (VSE) target is also a target of a (fully copied) regular native clone.

The following example shows the clone relationship that exists between A and B, and the VP Snap relationship that exists between B and C.

![VP Snap Restore To Target](image)

Figure 3  VP Snap Restore to Target (VRTT)

Previously, you could not restore C to B without terminating the relationship between A and B.

With this VRTT enhancement, the restore of C to B might occur without affecting the relationship between A and B, but A and B must be copied, and all devices must be thin devices.

VRTT requirements and limitations

The following requirements and limitations apply to VRTT actions:

- Requires Enginuity 5876 on VMAX 20K and VMAX 40K platforms.
- MFE does not support this feature on the VMAX 10K platform, but it is supported by the Dell EMC Solutions Enabler product.
- A, B and C are all FBA (B is a clone copy and C is a VSE relationship).
- The A to B clone copy must be completed.
- B cannot own virtual or multivirtual snap sessions (VDEV).
- No other kind of restore to device B is tolerated when the user issues a VRTT.
- When any other restore sessions are in progress, VRTT is rejected.
- Resnap of A to B leg is not allowed if a VRTT session exists. You must terminate the VRTT session before resnaping the A to B leg.
- All other limitations related to VP Snap Restore apply.
Introduction

Incremental clone refresh/resnap

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, you can perform the following incremental clone refresh/resnap operations:

- Incremental refresh/resnap of a clone during a persistent TimeFinder cascaded VP Snap session.
- Incremental refresh/resnap during a cascaded virtual snap session.

These features enable you to do a differential resnap of an intermediate TF/Clone when a cascaded VP Snap session exists. For example, if A->B is a regular clone session, and B->V is an active VP Snap session, then users are allowed to resnap A->B and the operating environment maintains the persistence of the VP Snap copies from these clones.

![Incremental Refresh of Clone](image)

**Figure 4** Incremental refresh/resnap of clone

Incremental clone refresh/resnap requirements and limitations

The following requirements and limitations apply to incremental clone refresh/resnap actions:

- Resnap will not be allowed if B—>C VSE/Snap is not active
- A—>B resnap session can be activated only when the precopy of A—>B completes and the state moves to precopy sync.

Persistent restore operations

Virtual restores cause the virtual device to be removed after the restore. Persistent restore allows you to restore from a virtual device to a standard device, either the original source, a different device, without losing the source virtual device or terminating other snapshots in the session.

You specify persistent restore with the PERSISTENT parameter of the GLOBAL and RESTORE VOLUME commands. The maximum number of virtual devices off a source volume is eight (8). However, persistent restore has a maximum of seven (7).

**Note:** “GLOBAL” on page 226 and “RESTORE VOLUME (TF/Snap)” on page 273 provide more information.
Persistent restores do not require all of the other VDEV sessions (or virtual devices assigned to the restore device) to be terminated.

Persistent virtual restore to a cloned target

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, users can perform a persistent virtual restore (PVR) to a cloned target. When performed, this feature is referred to as PTT and can support a restore to the snap source which is also a clone target.

Planning for virtual device implementations

A planning phase is essential for virtual device snap implementations. Consider the following as part of a virtual snap implementation:

- Both virtual devices and snap pool devices must be configured in the storage system.
- The right number of virtual devices must be configured for the intended use.
- Virtual devices must have the same geometry, track size, and number of cylinders as any source device that is going to be used in virtual snaps.

For example, you can only use a virtual 3390-1 with a real 3390-1. You can only use a virtual 3390-9 with a real 3390-9, and so forth.

- Virtual devices are configured with host addresses (ccuu) and consume PowerMax/VMAX device numbers.
- Snap pool devices are not host addressable but do consume PowerMax/VMAX device numbers.
- Sufficient storage space must be allocated to the snap device pool. This is crucial because if the snap device pool fills up, pre-update images of the newly changed tracks arriving on source volumes that have active virtual device sessions or writes to existing virtual devices are lost. Therefore, point-in-time copies are lost for any source device receiving newly changed tracks or writes to the virtual device if the snap device pool is full.

Dell EMC provides a monitor function within the ResourcePak Base product that advises you if and when you are reaching predetermined thresholds within the snap device pool(s). (All pools can be individually monitored.) User exits can be created based on site-specific policies.

**Note:** The ResourcePak Base for z/OS Product Guide describes the monitor function in detail.
TimeFinder/Consistency Group

Using the Enginuity Consistency Assist (ECA) feature, TF/Consistency Group allows you to perform consistent snap operations on volumes so that the target is dependent-write consistent.¹ TF/Consistency Group is available for full device, virtual device, remote full device and dataset snaps.

The source and target device pairs must reside in the same storage system. Consistency can be preserved over multiple volumes and multiple storage systems.

**Note:** For a consistent dataset snap, you must have PowerMaxOS 5978, HYPERMAX OS 5977, or Enginuity 5876.

Keep in mind the following device type consistency considerations when performing a remote, full-device, consistent snap.

TF/Consistency Group enforces the consistency parameter (CONSISTENT) in conjunction with the ACTIVATE command. The ACTIVATE command applies to SNAP VOLUME and SNAP DATASET commands preceding it in the input stream. Like a full device snap, you first code all of the SNAP statements and then code the ACTIVATE CONSISTENT(YES) statement.

There are a variety of SNAP statements, involving the following types of devices: local, remote (non-SRDF), remote (SRDF/S), and remote (SRDF/A). Consistency among the various types of devices occurs, but may not be what you expect.

For instance, if you have some local and some remote (non-SRDF) devices in the same ACTIVATE group, the remote devices are consistent with each other, but they may not be consistent with the local devices. Likewise, the local devices are consistent with each other, but they may not be consistent with the remote devices. Basically, devices of the same type are consistent with each other, but not with devices of another type.

---

¹ Enginuity Consistency Assist (ECA) is a feature of the storage system operating environment. ECA (often called RDF-ECA, a part of SRDF Consistency) provides an enterprise solution for ensuring dependent-write consistency in SRDF/S configurations with more than one SRDF group. 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.
CHAPTER 2
Getting Started

This chapter describes the main TimeFinder post-installation activities.

- Prerequisites ................................................................. 36
- Running TimeFinder (EMCSNAP) ..................................... 37
- TimeFinder and protection sessions ............................... 38
- TimeFinder Vary processing exit .................................... 40
Prerequisites

Running ResourcePak Base (EMCSCF)

TimeFinder requires that the Mainframe Enablers’ ResourcePak Base (EMCSCF) is installed and running. If EMCSCF is not running, all TimeFinder operations receive the following message:

EMC SCF IS NOT AVAILABLE - reason
IEF450I STON01B EMCSMMP - ABEND=000 U0806 REASON=00000000

Where reason is one of the following:

- SERVICE EMCSAI FAILED
- SERVICE SAICALL FAILED

You can run multiple instances of EMCSCF as separate z/OS sub-systems. You might want to do this when you are testing new versions of EMCSCF or EMCSCF-enabled products. For instructions, see the ResourcePak Base for z/OS Product Guide.

Software interoperability considerations

This section provides methods for avoiding possible interoperability problems between TimeFinder and other software products:

- 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.

- 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

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

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

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

Note: FlashCopy and Snap compatibility is not supported for cascading configurations in which the target device of one technology is used at the source device for the other.
Running TimeFinder (EMCSNAP)

The interface to TimeFinder is through the program EMCSNAP. You normally execute EMCSNAP as a batch job.

The following is example JCL for running EMCSNAP as a batch job:

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

Where:

- The STEPLIB DD statement is optional if you have copied EMCSNAP to a system LINKLIST library.
- The SCF$nnnn statement is optional. It is used to match the batch job to the SCF task that you would like to run against. If you do not specify it, the default SCF subsystem name is SCF$EMC.
- The QCINPUT DD statement can reference a disk file: DISP=SHR, DSN=dsn.
- The QCOUTPUT file contains the summary report that is produced at the end for each run.

```
RQST RC SOURCE TARGET DISP TRACKS EXTENTS
2 04 SNAP VOLUME U6A236  *6EF7*  16695
```

Within that report are the following fields:

- **RQST** Maps back to the STATEMENT# when actions are parsed.
- **RC** Is the return code for that statement.
- **SOURCE** Identifies two things: the action being performed and the entity that the action is being performed upon.
- **TARGET** Identifies any target entity
- **DISP** DISP is used with SNAP DATASET requests to identify whether the target dataset was created or already existed. Additionally, if a command is skipped in processing, it says SKIP.
- **TRACKS** Lists the number of SOURCE tracks being operated upon. In the TRACKS column, the following symbols are used to indicate whether the target dataset has more or less tracks than the source dataset:
  - “>” indicates the target dataset has fewer tracks than the source dataset.
  - “<” indicates the target dataset has more tracks than the source dataset.
  - “ ” (blank - no symbol) indicates the target dataset has the same number of tracks as the source dataset.
- **EXTENTS** The number of SOURCE extents and the number of TARGET extents.

- The QCERROR DD statement (optional) aids in resolving any problems you encounter. When you add QCERROR to the EMCSNAP JCL, any error messages generated are written to this file, as well as to the regular QCOUTPUT file.
TimeFinder and protection sessions

For protection sessions, TimeFinder takes the following steps internally:

- Establishes a protection session for the source dataset or volume
- Copies the tracks with a background process

This allows the target to be immediately available without waiting for the copy process to complete. There is a limit of 64 protection sessions within an SSID, further limited by a maximum of 16 protection sessions for any given device. TimeFinder limits the maximum number of sessions it uses to four per logical device for physical TimeFinder and eight for Virtual Device Snapshot.

Table 1 provides details on protection session limits:

<table>
<thead>
<tr>
<th>Protection sessions allowed</th>
<th>Protection sessions used indirectly</th>
<th>Multivirtual sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native Extents</td>
<td>4</td>
<td>Max 4 combined with native extents (differential)</td>
</tr>
<tr>
<td>Native Extents (Differential)</td>
<td>3</td>
<td>1 SDDF on source and target for each differential session Max 4 combined with native extents</td>
</tr>
<tr>
<td>Full Device Clone</td>
<td>4</td>
<td>1 SDDF on source and target for each differential session</td>
</tr>
<tr>
<td>Native Flash Copy</td>
<td>12</td>
<td>1 SDDF on source and target for each differential session</td>
</tr>
<tr>
<td></td>
<td>Note: For volume clone, only one session can be differential. Differential dataset (extent) sessions are not supported.</td>
<td></td>
</tr>
<tr>
<td>Virtual Device</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Multivirtual Device</td>
<td>1 Required</td>
<td>0</td>
</tr>
<tr>
<td>TF/Mirror (native)</td>
<td>8 (Multi-BCV)</td>
<td>N/A</td>
</tr>
<tr>
<td>TF/Mirror Clone Emulation</td>
<td>6</td>
<td>1 SDDF on source and target for each clone emulation session</td>
</tr>
<tr>
<td>VSE (Virtual Space Efficient)</td>
<td>32</td>
<td>There is a limit of 32 VSE sessions in addition to the existing limit of 16 non-VSE sessions. If differential is used, no more than 16 VSE differential session can be established because the differential SDDF session occupies one of the 16 non-VSE sessions.</td>
</tr>
</tbody>
</table>

Note: With PowerMaxOS 5978 and HYPERMAX OS 5977, SnapVX is used for Full Device Clone and TF/Mirror Clone Emulation.
Keep in mind that TimeFinder’s use of protection sessions is quite efficient at both the volume and the SSID level. When there is no source extent conflict/overlap, TimeFinder uses the same protection session for multiple snap operations for different datasets on the same source volume as well as across multiple volumes in an SSID.

You can perform a full-device resnap operation while there are still protected and indirect tracks present.

When establishing a session for copying a dataset or volume, TimeFinder Clone is using one of these sessions and must coexist with any regular protection session activities taking place. TimeFinder establishes a session, identifies the tracks to be copied, and then terminates, leaving the actual copy process to complete in the background.

Because TimeFinder is no longer running, cleanup of the TimeFinder protection sessions does not take place when the background copy process completes. Instead, the TimeFinder protection sessions remain registered (although idle) until one of the following events occurs:

- A CLEANUP command is issued.
- Another snap involving the same source device is run.
- The TERMINATE_SESSION_WHEN_COMPLETE parameter is specified. (“DELETE GROUP” on page 322 and “SNAP VOLUME” on page 300 provide more information.)

The CLEANUP command is designed to be executed against the source device and to analyze all of the outstanding requests to be copied. Any requests that have completed are removed and any protection sessions no longer needed are released.

Each time TimeFinder initiates a new request for a source device, it automatically performs the CLEANUP command. This might result in additional protection sessions being released for other uses.

Note that if you use TimeFinder against many source devices within a storage system without using the CLEANUP command, a large number of protection sessions can remain registered for a long period of time until you perform the next TimeFinder request or CLEANUP. While this does not affect the storage system, it does impact the availability of the protection sessions for other purposes.
TimeFinder Vary processing exit

The TimeFinder Vary processing exit gives a user exit routine the opportunity to influence the VARY ONLINE and VARY OFFLINE processing that occurs when a complete volume is the target of a SNAP VOLUME request.

The exit routine is called before the VARY command is issued, and can determine whether the VARY command is indeed issued. TimeFinder is an authorized program and the exit routine is authorized when it is invoked. The exit routine should return control to TimeFinder in the same mode as when it was invoked.

The exit must name a CSECT or ENTRY point name of SNAPVARY and must be link edited with the SCFGBLSN load module. A sample exit and the necessaryLKED statements are contained in member SNAPVARY in the Mainframe Enablers SAMPLIB.

Parameters

Five parameters are passed to the exit in a standard parameter list. The parameters are:

1. The address of an eight character field containing the word “ONLINE” or “OFFLINE.” This can be used to determine whether a VARY ONLINE or VARY OFFLINE is about to be performed.

2. Address of a 36-character field containing the command string about to be issued. The actual text is something like:

   ```
   V ccuu, OFFLINE
   ```

   or

   ```
   V ccuu, ONLINE
   ```

   You can modify this field in which case the modified field is used and not the original.

3. Address of a six-character field containing the volser of the device about to be varied offline or online.

4. Address of the mainframe UCB for the device about to be varied offline or online.

5. Address of a 256-byte work area available to the exit.

Return codes

Three return codes are accepted from the exit. The return code values are:

0  TimeFinder is to continue and issue the command in the field pointed to by the second parameter. The contents of the field can be modified by the user exit routine.

4  TimeFinder is to continue, but no command is to be issued. The exit routine is responsible for ensuring the proper device status.

8  TimeFinder is to stop processing the SNAP VOLUME command and the contents of the 36-character field pointed to by the second parameter is written to the message log.
Register contents

Register contents upon entry to the TimeFinder Vary Processing Exit:

R0    Unknown.
R1    Points to a five word parameter list, the fifth word in the list has the VL indicator set.
R2 - R12    Unknown, these registers must be restored by the exit.
R13    Points to an 18-word save area. The first word of the save area is important and must be restored by the exit.
R14    Contains the entry point address of the exit routine.
R15    Contains the return address for the exit routine.
Getting Started
CHAPTER 3
Configuration

This chapter covers the following topics:

- TimeFinder configuration layers ................................................................. 44
- Editing the EMCSNAPO macro ................................................................. 46
- EMCSNAPO site options ........................................................................ 47
TimeFinder 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.

For Mainframe Enablers 8.0 and later, 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. See “TimeFinder REXX EXITS” on page 367 for details.

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 47 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.
Example In this example, GLOBAL parameter.x applies to both the SNAP DATASET and SNAP VOLUME commands, because it precedes both. GLOBAL parameter.y applies only to SNAP VOLUME because it comes after SNAP DATASET, but before SNAP VOLUME.

```bash
//QCINPUT DD *
GLOBAL parameter.x
SNAP DATASET
GLOBAL parameter.y
SNAP VOLUME
//
```

In one job step, a GLOBAL command can override a preceding GLOBAL command.

Example In the following example, GLOBAL parameter.z applies to SNAP DATASET while GLOBAL Newparameter.z (same parameter, different value) applies to SNAP VOLUME.

```bash
//QCINPUT DD *
GLOBAL parameter.z
SNAP DATASET
GLOBAL Newparameter.z
SNAP VOLUME
//
```

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. To override an EMCSNAPO site option or a GLOBAL parameter, 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.

Example In this example, TimeFinder has a &DATAMOVR site option. This option specifies a default datamover utility to be used for SNAP DATASET and SNAP VOLUME operations. The default value for &DATAMOVR is “None.” However, you can set &DATAMOVR to the name of the datamover you normally want to use.

The matching GLOBAL parameter is DataMoverNaMe. If you set DataMoverNaMe to a different value than you set &DATAMOVR, the value you use overrides the &DATAMOVR site option for all the commands that follow in the job step.

If you then want to change the datamover again for a specific SNAP DATASET command, you can specify DATAMOVERNAME as an argument to that SNAP DATASET command. TimeFinder uses the datamover you specify for that SNAP DATASET operation.

After that SNAP DATASET is complete, TimeFinder then uses the GLOBAL specification for the rest of the commands in the job step. After the job step completes, TimeFinder returns to the set (or default) value of &DATAMOVR.
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.

**WARNING**

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

Before:

```
EMCSNAPO DSECT=NO
```

After:

```
EMCSNAPO AUTODEAL=NO, DSECT=NO
```

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

For example:

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

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

Summary

Table 2 lists the EMCSNAPO site options and their default values together with the corresponding GLOBAL parameters.

<table>
<thead>
<tr>
<th>#</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>54</td>
<td>ACT_SCF_GATEKEEPER</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>54</td>
<td>ACTIVATE_SUBTASK#</td>
<td>3</td>
<td>0-255</td>
<td>ACTIVATE_SUBTASK#</td>
</tr>
<tr>
<td>55</td>
<td>ADMIN</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>ALLOFAIL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>ALLOSEQ</td>
<td>DSTYPE</td>
<td>DSTYPE</td>
<td>SIZE</td>
</tr>
<tr>
<td>55</td>
<td>ALLOUNIT</td>
<td>SYSALLODA</td>
<td>SYSALLODA</td>
<td>ALLOCATION_UNITNAME</td>
</tr>
<tr>
<td>55</td>
<td>ALLOWCANCEL_LOCKED</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>ALLOW_FBA_META</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>ALLOW_SYMDV#</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>ALUNUSED</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>AUTOACTIVATE</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>55</td>
<td>AUTO_BIND_TDEV</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>AUTOCLN</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>AUTODEAL</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>AUTORLSE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>AUTO_UNBIND_TDEV</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>AUTOXPND</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>BACKGRND</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>57</td>
<td>BCVONLY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>58</td>
<td>CACHESYM</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>58</td>
<td>CATALOG</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>58</td>
<td>CHECKBCV</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>58</td>
<td>CHKONLIN</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>58</td>
<td>CLEAN_R2</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>58</td>
<td>CLEANDIFF</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
### Table 2: EMCSNAPO site options (page 2 of 7)

<table>
<thead>
<tr>
<th>#</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>58</td>
<td>CMPLT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>59</td>
<td>CMPLTMSG</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>59</td>
<td>COLLAPSE</td>
<td>n/a</td>
<td>VSAM</td>
<td>NOVSAM</td>
</tr>
<tr>
<td>59</td>
<td>COMPACT_MISMATCH</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>59</td>
<td>CONDVOL</td>
<td>ALL</td>
<td>ALL</td>
<td>DUMP</td>
</tr>
<tr>
<td>59</td>
<td>CONGROUP</td>
<td>IGNORE</td>
<td>IGNORE</td>
<td>REQUIRED_SAME</td>
</tr>
<tr>
<td>60</td>
<td>CONGROUP_LDMP</td>
<td>IGNORE</td>
<td>IGNORE</td>
<td>REQUIRED_SAME</td>
</tr>
<tr>
<td>61</td>
<td>CONSALL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>CONSIST</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>CONSVOL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>COPYCYL</td>
<td>10</td>
<td>number of I/O operations</td>
<td>COPYCYL COUNT</td>
</tr>
<tr>
<td>61</td>
<td>COPYFAIL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>COPYVOL</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>CSMSDATA</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>CSMSMGR</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>61</td>
<td>CSMSSTOR</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>62</td>
<td>DATACLASS</td>
<td>n/a</td>
<td>classname</td>
<td>DATACLASS</td>
</tr>
<tr>
<td>62</td>
<td>DATAMOVER</td>
<td>n/a</td>
<td>datamover_name</td>
<td>DATAMOVERNAME</td>
</tr>
<tr>
<td>63</td>
<td>DS1DSCHA</td>
<td>LEAVE</td>
<td>SET</td>
<td>RESET</td>
</tr>
<tr>
<td>63</td>
<td>DEALLOC</td>
<td>DEALLOC</td>
<td>taskname</td>
<td>DEALLOC</td>
</tr>
<tr>
<td>63</td>
<td>DEBUG_ERROR</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>63</td>
<td>DEBUG_SDUMP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>63</td>
<td>DFDSS_ADMIN</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>64</td>
<td>DFDSS_CC</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>64</td>
<td>DFDSS_OP</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>64</td>
<td>DIFF</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>64</td>
<td>DIFFDSN</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>#</td>
<td>Site option (without the macro “&amp;” designation)</td>
<td>Site option default</td>
<td>Site option valid values</td>
<td>Site option name in QUERY GLOBAL output</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------</td>
<td>---------------------</td>
<td>-------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>64</td>
<td>DMIDCAMS</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>65</td>
<td>EATTR</td>
<td>n/a</td>
<td>NO</td>
<td>OPT</td>
</tr>
<tr>
<td>66</td>
<td>EMCCOPY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>66</td>
<td>EMCDSSU_FLASH_SNAP</td>
<td>SNAP</td>
<td>SNAP</td>
<td>FLASHCOPY</td>
</tr>
<tr>
<td>66</td>
<td>EMCDSSU_TARGET</td>
<td>MATCH</td>
<td>MATCH</td>
<td>IGNORE</td>
</tr>
<tr>
<td>67</td>
<td>EMCONLY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>68</td>
<td>EMUL_TYPE</td>
<td>ALL</td>
<td>ALL</td>
<td>HARDLINK</td>
</tr>
<tr>
<td>67</td>
<td>ENQFAIL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>67</td>
<td>ENQSCOPE</td>
<td>REQUEST</td>
<td>REQUEST</td>
<td>STEP</td>
</tr>
<tr>
<td>67</td>
<td>ENQWAIT</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>67</td>
<td>ERRCHK</td>
<td>NORMAL</td>
<td>NORMAL</td>
<td>REDUCED</td>
</tr>
<tr>
<td>68</td>
<td>ERRDISP</td>
<td>DELETE</td>
<td>DELETE</td>
<td>KEEP</td>
</tr>
<tr>
<td>68</td>
<td>ERRREC</td>
<td>NORMAL</td>
<td>NORMAL</td>
<td>ENHANCED</td>
</tr>
<tr>
<td>68</td>
<td>ESNP119</td>
<td>WARNING</td>
<td>WARNING</td>
<td>ERROR</td>
</tr>
<tr>
<td>68</td>
<td>ESNP220</td>
<td>ERROR</td>
<td>ERROR</td>
<td>WARNING</td>
</tr>
<tr>
<td>68</td>
<td>ESNP231E</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>69</td>
<td>EXAMINE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>69</td>
<td>EXPATHGRP</td>
<td>n/a</td>
<td>pathlist</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>EXPLAIN_VOL_SEL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>69</td>
<td>EXTADDNEW</td>
<td>n/a</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>69</td>
<td>EXTALLOC</td>
<td>n/a</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>70</td>
<td>EXTENDED_MISMATCH</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>70</td>
<td>EXTXPAND</td>
<td>n/a</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>70</td>
<td>EXTXPVOL</td>
<td>n/a</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>71</td>
<td>FBA</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
<td>INCLUDE</td>
</tr>
<tr>
<td>71</td>
<td>FLASH_SNAP</td>
<td>SNAP</td>
<td>FLASHCOPY</td>
<td>SNAP</td>
</tr>
<tr>
<td>71</td>
<td>FORCE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>71</td>
<td>FORCECMP</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>71</td>
<td>FREESPC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
### Table 2: EMCSNAPO site options (page 4 of 7)

<table>
<thead>
<tr>
<th>#</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>71</td>
<td>FULL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>71</td>
<td>GROUP_DEVICE_READY_STATE</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
<tr>
<td>72</td>
<td>GROUP_DSNAMEN/A</td>
<td>dataset name</td>
<td>GROUP_DSNAMEN(LOCAL)</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>GROUP_EMCQCAPI_VERIFY</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>72</td>
<td>GROUP_HISTORY_LIMIT</td>
<td>100</td>
<td>number of records</td>
<td>GROUP_HISTORY_LIMIT</td>
</tr>
<tr>
<td>72</td>
<td>HOSTCOPYMODE</td>
<td>OLD</td>
<td>OLD</td>
<td>SHARED</td>
</tr>
<tr>
<td>72</td>
<td>IGNORERDF</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>72</td>
<td>INVALIDATE_PDSE</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>73</td>
<td>MAXDSSU</td>
<td>10</td>
<td>number of asids</td>
<td>MAXDSSU</td>
</tr>
<tr>
<td>73</td>
<td>MAXTASK2</td>
<td>999</td>
<td>number of tasks</td>
<td>MAXTASK(ALL)</td>
</tr>
<tr>
<td>73</td>
<td>MAXTASKR</td>
<td>99</td>
<td>number of tasks</td>
<td>MAXTASK(REQ)</td>
</tr>
<tr>
<td>73</td>
<td>MESSAGE</td>
<td>NONE</td>
<td>DISPLAY</td>
<td>PROMPT</td>
</tr>
<tr>
<td>73</td>
<td>MGMTCLASS</td>
<td>n/a</td>
<td>classname</td>
<td>MANAGEMENTCLASS</td>
</tr>
<tr>
<td>74</td>
<td>MIGRATRC</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>74</td>
<td>MINSNAP</td>
<td>n/a</td>
<td>number of tracks</td>
<td>MINSNAP</td>
</tr>
<tr>
<td>74</td>
<td>MLQ</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>74</td>
<td>MRGEXIST</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>74</td>
<td>MULTI_VIRTUAL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>75</td>
<td>NTFYGRPN/A</td>
<td>groupname</td>
<td>NOTIFY GROUP</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>NTFYLVL</td>
<td>n/a</td>
<td>DATASET</td>
<td>JOB</td>
</tr>
<tr>
<td>75</td>
<td>OFFLINE</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>76</td>
<td>OPEN_SOURCE_RC</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>76</td>
<td>OPT_CKD</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>76</td>
<td>OPT_FBA</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>77</td>
<td>OPT_NOTREADY</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>77</td>
<td>OPT_RAID</td>
<td>ALL</td>
<td>ALL</td>
<td>N</td>
</tr>
<tr>
<td>77</td>
<td>OPT_READY</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>#</td>
<td>Site option (without the macro “&amp;” designation)</td>
<td>Site option default</td>
<td>Site option valid values</td>
<td>Site option name in QUERY GLOBAL output</td>
</tr>
<tr>
<td>----</td>
<td>------------------------------------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>78</td>
<td>OPT_SAVEDEV</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>78</td>
<td>OPT_TDEV</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>78</td>
<td>OPT_THINPOOL</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>79</td>
<td>PARALLEL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>79</td>
<td>PARALLEL_CLONE</td>
<td>n/a</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>79</td>
<td>PERSIST</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>79</td>
<td>POOL</td>
<td>n/a</td>
<td>poolname</td>
<td>POOL</td>
</tr>
<tr>
<td>79</td>
<td>POOLUSE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>80</td>
<td>PRECOPY</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>80</td>
<td>PREPARE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>80</td>
<td>PROCESS_COPYCYL_DATAMOVER</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>80</td>
<td>PURGE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>80</td>
<td>QCAPIMSG</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>R1FULLCOPY</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>R1R2SYNC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>RECALC_FREE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>RECALL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>REFTOC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>81</td>
<td>REMOVE_REMOTE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>82</td>
<td>REPLACE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>82</td>
<td>RESERVE</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>82</td>
<td>RESERVE.Serialization</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>82</td>
<td>RETAIN_SOURCE_REFDT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>82</td>
<td>RETRY1731</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>83</td>
<td>RETRY1756</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>83</td>
<td>REUSE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>83</td>
<td>REUSFAIL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>83</td>
<td>SAMEONLY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>84</td>
<td>SAVEFULL</td>
<td>READY</td>
<td>READY</td>
<td>NOTREADY</td>
</tr>
<tr>
<td>84</td>
<td>SCRATCH</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
### Table 2 EMCSNAPO site options (page 6 of 7)

<table>
<thead>
<tr>
<th>#</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>84</td>
<td>SESSDETL</td>
<td>NODETAIL</td>
<td>DETAIL</td>
<td>NODETAIL</td>
</tr>
<tr>
<td>84</td>
<td>SESSDIFF</td>
<td>NODIFF</td>
<td>DIFF</td>
<td>NODIFF</td>
</tr>
<tr>
<td>85</td>
<td>SESSLIST</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>85</td>
<td>SMFRID</td>
<td>0</td>
<td>0</td>
<td>128-255</td>
</tr>
<tr>
<td>85</td>
<td>SMSKSDS</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>85</td>
<td>SMSPASSVOL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>86</td>
<td>SNAPSHOT_LIST</td>
<td>ALL</td>
<td>ALL</td>
<td>LINKED</td>
</tr>
<tr>
<td>86</td>
<td>SNAPSHOT_NAME</td>
<td>snapshot_name</td>
<td>NAME</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>SNAPUNUSED</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>86</td>
<td>SOFTLINK</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>86</td>
<td>SRDFAR1</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>86</td>
<td>SRDFAR2</td>
<td>WARNING</td>
<td>WARNING</td>
<td>R1R2SYNC</td>
</tr>
<tr>
<td>87</td>
<td>SRDFAR2_PRECOPY</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>87</td>
<td>SRDFA_RETRY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>87</td>
<td>SRDFSR1</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>87</td>
<td>STORCLAS</td>
<td>n/a</td>
<td>classname</td>
<td>STORAGECLASS</td>
</tr>
<tr>
<td>87</td>
<td>STORED_LOG_SIZE</td>
<td>25000</td>
<td>number of lines</td>
<td>STORED_LOG_SIZE</td>
</tr>
<tr>
<td>88</td>
<td>SUBTNAME</td>
<td>EMCSNAPO</td>
<td>subtask name</td>
<td>SUBTASK NAME</td>
</tr>
<tr>
<td>88</td>
<td>SYSCALL_RETRY</td>
<td>1600</td>
<td>count</td>
<td>SYSCALL_RETRY</td>
</tr>
<tr>
<td>88</td>
<td>TARGET_WAIT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>88</td>
<td>TDEV_RECLAIM</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>88</td>
<td>TERMINSESS</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>89</td>
<td>TIMEOUT</td>
<td>15</td>
<td>0-128</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>89</td>
<td>TRKALIGN</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>89</td>
<td>TRUNC</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>89</td>
<td>VALIDATE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>89</td>
<td>VALFIRST</td>
<td>0</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>90</td>
<td>VALLAST</td>
<td>n/a</td>
<td>number of tracks</td>
<td>VALIDATE(LAST)</td>
</tr>
<tr>
<td>90</td>
<td>VALRANGE_LOCAL</td>
<td>AUTO</td>
<td>AUTO</td>
<td>IGNORE</td>
</tr>
<tr>
<td>90</td>
<td>VALRANGE_REMOTE</td>
<td>AUTO</td>
<td>AUTO</td>
<td>IGNORE</td>
</tr>
</tbody>
</table>
### Table 2  EMCSNAPO site options  (page 7 of 7)

<table>
<thead>
<tr>
<th>#</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>90</td>
<td>VALSMS</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>91</td>
<td>VARYOFF</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
<tr>
<td>91</td>
<td>VARYON</td>
<td>AUTO</td>
<td>AUTO</td>
<td>NEVER</td>
</tr>
<tr>
<td>91</td>
<td>VCLOSE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>91</td>
<td>VDEV_REUSE</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>VDEVWAIT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>VERIFY_OPEN_SOURCE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>VERIFY</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>VSAMENQ</td>
<td>NONE</td>
<td>OLD</td>
<td>SHARED</td>
</tr>
<tr>
<td>92</td>
<td>VSAMFAIL</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>VTOCIX</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>92</td>
<td>WAIT_OFFLINE_LIMIT</td>
<td>5</td>
<td>number of seconds</td>
<td>WAIT_OFFLINE_TIME</td>
</tr>
<tr>
<td>93</td>
<td>WAIT_ONLINE_LIMIT</td>
<td>5</td>
<td>number of seconds</td>
<td>WAIT_ONLINE_TIME</td>
</tr>
<tr>
<td>93</td>
<td>WAIT_PRECOPY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>93</td>
<td>WAIT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>93</td>
<td>WFDEF</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>93</td>
<td>XTENDBNDRY</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

a.  For Mainframe Enablers 8.1, 8.2, 8.3  
b.  For Mainframe Enablers 8.4 and later
Configuration

**ACT_SCF_GATEKEEPER**

This option allows you to automatically assign a gatekeeper device from the previously installed and licensed application ResourcePak Base (EMCSCF). EMCSCF is a Mainframe Enablers component and maintains a list of available devices that can be used as gatekeepers.

For detailed information about management, see the *ResourcePak Base for z/OS Product Guide*. For specific sizing recommendations for all storage system configurations, see Knowledgebase article EMC255976 available on Dell EMC Support website.

*Note:* This parameter cannot be used for group processing.

**Syntax**

ACT_SCF_GATEKEEPER=YES|NO

Where:

YES

*(Default)* Assign a gatekeeper from EMCSCF for I/O operations.

NO

Use one of the participating devices in the Snap command.

For example, in this command:

```plaintext
SNAP VOLUME (SOURCE (UNIT(xxx)) TARGET(UNIT(yyy)))
```

If it is followed by an ACTIVATE command, then `xxx` or `yyy` is assigned as the gatekeeper.

If you have five "SNAP VOLUME" statements with an ACTIVATE after them all, you have 10 potential devices to be used as the gatekeeper.

**ACTIVATE_SUBTASK#**

This site option sets the minimum number of storage systems being activated to invoke the subtasking feature. The subtasking feature assigns one subtask for each storage system to minimize the ECA window when multiple syscalls are required.

When the ECA window is opened, the subtasks are posted to perform the ACTIVATE at the same time. As each subtask completes, it posts to the maintask. Once all subtasks have completed, the maintask closes the ECA window, and the subtasks are terminated.

Valid number values are from 0 to 255. Zero turns off the feature. When subtasking is used, there is one subtask attached for each storage system.

**Syntax**

ACTIVATE_SUBTASK#=nnn
Where:

\[ \text{nnn} \]

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

**ADMIN**

See “ADMINISTRATOR(Yes|No)” on page 153.

Syntax

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

**ALLOFAIL**

See “TOLERateALLOcationFailure(Yes|No)” on page 205.

Syntax

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

**ALLOSEQ**

See “ALLOCATION_SEQUENCE(DATASET|NONE|SIZE)” on page 154.

Syntax

\[
\text{ALLOSEQ=} \text{DSNAME | SIZE | NONE}
\]

**ALLOUNIT**

This option assigns a default unit name for the dataset location.

Syntax

\[
\text{ALLOUNIT=} \text{SYSALLDA}
\]

**Note:** SYSALLDA is the default name.

**ALLOW_CANCEL_LOCKED**

This option allows the execution of the CANCEL command when the device lock is held by a QCAPI instruction.

Syntax

\[
\text{ALLOW\_CANCEL\_LOCKED=} \text{YES | NO}
\]

Where:

\[
\text{YES}
\]

(\text{Default}) Allow the CANCEL command even while the device lock is held by a QCAPI instruction.
Configuration

NO

Disable the CANCEL command while the device lock is held in QCAPI instruction.

ALLOW_FBA_META¹

See “ALLOW_FBA_META(Yes|No)” on page 233.

Syntax

ALLOW_FBA_META=YES|NO

ALLOW_SYMDEV#

This option allows you to prevent the SYMDV# parameter from being used. (In some situations where multiple users are sharing the same storage system, using the SYMDV# parameter might be considered a security exposure.)

Syntax

ALLOW_SYMDEV#=YES|NO

Where:

YES

(Default) Allow full usage of the SYMDV# parameter.

NO

The SYMDV# parameter is not allowed.

ALUNUSED

See “ALLOCATE_UNUSED_SPACE(Yes|No)” on page 153.

Syntax

ALUNUSED=YES|NO

AUTOACTIVATE

See “AUTOMATIC_ACTivate(Yes|No)” on page 234.

Syntax

AUTOACTIVATE=YES|NO

AUTO_BIND_TDEV

See “AUTO_BIND_thin_device(Yes|No)” on page 307.

Syntax

AUTO_BIND_TDEV=YES|NO

¹ Available starting with Mainframe Enablers 8.2.
AUTOCLN

See “AUTOMATIC_CLEANup(Yes|No)” on page 154.

Syntax

AUTOCLN=\texttt{YES} | \texttt{NO}

AUTODEAL

See “AUTOMATIC_DEALLOC(Yes|No)” on page 154.

Syntax

AUTODEAL=\texttt{YES} | \texttt{NO}

AUTORLSE

See “AUTOMATIC_RELEASE_hold(Yes|No)” on page 155.

Syntax

AUTORLSE=\texttt{YES} | \texttt{NO}

AUTO_UNBIND_TDEV

See “AUTO_UNBIND_thin_device(Yes|No)” on page 318.

Syntax

AUTO_UNBIND_TDEV=\texttt{YES} | \texttt{NO}

AUTOXPND

See “REUSE_AUTO_expand(Yes|No)” on page 194.

Syntax

AUTOXPND=\texttt{YES} | \texttt{NO}

BACKGRND

See “BACKGROUNDCOPY(Yes|No|NOCOPYRD|VSE)” on page 155.

Syntax

BACKGRND=\texttt{YES} | \texttt{NO} | \texttt{NOCOPYRD} | \texttt{VSE}

BCVONLY

See “BCVOnly(Yes|No)” on page 156.

Syntax

BCVONLY=\texttt{YES} | \texttt{NO}
Configuration

CACHESYM

See “CACHE_FULL_SYM(Yes|No)” on page 235.

Syntax

CACHESYM=YES|NO

For Mainframe Enablers 8.1, 8.2, and 8.3, the default value is YES. For Mainframe Enablers 8.4 and later, the default value is NO.

CATALOG

See “CATalog(Yes|No)” on page 156.

Syntax

CATALOG=YES | NO

CHECKBCV

See “CHECKBCVholdstatus(Yes|No)” on page 157.

Syntax

CHECKBCV=YES | NO

CHKONLIN

See “CHECKONLINEpathstatus(Yes|No|NEVER)” on page 157.

Syntax

CHKONLIN=YES | NO | NEVER

CLEAN_R2

See “AUTOmatic_CLEANUP_R2(Yes|No)” on page 154.

Syntax

CLEAN_R2=YES | NO

CLEANDIFF

See “CLEANup_DIFFerential(Yes|No)” on page 158.

Syntax

CLEANDIFF=YES | NO

CMPLT

See “WAITFORCOMPLETION([Yes|No][hh:mm:ss] [,MeSsaGes],[.R1R2SYNC] [TIMEOUT(INformational|WARNing|ERRor)])” on page 211, the [YES|NO][hh:mm:ss] values
Syntax

CMPLT=YES | NO | \textit{wait\_seconds}

\textbf{CMPLTMSG}

See “\texttt{WAITFORCOMPLETION([Yes|No][hh:mm:ss] [,MeSsaGes][,R1R2SYNC] [TIMEOUT(INFormational|WARNing|ERRor)])}” on page 211, the \texttt{MeSsaGes} option

Syntax

CMPLTMSG= YES | NO

\textbf{COLLAPSE}

See “\texttt{COLLAPSE\_dataset\_extents(VSAM|NONVSAM|VSAM,NONVSAM)}” on page 158.

Syntax

COLLAPSE= VSAM | NOVSAM

\textbf{Note: This site option has no default value.}

\textbf{COMPACT\_MISMATCH}

See “\texttt{TOLERATE\_DATACLASS\_COMPACTION\_MISMATCH(Yes|No)}” on page 254.

Syntax

COMPACT\_MISMATCH= YES | NO

\textbf{CONDVOL}

See “\texttt{CONDitionVOLUMe(ALL|LaBeL|DUMP)}” on page 158.

Syntax

CONDVOL= ALL | DUMP | LABEL

\textbf{CONGROUP}

This option enforces the use of devices that match ConGroup (Consistency Groups for z/OS) criteria. This site option is for normal TF/Snap activity.

\textbf{Note: There are two site options that control TimeFinder interaction with ConGroup (Consistency Groups) and z/OS Migrator. They are CONGROUP and CONGROUP\_LDMF. Although both site options have the same values, they allow you to set one value for TimeFinder-ConGroup operations and the other for operations when ConGroup was invoked by z/OS Migrator.}

Syntax

CONGROUP= \textit{option}
Configuration

**Where option is one of the following:**

**IGNORE**

*Default* Do not use any special ConGroup processing.

**NONE**

The target must not be in a consistency group.

**REQUIRED_ANY**

If the source is in a consistency group, the target must be in a consistency group, but the target does not have to be in the same consistency group as the source.

**REQUIRED_SAME**

If the source is in a consistency group, the target must be in the same consistency group.

**REQUIRED_TARGET**

The target must be in a consistency group. However, the source does not have to be in a consistency group.

**WARNING**

Check and issue a warning if the target is not in the same consistency group as the source.

**CONGROUP_LDMF**

This option enforces the use of devices that match ConGroup (Consistency Groups for z/OS) criteria. This site option is for z/OS Migrator (formerly named LDMF) activity.

**Note:** There are two site options that control TimeFinder interaction with ConGroup (Consistency Groups) and z/OS Migrator. They are CONGROUP and CONGROUP_LDMF. Although both site options have the same values, they allow you to set one value for TimeFinder-ConGroup operations and the other for operations when ConGroup was invoked by z/OS Migrator.

**Syntax**

CONGROUP_LDMF=option

Where *option* is one of the following:

**IGNORE**

*Default* Do not use any special ConGroup processing for z/OS Migrator.

**NONE**

The target must not be in a consistency group.

**REQUIRED_ANY**

If the source is in a consistency group, the target must be in a consistency group; but, the target does not have to be in the same consistency group as the source.

**REQUIRED_SAME**
If the source is in a consistency group, the target must be in the same consistency group.

REQUIRED_TARGET

The target must be in a consistency group. However, the source does not have to be in a consistency group.

WARNING

Check and issue a warning if the target is not in the same consistency group as the source.

CONSALL

See “EXTENT_ALLOCAtion(Yes[,CONSOLIDATE_VOLTume],CONSOLIDATE_ALL][No)” on page 170, the CONSOLIDATE_ALL option.

Syntax

CONSALL=YES | NO

CONSIST

See “CONSISTENT(Yes|No)” on page 159.

Syntax

CONSIST=YES | NO

CONSVOL

See “EXTENT_ALLOCAtion(Yes[,CONSOLIDATE_VOLTume],CONSOLIDATE_ALL][No)” on page 170, the CONSOLIDATE_VOLTume option.

Syntax

CONSVOL=YES | NO

COPYCYL

This option allows you to specify the number of cylinders used for simultaneous I/O during a datamover cylinder copy operation.

Syntax

COPYCYL\_COUNT=number

Where:

number

Simultaneous I/O operations for COPYCYL datamover. The default value is 10.
Configuration

COPYFAIL

See “TOLerate_COPY_Failure(Yes|No)” on page 205.

Syntax

COPYFAIL=YES | NO

COPYVOL

See “COPYVolid(Yes|No)” on page 161.

Syntax

COPYVOL=YES | NO

CSMSDATA

See “COPYsourceSMSclasses([DATACLASSs] [ManaGeMenTCLASSs] [STORageCLASSs] [ALL])” on page 160, the DATACLASSs option.

Syntax

CSMSDATA=YES | NO

CSMSMGMT

See “COPYsourceSMSclasses([DATACLASSs] [ManaGeMenTCLASSs] [STORageCLASSs] [ALL])” on page 160, the MANAGECLASSs option.

Syntax

CSMSMGMT=YES | NO

CSMSSTOR

See “COPYsourceSMSclasses([DATACLASSs] [ ManaGeMenTCLASSs] [STORageCLASSs] [ALL])” on page 160, the STORageCLASSs option.

Syntax

CSMSSTOR=YES | NO

DATACLASS

See “DATACLASSs(classname)” on page 162.

Syntax

DATACLASS=classname

DATAMOVR

See “DaTaMoverNaMe(ADRdSsu|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)” on page 162.
Syntax

\texttt{DATAMOVR=name}

**DS1DSCHA**

See “\texttt{DATASET\_CHANGED\_indicator(SET|RESET|LEAVE)}” on page 164.

Syntax

\texttt{DS1DSCHA=SET | RESET | LEAVE}

**DEALLOC**

This option specifies the default task name for the "S DEALLOC" parameter if a device fails to go offline or online.

Syntax

\texttt{DEALLOC=name}

Where:

\texttt{name}

The task name. The default value is DEALLOC.

\textit{Note:} If a name other than DEALLOC is used, it must be present in your STC proclib.

**DEBUG\_ERROR**

See “\texttt{DEBUG(ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP)}” on page 238, the ERROR option.

Syntax

\texttt{DEBUG\_ERROR=YES | NO}

**DEBUG\_SDUMP**

See “\texttt{DEBUG(ALL|EXTRA|TRACE|DUMP|ERROR|SDUMP)}” on page 238, the SDUMP option.

Syntax

\texttt{DEBUG\_SDUMP=YES | NO}

**DFDSS\_ADMIN**

See “\texttt{DFDSS\_ADMIN(Yes|No)}” on page 165.
**Syntax**

```
DFDSS_ADMIN=YES | NO
```

**DFDSS_CC**

See “DFDSS_CC(Yes|No)” on page 165.

**Syntax**

```
DFDSS_CC=YES | NO
```

**DFDSS_OP**

See “DFDSS_OPTimize(n)” on page 239.

**Syntax**

```
DFDSS_OP=1 | 2 | 3 | 4
```

**DIFF**

See “DIFFerential(Yes|No)” on page 165.

**Syntax**

```
DIFF=YES | NO
```

**DIFFDSN**

See “DIFFERENTIAL_DATASET(Yes|No)” on page 166.

**Syntax**

```
DIFFDSN=YES | NO
```

**DMIDCAMS**

See “DaTaMoverNaMe(ADR|SSU|CO|PY|CYL|COPY|TRK|DFDSS|SSS|FDR|FDR|DSF|IDCA|MS|NONE)” on page 162, the IDCAMS option.

**Syntax**

```
DMIDCAMS=YES | NO
```

**EATTR**

See “EATTR(NO|OPT)” on page 166.

**Syntax**

```
EATTR=NO | OPT
```

**Note:** This site option has no default value.
EMCALLOC_TRACE

This option determines whether trace is on or off during a EMCALLOC operation. Tracing is captured and included in the QCOUTPUT log.

Syntax

EMCALLOC_TRACE=YES | NO

Where:

YES

The trace is on.

NO

(Default) The trace is off.

EMCCOPY

This option determines whether EMCOPY microcode can be used with Enginuity 5x65 if the source is not an STD or the target is not a BCV.

Syntax

EMCCOPY=YES | NO

Where:

YES

EMCCOPY microcode may be used.

NO

(Default) EMCCOPY microcode may not be used.
EMCDSSU_FLASH_SNAP

This option allows you to specify IBM's FLASHCOPY or Dell EMC's SNAP as the preferred copy method on the device.

If multiple extents are grouped in datasets, TF/Clone replicates source datasets to target datasets as long as the source device and the target device reside on the same storage system. If an external data mover such as IBM FlashCopy is required to replicate datasets between control units or storage systems, the operating environment provides support for IBM FlashCopy and invokes a FlashCopy session to replicate within an array, and not between arrays.

Syntax

EMCDSSU_FLASH_SNAP=SNAP | FLASHCOPY
Where:
SNAP
  (Default) Use the SNAP microcode.
FLASHCOPY
  Use the FLASHCOPY microcode.

EMCDSSU_TARGET

This option allows you to decide whether EMCQCAPI (Dell EMC's TF Snap interface) or ADRDSSU (IBM datamover utility program) is used for a copy/damamover operation.

Syntax

EMCDSSU_TARGET=MATCH | IGNORE
Where:
MATCH
  (Default) The Symmetric source and target volume must match to invoke EMCSNAPI instead of ADRDSSU.
IGNORE
  Only the source volume must be a PowerMax/VMAX device to invoke EMCQCAPI.

EMCONLY

See “EXTALLOC_EMC_ONLY(Yes|No)” on page 171.

Syntax

EMCONLY=YES | NO
**EMCQCAPI_TRACE**

This option allows you to turn debug tracing on or off for EMCQCAPI operations. Tracing is captured and included in the QCOUTPUT log.

**Syntax**

```plaintext
EMCQCAPI_TRACE=YES|NO
```

**Where:**

- **YES**
  - Debug tracing is on.
- **NO**
  - (Default) Debug tracing is off.

**EMUL_TYPE**

See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 166.

**Syntax**

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

**ENQFAIL**

See “TOLERateENQFailure(Yes|No)” on page 205.

**Syntax**

```plaintext
ENQFAIL=YES|NO
```

**ENQSCOPE**

See “ENQSCOPE(REquest|STEP)” on page 167.

**Syntax**

```plaintext
ENQSCOPE=REQUEST|STEP
```

**ENQWAIT**

See “ENQWAIT(Yes|No)” on page 167.

**Syntax**

```plaintext
ENQWAIT=YES|NO
```

**ERRCHK**

See “ERROR_Checking(NORmal|REDUCED)” on page 167.

---

1. Available starting with Mainframe Enablers 8.2.
Syntax

ERRCHK=\texttt{NORMAL} | \texttt{REDUCED}

\textbf{ERRDISP}

See “\texttt{ERRor\_DISPosition(DELETE|KEEP)}” on page 168.

Syntax

ERRDISP=\texttt{DELETE} | \texttt{KEEP}

\textbf{ERRREC}

This option determines how error recovery is handled when a copy operation fails.

Syntax

\begin{Verbatim}
ERRREC=\texttt{NORMAL} | \texttt{ENHANCED}
\end{Verbatim}

Where:

\texttt{NORMAL}

\begin{itemize}
  \item \textbf{(Default) Perform normal error recovery after a copy operation fails.}
\end{itemize}

\texttt{ENHANCED}

\begin{itemize}
  \item Perform enhanced error recovery when a copy operation fails. Used in conjunction with \texttt{ERROR\_CHECKING} in order to recover from an error that would normally be caught before the copy operation is processed.
\end{itemize}

\textbf{ESNP119}

See “\texttt{ESNP119(WARNING|ERROR)}” on page 241.

Syntax

\begin{Verbatim}
ESNP119=\texttt{WARNING} | \texttt{ERROR}
\end{Verbatim}

\textbf{ESNP220}

See “\texttt{ESNP220(ERROR|WARNING)}” on page 169.

Syntax

\begin{Verbatim}
ESNP220=\texttt{WARNING} | \texttt{ERROR}
\end{Verbatim}

\textbf{ESNP231E}

This is a site-options parameter that has no GLOBAL equivalent. Normally, when the ESNP231E message is issued (dataset not found) during the parse phase, execution of all statements is bypassed.
Syntax

\texttt{ESNP231E=\{YES\mid NO\}}

Where:

\texttt{YES}

Change the severity of message ESNP231E to ESNP231W and allow the execution of other statements to occur.

\texttt{NO}

(\texttt{Default}) If ESNP231E is issued during the parse phase, skip the execution phase.

**EXAMINE**

See “\texttt{EXAMINE(Yes\mid No)}” on page 169.

Syntax

\texttt{EXAMINE=\{YES\mid NO\}}

**EXPATHGRP**

See “\texttt{EXclude\_PathGroupID(pathlist)}” on page 169.

Syntax

\texttt{EXPATHGRP=pathlist}

**EXPLAIN\_VOL\_SEL**

See “\texttt{EXPlain(VOLUME\_SELection(Yes\mid No))}” on page 170.

Syntax

\texttt{EXPLAIN\_VOL\_SEL=\{YES\mid NO\}}

**EXTADDNEW**

See “\texttt{EXTENT\_EXPAND(Yes\mid No,\{ADDNEW(Yes\mid No)\mid SAMEVOL\mid NEWVOL\})}” on page 171, the ADDNEW option.

Syntax

\texttt{EXTADDNEW=\{YES\mid NO\}}

\textbf{Note:} This site option has no default value.

**EXTALLOC**

This option determines if new target datasets are defined through extent allocation or by another means.
Configuration

Syntax

EXTALLOC=YES|NO|OK

**Note:** This site option has no default value.

Where:

YES

Always use extent allocation to create new target datasets.

NO

Never use extent allocation.

OK

Use IDCAMS/SVC99 or Extent allocation to create new target datasets.

**EXTENDED_MISMATCH**

See “TOLERATE_DATACLASS_EXTENDED_MISMATCH(Yes|No)” on page 254.

Syntax

EXTENDED_MISMATCH=YES|NO

**EXTXPAND**

See “EXTENT_EXPAND(Yes|No,[ADDNEW(Yes|No)][,SAMEVOL][,NEWVOL])” on page 171.

Syntax

EXTXPAND=YES|NO

**Note:** This site option has no default value.

**EXTXPVOL**

See “EXTENT_EXPAND(Yes|No,[ADDNEW(Yes|No)][,SAMEVOL][,NEWVOL])” on page 171, the SAMEVOL/NEWVOL options.

Syntax

EXTXPVOL=YES|NO

**Note:** This site option has no default value.

**FBA**

See “FBA(EXCLUDE|INCLUDE)” on page 172.

Syntax

FBA=INCLUDE|EXCLUDE
FLASH_SNAP

See “FLASH_SNAP(FLASHCOPY|SNAP)” on page 172.

Syntax

FLASH_SNAP=FLASHCOPY | SNAP

FORCE

See “FORCE(Yes|No)” on page 172.

Syntax

FORCE=YES | NO

FORCECMP

See “FORCE_COMPLETION(Yes|No)” on page 172.

Syntax

FORCECMP=YES | NO

FREESPC

See “FREESPAC(Yes|No)” on page 173.

Syntax

FREESPC=YES | NO

FULL

This option determines if SNAP is operational on a full device or on a defined extent device.

Syntax

FULL=YES | NO

Where:

YES

Full device microcode SNAP.

NO

(Default) Extent device microcode SNAP.

GROUP_DEVICE_READY_STATE

See “GROUP_DEVic_e_ready_state(AUTO|NEVER)” on page 243.

Syntax

GROUP_DEVICE READY_STATE=AUTO | NEVER
GROUP_DSNAME

See “GROUP_DATASET_name(‘dataset_name’)” on page 242.

Syntax

GROUP_DSNAME=dataset_name

GROUP_EMCQCAPI_VERIFY

See “GROUP_EMCQCAPI_VERIFY(Yes|No)” on page 243.

Syntax

GROUP_EMCQCAPI_VERIFY=YES|NO

GROUP_HISTORY_LIMIT

This option sets a limit for the number of history records that are kept for a group.

Syntax

GROUP_HISTORY_LIMIT=number

Where:

number

The number of history records to keep. The default value is 100.

HOSTCOPY

See “HostcoPYMODE(SHaReD|EXClusive|NONE)” on page 174.

Syntax

HOSTCOPY=OLD|SHARED|NONE

IGNORERDF

This option allows you to ignore any R1/R2 relationships in a SNAP operation.

Syntax

IGNORERDF=YES|NO

Where:

YES

Ignore the R1/R2 relationship.

NO (Default) Take advantage of the R1/R2 relationship.

INVALIDATE_PDSE

See “INVALIDATE_PDSE_buffers(Yes|No)” on page 174.
Syntax

**INVALIDATE_PDSE=**YES|NO

**MAXDSSU**

See “MAXIMUM_ADRDSSU_address_spaces(number)” on page 244.

Syntax

**MAXDSSU=**number_of_asids

*Note: The default value is 10.*

**MAXTASK2**

See “MAXIMUM_SUBTASKS(number1,number2)” on page 245, the number1 option.

Syntax

**MAXTASK2=**number_of_tasks

*Note: The default value is 999.*

**MAXTASKR**

See “MAXIMUM_SUBTASKS(number1,number2)” on page 245, the number2 option.

Syntax

**MAXTASKR=**number_of_tasks

*Note: The default value is 99.*

**MESSAGE**

See “MESSages(DISplay|PROMpt|NONE|DETAIL)” on page 176.

Syntax

**MESSAGE=**DISPLAY|PROMPT|NONE|DETAIL

**MGMTCLAS**

See “ManaGeMenTCLASs(classname)” on page 176.

Syntax

**MGMTCLAS=**classname

**MIGRATRC**

See “MiGrate([PURge(Yes|No)] [RECall(Yes|No)])” on page 177.
Configuration

Syntax

MIGRATRC=4 | 8

**MINSNAP**

This option allows you to decide on the minimum number of tracks to be SNAPPED before microcode is used.

Syntax

MINSNAP=number

Where:

number

Sets the minimum number of tracks. If the number of tracks is less than this number, then a physical copy of the tracks occurs instead of using microcode to complete the SNAP.

The default value is 5.

**MLQ**

See “MULTI_LINE_query(Yes|No)” on page 181.

Syntax

MLQ=YES | NO

**MRGEXIST**

This options allows you to consolidate, or not consolidate, extents on a volume.

Syntax

MRGEXIST=YES | NO

Where:

YES

(Default) Consolidate extents on a volume.

NO

Do not consolidate extents on a volume.

**MULTIVIRTUAL**

See “MULTI_VIRTUAL(Yes|No)” on page 182.

Syntax

MULTI_VIRTUAL=YES | NO
**NTFYGRP**

See “NOTIFYwhencomplete[([GROUP(name)])[DATASET|JOB|STEP|SNAP]])” on page 183, the GROUP option.

Syntax

NTFYGRP=\texttt{\textit{groupname}}

**NTFYLVL**

This option allows you to request ResourcePak Base (EMCSCF) to issue a WTO when the SNAP operation is complete.

No notification is performed by default.

See “NOTIFYwhencomplete[([GROUP(name)])[DATASET|JOB|STEP|SNAP]])” on page 183 for more information.

Syntax

NTFYLVL=\texttt{\textit{DATASET} | \textit{JOB} | \textit{SNAP} | \textit{STEP}}

Where:

**DATASET**

EMCSCF issues WTO for each dataset or volume when SNAP is completed.

**JOB**

EMCSCF issues WTO for all datasets or volumes in the JOB when SNAP is completed.

**SNAP**

EMCSCF issues WTO for all datasets or volumes in a single SNAP statement when it is completed.

**STEP**

EMCSCF issues WTO for all datasets or volumes in the JOB STEP when SNAP is completed.

**OFFLINE**

This option determines whether you can allow offline devices to participate in a SNAP VOLUME operation.

Syntax

OFFLINE=\texttt{\textit{YES} | NO}

Where:

**YES**

(Default) SNAP VOLUME operation may specify offline devices.

**NO**

SNAP VOLUME operation may not specify any offline devices.
**OPEN_SOURCE_RC**

This option lists the message severity (0, 4, or 8) and code type (informational, warning, or error) if the source dataset is open.

**Syntax**

```
OPEN_SOURCE_RC=0 | 4 | 8
```

*Where:*

0  
Message severity = 0, message code = I (Informational).

4  
*(Default) Message severity = 4, message code = W (Warning).*

8  
Message severity = 8, message code = E (Error).

**OPT_CKD**

This option allows you to include or exclude CKD devices in the outcome report of a QUERY.

**Syntax**

```
OPT_CKD=INCLUDE | EXCLUDE
```

*Where:*

INCLUDE  
*(Default) Include CKD devices in the query report.*

EXCLUDE  
Do not include CKD devices in the query report.

**OPT_FBA**

This option allows you to include or exclude FBA devices in the outcome report of a QUERY.

**Syntax**

```
OPT_FBA=INCLUDE | EXCLUDE
```

*Where:*

INCLUDE  
*(Default) Include FBA devices in the query report.*

EXCLUDE  
Do not include FBA devices in the query report.
OPT_NOTREADY

This option allows you to include or exclude NOTREADY devices in the outcome report of a QUERY.

Syntax

```
OPT_NOTREADY=INCLUDE | EXCLUDE
```

Where:

INCLUDE

(Default) Include NOTREADY devices in the query report.

EXCLUDE

Do not include NOTREADY devices in the query report.

OPT_RAID

This option allows you to include or exclude all or specific RAID devices in the outcome report of a QUERY.

Syntax

```
OPT_RAID=ALL | N | S | 1 | 5 | 6 | 10
```

Where:

ALL

(Default) Include all RAID devices.

N

Do not include any RAID devices.

S

Include RAID 'S' devices.

1

Include RAID '1' devices.

5

Include RAID '5' devices.

6

Include RAID '6' devices.

10

Include RAID '10' devices.

OPT_READY

This option allows you to include or exclude READY devices in the outcome report of a QUERY.
Configuration

Syntax

OPT_READY=INCLUDE | EXCLUDE
Where:

INCLUDE

(Default) Include READY devices in the query report.

EXCLUDE

Do not include READY devices in the query report.

OPT_SAVEDEV

This option allows you to include or exclude SAVEDEV devices in the outcome report of a QUERY.

Syntax

QUERYVOL(SAVEDEV)=INCLUDE | EXCLUDE
Where:

INCLUDE

(Default) Include SAVEDEV devices in the query report.

EXCLUDE

Do not include SAVEDEV devices in the query report.

OPT_TDEV

This option determines whether thin devices are to be included in reports generated by the QUERY VOLUME command.

Syntax

OPT_TDEV=EXCLUDE | INCLUDE
Where:

EXCLUDE

Exclude thin devices on QUERY VOLUME reports.

INCLUDE

(Default) Include thin devices on QUERY VOLUME reports.

OPT_THINPOOL

See “THINPOOL(EXCLUDE|INCLUDE)” on page 204.

Syntax

OPT_THINPOOL=INCLUDE | EXCLUDE
**OPT_VDEV**

This option allows you to include or exclude VDEV devices in the outcome report of a QUERY.

**Syntax**

```plaintext
QUERYVOL(VDEV)=INCLUDE|EXCLUDE
```

Where:

- **INCLUDE**
  - *(Default)* Include VDEV devices in the query report.
- **EXCLUDE**
  - Do not include VDEV devices in the query report.

**PARALLEL**

See “PARallel(Yes|No)” on page 248.

**Syntax**

```plaintext
PARALLEL=YES|NO
```

**PARALLEL_CLONE**

See “PARALLEL_CLONE(Yes|No|PREFerred|REQuired)” on page 185.

**Syntax**

```plaintext
PARALLEL_CLONE=YES|NO|PREF|REQ
```

**Note:** This site option has no default value.

**PERSIST**

See “PERSISTent(Yes|No)” on page 186.

**Syntax**

```plaintext
PERSIST=YES|NO
```

**POOL**

See “POOL(poolname)” on page 186.

**Syntax**

```plaintext
POOL=poolname
```

**POOLUSE**

See “CHecK_POOL_usable(Yes|No)” on page 157.
POOLUSE=YES|NO

PRECOPY
See “PRECOPY(Yes|No)” on page 187.

PREPARE
See “PREPARE_FOR_SNAP(Yes|No)” on page 248.

PROCESS_COPYCYL_DATAMOVER
This option controls whether an internal datamover may be used when a snap target device is an z/OS Migrator source device.

PROCESS_COPYCYL_DATAMOVER=YES|NO
Where:
YES
Instead of failing the request, use the internal datamover to copy the track images.
NO (Default) Fail the request when the snap target device is an z/OS Migrator source device.

PURGE
See “MIGrate([PURge(Yes|No)] [RECall(Yes|No)])” on page 177, the PURGE option.

PURGE=YES|NO

QCAPIMSG
This option allows you to add the job name to each message generated by EMCQCAPI.

QCAPIMSG=YES|NO
Where:
YES
(Default) Add the job name prefix to each message.

NO

Do not add the job name prefix to each message.

R1FULLCOPY

See “R1FULLCOPYonly(Yes|No)” on page 188.

Syntax

R1FULLCOPY=YES | NO

R1R2SYNC

See “WAITFORCOMPLETION([Yes|No][hh:mm:ss] [,Messages][,R1R2SYNC][TIMEOUT(INformational|WARNing|ERRor)])” on page 211, the R1R2SYNC option.

Syntax

R1R2SYNC=YES | NO

RECALC_FREE

See “RECALCULATE_FREESPACE(Yes|No)” on page 189.

Syntax

RECALC_FREE=YES | NO

RECALL

See “MIgrate([PURge(Yes|No)] [RECall(Yes|No)])” on page 177, the RECall option.

Syntax

RECALL=YES | NO

REFVTOC

See “REFVTOC(Yes|No)” on page 190.

Syntax

REFVTOC=YES | NO

REMOVE_REMOTE

See “REMOVE_REMOTE_extent_sessions(Yes|No)” on page 191.

Syntax

REMOVE_REMOTE=YES | NO
**REPLACE**

See “REPLace(Yes|No)” on page 192.

Syntax

REPLACE=\texttt{YES} | \texttt{NO}

**RESERVE**

See “RESERVE(Yes|No)” on page 250.

Syntax

RESERVE=\texttt{YES} | \texttt{NO}

**RESERVE\textunderscore SERIALIZATION**

This option allows you to specify the enqueue area when serializing devices for syscalls.

Syntax

RESERVE\textunderscore SERIALIZATION=\texttt{YES} | \texttt{NO}

Where:

\texttt{YES}

When serializing devices for syscalls, use a reserve instead of a system enqueue.

\texttt{NO}

(\texttt{Default}) Only use the system enqueue and not a reserve for serialization.

**RETAIN\textunderscore SOURCE \_REFDT**

This option allows you to keep the original date of the source dataset or reset to the current date.

Syntax

RETAIN\textunderscore SOURCE\_REFDT=\texttt{YES} | \texttt{NO}

Where:

\texttt{YES}

Retain the original source reference date of the dataset.

\texttt{NO}

(\texttt{Default}) Set the reference date of the dataset to the current date.

**RETRY1731**

This option allows you to set the retry logic when a 1731 and 1767 type of error is encountered.
Syntax

RETRY1731=YES|NO|minutes

Where:

YES

Issue a retry message to the console and continue to retry.

NO

(Default) Accept a failure with the error.

minutes

Issue a retry message to the console and continue to retry for ## of minutes.

RETRY1756

This option allows you to set the retry logic when a 1756 type of error is encountered.

Syntax

RETRY1756=YES|NO

Where:

YES

(Default) Attempt to release the processing hold status and retry the TimeFinder command.

NO

Fail and generate an error.

REUSE

See “REUSE(Yes|No[,WAIT])” on page 194.

Syntax

REUSE=YES|NO

REUSFAIL

See “TOLerate_REUSE_Failure(Yes|No)” on page 204.

Syntax

REUSFAIL=YES|NO

SAMEONLY

There are several conditions which must be met for SAMEONLY to take effect:
Configuration

- You must specify SAMEONLY(YES) in the site options table.
- You must select EXTENT_ALLOCATION for allocation. SVC99 and IDCAMS allocation ignore the SAMEONLY site option.
- You must specify DATAMOVERNAME(NONE). If a datamover is specified, then SAMEONLY is ignored.

Syntax

SAMEONLY=YES| NO

SAVEFULL

See “WHEN_SAVEDEV_FULL(READY|NOTREADY)” on page 213.

Syntax

SAVEFULL=READY| NOTREADY

SCRATCH

With this option, you can choose to delete the dataset when a STOP SNAP TO DATASET command is issued.

Syntax

SCRATCH=YES| NO

Where:

YES

Delete the dataset after issuing the STOP SNAP TO DATASET command.

NO

(Default) Do not delete the dataset after issuing a STOP SNAP TO DATASET command.

SESSDETL

See “SESSION_LIST(Yes|No[,DETAIL|NODETAIL|DIFFerential])” on page 195, the DETAIL option.

Syntax

SESSDETL=DETAIL| NODETAIL

SESSDIFF

See “SESSION_LIST(Yes|No[,DETAIL|NODETAIL|DIFFerential])” on page 195, the DIFFerential option.

Syntax

SESSDIFF=DIFF| NODIFF
SESSLIST

See “SESSION_LIST(Yes|No[,DETAIL|NODETAIL[,DIFFerential]])” on page 195.

Syntax

SESSLIST=YES|NO

SMFRID

Enables SMF records for each command executed. The values for the record ID may be 0, meaning omit this feature, or a valid integer between 128 and 255, inclusive. Since the records are written for EMCSNAP commands, if the command is executed on a non-Dell EMC device, it is still recorded.

Syntax

SMFRID=number

Where:

0

(Default) Do not enable recording of SMF records.

128–255

Values for the SMF record ID. (IBM reserves the values 1-127.)

SMSKSDS

With this option you can force the key sequential dataset (KSDS) components to be stored on separate volumes. The INDEX and DATA components is separated.

Syntax

SMSKSDS=YES|NO

Where:

YES

Force the KSDS components to be stored on separate volumes.

NO

(Default) Allow SMS to determine how the KSDS is stored.

SMSPASSVOL

This option allows you to pass user-suggested volumes to SMS for allocation.

Syntax

SMSPASSVOL=YES|NO

Where:

YES
Configuration

Pass suggested volumes to SMS for allocation

NO

(Default) Let SMS determine the volume candidates.

SNAPSHOT_LIST¹

See “SNAPSHOT_LIST(ALL|LINKED|NOT_LINKED|SNAPSHOT)” on page 196.

Syntax

SNAPSHOT_LIST=ALL|LINKED|NOT_LINKED|SNAPSHOT

SNAPSHOT_NAME

See “NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%%])” on page 182.

Syntax

SNAPSHOT_NAME=snapshot_name

SNUNUSED

See “SNAP_UNUSED_SPACE(Yes|No)” on page 197.

Syntax

SNUNUSED=YES|NO

SOFTLINK

See “SOFTlink(Yes|No)” on page 197.

Syntax

SOFTLINK=YES|NO

SRDFAR1

See “SRDFA_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormational)” on page 199.

Syntax

SRDFAR1=YES|NO|PHYSICAL|DATAMOVERNAME

SRDFAR2

See “SRDFA_R2_sync(WARNING|R1R2SYNC|DATAMOVER)” on page 200.

Syntax

SRDFAR2=WARNING|DATAMOVER|R1R2SYNC

¹ Available starting with Mainframe Enablers 8.2.
**SRDFAR2_PRECOPY**

This option allows you to determine if the SRDFA/R2 precopy operation can be overridden, allowing the SNAP to occur.

**Syntax**

```
SRDFAR2_PRECOPY=YES|NO
```

**Where:**

**YES**

*(Default)* Allow the SRDFA/R2 precopy wait time to be overridden allowing the SNAP to occur.

**NO**

Do not allow the SNAP unless the precopy has completed. Wait if necessary.

**SRDFA_RETRY**

See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 199.

**Syntax**

```
SRDFA_RETRY=YES|NO|nn
```

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

**SRDFS_R1**

See “SRDFS_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormational)” on page 201.

**Syntax**

```
SRDFS_R1=YES|NO|PHYSICAL|DATAMOVERNAME
```

**STORCLAS**

See “STORageCLASs(classname)” on page 201.

**Syntax**

```
STORCLAS=classname
```

**STORED_LOG_SIZE**

See “STORED_LOG_SIZE(size)” on page 253.

**Syntax**

```
STORED_LOG_SIZE=number_of_lines
```
Configuration

**SUBTNAME**

Establish a subtask name or accept the default of EMCSNAPI.

Syntax

```
SUBTNAME=subtask_name
```

Where:

- `subtask_name`
  The name of the subtask. The default value is EMCSNAPI.

**SYSCALL_RETRY**

Specifies logic for handling generic syscalls.

Syntax

```
SYSCALL_RETRY=count
```

Where:

- `count`
  Issue a retry message to the console, but only for this number of attempts. The default value is 1600.

**TARGET_WAIT**

See “TARGET_ENQ_dataset_wait(Yes|No|hh:mm:ss)” on page 296.

Syntax

```
TARGET_WAIT=YES|NO
```

**TDEV_RECLAIM**

See “TDEV_RECLAIM(Yes|No)” on page 203.

Syntax

```
TDEV_RECLAIM=YES|NO
```

**TERMSESS**

See “TERMINATE_SESSION_when_complete(Yes|No)” on page 203.

Syntax

```
TERMSESS=YES|NO
```
**TIMEOUT**

See “TIMEOUT(nnn)” on page 203.

**Syntax**

TIMEOUT=number_of_seconds

**TRKALIGN**

Specifies whether or not to match the source dataset track alignment.

**Syntax**

TRKALIGN=YES|NO

Where:

YES

Match source dataset track alignment.

NO

(Default) Do not match source track alignment.

**TRUNC**

See “TOLerateTRUNCation(Yes|No)” on page 206.

**Syntax**

TRUNC=YES|NO

**VALIDATE**

This option allows you to read and compare all the source and target tracks after a SNAP is executed.

**Syntax**

VALIDATE=YES|NO

Where:

YES

Read and compare all source and target tracks.

NO

(Default) Do not read and compare source and target tracks.

**VALFIRST**

This option allows you to read and compare the first track numbers from each extent after the SNAP is started. This overrides the VALIDATE=YES to limit it to just the first number of tracks.
Configuration

Syntax

VALFIRST=number_of_tracks

Where:

number_of_tracks

Number of tracks to compare. The default value is 0.

VALLAST

This option allows you to read and compare the last track numbers from each extent after the SNAP is started. This overrides the VALIDATE=YES to limit it to just the last number of tracks.

Syntax

VALLAST=number_of_tracks

Where:

number_of_tracks

Number of tracks to compare. The default value is 0.

VALRANGE_LOCAL

See “VALIDATE_RANGE({LOCAL|REMOTE}({AUTO|IGNORE}))” on page 255.

Syntax

VALRANGE_LOCAL=AUTO | IGNORE | FORCE

VALRANGE_REMOTE

See “VALIDATE_RANGE({LOCAL|REMOTE}({AUTO|IGNORE}))” on page 255.

Syntax

VALRANGE_REMOTE=AUTO | IGNORE | FORCE

VALSMS

This option allows you to validate SMS class names when supplied by Parser or API.

Syntax

VALSMS=YES | NO

Where:

YES

(Default) Validate class names during parsing.

NO

Do not validate SMS class names during parsing.
**VARYOFF**

See “VARY_OFFline(AUTO|NEVER)” on page 207.

Syntax

VARYOFF=\texttt{AUTO|NEVER}

**VARYON**

See “VARY_ONline(AUTO|Yes|No)” on page 208.

Syntax

VARYON=\texttt{AUTO|NEVER}

**VCLOSE**

See “VCLOSE(Yes|No)” on page 208.

Syntax

VCLOSE=\texttt{YES|NO}

**VDEV_REUSE**

This option permits you to allow, or prevent, a VDEV from being reused unless a STOP SNAP has been done to the device.

By default, VDEV_REUSE=\texttt{YES} allows multiple, separate SNAP VOLUME to VDEV requests to be issued with the same VDEV specified. TF/Snap automatically detects and reuses the VDEV by internally issuing a STOP SNAP to the VDEV and then letting the SNAP VOLUME continue.

With VDEV_REUSE=\texttt{NO}, TF/Snap fails the new SNAP VOLUME request when it detects that the VDEV is already being used. This forces you to manually issue a STOP SNAP to free the virtual device. After the virtual device is freed, a SNAP VOLUME request is successful.

Syntax

VDEV_REUSE=\texttt{YES|NO}

Where:

\texttt{YES} \hspace{1cm} \texttt{(Default)} Allow multiple, separate SNAP VOLUME to VDEV requests to be issued with the same VDEV specified. TimeFinder automatically detects and reuses the VDEV by internally issuing a STOP SNAP to the VDEV and then letting the SNAP VOLUME continue.

\texttt{NO} \hspace{1cm} Do not allow multiple, separate SNAP VOLUME to VDEV requests to be issued with the same VDEV specified. Instead, a STOP SNAP request must be issued to release the VDEV device before another SNAP VOLUME is used with the device.
**VDEVWAIT**

See “VDEVWAIT(Yes|No)” on page 209.

Syntax

```
VDEVWAIT=YES | NO
```

**VERIFY_OPEN_SOURCE**

See “VERIFY_OPEN_SOURCE(Yes|No)” on page 210.

Syntax

```
VERIFY_OPEN_SOURCE=YES | NO
```

**VERIFY**

See “VERIFY(Yes|No|NEVER)” on page 209.

Syntax

```
VERIFY=YES | NO | NEVER
```

**VSAMENQ**

See “VSAMENQMODE(SHAREd|EXClusive|NONE)” on page 210.

Syntax

```
VSAMENQ=OLD | SHARED | NONE
```

**VSAMFAIL**

See “TOLerateVSAMENQFailure(Yes|No)” on page 206.

Syntax

```
VSAMFAIL=YES | NO
```

**VTOCIX**

See “BUILD_VTOCIX(Yes|No)” on page 156.

Syntax

```
VTOCIX=YES | NO
```

**WAIT_OFFLINE_LIMIT**

This option allows you to governs the time to wait for a device to go offline when a VARY OFFLINE command has been issued to a device.

Syntax

```
WAIT_OFFLINE_LIMIT=time
```
WAIT_ONLINE_LIMIT

This option allows you to govern the time to wait for a device to come online when a VARY OFFLINE command has been issued to a device.

Syntax

\[
\text{WAIT\_ONLINE\_LIMIT} = \text{time}
\]

Where:

\text{time}

Number of minutes to wait (0 to infinity). The default value is 5.

WAIT_PRECOPY

See “WAIT\_FOR\_PRECOPY\_PASS1\(Yes|No\)” on page 212.

Syntax

\[
\text{WAIT\_PRECOPY} = \text{YES|NO}
\]

WAIT

See “WAIT\_forsession\(Yes|No|hh:mm:ss\)” on page 212.

Syntax

\[
\text{WAIT} = \text{YES|NO|seconds}
\]

WFDEF

See “WAIT\_FOR\_Definition\(Yes|No\)” on page 212.

Syntax

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

XTNTBNDRY

This option allows you to determine the handling of extent boundaries.

Syntax

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

Where:
Configuration

YES
Target extent matches source extent and source extent boundary.

NO
(Default) Do not worry about extent boundaries.
CHAPTER 4
Operations and Examples

This chapter describes traditional TimeFinder operations and examples.

◆ Specifying devices .................................................................................................. 96
◆ Defining a group of statements .............................................................................. 97
◆ Performing a SNAP VOLUME copy ..................................................................... 101
◆ Performing a SNAP VOLUME using virtual devices .............................................. 111
◆ Performing Cascaded clone operations ............................................................... 114
◆ Influencing SMS volume selection ..................................................................... 117
◆ Selecting volume processing by phases .............................................................. 119
◆ Performing a SNAP DATASET copy .................................................................. 122
◆ Performing a Parallel Snap ................................................................................ 142
◆ Performing queries .............................................................................................. 145
◆ Cleaning up volumes .......................................................................................... 146
◆ Using SRDF/A R2 Wait for Precopy .................................................................... 147
◆ Viewing GCM status ........................................................................................... 148

IMPORTANT

Traditional TimeFinder operations refers to the commands, syntax, and processes that require a physical target device to be specified in the command to execute a full volume copy. For information about SnapVX targetless snapshots, see the TimeFinder SnapVX and zDP Product Guide.
Specifying devices

SNAP VOLUME and various other TimeFinder commands have a series of keywords that you can use to identify the device on which you want the command to operate. The usual syntax (employed in Chapter 5, “Command Reference”) is as follows:

\[ \text{VOLUME}(\text{volser}) | \text{UNIT}(\text{cuu}) | \text{SYMDV#}(\text{symdv#}) | \text{GROUP}(\text{grpname}, \text{grpname}...) \]

- VOLUME(\text{volser}) and UNIT(\text{cuu}) identify a device that is known to z/OS. When you query z/OS about the device or devices, TimeFinder returns both the device and what storage system it is in.

\[ \textbf{Note:} \text{When predefined and stored in a group, the VOLUME subparameter can be used within the SOURCE parameter, but cannot be used within the TARGET parameter.} \]

The UNIT or SYMDV# subparameter must be used within the TARGET parameter to identify a device when predefined and stored in a group.

- SYMDV#(\text{symdv#}) identifies a PowerMax/VMAX device number in a storage system. But, it does not identify which storage system.

For that reason, most commands on which a particular device is to be specified, require you to use the LOCAL, the REMOTE, or the CONTROLLER parameter to specify the gatekeeper for the SYMDV# device.

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

The CONTROLLER parameter specifies the storage system to be queried. You can use the 5 or 12 digit serial number of the storage system or a logical storage system name if you previously defined that name to ResourcePak Base.

The CONTROLLER parameter is available as an optional subparameter of the LOCAL and REMOTE parameters. It is also available as a separate parameter. If you use the separate CONTROLLER parameter, do not include the LOCAL and REMOTE parameters.

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.

- GROUP(\text{grpname}, \text{grpname}...) identifies one or more groups of TimeFinder statements. The commands and parameters you include in these groups identify a particular device or devices in a particular storage system.

\[ \textbf{Note:} \text{“Defining a group of statements” on page 97 provides more information.} \]
Defining a group of statements

TimeFinder allows you to define a group of TF SNAP VOLUME or GLOBAL statements, store them in a group dataset, and then use that group as an argument to ACTIVATE, CLEANUP, CONFIG, SNAP VOLUME, and STOP SNAP TO VOLUME 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 TimeFinder commands
4. Selecting processing by phases (if applicable)

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_DSNAME (dataset name) parameter of the GLOBAL command.

  Note: “GROUP_DATaset_name('dataset_name’)” on page 242 provides more information about this parameter.

- Use the DD statement //EMCGROUP DD to point to the group dataset.

  Note: You can concatenate multiple datasets together with EMCGROUP.

- Define GROUP_DSNAME in the site options table EMCSNAPO.
The format is EMCSNAPO GROUP_DSNAMES\(dataset\ name, dataset\ name, \ldots\) You can specify an unlimited number of dataset names. When allocated, they are concatenated in order.

**Note:** Table 2 on page 47 describes the site options table, EMCSNAPO.

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

**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.

```bash
//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 the groups

You now create groups using the GROUP commands. There are four commands for group processing:

- DEFINE GROUP
- END GROUP
- DELETE GROUP
- QUERY GROUP

**Note:** Do not edit either member of the PDS to change a group definition.

You use DEFINE GROUP 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 SNAP VOLUME 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. The QUERY GROUP command allows you to display information about a group.

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.

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)
  .
  .
Using groups as arguments to TimeFinder commands

After you define such a group, you can handle the devices as a group in an argument of the GROUP parameter with the ACTIVATE, CLEANUP, CONFIG, SNAP VOLUME, and STOP SNAP TO VOLUME commands. The GROUP parameter allows you to specify multiple groups at once. This is most important for consistent activate—allowing multiple groups to be activated together.

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.
Performing a SNAP VOLUME copy

You can copy a volume to a target with the SNAP VOLUME command. SNAP VOLUME creates an exact copy of the source volume on the target volume (requires the TF/Clone Licensed Feature Code).

You can also perform copies to virtual devices. (This requires the TF/Snap Licensed Feature Code).

Options to consider

The SNAP VOLUME command enables you to specify:

- Whether to wait for a session if all four sessions are in use.
- Whether to wait for the volume to be completely copied prior to the program terminating.
- A device to be snapped, identified by volser (SOURCE and VOLUME parameters).
- An existing target device identified by volser (TARGET and VOLUME parameters).
- Whether data from the source device is to overwrite data on the target device (REPLACE parameter).
- Whether the source volser is to be copied to the target with the SNAP (COPYVOLID parameter).
- Whether a background copy begins after the source and target are selected, prior to the activate operation (PRECOPY parameter).
- Whether the snap session should be automatically terminated as soon as the background copy is complete. (TERMINATE_SESSION_WHEN_COMPLETE parameter).
- That diagnostic messages and trace records are to be generated (TRACE and DEBUG parameters).
- Whether to request asynchronous notification upon completion of the SNAP VOLUME operation using the NOTIFY parameter.
- Whether the snap is to a virtual device.
- Whether the snap is differential, which only copies tracks changed since the last differential snap.
- Whether the virtual device is mounted and online at the end of the snap.
- Whether a snap can occur in a remote storage system that is connected by network to a local storage system (SYMDV# and REMOTE parameters).
Important points

Keep the following points in mind when you use SNAP VOLUME:

- You must define source and target volumes to emulate identical models. For example, you can snap a 3390 volume to another 3390; but, you cannot snap a 3390 to a 3380 device.

- The source and target volumes must be located in the same storage system for the internal snap operation to be effective. Otherwise, you must specify, and have available, a datamover utility to perform the actual track copy operation. The target can be a virtual device.

- You can use the COPYVOLID(YES) parameter to copy the source volume label to the target volume and have the target volume varied offline after the snap completes. Use the COPYVOLID(NO) to:
  - Restore the target volume label.
  - Vary online the target volume.

- If you use the MODE(NOCOPY) parameter, you may find it necessary to cause the MODE(NOCOPY) snap to normally complete by running the original snap job (JCL and control cards) and adding PARM='GLOBAL MODECOPYFINISH' to the PGM=SNAP execute statement.

- If a target volume is shared by more than one host, that target volume should be offline to all other hosts but the one from which you issue the command.

- A SNAP VOLUME command places the target volume in a Hold status. To remove the Hold status, use the RELEASE(YES) option on the CONFIG command after the snap is complete or has been stopped. The Hold status cannot be released while there are any indirect tracks on the volume.

**Note:** You can use the AUTOMATIC_RELEASE_HOLD parameter to request ResourcePak Base to monitor the background copy and to automatically release the Hold when the copy is complete. "SNAP VOLUME" on page 300 provides more information about the AUTOMATIC_RELEASE_HOLD parameter.

- If the target devices are FBA, they are offline to all mainframe systems. This ensures that the cache information kept on the target volume by other hosts remains unaffected by the snap operation.

- You can replace existing target volumes that contain datasets using optional SNAP VOLUME command parameters.
◆ TimeFinder provides a SRDF/A R2 Wait for Precopy feature with SNAP VOLUME. SRDF/A R2 Wait for Precopy is intended to address a situation when too many protected tracks occur on an SRDF/A R2 device. To minimize any possible issues, TimeFinder now requires that you specify the following parameters to snap from an SRDF/A R2 device:
  ■ PRECOPY(YES)
  ■ MODE(COPY)
  ■ WAIT_FOR PRECOPY_PASS1(YES)

◆ With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, a TF/Clone and TF/Snap off an active SRDF/A R2 device is supported with device level pacing set by SRDF HC commands. See the SRDF Host Component for z/OS Product Guide for more information.

◆ All active TF/Snap operations with SRDF/Metro devices will be blocked.

◆ SNAP VOLUME is blocked if Dynamic Volume Expansion (DVE)\(^1\) is active on a requested device.

Thick and thin device support

TimeFinder currently supports both thick and thin FBA and CKD devices for clone operations.

For both FBA and CKD devices, TimeFinder allows:

◆ Thick or thin device to thick or thin device operations

◆ Thick or thin device to virtual device operations

You can perform these thick and thin device operations with SNAP VOLUME, STOP SNAP VOLUME, and CONFIG. There is no new device syntax. Using regular source and target notation, if a thin device is selected, it is utilized.

Thin devices may be used as gatekeepers in PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876.

TF/Clone and TF/Snap supports operations between thin and non-thin (thick) devices.

There are some restrictions on using thin devices:

◆ Data devices are not allowed to be the source or target of any TF/Clone or TF/Snap operations.

---

Note: TimeFinder provides a OPT_TDEV site option that determines whether thin devices are to be included in reports generated by the QUERY VOLUME command. “OPT_TDEV” on page 78 provides more information about this site option.

---

\(^1\) For Mainframe Enablers 8.0, 8.1, and 8.2: you can perform DVE operations using Dell EMC Solutions Enabler, Unisphere for PowerMax, or Unisphere for VMAX.
Extended address volumes

SNAP VOLUME and all TimeFinder components can perform operations against extended address volumes (EAVs).

Diskless SRDF devices

SNAP VOLUME and all TimeFinder components recognize diskless SRDF devices; but, do not perform operations against diskless SRDF devices. If you attempt to issue a command against a diskless SRDF device, TimeFinder logs an error.

Offline volume support

TimeFinder can process offline volumes. The volumes may be CKD or FBA devices. CKD volumes can only be snapped to CKD volumes of the same size or larger. FBA volumes can only be snapped to FBA volumes of the same size.

The syntax of the SNAP VOLUME and STOP SNAP TO VOLUME statements remains unchanged. The CLEANUP [EXTENT TRACK ON] has been enhanced with the UNIT parameter.

When you use the UNIT parameter, the indicated device may be offline. If you use both the UNIT and VOLUME parameter in a SNAP VOLUME request for a CKD device and the volume is offline, the volume label is read and verified before the execution of the snap.

IMPORTANT
You should not use the VOLUME parameter with a FBA device.

TimeFinder ignores the CONDVOL, COPYVOLID, REFVTIOC and REPLACE parameters if you specify them when snapping a FBA device. The ADRDSSU, DFDSS, DSS, FDR and FDRDSF DATAMOVERNAMEs do not work correctly if you specify them with FBA devices. A DATAMOVERNAME of COPYCYL or COPYTRK can be used with FBA devices.

SNAP DATASET supports offline source devices when SOURCE_VOLUME_LIST is used. The target device must still be online. See the section “Snaps from offline or cloned volumes” on page 136 for more information.

Full-device resnap operations

A resnap is basically any snap operation of a source dataset, or device to a target dataset, or devices that were the source and target of a previous snap. A requirement of this feature is the original snap operation, as well as subsequent resnap operations that are differential.

You can perform a full-device resnap operation from the same source device to the same target device as used in an original snap operation while there are still protected and indirect tracks present.

However, you cannot perform full-device resnap operations using the original target device as the source device and another device as the target until the original background copy has completed.
For example, you can do a full-device resnap of device A to device B before a previous snap of device A to device B is complete. However, you cannot execute a full-device resnap of device B to device A or to device C before the previous snap of device A to device B is complete.

Reminder regarding license requirements

- To use SNAP VOLUME to perform full-volume snaps, install the TF/Clone licensed feature code.
- To use SNAP VOLUME to perform virtual-device operations, install the TF/Snap licensed feature code.
- To use the TARGET parameter with a clone or a virtual-device snap, install the TF/Clone licensed feature code.

Incremental clone restore

Although TF/Snap and TF/Mirror both have a RESTORE command, TF/Clone does not have a separate RESTORE command. A TF/Clone restore is achieved by reversing the source and target volumes and performing a “snap back”.

In the situation where you have created a full clone or increment using the DIFF option, and at some point you want to copy the clone target volume back to the original source device, you can take advantage of the automatic restore feature. PowerMaxOS 5978, HYPERMAX OS 5977, or Enginuity 5876 is required, and the background copy must be complete prior to starting the restore.

By executing a SNAP VOLUME with DIFF (YES) in the opposite direction, the need for a differential synchronization is automatically recognized.

For example, if the original SNAP command was:

```plaintext
SNAP VOLUME (SOURCE (UNIT(1234))
          TARGET (UNIT (3456))DIFFERENTIAL(YES))
```

Once the copy from the source to the target is complete, and there are no protected or indirect tracks on the source and target drives, you can copy back the volume using SNAP’s incremental clone restore feature by executing the following command:

```plaintext
SNAP VOLUME (TARGET (UNIT(1234))
          SOURCE (UNIT (3456))DIFFERENTIAL(YES))
```

After the restore (snap back) is completed, the original SNAP VOLUME statement can be used to reverse the direction again.

**Note:** When the DIFFERENTIAL keyword is used, a full copy is avoided after the initial synchronization.
Multidevice operations

Standard thin and thick devices are supported for multidevice operations.

TimeFinder has a site option, &AUTOACTIVATE, and parameter, AUTOMATIC_ACTIVATE, that:

- Automatically performs an ACTIVATE operation when there are two or more SNAP VOLUME commands in the input stream and no ACTIVATE command was issued.
- Causes the SNAP VOLUME requests to be processed together.

The default for AUTOMATIC_ACTIVATE is YES. If you do not want to use &AUTOACTIVATE or AUTOMATIC_ACTIVATE, set the value to NO.

Even if you use the YES default, there are some limitations:

- SNAP VOLUME ignores AUTOMATIC_ACTIVATE(YES) for any requests that specify a group name.
- SNAP VOLUME ignores AUTOMATIC_ACTIVATE(YES) for any requests with a VDEV.

Note that AUTOMATIC_ACTIVATE does not provide a consistent activate. For the activate to be consistent:

- Issue a separate ACTIVATE command with the CONSISTENT parameter.
- Specify the CONSISTENT parameter on the GLOBAL command.

Note: “CONSISTENT(Yes|No)” on page 215 provides more information about the CONSISTENT parameter with the ACTIVATE command. “CONSISTENT(Yes|No)” on page 237 provides more information about the CONSISTENT parameter with the GLOBAL command.

SNAP/FlashCopy coexistence

TimeFinder allows SNAP and FlashCopy sessions to exist on the same volume. Previously, TimeFinder would detect whether a Snap or FlashCopy session already existed and would then use the appropriate method to ensure that the session types were consistent.

Sometimes this procedure would go against the required session setting in the site options table. For instance if the site options table has Snap as the preferred copy method, but a FlashCopy session already existed on the device, TimeFinder would use FlashCopy. Now that the sessions can coexist, the preferred method for copying as set in the site options table (the &EMCDSSU_FLASH_SNAP option) is always used.

Note: Table 2 on page 47 in this document provides more information about the site options table.

FlashCopy and Snap compatibility is not supported for cascading configurations in which the target device of one technology is used at the source device for the other.
**R21 device recognition**

TimeFinder recognizes R21 devices. An R21 device is a dual-role SRDF R1/R2 device used in Cascaded SRDF operations.

Cascaded SRDF is a three-site disaster recovery configuration where data from a primary site is synchronously replicated to a secondary site, and then asynchronously replicated to a tertiary site. The core benefit behind a “cascaded” configuration is its inherent capability to continue replicating from the secondary site to the tertiary sites in the event that the primary site goes down. This enables a faster recovery at the tertiary site.

Located at the secondary site, the R21 device simultaneously acts as an R2 device to the primary site and as an R1 to the tertiary site.

**Note:** Although you can perform a snap from a regular R21 device, you cannot perform a snap from a R21 “diskless” device.

**Note:** The *SRDF Host Component for z/OS Product Guide* presents more information about Cascaded SRDF.

Although you can perform snaps from R1, R2 and R21 devices, you cannot perform snaps to an R2 or R21 device, only to an R1 device.

**Concurrent R2 (R22) device recognition**

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, TimeFinder recognizes concurrent R2 devices. Concurrent R2 is an operating environment feature that allows an R2 device to have two SRDF mirrors. Each R2 mirror is paired with a different R1 mirror and only one of the R2 mirrors can be Read-Write on the link at a time.

**Note:** The *SRDF Host Component for z/OS Product Guide* provides more information about diskless R22 devices.

**Security considerations**

No change is required to the existing security process when implementing the SNAP VOLUME command. DASDVOL requests made to SAF verify access at the device level.

You can also make use of the EMCSAFI Security Interface and the SAF command security. The EMCSAFI Security Interface feature provides additional security checks for environments where multiple groups of users are using different devices in a single storage system.

**Note:** The *Mainframe Enablers Installation and Customization Guide* provides more information about the EMCSAFI Security Interface.
SNAP VOLUME with the COPYVOLID(NO) parameter

When a full device volume copy is performed, the microcode accepts a simple instruction to copy all of the tracks from the source device to the target device. When COPYVOLID(NO) is specified, it is the intent that the original target volser is retained on the target device. Physically speaking, this cannot happen when all of the tracks are being copied from the source device to the target device. So the retention of the original volser takes place logically, as follows:

- Prior to initiating the copy, the volser of the target device is read and retained in memory.
- Because the target device is physically changing identities, it is varied offline in order to reduce confusion.
- The microcode initiates the copy of the source device to the target device. Physically, this means that the target device now has the same volser as the source device.
- Once the microcode copy is initiated, the target device label is read. The contents are verified to ensure that they match the original source device. Then the target device label is updated with the original target device volser. This restores the target device volser to its original contents.
- The target device is now varied back online with its original volser.

SNAP VOLUME with the COPYV(N) and CONDVOL(ALL) parameters

When you specify COPYV(N) and CONDVOL(ALL) with SNAP VOLUME, the following additional changes are made after successful completion of the SNAP VOLUME command:

- If a VTOC index and VVDS are present and active on the target volume, TF/Snap 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 volumes.
- 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.
- TF/Snap 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.
- If a RESTORE VOLUME command with COPYVOLID(YES) parameter occurs in a JES3 environment, the target volume must be manually varied offline to JES3 after the RESTORE VOLUME completes.

During the RESTORE VOLUME operation with COPYVOLID(NO), the target volume is normally varied offline during the operation and varied online after the RESTORE VOLUME completes. You can link edit a user exit into TF/Snap to be invoked prior to the VARY ONLINE and VARY OFFLINE commands being issued. This exit is available for automating JES3 operations.
GCM support

Geometry Compatible Mode (GCM) allows SRDF relationships to be established between an FBA device on a storage system running Enginuity 5876 and an FBA device on a storage system running PowerMaxOS 5978 or HYPERMAX OS 5977, where the device under PowerMaxOS 5978 or HYPERMAX OS 5977 is exactly a half cylinder larger than the device under Enginuity 5876.

During snap copy operations, GCM attributes will be copied from source to target and target to source (restore) automatically.

Examples

Example 1 This example demonstrates snapping a volume:

```
SNAP VOLUME (SOURCE (VOL(USER00)) TARGET (VOL(BKUP75)))
```

Example 2 This example shows both a GLOBAL statement and a SNAP VOLUME statement. In this example:

- The GLOBAL statement sets the maximum acceptable return code to 4 and only issues a warning statement if the target volume is currently ONLINE to any other z/OS image in the complex.
- The SNAP VOLUME statement tells TimeFinder which volumes to use as the source and the target. Both the source and target volumes must be online to this z/OS image.
- The REPLACE(Y) parameter indicates that data on the target volume is to be completely overwritten.
- CONDVOL(ALL) and COPYVOLID(NO) cause the target volume serial number to remain MV3497. In addition, all of the pointers in the VTOC, IXVTOC and VVDS (if applicable on the target volume) are updated for all of the datasets snapped to the target. None of the datasets on the target volume are cataloged.
- The WAITFORCOMPLETION (Y,MESSAGES) cause the SNAP VOLUME step to remain active until the storage system completes the background copy of the source volume to the target volume and to issue status messages of the number of remaining tracks to be copied to the target volume.

```
GLOBAL MAXRC(4) CHKO(N)
SNAP VOLUME (SOURCE (VOLUME (MV3417)) TARGET (VOLUME (MV3497)) REPLACE(Y) CONDVOL(ALL) COPYVOLID(NO) WAITFORCOMPLETION(Y,MESSAGES)
```

Example 3 This example demonstrates a remote full device snap:

- The source volume at PowerMax/VMAX device number 00CE is copied to the target volume at PowerMax/VMAX device number 032E in the remote storage system.
- The remote storage system is found by using the gatekeeper found by using volume serial UMC001 in a local storage system, and then using RAGROUP(21) to determine the remote storage system.
The storage system serial number is an extra check to ensure that the correct storage system is being used for the remote full volume snap.

SNAP VOLUME (SOURCE (SYMDV# (00CE)) TARGET(SYMDV#(032E)) - REMOTE(VOL(UMC001) RAGROUP(21) CONTROLLER(0001879-90171) ) )

**Example 4**

This example employs SNAP VOLUME to snap a volume from source to target. The target volume retains its original volser and is made available to the host.

**Note:** You can use this example only if you have purchased the licensed feature code for the keyword (parameter) TARGET.

```plaintext
// JOB
//QCOPYRUN EXEC PGM=EMCSNAP
//STEPLIB DD DISP=SHR,DSN=DS-PREFIX.LINKLIB
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
GLOBAL MAXRC(4) 
*
SNAP VOLUME ( SOURCE( VOLUME ( YOUR-SOURCE-VOLUME) ) - TARGET( VOLUME (BCV-VOLUME) ) - COPYVOLID(N) )
/*
```

**Example 5**

This example performs a remote SNAP VOLUME. This is an operational job that shows some of the options.

```plaintext
//RMTSNAP EXEC PGM=EMCSNAP
//STEPLIB DD DISP=SHR,DSN=EMC.SSNP.V580.LINKLIB
//MV6C00 DD DISP=SHR,UNIT=3390,VOL=SER=MV6C00
//SYSPRINT DD SYSOUT=* 
//EMCQCAPI DD SYSOUT=* 
//SYSABEND DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
GLOBAL MAXRC(4) CHKO(N) AUTOMATIC_RELEASE_HOLD(YES) - CHECKBCVHOLDSTATUS(NO)
* 
* SNAP REMOTE DEVICE # 0000 to DEVICE # 0001 USING the DMX 
* INTERNAL PATH THROUGH RAGROUP 17 WITH UNIT 6C00 
* THE REMOTE DMX SERIAL NUMBER IS 90132 
* SNAP VOLUME (SOURCE (SYMDV# ( 0000)) - TARGET (SYMDV# ( 0001)) - REMOTE (RAGROUP(17) CONTROLLER (90132)) - REMOTE (UNIT (6C00)) - TOLERATEENQFAILURE(YES) - COPYVOLID(NO) - REPLACE (YES) - THIS STATEMENT AND THE NEXT ONE ARE BOTH COMMENTS.. 
* DATAMOVERNAME(DFDSS) - WAITFORSESSION(YES) -
* 
* 
/*
```
Performing a SNAP VOLUME using virtual devices

Virtual devices (VDEV) are PowerMax/VMAX devices that are represented by a collection of pointers, as shown in Figure 5.

![Figure 5](image)

Virtual devices have the following characteristics:
- Are configured in the storage system
- Have PowerMax/VMAX device numbers and host channel addresses
- Do not reserve space equal to their size
- Share common snap pool devices to store new writes to the source or target virtual device

A virtual device snap creates a point-in-time image of the source device that only consumes space for new writes to the source or the target virtual device. As a result, virtual device snaps can consume much less space than full device snap.

Tracks that are updated on the source after the snap cause the pre-update image of the updated tracks to be copied from the source to the snap pool device. Tracks that are updated on the virtual device have the updated, or post-image track, written to the snap pool device.

For devices with very low change rates (total number of tracks changed, not total write activity for the volume), virtual devices can provide a space-efficient way to capture one or more point-in-time copies of a logical volume. They are best used as a complement to TF/Clone full volume copies.

Virtual devices that are associated with a source device can be mounted, read from, written to, and varied online or offline. Virtual devices that are not associated with a source device remain offline and not ready.
SNAP VDEVice example

You can use this example only if you have purchased TF/Snap for z/OS and its licensed feature code for the keyword (parameter) VDEV, along with the LFC for ECA.

This example employs SNAP VOLUME VDEVice to snap from a volume to a virtual device. The virtual device has a new volser and is made available to the host.

The example contains two snap operations:

**In the first snap operation:**
- There is a query of all of the VDEVS (virtual devices) and of all of the snap pool devices before the SNAP VOLUME VDEVice and after the SNAP VOLUME VDEVice.
- The source volume on A00A is snapped to virtual volume A04A.
- The BCV HOLD status is not checked.
- Freespace is not copied from the source to the target.
- If no session is available for the source volume copy, TF/Snap waits for a session.
- The session is completed with messages before the step is completed.
- The new volume ID is UWC0AA.
- The data on A04A is replaced by the data on A00A.

**In the second snap operation:**
- The source volume at A00B is snapped to a virtual volume at A04B.
- The BCV HOLD status is not checked.
- Freespace is not to be copied.
- If no session is available for the source volume copy, TF/Snap waits for a session.
- The session is completed with messages before the step is completed.
- The new volume ID is UWC04B.
- The data is replaced on the VDEV.

**For the ACTIVATE command:**
- These two commands are activated at the same time using ECA assist to ensure that each volume is consistent.
- A message is displayed to show when consistency is completed.
- An ECA time-out value of 15 seconds maximum is set.
Performing a SNAP VOLUME using virtual devices

// JOB
//QCOPYRUN EXEC PGM=EMCSNAP, REGION=0M
//STEPLIB DD DISP=SHR, DSN=DS-PREFIX.LINKLIB
//SYSPRINT DD SYSOUT=*  
//SYSDUMP DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
/*

QUERY VDEV (LOCAL (UNIT (A04A)))
QUERY SNAPPool (UNIT (A04A))
SNAP VOL(SOURCE (UNIT (A00A)) -
CHECKBCVHOLDSTATUS (N) -
FREESPACE (N) -
WAITFORSESSION (Y) -
WAITFORCOMPLETION (Y, MSG) -
NEWVOLID (UWC0AA) -
REPLACE (Y) -
VDEV (UNIT (A04A)) )

SNAP VOL (SOURCE (UNIT (A00B)) -
CHECKBCVHOLDSTATUS (N) -
FREESPACE (N) -
WAITFORSESSION (Y) -
WAITFORCOMPLETION (Y, MSG) -
NEWVOLID (UWC0AB) -
REPLACE (Y) -
VDEV (UNIT (A04B)) )
ACTIVATE (CONSISTENT (YES) MSG (DIS) TIMEOUT (15))
QUERY VDEV (UNIT (A04B))
QUERY SNAPPool (UNIT (A04A))
*/
Performing Cascaded clone operations

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, TimeFinder allows for *cascaded clone* operations. This allows a clone operation to take place with a device that is already involved in a clone operation without ending the first clone session.

For instance, as shown in Figure 6, you can use TimeFinder to clone device A to device B. Then, while the relationship between A and B is preserved, clone device B to device C.

A cascaded relationship is implied in a snap from A to B and then from B to C. After the A to B snap has finished, the A to B cascaded operation may still be active. In this situation, you can encounter a problem when you try to perform a cascaded-like snap from B to C and then from A to B.

In both cases, B becomes the “middle” of an extended relationship. In the traditional cascaded situation, B is the target of a persistent relationship (like differential). In the second, cascaded-like case, B is the source of a persistent relationship (like differential). Neither situation is allowed by the operating environment.

TimeFinder always attempts to ensure that both situations work. However, there are times that one must fail. For instance, if C is a virtual device (VDEV), in a snap of B to C and A to B, TimeFinder does not delete virtual device C. C may be used for more than one purpose. Instead, TimeFinder fails the snap of A to B.
Cascaded clone emulation

PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876 allow a cascaded-like operation for clone emulation. With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, you can cascade from a device involved in a clone operation to a device involved in a clone emulation operation, as shown in Figure 7 on page 115.

**Note:** Cascaded clone emulation to clone is allowed with PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876.

![Figure 7: Cascaded clone to cascaded clone emulation](image)

Clone 1: The relationship between A and B is preserved, allowing differential resynchronization following Clone Emulation 2.

Clone Emulation 2: Operation is allowed after Clone 1 copy is finished.

However, as shown in Figure 8, you cannot cascade from a device involved in a clone emulation operation to a device involved in a clone operation.

![Figure 8: Cascaded clone emulation to cascaded clone](image)

Clone Emulation 1: Operation is allowed.

Clone 2: Operation is not allowed.

For instance, you can use TimeFinder to clone device xxx to A (an STD device). Then, after the xxx to A copy has completed but while the xxx to A session is still in effect, use TF/Mirror clone emulation to clone A to B (a BCV device). However, you cannot then use TF/Clone Mainframe SNAP Facility to clone B to C.

In addition, as shown in Figure 9, operations A to B and B to C cannot both be clone emulation.

![Figure 9: Cascaded clone emulation to cascaded clone emulation](image)

Clone Emulation 1: Operation is allowed.

Clone Emulation 2: Operation is not allowed.
Operations and Examples

Requirements

There are a few requirements to keep in mind:

◆ There cannot be any indirect tracks on the source device.
◆ There cannot be any protected tracks on the target device.
◆ A virtual device (VDEV) cannot have a session with any target device.
◆ If the new source device is already a target of another operation, the corresponding session must be active.

Limitations

Although circular cascading (A→B→A) is not allowed, devices A and B can have multiple targets. For example: A→B→C(1) and A→B→C(2) and A→B→C(3).

There is also a limitation on full-volume, Incremental FlashCopy. If A→B is full-volume FlashCopy, device B cannot be used as the source of a FlashCopy command.

Example

Consider the following example of cascading three devices:

1. You issue SNAP VOLUME with the DIFFERENTIAL parameter set to snap device AAA to device BBB.
2. After ensuring that all background copy operations have completed, you issue SNAP VOLUME with the DIFFERENTIAL parameter set to snap device BBB to device CCC. You then ensure that all background copy operations have completed.
3. You make changes to the data on devices AAA and BBB.
4. You issue SNAP VOLUME with the DIFFERENTIAL parameter set to snap device AAA to device BBB.
5. After ensuring that the changes made to device AAA are on device BBB and that all background copy operations have completed, you issue SNAP VOLUME with the DIFFERENTIAL parameter set to snap device BBB to CCC.
6. After ensuring that the changes made to device AAA are on device CCC, you issue a STOP SNAP to all volumes.
Influencing SMS volume selection

When the Dell EMC volume preferencing exit routine is activated, TimeFinder attempts to influence SMS volume selection. Volume preferencing causes a target volume to be selected from a device that is on the same storage system as the source, even though the SMS storage group contains target volumes on multiple storage systems.

The following Dell EMC components are involved when influencing SMS:

- Utility program—Used to activate, inactivate, and query the status of the exit routine.
- Exit routine—Used to tailor the SMS volume candidate list in an attempt to limit allocations to the storage system containing the source dataset/volume.
- TimeFinder—Recognizes when the exit routine is active and prepares information about the target allocation and the target storage system for use by the exit routine.

**EXTENT_ALLOCATION**

The result of influencing target device selection is dependent on whether EXTENT_ALLOCATION(YES) is specified or internally selected. Volume preferencing is **not** used with EXTENT_ALLOCATION. It does not even need to be activated.

**Note:** When necessary, you may select EXTENT_ALLOCATION internally for some SMS controlled dataset types, such as striped extended format datasets.

When requested or selected, EXTENT_ALLOCATION(YES) causes TimeFinder to attempt to select a target device using the following sequence:

1. Target device in the same storage system or control unit as the source device.
2. Target device in any storage system.
3. Any eligible device.

If required, a TimeFinder site default is available, SAMEONLY, to fail the allocation if the target BCV or STD is not in the same storage system or control unit.

**Note:** Table 2 on page 47 lists the site options.

**Internal selection**

When you do not use EXTENT_ALLOCATION but activate volume preferencing, TimeFinder attempts to:

- Influence the allocation by examining the SMS candidate device list.
- Select the same storage system or control unit for the target device as the source device.

This mode of allocation offers no opportunity to fail the request if the same storage system condition cannot be met. Eventually, allocation continues on any eligible target device where space is available.
Installation and activation of the Dell EMC volume preferencing exit routine (EMCVLPRF) is performed by executing the utility program (EMCSNPVS). EMCSNPVS accepts its commands through the parameter field and displays its responses on the console. Each execution of EMCSNPVS performs one command.

Exit routine activation should be done automatically after system IPL. It is only necessary to activate the exit routine once after an IPL.

**Note:** These modules are shipped as LINKLIB members of the EMCSCF component of ResourcePak Base for z/OS.

## Set and query volume preferring

The syntax of the EMCSNPVS volume preferring commands are:

**Query Volume Preferencing**

```plaintext
QUERY VOLumePREFerencing
```

**Set Volume Preferencing**

```plaintext
SET VOLumePREFerencing ( [ STATus ( Active | Inactive ) ]
[ DEBUG ( OFF | ON ) ]
[ TRACE ( OFF | ON ) ] )
```

The QUERY command displays whether the exit routine is installed and enabled. The SET command can activate and inactivate the exit routine. Normally, the exit code is completely removed when the exit routine is deactivated.

### Examples

The following example displays the status of the Dell EMC volume preferencing exit:

```plaintext
//TSTSNVPS JOB (EMC),,MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//VOLPROF EXEC PGM=EMCSNVPS,PARM=' QUERY VOLPREF '
//STEPLIB DD DSN=EMC.SCFvrm.LINKLIB,DISP=SHR
```

The following example activates the Dell EMC volume preferencing exit. Note that the STEPLIB DD-statement must point to the library containing the EMCVLPRF program. This may be run as a batch job stream or a started task:

```plaintext
//USRSNVPS JOB (EMC),,MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//VOLPROF EXEC PGM=EMCSNVPS,PARM=' SET VOLPREF (STATUS(ACTIVE))'
//STEPLIB DD DSN=EMC.SCFvrm.LINKLIB,DISP=SHR
```

The following example inactivates the Dell EMC volume preferencing exit:

```plaintext
//USRSNVPS JOB (EMC),,MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//VOLPROF EXEC PGM=EMCSNVPS,PARM=' SET VOLPREF (STATUS(INACTIVE))'
//STEPLIB DD DSN=EMC.SCFvrm.LINKLIB,DISP=SHR
```
Selecting volume processing by phases

The SNAP VOLUME and ACTIVATE commands have two parameters, PRESNAP and POSTSNAP, that allow you to select SNAP VOLUME processing by phases:

- Presnap
- Activate
- Postsnap

By using groups, you can run these phases individually to ensure that:

- The correct volume list is used in all phases.
- The processing phases are scheduled appropriately.

For example, before a nightly backup, you can execute the presnap phase and allow the precopy to take place in the background. Then, later, you can execute the short activate phase and follow it with the postsnap phase to make the snapped devices available.

All three phases must complete before the target volume(s) are available; but, this allows them to be scheduled in a way to minimize impact on other workloads.

PRESNAP and POSTSNAP parameters only apply to regular input (after a //QCINPUT DD * JCL statement) SNAP VOLUME statement that references a GROUP, and are only valid if GROUP is also specified. The GROUP parameter identifies a set of stored statements that are to be executed, while the PRESNAP and POSTSNAP indicate some special processing for the GROUP. This is why these parameters cannot be stored within a group definition.

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

Presnap processing

Presnap processing 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.

Usually, you would also specify (or default) the PRECOPY parameter and the background copy would begin after the “establish” is accepted by the storage system. The target device would not be available from this point until the postsnap phase is executed.
Activate processing

Activate processing involves making sure that the source and target device pairs have an existing session that has been “established” but not “activated.” Then the “activate” operating environment request is used to enable the session.

Note: “ACTIVATE” on page 214 provides more information about the ACTIVATE command.

Postsnap processing

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

Phase processing and group status

Table 3 shows how the group status interacts with various phase processing. For instance, postsnap processing is not allowed unless activate has already been performed. As in the past, a simple SNAP VOLUME with no PRESNAP or POSTSNAP parameters performs all three phases together. An ACTIVATE group with PRESNAP and POSTSNAP also performs all three phases together.

<table>
<thead>
<tr>
<th>Group status (before)</th>
<th>Action</th>
<th>PRESNAP</th>
<th>POSTSNAP</th>
<th>Group status (after)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL</td>
<td>ACTIVATE</td>
<td>YES</td>
<td>NO</td>
<td>ACTIVATE</td>
</tr>
<tr>
<td>INITIAL</td>
<td>ACTIVATE</td>
<td>YES</td>
<td>YES</td>
<td>POSTSNAP</td>
</tr>
<tr>
<td>PRESNAP</td>
<td>ACTIVATE</td>
<td>YES or NO (Ignored when PRESNAP is already group status.)</td>
<td>NO</td>
<td>ACTIVATE</td>
</tr>
<tr>
<td>PRESNAP</td>
<td>ACTIVATE</td>
<td>YES or NO (Ignored when PRESNAP is already group status.)</td>
<td>YES</td>
<td>POSTSNAP</td>
</tr>
<tr>
<td>DEFINE GROUP</td>
<td>-N/A-</td>
<td>-N/A-</td>
<td>INITIAL</td>
<td></td>
</tr>
<tr>
<td>-any-</td>
<td>CLEANUP</td>
<td>-N/A-</td>
<td>-N/A-</td>
<td>INITIAL</td>
</tr>
<tr>
<td>-any-</td>
<td>CONFIG</td>
<td>-N/A-</td>
<td>-N/A-</td>
<td>INITIAL</td>
</tr>
<tr>
<td>INITIAL, FAILED, POSTSNAP</td>
<td>SNAP VOLUME</td>
<td>NO</td>
<td>NO</td>
<td>POSTSNAP</td>
</tr>
<tr>
<td>INITIAL, FAILED, POSTSNAP</td>
<td>SNAP VOLUME</td>
<td>YES</td>
<td>NO</td>
<td>PRESNAP</td>
</tr>
</tbody>
</table>
Examples

**Example 1**

The following example defines the groups TEST and TEST2:

```plaintext
DEFINE GROUP TEST (DESC ('SNAP TWO VOLUMES'))
GLOBAL FREESPACE(YES)
SNAP VOLUME (SOURCE (VOLUME(U6A230)) -
TARGET(UNIT(6FE6)) NEWVOLID(BAP000) )
SNAP VOLUME (SOURCE (VOLUME(U6A231)) -
TARGET(UNIT(6FE7)) NEWVOLID(BAP001) )
END GROUP

DEFINE GROUP TEST2 (DESC ('MAKE VIRTUAL COPIES')
SNAP VOLUME (SOURCE (VOLUME(U6A232)) -
VDEV (UNIT(6DC0)) NEWVOLID(VBAP00) )
SNAP VOLUME (SOURCE (VOLUME(U6A233)) -
VDEV (UNIT(6DC1)) NEWVOLID(VBAP01) )
END GROUP
```

**Note:** You can also use the same source to make up to 8 copies.

**Note:** In the previous example The VDEV parameter is only available if you purchase the TF/Snap Licensed Feature Code.

**Example 2**

The following example snaps TEST and TEST2 in three separate actions:

```plaintext
SNAP VOLUME ( GROUP ( TEST,TEST2 ) PRESNAP(YES))
**
ACTIVATE ( GROUP ( TEST,TEST2 ) -
PRESNAP(NO) POSTSNAP(NO) -
CONSISTENT(YES) MESSAGE(DISPLAY))
**
SNAP VOLUME ( GROUP ( TEST,TEST2 ) POSTSNAP(YES))
```

The following example stops the snap and performs cleanup on the groups:

```plaintext
STOP SNAP TO VOLUME ( GROUP ( TEST ,TEST2) )
**
CLEANUP EXTENT TRACK FOR GROUP ( TEST ) CLEANDIFF(YES)
**
CONFIG (GROUP(TEST,TEST2) RELEASE(YES))
```
Performing a SNAP DATASET copy

Use the SNAP DATASET command to create a copy of the specified dataset. Source and target devices must be the identical models. That is, you can snap a 3390 device to another 3390 device, but you can not snap a 3390 to a 3380 device.

Supported dataset types

The SNAP DATASET command can snap the following types of datasets:

- Direct access (DA) datasets
- Extended format sequential datasets
- Extended Partitioned datasets (PDSE)
- GDG base names and GDG datasets
- Extended format VSAM Keyed Sequential Datasets (KSDS)
- Partitioned (PO) datasets (TYPE=HFS is not supported)
- Physical Sequential (PS) datasets
- Striped Sequential datasets
- BDAM datasets
- VSAM datasets:
  - Alternate Index (AIX)
  - ESDS
  - KSDS
  - Linear
  - RRDS
  - Spheres (KSDS | ESDS + PATHS + AIX)
  - VRRDS
  - Logical copy operations of IMBED, KEYRANGE and REPLICATE datasets are supported with DATAMOVER(DFDSS)
  - Datasets on offline volumes
- Undefined datasets (only with the FORCE(YES) option on the SNAP DATASET command)

Note: zFS datasets are supported by SNAP DATASET; however, it is recommended to quiesce the zFS file prior to issuing a SNAP DATASET command against it.
Unsupported dataset types

SNAP DATASET does not support snapping the following types of datasets:

- Concatenated datasets
- ISAM datasets
- Individual members of partitioned datasets
- Open Edition HFS datasets
- Page datasets
- VSAM Volume datasets (VVDS)
- VTOCs
- VTOC indexes
- Encrypted datasets

The SNAP DATASET command allows you to use a ddname or dataset name to designate the source dataset and the target dataset. If the target dataset already exists, it may be reused or erased and a new one allocated, depending upon the replace and reuse parameters.

In most circumstances, the SNAP DATASET operation is not affected by the size of a single extent. However, the directory must reside within the first extent of a partitioned dataset. As a performance consideration, SNAP DATASET does not check the member directory size, but issues a warning message when the first extent of the target PDS is smaller than the first extent of the source.

Note: SNAP DATASET is blocked if Dynamic Volume Expansion (DVE)¹ is active on a requested device.

SNAP DATASET options and operations

The following sections discuss SNAP DATASET options and operations.

Source and target datasets

Source datasets are identified by either:

- The SOURCE parameter
- The INDDname parameter

Unless SOURCE_VOLUME_LIST is specified, all datasets identified by the SOURCE parameter must be cataloged. You can only specify uncataloged datasets with the INDDname parameter, or by specifying the SOURCE_VOLUME_LIST parameter.

Target datasets are identified by either:

- The TARGET parameter
- The OUTDDname parameter

A dataset identified by the OUTDDname parameter is always reused. An existing dataset identified by the TARGET parameter may be reused or replaced, depending on the REUSE and REPLACE parameter settings. A new target dataset may be created and not cataloged, except in an SMS environment, where non-cataloged datasets are not allowed.

¹ For Mainframe Enablers 8.0, 8.1, and 8.2: you can perform DVE operations using Dell EMC Solutions Enabler, Unisphere for PowerMax, or Unisphere for VMAX.
Operations and Examples

SMS classes

SNAP DATASET allows you to specify SMS classes. If you specify SMS storage, data, or management classes on the SNAP DATASET command, TimeFinder supplies the classes to DYNALLOC or IDCAMS during allocation. You can use the COPYSOURCESMSCLASSES (COPYSMS) parameter to indicate which classes are to be obtained from the source (SMS managed) dataset.

---

Note: The *TimeFinder Utility for z/OS Product Guide* provides more information about IDCAMS.

---

The COPYSMS parameter does not work with TimeFinder and an alternate index dataset. This is because SMS does not record the class information when an alternate index dataset is created.

SMS ACS rules are not modified to accomplish a snap operation.

TimeFinder honors additional IBM Systems Managed Storage volume states, DISNEW and QUINEW.

---

Note: “Influencing SMS volume selection” on page 117 provides more information regarding volume selection.

---

WAIT options

SNAP DATASET also allows you to specify wait options. There are three circumstances where a built-in wait may be desirable. Each of these must be addressed individually:

- Waiting for a source dataset enqueue to become available, ensuring that the snap occurs *cleanly*.
- When the volume or dataset has four snaps occurring for the same track range.
- When a background snap operation has started and it is desirable to wait for the background snap operation to complete.

**Waiting for a source dataset enqueue**

Waiting for a source dataset enqueue only applies to the SNAP DATASET command. By default, the snap operation waits for the source dataset until it becomes available.

The type of enqueue is determined by the HOSTCOPYMODE parameter. You may set this parameter to:

- EXCLUSIVE
- SHARED
- NONE

The parameter ENQWAIT controls whether the enqueue must successfully complete before proceeding. If you specify ENQWAIT(NO) and the HOSTCOPYMODE indicates EXCLUSIVE or SHARED, then failure to obtain the enqueue results in an action based upon the TOLERATEENQFAILURE parameter.

The following list outlines this flow of events.

1. If you specify HOSTCOPYMODE(NONE), no additional processing is required.
2. If you specify ENQWAIT(YES), an enqueue is issued that must be satisfied before proceeding. After the enqueue is satisfied, no additional processing is required.

3. If you specify ENQWAIT(NO), issue an enqueue to acquire/test the availability of the source dataset.
   - If you acquire the enqueue for the dataset, no additional processing is required.
   - If the enqueue for the dataset is not available, you must test the TOLERATEENQFAILURE parameter.
     - If you specify TOLERATEENQFAILURE(NO), an error message is written and the SNAP DATASET action terminates.
     - If you specify TOLERATEENQFAILURE(YES), a warning message is written and processing continues.

Four snaps occurring in the same track range

A maximum of four snap operations may be active for a range of tracks at any given moment. For example, you can snap a single source dataset to four new datasets without any problems. A fifth snap of that same source dataset may not begin until one of the four previous snaps has completed.

Note: This limitation also applies to SNAP VOLUME.

By default, if a range of tracks is already involved in four snap operations, a request to snap it a fifth time fails. You can control this by using the WAITFORSESSION parameter. The WAITFORSESSION parameter indicates how to handle the fifth and succeeding snap operations. WAITFORSESSION(NO) is the default setting. However, if you want to wait for one of the prior snap operations to complete, you can specify WAITFORSESSION(YES). In addition, you may indicate a time value and the operation is checked until the time period expires.

For example, WAITFORSESSION(5:0) indicates that the snap operation waits up to 5 minutes for a prior operation to complete. At the end of that time period, if the snap operation is unable to start the new snap operation, it fails with an error message.

TimeFinder may optionally wait for the actual copy operations to complete. This is done by polling the storage system periodically and checking the status of the copy operations. You can specify the parameter WAITFORCOMPLETION on:

- The GLOBAL command
- The RESTORE VOLUME command
- The SNAP DATASET command
- The SNAP VOLUME command

At program termination, after all copy operations have been initiated, TimeFinder makes a final check to wait for the copy operations. An optional subparameter indicates whether the remaining tracks to be copied are logged as each check is made. The amount of time between checks is dependent upon the number of tracks remaining to be copied. The more tracks, the longer the time period.
All datasets or volumes are checked one at a time in the same sequence as the original copy operation. After a dataset or volume copy is complete, then the next dataset or volume copy is checked.

**Session limit**

Under PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, TimeFinder no longer strictly enforces the limit of four full-device sessions for any one device. This means that you can create up to 16 simultaneous, full-device copies of a single source device as long as no other sessions exist on the device.

A single device is limited to an absolute maximum of 16 sessions of various types. This does not affect the current limit of four extent sessions or eight virtual device sessions, except to limit the total number of full device sessions for a single volume to 16 or less.

This does not affect the current limit of four full-device sessions of Enginuity 5773.

**Multivolume datasets**

TimeFinder can snap:

- A single volume source dataset to a single volume target dataset
- A multivolume source dataset to a multivolume target dataset
- A single volume source dataset to a multivolume target dataset
- A multivolume source dataset to a single volume target dataset

Special consideration is required when snapping a multivolume, extended-format, non-VSAM dataset with a stripe count of one. In this situation, the target dataset must have the exact same number of volumes and tracks allocated to each volume, corresponding to the source dataset.

For instance, if the source dataset is allocated to three volumes containing 3000, 2000, and 1000 tracks, the target dataset must also be allocated to three volumes containing 3000, 2000, and 1000 tracks. Otherwise, the snap operation fails.

Multistripe datasets must have the same number of stripes for the source and target dataset.

**Summary of multivolume SNAP DATASET scenarios**

The target attributes of all *dynamically allocated* datasets are influenced by z/OS allocation (UNITNAME, BCVGROUP, SCFGROUP, and/or VOLUME parameters) or SMS allocation (STORCLAS, DATACLAS, MGMTCLAS and/or ACS routines).
Table 4 summarizes the various possible SNAP DATASET scenarios.

### Table 4: Summary of multivolume SNAP DATASET scenarios (page 1 of 4)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same storage system</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset on the same storage system. Extent sizes may be different from source and may be consolidated.</td>
</tr>
<tr>
<td>Same storage system</td>
<td>Preallocated</td>
<td>Multivolume dataset on the same storage system. Determined by pre-allocation. If necessary to extend dataset, then z/OS automatically selects the last allocated or next candidate volume.</td>
</tr>
<tr>
<td>Same storage system</td>
<td>Extent Allocation</td>
<td>Multivolume dataset on the same storage system. Is determined by candidate list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES): Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_VOLUME): Each target volume contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_ALL): The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source.</td>
</tr>
<tr>
<td>Different storage system</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset on available volumes in one storage system. Is determined by candidate list. Extent sizes may be different from source and may be consolidated. A datamover is required to copy extents between different storage systems.</td>
</tr>
<tr>
<td>Different storage system</td>
<td>Preallocated</td>
<td>Multivolume dataset whose placement is determined by pre-allocation. If necessary to extend dataset, z/OS automatically selects the last allocated or next candidate volume. A datamover is required to copy extents between different storage systems.</td>
</tr>
<tr>
<td>Different storage system</td>
<td>Extent Allocation</td>
<td>Multivolume dataset on the different storage systems. Is determined by candidate list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES): Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent. Each target volume is also in the same storage system as its respective source volume.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_VOLUME): Each target volume contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume. Each target volume is also in the same storage system as its respective source volume.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_ALL): The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source. Each target volume is also in the same storage system as its respective source volume.</td>
</tr>
</tbody>
</table>
### Table 4 Summary of multivolume SNAP DATASET scenarios (page 2 of 4)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixed storage system and RVA or ESS</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset on available volumes in storage systems and RVA or ESS. Is determined by candidate list. Extent sizes may be different from source and may be consolidated. A datamover is required to copy extents between units. For RVA or ESS, TimeFinder invokes SNAPSHOT and/or FlashCopy if possible.</td>
</tr>
<tr>
<td>Mixed storage system and RVA or ESS</td>
<td>Preallocated</td>
<td>Multivolume dataset whose placement is determined by pre-allocation. If necessary to extend dataset, z/OS automatically selects the last allocated or next candidate volume. A datamover is required to copy extents between units. For RVA or ESS, TimeFinder invokes SNAPSHOT and/or FlashCopy if possible.</td>
</tr>
<tr>
<td>Mixed PowerMax/VMAX system and RVA or ESS</td>
<td>Extent Allocation</td>
<td>Multivolume dataset on the different storage systems and RVA or ESS. Is determined by candidate list. <strong>Using EXTENT_ALLOCATION(YES):</strong> Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent. Each target volume is in the same storage system or RVA and/or ESS as its respective source volume. If enough target volumes exist in the respective storage systems and/or RVA or ESS and IBM SNAPSHOT is available, it is not necessary to code a datamover. For RVA, TimeFinder invokes SNAPSHOT or FlashCopy if available. <strong>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_VOLUME):</strong> Each target volume contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume. Each target volume is in the same storage system or RVA and/or ESS as its respective source volume. If enough target volumes exist in the respective storage systems and/or RVA or ESS and IBM SNAPSHOT is available, it is not necessary to code a datamover. For RVA, TimeFinder invokes SNAPSHOT or FlashCopy if available. <strong>Using EXTENT_ALLOCATION(YES, CONSOLIDATE_ALL):</strong> The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source. Each target volume is in the same storage system or RVA and/or ESS as its respective source volume. If enough target volumes exist in the respective storage systems and/or RVA or ESS and IBM SNAPSHOT is available, it is not necessary to code a datamover. For RVA, TimeFinder invokes SNAPSHOT or FlashCopy if available.</td>
</tr>
<tr>
<td>Mixed PowerMax/VMAX system and non-PowerMax/VMAX</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset on available volumes in storage system. Is determined by candidate list. Extent sizes may be different from source and may be consolidated. A datamover is required to copy extents between storage systems.</td>
</tr>
<tr>
<td>Mixed PowerMax/VMAX system and non-PowerMax/VMAX</td>
<td>Preallocated</td>
<td>Multivolume dataset whose placement is determined by pre-allocation. If necessary to extend dataset, z/OS automatically selects the last allocated or next candidate volume. A datamover is required to copy extents between units.</td>
</tr>
</tbody>
</table>
### Table 4: Summary of multivolume SNAP DATASET scenarios (page 3 of 4)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixed PowerMax/VMAX system and non-PowerMax/VMAX</td>
<td>Extent Allocation</td>
<td>Multivolume dataset on the different storage systems and RVA and/or ESS. Is determined by candidate list. Using <strong>EXTENT_ALLOCATION(YES)</strong>: Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent. Each target is in the same storage system, RVA and/or ESS or other storage system as its respective source volume. A datamover is required to copy extents between non-PowerMax/VMAX system or RVA and/or ESS controllers. Using <strong>EXTENT_ALLOCATION(YES,CONSOLIDATE_VOLUME)</strong>: Each target volume contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume. Each target is in the same storage system, RVA and/or ESS or other controller as its respective source volume. A datamover is required to copy extents between non-PowerMax/VMAX system or RVA and/or ESS controllers. Using <strong>EXTENT_ALLOCATION(YES,CONSOLIDATE_ALL)</strong>: The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source. Each target is in the same storage system, RVA and/or ESS or other controller as its respective source volume. A datamover is required to copy extents between non-PowerMax/VMAX system or RVA and/or ESS controllers.</td>
</tr>
<tr>
<td>Same RVA and/or ESS</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset on the same RVA and/or ESS. Extent sizes may be different from source and may be consolidated. TimeFinder invokes SNAPSHOT and/or FlashCopy to copy extents.</td>
</tr>
<tr>
<td>Same RVA and/or ESS</td>
<td>Preallocated</td>
<td>Multivolume dataset on the same RVA and/or ESS. Determined by pre-allocation. If necessary to extend the dataset, z/OS automatically selects the last allocated or next candidate volume. TimeFinder invokes SNAPSHOT and/or FlashCopy to copy extents.</td>
</tr>
</tbody>
</table>
Operations and Examples

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same RVA and/or ESS</td>
<td>Extent Allocation</td>
<td>Multivolume dataset on the same RVA and/or ESS. Is determined by candidate list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent. TimeFinder invokes SNAPSHOT and/or FlashCopy to copy extents.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES,CONSOLIDATE_VOLUME):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each target volume contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume. TimeFinder invokes SNAPSHOT and/or FlashCopy to copy extents.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES,CONSOLIDATE_ALL):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source. TimeFinder invokes SNAPSHOT and/or FlashCopy to copy extents.</td>
</tr>
<tr>
<td>Non PowerMax/VMAX or RVA and/or ESS</td>
<td>Dynamically Allocated</td>
<td>Multivolume dataset anywhere. Extent sizes may be different from source and may be consolidated. A datamover is required to copy extents.</td>
</tr>
<tr>
<td>Non PowerMax/VMAX or RVA and/or ESS</td>
<td>Preallocated</td>
<td>Multivolume dataset anywhere. Determined by pre-allocation. If necessary to extend dataset, z/OS automatically selects the last allocated or next candidate volume. A datamover is required to copy extents.</td>
</tr>
<tr>
<td>Non PowerMax/VMAX or RVA and/or ESS</td>
<td>Extent Allocation</td>
<td>Multivolume dataset anywhere. Is determined by candidate list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each target volume has the same number of tracks used as its respective source volume. Each target extent matches the corresponding source extent. A datamover is required to copy extents.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES,CONSOLIDATE_VOLUME):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The target volumes contains the same number of tracks as the source volume; but, the number and size of extents on the target volume may be different from those on the source volume. A datamover is required to copy extents.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Using EXTENT_ALLOCATION(YES,CONSOLIDATE_ALL):</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The number of target volumes may be different from the number of source volumes and the number and size of the extents on the targets may not match those on the source. A datamover is required to copy extents.</td>
</tr>
</tbody>
</table>
Volser assignment for multivolume datasets defined with and without guaranteed space

The volser assignment for target multivolume datasets depends on whether the source and target dataset is:

- A VSAM or non-VSAM dataset
- Defined as having guaranteed or nonguaranteed space

The following tables show the resulting allocation following the snap of a multivolume non-VSAM and VSAM dataset where the source and target are guaranteed space or nonguaranteed space. Table 5 shows results of snapping non-VSAM datasets. shows the results of snapping VSAM datasets.

### Table 5  Snapping non-VSAM datasets

<table>
<thead>
<tr>
<th>If the source dataset</th>
<th>and the target dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>has guaranteed space</td>
<td>has guaranteed space, then</td>
</tr>
<tr>
<td></td>
<td>all volumes have specific volser.</td>
</tr>
<tr>
<td>has nonguaranteed space</td>
<td>the first volume has a specific volser. The other volumes are candidate volumes.</td>
</tr>
</tbody>
</table>

### Table 6  Snapping VSAM datasets

<table>
<thead>
<tr>
<th>If the source dataset</th>
<th>and the target dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>has guaranteed space</td>
<td>has guaranteed space, then</td>
</tr>
<tr>
<td></td>
<td>all volumes have specific volser.</td>
</tr>
<tr>
<td>has nonguaranteed space</td>
<td>all volumes have specific volser.</td>
</tr>
<tr>
<td></td>
<td>the first volume has a specific volser. The other volumes are candidate volumes.</td>
</tr>
</tbody>
</table>

Relative GDGs

You can use TimeFinder to manipulate a relative GDG if you use the INDDname and/or OUTDDname options instead of the SOURCE and TARGET options for snap by dataset. The relative GDG number specified in the INDD or OUTDD parameter are converted to an absolute GDG number by z/OS. When using a relative GDG, then only that generation is considered.

**Note:** TimeFinder does not create the model GDG statement when using datasets addressed by a relative GDG number.

The following examples illustrate the capabilities available for INDD and OUTDD. These examples are not complete TimeFinder statements (only the input and/or output DD statements are shown); they are samples that show the different capabilities available when versions of a GDG are required.
Operations and Examples

The examples are based on the existence of a GDG base for both the source and target datasets, with a limit of 5 generations. The source and target datasets have cataloged generations G0001V00 through G0005V00. The name of the source GDG is EMC.SOURCE, and the name of the target GDG is EMC.TARGET.

Example 1
Use the relative (+0) source dataset and create its associated target dataset.

```bash
//STDIN DD DISP=SHR,DSN=EMC.SOURCE(+0)
//QCINPUT DD *
SNAP (INDD(STDIN) -
  TARGET(EMC.TARGET.* ) -
other SNAP parameters)
```

Result: EMC.TARGET.G0005V00 is replaced.

Example 2
Create the relative (+1) source dataset and in another step in the same job, create its associated target dataset.

```bash
//SNAPSTEP EXEC PGM=EMCSNAP
//CREATE DD DISP=(NEW, CATLG),DSN=EMC.SOURCE(+1),UNIT=3390,
//  VOL=SER=STDVOL,SPACE=(CYL,(3,1)),.....
//*
//STDIN DD DISP=SHR,DSN=EMC.SOURCE(+1)
//QCINPUT DD *
SNAP (INDD(STDIN) -
  TARGET(EMC.TARGET.* ) -
other SNAP parameters)
```

Result: EMC.SOURCE.G0006V00 and EMC.TARGET.G0006V00 are created.

Example 3
Create the relative (+1) source dataset and in another step in the same job create a relative (+1) target dataset.

```bash
//SNAPSTEP EXEC PGM=EMCSNAP
//CREATE DD DISP=(NEW, CATLG),DSN=EMC.SOURCE(+1),UNIT=3390,
//  VOL=SER=STDVOL,SPACE=(CYL,(3,1)),.....
//*
//STDIN DD DISP=SHR,DSN=EMC.SOURCE(+1)
//OUTPUT DD DISP=(NEW, CATLG),DSN=EMC.TARGET(+1),UNIT=3390,
//  VOL=SER=OUTVOL,SPACE=(CYL,(3,1)),.....
//QCINPUT DD *
SNAP (INDD(STDIN) -
  OUTDD(OUTPUT ) -
other SNAP parameters)
```

Result: Assuming Example 2 completes before Example 3, EMC.SOURCE.G0007V00 and EMC.TARGET.G0007V00 are created.

Note: Keep in mind that Example 2 is a different job from Example 3.

The use of the INDD and OUTDD parameters in TimeFinder allows for many other options for both input to the snap and output from the snap process.
Candidate volume list

SNAP DATASET allows you to specify the ESOTERIC(UNITNAME), VOLUME, SCFGROUP and BCVGROUP parameters together. SNAP DATASET then uses the resulting list of volumes as a candidate volume list.

As SNAP DATASET processes each source and target dataset pair, it chooses the volumes used for target dataset allocation from the candidate volume list. SNAP DATASET determines the eligibility of a particular volume using the following criteria:

- A matching volume is found online.
- The volume is a BCV volume.
- The volume is on the same storage system as the source volume.
- The volume is of the same device type as the source volume.
- The volume track size is the same as the source volume track size.

After compiling a list of eligible volumes, SNAP DATASET determines the amount of free space for each of the eligible volumes. Then SNAP DATASET sorts the list based upon the amount of free space. Finally, TimeFinder selects the first VOLUMECOUNT number of volumes for use when allocating the target dataset.

A SNAP DATASET command places the target volume in a Hold status. To remove the Hold status, use the RELEASE option on the CONFIG command after the snap is complete or has been stopped. You cannot release the Hold status while there are any indirect tracks on the volume.

If you used the MODE(NOCOPY) parameter, you may find it necessary to cause the MODE(NOCOPY) snap to complete normally by:

- Running the original snap job (JCL and control cards)
- Adding PARM="GLOBAL MODECOPYFINISH" to the PGM=snap execute statement

**Note:** In an SMS environment, TimeFinder ignores the candidate volume list.

VSAM ENQ support

If you specify (or default) VSAMENQMODE (NONE) on SNAP DATASET, then TimeFinder performs no testing of the SYVSAM ENQ.

If you specify VSAMENQMODE (SHARED), an ENQ is issued with the SHR attribute. If you specify VSAMENQMODE (EXCLUSIVE), an ENQ is issued with the EXC attribute. If the ENQ is satisfied, processing continues normally.

After the request is processed, then a DEQ is issued to release the resource.

- If you specified (or defaulted) TOLERATEVSAMENQFAILURE(NO), an error message is issued and processing of the request terminates.
- If you specified TOLERATEVSAMENQFAILURE(YES), a warning message is issued and processing of the request is continues.

After the request is processed, then a DEQ is issued to release the resource. If the ENQ cannot be satisfied, the value of the TOLERATEVSAMENQFAILURE parameter determines what happens.
◆ If you specified (or defaulted) TOLERATEVSAMENQFAILURE(NO), an error message is issued and processing of the request terminates.

◆ If you specified TOLERATEVSAMENQFAILURE(YES), a warning message is issued and processing of the request continues.

The QNAME used is 'SYSVSAM'. The RNAME used is 'dsname|catalogname|L1|L2|L3|O', the same as used by VSAM for protecting resources opened for update purposes.

**Note:** IBM documentation provides more information about the QNAME and RNAME.

When a different job attempts to use the VSAM dataset while it is being snapped, results vary depending on the VSAM dataset share options and the VSAMENQMODE. **Table 7** provides more information.

**Table 7** VSAM dataset share options and the VSAMENQMODE

<table>
<thead>
<tr>
<th>VSAMENQMODE</th>
<th>Share options</th>
<th>Open mode</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR (1,3) or (1,4)</td>
<td>Input</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>SHR (1,3) or (1,4)</td>
<td>Update</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>EXC (1,3) or (1,4)</td>
<td>Input</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>EXC (1,3) or (1,4)</td>
<td>Update</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>SHR (2,3) or (2,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>SHR (2,3) or (2,4)</td>
<td>Update</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (2,3) or (2,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (2,3) or (2,4)</td>
<td>Update</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>SHR (3,3) or (3,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>SHR (3,3) or (3,4)</td>
<td>Update</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (3,3) or (3,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (3,3) or (3,4)</td>
<td>Update</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
<tr>
<td>SHR (4,3) or (4,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>SHR (4,3) or (4,4)</td>
<td>Update</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (4,3) or (4,4)</td>
<td>Input</td>
<td>File opened successfully.</td>
<td></td>
</tr>
<tr>
<td>EXC (4,3) or (4,4)</td>
<td>Update</td>
<td>File Open Error – IEC161I 052-084.</td>
<td></td>
</tr>
</tbody>
</table>
VSAM open indicator support

The VSAM open (for update) indicator is located in the VVDS. Whenever you open a VSAM cluster for update, the open (for update) indicator is set. When the VSAM cluster is successfully closed, the open (for update) indicator is reset.

If multiple jobs open the VSAM cluster for update, the last job to close the cluster resets the open (for update) indicator. If the last (or only) job fails, the open (for update) indicator is left set.

It is the responsibility of the next job that opens the VSAM cluster for update to perform a VERIFY. The VERIFY ensures that the metadata for the cluster matches the contents of the VSAM cluster.

A warning message is issued if a VSAM cluster is snapped and the VSAM open (for update) indicator is set.

Note: The number of extents does not matter. You can use the EXTENT ALLOCATION parameter on the GLOBAL or SNAP DATASET commands to snap non-VSAM datasets with a stripe count of one (1).

Dataset name masking

SNAP DATASET supports dataset masking for the SOURCE, EXCLUDE, TARGET, and RENAMEUNCONDITIONAL parameters following the DFDSS masking rules. Table 8 shows the DFDSS masking rules.

Table 8 DFDSS masking rules

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>* (single asterisk)</td>
<td>Takes the place of exactly one qualifier or indicates that you are specifying one part of a qualifier.</td>
</tr>
<tr>
<td>** (double asterisk)</td>
<td>Used with other qualifiers, denotes the nonexistence of leading, trailing and middle qualifiers, or that they play no role in the selection.</td>
</tr>
<tr>
<td>% (percent character)</td>
<td>Indicates a single-character.</td>
</tr>
</tbody>
</table>

Follow these rules when using asterisks in a qualifier:

- The maximum permissible asterisks in a qualifier is two.
- When two asterisks are present in a qualifier, they must be the first and last.

For example:

Valid qualifiers:

- **
- *A*

Invalid qualifiers:

- ***A*
- *A*B*
- *A*B
- A*B*C

Performing a SNAP DATASET copy
Keep in mind the following masking rules:

- **SOURCE, EXCLUDE, TARGET and RENAMEUNCONDITIONAL (RENUNC)**
  dataset name mask rules are different. The **SOURCE, EXCLUDE** and **RENUNC** (oldname) allow partial index level masking. For example, while EMC.DATA*.ABC** is valid for **SOURCE, EXCLUDE** or **RENUNC** (oldname), it is not valid as a **TARGET** or **RENUNC** (newname).

- The **SOURCE, EXCLUDE** and **RENUNC** (oldname) dataset name masks follows DFDSS rules for dataset selection.

  **Note:** The first qualifier of the source dataset must not include any wildcards.

- The **TARGET** and **RENUNC** (newname) masks only allow wildcarding for complete index levels. For example, while EMC.DATA*.ABC** is invalid for a **TARGET** or **RENUNC** (newname), EMC.DATA.** is valid as a target.

- The **TARGET** and **RENUNC** (newname) dataset follow DFDSS rules for RENAME processing. The **RENUNC** dataset name masks follow DFDSS rules for RENAME.

Some examples of valid **TARGET** masks are:

- A.** Replace the first index level of the source dataset name with “A.”
- A.B.** Replace the first two index levels of the source dataset name with “A.B.”
- *.A.** Replace the second index level of the source dataset name with “A.”
- **.BCD Replace the last index level of the source dataset name with “BCD.”
- **.*EFG Copy the first two index levels from the source dataset name and make the third index level “EFG.”

Some examples of invalid **TARGET** masks are:

- **.DEF.** INVALID, don’t know which level to replace.
- A.*BCD* INVALID, the entire index level must be wild, or not at all.

**Waiting for full track definition**

If a source device of the SNAP DATASET command is a LINK target, TF/Clone waits for the tracks to be fully defined on the source device.

**Snaps from offline or cloned volumes**

TF/Clone can now perform snaps from devices that are offline or cloned. To do this, you must identify all of the volumes to be scanned. Then, TF/Clone performs the following steps:

1. Starting with Mainframe Enablers 8.4.
1. Examines the VVDS and VTOC for each device specified in the SOURCE_VOLUME_LIST parameter, looking for datasets that match the SOURCE(DATASET xx) parameter.

   **Note:** You can use wildcards for dataset names.

   For non-VSAM, primary information is acquired from the VTOC. For VSAM, cluster and component relationships and names are acquired from the VVDS and VTOC. The VVDS and VTOC records are cached in memory for future reference, ensuring code compatibility.

2. Verifies that all pieces of a dataset are present. This includes
   - Checking for missing volumes on multivolume non-VSAM datasets. If the end-of-dataset indicator is not set for a multivolume non-VSAM dataset, the dataset cannot be copied.
   - Analyzing the RBA values for VSAM datasets to ensure that the entire range of RBAs is represented by the components found.

3. Resumes normal processing. Since the VVDS and VTOCIX entries are cached, no further reference to the VTOC or VVDS is made.

   A single SNAP DATASET statement may reference catalogued datasets, or datasets on offline volumes, but not both together.

### SNAP offline volumes example

This example uses groups to snap multivolume, VSAM KSDS (with AIX) from offline volumes.

1. The example defines a source volume list called OFFVOLS for the offline volumes.
2. The example performs a SNAP DATASET specifying a target of HLQ.TESTING.NEWKSDS. It also uses SOURCE_VOLUME_LIST to specify the source volume list created previously.

   **Note:** You can use this example only if you have purchased the licensed feature code for the keyword (parameter) TARGET.

```plaintext
* DEFINE
* DEFINE SOURCE_VOLUME_LIST OFFVOLS ( -
   UNIT(6EF0) -
   UNIT(6EF6-6EF7) -
   VOL(U6A230) -
   VOL(U6A23*) -
 )
* SNAP
* SNAP_DATASET (SOURCE(HLQ.TESTING.TWOKSDS) -
   TARGET(HLQ.TESTING.NEWKSDS) -
   HOSTCOPYMODE(NONE) -
   SOURCE_VOLUME_LIST (OFFVOLS) -
   REPLACE(Y) -
   REUSE(N) -
   SPHERE(YES) -
```
Snapping to GSPACE datasets

Starting with Mainframe Enablers 8.4, when the target is a GSPACE (Guaranteed Space) dataset and the VOLUME parameter is specified, the allocation can only be on the specified devices. If the VOLUME parameter is not specified, the candidate volumes are selected through SMS.

Cascaded operations

"SNAP/FlashCopy coexistence" on page 106 provides information about cascaded operations with both SNAP VOLUME and SNAP DATASET.

Security considerations

No change is required to the existing security process when implementing the SNAP DATASET command. To verify that the users have proper access to perform the snap, a request is made to SAF with a resource ID of “DATASET” and the dataset name. The source dataset is checked for READ access and the target is checked for ALTER access.

You can also make use of the EMCSAFI Security Interface and the SAF command security. The EMCSAFI Security Interface feature provides additional security checks for environments where multiple groups of users are using different devices in a single storage system.

Note: The Mainframe Enablers Installation and Customization Guide provides more information about the EMCSAFI Security Interface.

Examples

Example 1  This example snaps a dataset:

```
SNAP DATASET (SOURCE ('PROD.R1.DATA') -
TARGET (PROD.R1.DATA.SNA))
```

Example 2  This example snaps a dataset to a target to be allocated using the storage class PROJ1. The snap is to proceed even if exclusive serialization is not obtained for the source.

```
SNAP DATASET (SOURCE ('PROJECT1.OUTPUT.FILE') -
TARGET (SNAP.PROJECT1) STORCLAS (PROJ1) -
TOLENQF (YES))
```

Example 3  This example snaps a dataset, that is being shared by another job to a target that is allocated on volume PACK01.

```
SNAP DATASET (SOURCE ('DATA.RVA#1.FILE') -
TARGET (BACKUP.RVA#1.FILE) VOLUME(PACK01) -
HCPYMODE (SHR))
```

Example 4  This example snaps a VSAM KSDS and its SPHERE records (alternate index and path records). In the example:
Wildcards are used in the RENAMEUNCONDITIONAL statement to ensure that only the first two HLQ are changed.

A BCVGROUP and a VOLUME are both used, which allows TF/Clone to choose the best target volume from the combination of the BCVGROUP and the volume statements.

The HOSTCOPYMODE (SHR) allows other users to access the source dataset while the snap is in progress.

The REPLACE(Y) enables the overwriting of target datasets if they exist.

The WAITFORCOMPLETION(YES,MESSAGES) parameter tells TF/Clone to continue this job step until this snap is complete.

The MESSAGES sub-parameter activates a progress report with the number of remaining tracks for the storage system to copy. This status report is given from time to time until this snap is finally completed.

```*  SNAP     MULTI-AIX VSAM KSDS WITH UPGRADE PATH  *
*  SNAP DATASET ( SOURCE(STANDARD.MV3404.VSAM.CL1   ) -
  TARGET(BCV.MV3424.VSAM.CL1       )               -
  VOL   (MV3424                   )                -
  BCVGROUP(BCV_GRP1)                               -
  CATALOG(YES)                                     -
  SPHERE(YES)                                      -
  RENAMEU((STANDARD.MV3404.**,BCV.MV3424.**) )     -
  HOSTCOPYMODE(SHR   )                             -
  REPLACE(Y)                                       -
  WAITFORCOMPLETION(YES,MESSAGES)                  -
)  *
```

**Example 5**

This example uses wildcards. In the example:

- In the SOURCE statement, the % symbol represents a single character. All selected datasets with a suffix of G00x1V00 (where x is a number from 0 to 9 respectively) is selected.

- Wildcarding in the target dataset changes only the HLQ of the target dataset names to BG5. The rest of the dataset name is like the source dataset name. This is a SMS managed source and target, and the suggested volumes come from BCVGROUP (SMS_GRP1), but the ACS routines determine the DFSMS approved candidate volume or volumes.

- The DATACLASS is to be copied from the source dataset. The target dataset is cataloged.

- HOSTCOPYMODE(SHARED) means that exclusive control of the source is not required, and the you manage multiple access to the source dataset during the snap operation. The snap tolerates an allocation of the target failure, a target copy failure, and an ENQUEUE failure of the source and completes this operation. The target is to be replaced.

- If this job causes more than a maximum of the sessions allowable for a dataset, TF/Clone waits for a completed session before starting another session. If the target dataset already exists, then the old target dataset is deleted and a new one is allocated.
Note: If the GDG base does not exist, a GDG target base is created with the same attributes as the GDG source base. If the GDG target base does exist, it is updated with the GDG source base attributes.

* * SNAP GDG USING WILDCARDS
* SNAP DATASET (SOURCE(SG5.SNAP.GDG.G00%1V00) -
TARGET (BG5.** ) -
BCVGROUP(SMS_GRP1) -
COPYSMS(DATACLAS) -
DATAMOVERNAME(EMCCOPY) -
CATALOG(YES) -
HOSTCOPYMODE(SHARED) -
TOLERATEALLOCATIONFAILURE(YES) -
TOLERATECOPYFAILURE(YES) -
TOLERATEENQFAILURE(YES) -
REPLACE(Y) -
WAITFORSESSION(YES) -
REUSE(NO ) -
)

Example 6
The following example uses the DEFINE_SOURCE_VOLUME_LIST to define a list of offline volumes. Then, it uses SNAP DATASET to perform a snap from the volumes.

* DEFINE
* DEFINE SOURCE_VOLUME_LIST OFFVOLS ( -
UNIT(6EF0) -
UNIT(6EF6-6EF7) -
VOL(U6A230) -
VOL(U6A23*) -
)
* SNAP
* SNAP DATASET (SOURCE(HLQ.TESTING.TWOKSDS) -
TARGET(HLQ.TESTING.NEWKSDS) -
HOSTCOPYMODE(NONE) -
SOURCE_VOLUME_LIST (OFFVOLS) -
REPLACE(Y) -
REUSE(N) -
SPHERE(YES) -
VOL(U6A231,U6A230) -
)

Example 7
This example employs SNAP DATASET to snap two datasets from source to target. A BCV group is identified for inclusion in the volume candidate list and existing target datasets may be erased.

// JOB
//QCOPYRUN EXEC PGM=EMCSNAP
//STEPLIB DD DISP=SHR,DSN=DS-PREFIX.LINKLIB
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=*
//SYSCOPY DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
* GLOBAL MAXRC(8)
* SNAP DATASET (SOURCE(YOUR-SOURCE-DSN) -
TARGET ('YOUR-TARGET-DSN') BCVGROUP (SYMBCV) -
  REPLACE (Y) FORCE (N) HOSTCOPYMODE (NONE) ) *
SNAP DATASET (SOURCE ('YOUR-SOURCE-DSN') -
  TARGET ('YOUR-TARGET-DSN') BCVGROUP (SYMBCV) -
  REPLACE (Y) FORCE (N) HOSTCOPYMODE (NONE) ) */

//BCVGROUP DD *
BCVGROUP SYMBCV VOL (BCVVOL1 BCVVOL2 BCVVOL3 BCVVOL4)
//
Performing a Parallel Snap

**Note:** The Parallel SNAP solution should not be confused with the Simultaneous TF Clone feature. See “Simultaneous (parallel) clone” on page 24 for more information about the use of the PARALLEL_CLONE parameter.

Parallel Snap creates a snap copy of a dataset on both sides of an SRDF/S configuration. To allow for Parallel Snap, the configuration must meet the following conditions:

- The snap source volumes are the R1/R2 pair.
- The snap target volumes are non-SRDF devices.
- Data is not replicated across the SRDF link.
- TF/Mirror operations are supported on both sides of the SRDF relationship.

Figure 10 shows a Parallel Snap operation.

**Figure 10** Parallel Snap operation

**Parallel Snap software requirements**

The minimum levels of Dell EMC software required for Parallel Snap are:

- TF/Mirror for z/OS, PTF level – ST54009 or later.
- TimeFinder, PTF level – SN55008 or later.

**Parallel Snap operations**

Parallel Snap allows two independent snap operations in the same or different storage systems. These operations are driven by two independent SNAP DATASET commands. These commands specify the same source dataset name, but different target dataset names. Serialization can be guaranteed across both commands so as to achieve the same consistent point in time copy on each target dataset.
In a parallel operation, TimeFinder requires a channel to the storage system on which the R2 is located. Parallel Snap performs the following steps:

1. Obtains dataset information from the R1 device.
2. Allocates the snap target dataset on both the local and remote storage system.
3. After detecting that there is a synchronously mirrored R2 that is a partner of the source R1, performs the snap of the source datasets from the R2 mirror within the remote storage system.

In short, the R1 provides the dataset information to drive allocation, and SRDF provides a copy of the data remotely from which to perform the snap. The target dataset is cataloged on the R1 storage system, and (assuming the catalog volumes are remotely mirrored as they should normally be) the snap target dataset is properly cataloged and accessible if you need to perform an IPL operation on the z/OS system after a disaster occurs.

To achieve independent dataset snaps at the same point in time from both the R1 and R2 of a mirrored pair, you must write SMS ACS routines to direct the allocation of the snap target datasets to the appropriate storage systems.

These routines also must ensure that the target volumes selected on the local site are not R1 devices. If an R1 device is selected as the snap target a “redundant” copy of the data is propagated across the link, obviating the benefit of this solution.

**Invoking Parallel Snap**

Invoking Parallel Snap requires proper TimeFinder syntax if BCV operations are to be performed against the snap source or snap target volumes in the R2 storage system.

**Parallel Snap operation**

For both snap targets to represent a consistent point in time image of the source datasets, specify the following parameters:

- For non-VSAM datasets, set TOLERATEENQFAILURE to NO (the default) and set HOSTCOPYMODE(EXCLUSIVE).
  
  To ensure consistency for the snap target, these settings do not allow read access to occur on the source dataset during snap initiations. You can allow read access with HOSTCOPYMODE(SHARE), but doing so does not guarantee a consistent point-in-time snap.

  For VSAM datasets, set HOSTCOPYMODE(SHARED) with VSAMENQMODE(EXCLUSIVE). Also specify TOLERATEENQFAILURE(NO) and TOLERATEVSAMENQFAILURE(NO).

  **Note:** VSAM concurrency is controlled by a combination of VSAMENQMODE and the VSAM SHAREOPTIONS settings for the dataset. Consistent, point-in-time copies of a dataset can only be obtained using EXCLUSIVE.

- You must also use the new GLOBAL command parameter:

  ```
  GLOBAL ENQSCOPE(STEP)
  ```

  This option is required for the ENQ to be set for the dataset that is to be snapped.
Example

The following example shows parameter usage for Parallel Snap operation:

```
/*****************************************************/
/* USER WANTS TO SNAP TO A LOCAL TGT NOT IN THE */
/* SRDF GROUP, AND AT THE SAME TIME SNAP THE */
/* SAME DS THRU THE R2 TO A DIFFERENT TARGET */
/* ON THE REMOTE BOX, ALSO NOT IN THE SRDF GROUP, */
/* ALL OF THIS WHILE AN ESTABLISH/SPLIT */
/* RELATIONSHIP IS IN PROGRESS ON THE */
/* REMOTE BOX USING THE R2 AND A BCV THAT IS */
/* DIFFERENT FROM THE REMOTE SNAP TARGET VOLUME. */
/*

VOLUMES USED FOR SNAP

GLOBAL ENQSCOPE(STEP) -
HOSTCOPYMODE(EXCLUSIVE) - — The next 4 parameters ensure
VSAMENQMODE(EXCLUSIVE) - consistency of the snaps.
TOLERATEENQFAILURE(NO) -
TOLERATEVSAMENQFAILURE(NO) -

* SNAP FROM R1 TO LOCAL TGT *
SNAP DATASET ( SOURCE(LCLSRC.MV6600.** ) ) - — Same source.
TARGET(LCLTGT.MV6608.** ) - — Different target.
VOLUME(MV6608 ) -
REPLACE(Y) -
REUSE(NO ) -

* SNAP FROM R1 THRU R2 TO REMOTE TGT *
SNAP DATASET ( SOURCE(LCLSRC.MV6600.** ) ) - — Same source.
TARGET(RMTTGT.MVD750.** ) - — Different target.
VOLUME(MVD750 ) -
REPLACE(Y) -
REUSE(NO ) -
```

Operations and Examples

144  TimeFinder/Clone Mainframe Snap Facility 8.0 and Higher Product Guide
Performing queries

QUERY GROUP display example

The following example shows the output from a QUERY GROUP command.

```
PROCESSING FOR STATEMENT #2 BEGINNING, QUERY GROUP REQUEST FOR GROUP SNP7310
GROUPNAME - STATUS - DESCRIPTION
SNP7310 - INITIAL - SNP7310 - 8 VOL SNAP
HISTORY: RC DATE / TIME OLD STAT STATUS
DEFINE - 0000 2006-12-02 / 15:28:58 INITIAL -> INITIAL
DEFINE - 0000 2006-12-02 / 14:31:45 INITIAL -> INITIAL
SNAP vOL - 0008 2006-12-02 / 15:31:50 INITIAL -> FAILED
SNAP VOL - 0008 2006-12-02 / 15:54:59 FAILED -> FAILED
SNAP VOL - 0004 2006-12-02 / 15:57:37 FAILED -> PRESNAP
STOP VOL - 0000 2006-12-02 / 16:22:34 PRESNAP -> INITIAL

STATEMENTS:
+ *
+ SNAP VOLUME (SOURCE (VOLUME(mC0C10)) TARGET (uNIT(0C30)) ) -
+ NEWVOLID (MV0C30) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C11)) TARGET (UNIT(0C31)) ) -
+ NEWVOLID (MV0C31) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C12)) TARGET (UNIT(0C32)) ) -
+ NEWVOLID(MV0C32) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C13)) TARGET (UNIT(0C34)) ) -
+ NEWVOLID(MV0C34) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C14)) TARGET (UNIT(0C34)) ) -
+ NEWVOLID(MV0C34) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C15)) TARGET (UNIT(0C35)) ) -
+ NEWVOLID(MV0C35) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C16)) TARGET (UNIT(0C36)) ) -
+ NEWVOLID(MV0C36) )
+ SNAP VOLUME (SOURCE (VOLUME(MV0C17)) TARGET (UNIT(0C37)) ) -
+ NEWVOLID(MV0C37) )
+ *
+ END GROUP
```

The output is organized as follows:

- The group name (SNP7310), the status (INITIAL) and a description (SNP7310 - 8 VOL SNAP).
- A history of group usage, including:
  - Commands executed
  - Resulting return codes (RC)
  - Date of execution
  - Time of execution
  - Original status (OLD STAT)
  - Resulting status (STATUS)
- The commands in the group.
- The results.
Remote QUERY VOLUME example

This example employs three parameters available for the remote QUERY VOLUME command. Any of the three works.

```plaintext
//SNPQUERY EXEC PGM=EMCSNAP
//STEPLIB DD DISP=SHR, DSN=EMC.SSNI.P580.LINKLIB
//MV6C00 DD DISP=SHR, UNIT=3390, VOL=SER=MV6C00
//SYSPRINT DD SYSOUT=* 
//EMCQCAPI DD SYSOUT=* 
//SYSABEND DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
GLOBAL MAXRC(4)
*
* 1. QUERY REMOTE SYMM USING LOCAL VOLUME
   QUERY VOLUME (REMOTE (VOL (MV6C00) RAGROUP(17) ) )
* 
* 2. QUERY REMOTE SYMM USING LOCAL UNIT
   QUERY VOLUME (REMOTE (UNIT( 6C00) RAGROUP(17) ) )
* 
* 3. QUERY REMOTE SYMM USING LOCAL DDNAME
   QUERY VOLUME (REMOTE (DDNAME(MV6C00) RAGROUP(17) ) )
/`
/
```

Cleaning up volumes

Use the CLEANUP command to remove each completed extent in the extent track on the indicated volume.

Remote CLEANUP example

This example performs a remote CLEANUP to a source volume.

```plaintext
//CLEANUP EXEC PGM=EMCSNAP
//STEPLIB DD DISP=SHR, DSN=EMC.SSNI.P580.LINKLIB
//SYSPRINT DD SYSOUT=* 
//EMCQCAPI DD SYSOUT=* 
//SYSABEND DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//QCOUTPUT DD SYSOUT=* 
//QCINPUT DD *
GLOBAL MAXRC(4) CHK0(N) AUTOMATIC_RELEASE_HOLD(YES)
*
*   CLEANUP 2 REMOTE VOLUMES USING UNIT AS THE
* 
   CLEANUP REMOTE(UNIT (6C00) RAGROUP(17) CONTROLLER(90132)) SYMDV#(0000)
   CLEANUP REMOTE(UNIT (6C00) RAGROUP(17) CONTROLLER(90132)) SYMDV#(0001)
* 
/`
```
Using SRDF/A R2 Wait for Precopy

TimeFinder has a new SRDF/A R2 Wait for Precopy feature. SRDF/A R2 Wait for Precopy is intended to address a situation when too many protected tracks occur on an SRDF/A R2 device. To minimize any possible issues, TimeFinder now requires that you specify the following parameters to snap from an SRDF/A R2 device:

- PRECOPY(YES)
- MODE(COPY)
- WAIT_FOR PRECOPY_PASS1(YES)

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, a TF/Snap off an active SRDF/A R2 device is supported with group and device level pacing set by SRDF Host Component commands. See the SRDF Host Component for z/OS Product Guide for more information.
Viewing GCM status

Starting with Mainframe Enablers 8.4, TF/Clone supports Geometry Compatible Mode (GCM).

**Note:** The *SRDF Host Component for z/OS Product Guide* discusses GCM.

GCM status is displayed with the QUERY VOLUME command as follows:

```
ESNPP36I   GCM: YES
```
CHAPTER 5
Command Reference

This chapter provides a reference for the TimeFinder commands.

- Conventions ........................................................................................................ 150
- Traditional TimeFinder commands .................................................................. 151
- Grouping commands (TF/Snap) ........................................................................ 320
Conventions

This chapter provides reference sections for the TimeFinder commands. In the following sections:

◆ Those commands that apply to both TF/Clone and TF/Snap have headings that consist only of the command name.

◆ Those commands that only apply to TF/Clone have headings that consist of the command name and the term: (TF/Clone).

Note: Any command that is noted as (TF/Clone) requires a TF/Clone License before you can use it.

◆ The commands that only apply to TF/Snap have headings that consist of the command name and the term: (TF/Snap).

Note: Any command that is noted as (TF/Snap) requires a TF/Snap License before you can use it.

◆ The descriptions of parameters that are only available if you install the appropriate licensed feature code have notes that explain which licensed feature code is needed.

Note: When entering commands, do not code fields past column 71.

Syntax conventions

The commands in this chapter follow these syntax conventions:

◆ Conventions described in “Conventions used in this document” on page 17.

◆ Keywords appear in uppercase (for example, ALL). They must be spelled exactly as shown.

◆ For easy reference, command keywords can be supplemented by lowercase letters to form a meaningful word (for example, CoNTROLler). When typing a command, use only CAPITALIZED characters of any keyword.

◆ Aside from the characters described in “Conventions used in this document” on page 17, you must type all other characters that are shown in the syntax statements.
Traditional TimeFinder commands

Traditional TimeFinder TF/Clone and TF/Snap commands require a physical target device to be specified in the command to execute a full volume copy.

While still supported for downward compatibility, the traditional TimeFinder commands now use the SnapVX architecture under the covers. It is recommended to use SnapVX commands whenever possible.

Note: SnapVX commands are covered in the TimeFinder SnapVX and zDP Product Guide.

IMPORTANT
Some traditional TimeFinder commands may have parameters related to SnapVX. For a description of such parameters, see the TimeFinder SnapVX and zDP Product Guide.

Customer task guide for traditional TF commands

The customer task guide table allows you to quickly find the corresponding command for TimeFinder tasks. Follow the page references for a more complete description of each command.

Table 9 Customer task guide for traditional TF commands

<table>
<thead>
<tr>
<th>Task</th>
<th>Associated TF Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine when the preceding SNAP VOLUME or SNAP DATASET actions take place. Optionally specify whether the SNAP actions are to be performed using Enginuity Consistency Assist (ECA) to form consistent point-in-time volume snaps.</td>
<td>“ACTIVATE” on page 214</td>
</tr>
<tr>
<td>Check each extent track on the indicated volume to determine whether it is complete, and then remove each completed extent in the extent track.</td>
<td>“CLEANUP [EXTENT TRACK ON]” on page 217</td>
</tr>
<tr>
<td>Specify RELEASE, NR, and READY conditions for BCV and STD devices.</td>
<td>“CONFIG (TF/Clone)” on page 220</td>
</tr>
<tr>
<td>Define a group of SNAP VOLUME and GLOBAL statements.</td>
<td>“DEFINE GROUP” on page 320</td>
</tr>
<tr>
<td>Create a list of offline devices, and then specify that list (as an argument to the SOURCE_VOLUME_LIST parameter) to the QUERY DATASET and SNAP DATASET commands.</td>
<td>“DEFINE SOURCE_VOLUME_LIST (TF/Clone)” on page 225</td>
</tr>
<tr>
<td>Delete an existing group, that was defined with the DEFINE GROUP command.</td>
<td>“DELETE GROUP” on page 322</td>
</tr>
<tr>
<td>Complete the definition of a group. You must enter an END GROUP after you finish entering the SNAP VOLUME and GLOBAL statements that define the group.</td>
<td>“END GROUP” on page 322</td>
</tr>
<tr>
<td>Specify parameters that apply to all following commands, unless you override them using optional parameters specified for an individual command.</td>
<td>“GLOBAL” on page 226</td>
</tr>
<tr>
<td>Get dataset status information.</td>
<td>“QUERY DATASET (TF/Clone)” on page 259</td>
</tr>
</tbody>
</table>
GPM commands

Dell EMC ResourcePak Base provides a set of General Pool Management (GPM) commands that can be executed online, or in batch mode, to configure and manage a predefined set of devices that provide a pool of physical space.

For TimeFinder use, the devices for SNAPPOOLs come from a special pool called the DEFAULT_POOL. The DEFAULT_POOL contains snap pool devices that have not been assigned to any named pool, but are available for use.

Multiple SNAPPOOLs can be created to isolate workloads. This alleviates contention for device space among several users and lessens the possibility of a single pool consuming all the available space.

The ResourcePak Base for z/OS Product Guide describes the GPM commands and provides a complete description of creating pools and managing the pooling process.

Note: The GPM command CONFIGPOOL is no longer supported.
Common parameters

ACTIVATE_SUBTASK# (*nnn*)

This parameter sets the minimum number of storage systems being activated to invoke the subtasking feature. The subtasking feature assigns one subtask for each storage system to minimize the ECA window when multiple syscalls are required.

When the ECA window is opened, 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.

*nnn* is the number of storage systems. 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.

ADMINISTRATOR (Yes | No)

When you use DFDSS (ADRDSSU) as a datamover, it queries RACF for each dataset being copied. The ADMINISTRATOR parameter determines whether DFDSS avoids RACF calls. This action can make the DFDSS processing faster.

Yes    Specifies passing the ADMINISTRATOR parameter to DFDSS.

No     (Default) Specifies not passing the ADMINISTRATOR parameter.

*Note:* DFDSS requires that you have certain RACF privileges for ADMINISTRATOR to be accepted.

If you specify ADMINISTRATOR(YES), DFDSS_ADMIN(YES) is implied.

The ADMINISTRATOR parameter with a NO value instructs TimeFinder not to use the ADMINISTRATOR parameter.

*Note:* The IBM publication, *Implementing ESS Copy Services with IBM eServer zSeries* (SG24-5680) provides more information.

The ADMINISTRATOR parameter has a matching site option, &ADMIN.

ALLOCATE_UNUSED_SPACE (Yes | No)

The ALLOCATE_UNUSED_SPACE parameter determines whether the target dataset is allocated using the total space, both used and unused, of the source dataset or just the used space:

Yes     (Default) Specifies allocating the target dataset large enough to contain both the used and unused space of the source dataset.

No      Specifies allocating the target dataset only large enough to contain the used space of the source dataset.

This parameter only applies to sequential and standard partitioned datasets.

The ALLOCATE_UNUSED_SPACE parameter has a matching site option, &ALUNUSED.
**ALLOCATION_SEQUENCE** *(DATASET | NONE | SIZE)*

The `ALLOCATION_SEQUENCE` parameter specifies the processing order of datasets (VSAM clusters and non-VSAM files) in a wildcarded request:

- **DATASET** *(Default)* Specifies processing VSAM clusters and non-VSAM files in ascending name sequence.
- **NONE** Specifies processing VSAM clusters and non-VSAM files in the order they are selected for processing. This may appear random.
- **SIZE** Specifies processing VSAM clusters and non-VSAM files in descending size sequence. The largest datasets are processed first and the smallest are processed last.

`ALLOSEQ` is an alias of `ALLOCATION_SEQUENCE`.

The `ALLOCATION_SEQUENCE` parameter has a matching site option, `&ALLOSEQ`.

**AUTOMATIC_CLEANup** *(Yes | No)*

The `AUTOMATIC_CLEANup` parameter allows or disallows an automatic cleanup to be run as part of the `RESTORE VOLUME` command prior to the restore occurring. This cleans up the device and prevents some related errors from occurring.

- **Yes** *(Default)* CLEANUP is automatically run against the device.
- **No** CLEANUP is not automatically run against the device.

The `AUTOMATIC_CLEANup` parameter has a matching site option, `&AUTOCLN`.

**AUTOMATIC_CLEANup_R2** *(Yes | No)*

This parameter ensures that when an active R1 device with an R2 device is cleaned, the R2 device is cleaned also. This parameter works when the `CLEANUP` command specifies devices using `UNIT`, `VOLSER` or `SYMDV#` parameters.

The default value is `YES`.

`CLEANUP_R2` is an alias of `AUTOMATIC_CLEANup_R2`.

The `AUTOMATIC_CLEANup_R2` parameter has a matching site option, `&CLEAN_R2`.

**AUTOMATIC_DEALLOC** *(Yes | No)*

The `AUTOMATIC_DEALLOC` parameter allows or disallows automatic issuance of an `S DEALLOC` command to z/OS when a device `VARY ONLINE` or `VARY OFFLINE` appears to be hung. z/OS sometimes requires a job to go through allocation to handle these situations.

- **Yes** *(Default)* Allow TimeFinder to automatically issue an `S DEALLOC`.
- **No** Prevent TimeFinder from automatically issuing an `S DEALLOC`.

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

`AUTO_DEAlloc` is an alias of `AUTOMATIC_DEALLOC`.

The `AUTOMATIC_DEALLOC` parameter has a matching site option, `&AUTODEAL`.

**Example**

`AUTO_DEAL(NO)`
AUTOMATIC_RELEASE_hold(Yes|No)

The AUTOMATIC_RELEASE_HOLD parameter allows the Hold to be automatically released when the background snap of a volume is complete:

Yes  Allow Hold to be automatically released.
No   (Default) Disallow Hold from being automatically released.

This only applies to SNAP VOLUME. This feature is provided through the SNAP NOTIFY feature in EMCSCF.

If requested, then the SNAP NOTIFY subtask in EMCSCF monitors the volume progress and issues the CONFIG RELEASE command to the volume when the snap is complete.

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

AUTOMATIC_RELEASE is an alias of AUTOMATIC_RELEASE_HOLD.

The AUTOMATIC_RELEASE_HOLD parameter has a matching site option, &AUTORLSE.

Example

AUTOMATIC_RELEASE(YES)

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.

You can specify the NOCOPYRD keyword as:

- NOBACKGROUNDCOPYONREAD
- NOBGCOPYONREAD
- NOCOPYONREAD
- NOCOPYRD

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.

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

“MODE(COPY|NOCOPY|NOCOPYRD|VSE)” on page 178 provides additional information.

The BACKGROUNDCOPY parameter has a matching site option, &BACKGRND.
BCVOnly(Yes | No)

The BCVONLY parameter restricts allocation of new target devices to BCV devices:

- **Yes** Only BCV devices are to be considered for new target devices or datasets.
- **No** (Default) STD or BCV devices are to be considered for new target devices or datasets.

**Note:** This optional parameter is only valid when you do not specify the TARGET parameter using the SYMDV#, UNIT or the VOLUME parameter.

Exceptions to BCVONLY(YES) specification:

- If you specify a STD device in a BCVGROUP, TimeFinder honors BCVONLY(YES) and ignores STD devices.
- If you specify a specific target using the UNIT, SYMDV#, or VOLume parameter on the command, then TimeFinder ignores the BCVONLY(YES) request.
- If a target dataset is being reused, then TimeFinder ignores the volume type.
- If volume preferencing is used to influence SMS volume selection, then TimeFinder honors BCVONLY(YES) and relegates STD devices to the secondary list.

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

BUILD_VTOCIX(Yes | No)

The BUILD_VTOCIX parameter is employed when extent allocation is used and the device in question does not have a VTOC INDEX present on the device. If you specify BUILD_VTOCIX(YES), then extent allocation attempts to create a VTOC INDEX on the device.

If extent allocation is successful, the allocation proceeds normally.

Possible values are:

- **Yes** Attempt to create a VTOC INDEX on a device that does not have one.
- **No** (Default) Do not attempt to create a VTOC INDEX on a device that does not have one.

The BUILD_VTOCIX parameter has a matching site option, &VTOCIX.

CATalog(Yes | No)

The CATALOG parameter determines whether the new target dataset created by the SNAP DATASET command is to be cataloged:

- **Yes** (Default) Specifies that the allocated target dataset is to be cataloged.
- **No** Specifies that the allocated target dataset is not to be cataloged.

TF/Clone supports Integrated Catalog Facility (ICF) catalog entries.

The CATALOG parameter only applies to new non-VSAM datasets. Existing datasets is not cataloged. VSAM datasets are always be cataloged. Datasets managed by SMS are always be cataloged, because SMS allows only the creation of cataloged datasets on SMS-managed volumes.

The CATALOG parameter has a matching site option, &CATALOG.
CHECKBCVholdstatus (Yes | No)

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

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

Aliases of CHECKBCVholdstatus include CHECK_BCV_hold_status, CHKBCVhold, CHECK_DEvice_hold_status, CHKDEVicehold, CBHS, CHECK_TARGET_hold_status, CHECK_TGT_hold_status, CHECKTARGETholdstatus, CHECKTGTholdstatus, CHKTARGEThold, CHKTGThold, CTHS.

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

CHECKONLINEpathstatus (Yes | No | NEVER)

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

- **NEVER** Specifies that for all commands, no check or report is issued to indicate that the device is online to other systems.
- **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.
- **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.

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

CHECK_POOL_usable (Yes | No)

If CHECK_POOL_USABLE(YES) is specified, then the pool name and pool usability is checked during the parse phase to ensure that the pool is a valid name. Pool usability is defined by at least one enabled device, with one or more free tracks, with the same geometry as the virtual device (3380, 3390, fba512, fba520)

If CHECK_POOL_USABLE(NO) is specified, then the pool name and pool usability is not checked until the VDEV (or thin device if AUTO_BIND(YES) is specified) is actually being created in the storage system device.

**Note:** The AUTO_BIND and AUTO_UNBIND parameters are only compatible with Enginuity 5876 and 5773.

**Default value**

**NO**

The CHECK_POOL parameter has a matching site option, &POOLUSE.

CKD (EXCLUDE | INCLUDE)

The CKD parameter includes or excludes CKD devices from a QUERY VOLUME device list:

- **EXCLUDE** Exclude CKD devices from QUERY VOLUME device list.
- **INCLUDE** *(Default)* Include CKD devices on QUERY VOLUME device list.
CLEANup_DIFFerential(Yes | No)

The CLEANUP_DIFFERENTIAL parameter is used to ensure that differential sessions are cleaned up.

Yes  CLEANUP should examine and remove differential sessions.

No (Default) CLEANUP should ignore differential sessions.

When run without CLEANUP_DIFFERENTIAL, a CLEANUP command ignores differential sessions.

If you specify CLEANUP_DIFFERENTIAL(YES), then CLEANUP also examines and removes differential sessions, if all tracks have been copied.

The CLEANUP_DIFFERENTIAL parameter has a matching site option, &CLEANDIFF.

COLLAPSE_dataset_extents(VSAM | NONVSAM | VSAM, NONVSAM)

If EXTENT_ALLOCATION is not requested, TimeFinder uses a simple two-pass approach to allocation. The first pass attempts to allocate the dataset as one large single extent. Often this fails because z/OS is not able to find such free space on the available volume list. If the first pass fails, then a second attempt is made by:

1. Allocating a single small extent (approximately equal in size to the first extent of the source dataset).

2. Expanding the dataset until it is as large as the source dataset.

This parameter indicates whether the first pass should be attempted. Possible values are:

VSAM  Both passes are used for VSAM datasets.

NONVSAM  Both passes are used for non-VSAM datasets.

VSAM, NONVSAM (Default) Both passes are used for all dataset types.

The COLLAPSE_DATASET_EXTENTS parameter has a matching site option, &COLLAPSE.

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. This is equivalent to an ADRDSSU COPY VOLUME command with the DUMPCONDITIONING parameter.

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 has a matching site option, &CONDVOL.
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 the SNAP VOLUME command:

- If a VTOC index and VVDS are present and active on the target volume, TF/Snap 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.
- TF/Snap 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.
- If a RESTORE VOLUME command with COPYVOLID(YES) parameter occurs in a JES3 environment, the target volume must be manually varied offline to JES3 after the RESTORE VOLUME completes. During the RESTORE VOLUME operation with COPYVOLID(NO), the target volume is normally varied offline during the operation and varied online after the RESTORE VOLUME completes. You can link edit a user exit into TF/Snap to be invoked prior to the VARY ONLINE and VARY OFFLINE commands being issued. This exit is available for automating JES3 operations.

CONSISTENT(Yes | No)

The CONSISTENT parameter determines whether you use Enginuity Consistency Assist (ECA) for consistent SNAP VOLUME 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.

CONSISTENT(YES) with zBoost PAV Optimizer

For PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, issuing an TF/Clone ACTIVATE command with the CONSISTENT(YES) parameter suspends zBoost PAV Optimizer write processing before enabling ECA to achieve consistency. When ECA is disabled, the write processing is resumed.

CONTROLLER( [xxxxxxx-]xxxxx | name)

You can use CONTROLLER as a separate parameter to identify the storage system and automatically provide the address of the gatekeeper device and SRDF group.
This means that you can use the CONTROLLER parameter instead of the LOCAL and REMOTE parameters. 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.

You may specify a five-digit (xxxx) or a 12 digit (xxxxxxx-xxxxx) serial number.

name

A logical storage system name of up through 64 characters that was already assigned through ResourcePak Base. If the logical storage system name has a simple format (single-word string of uppercase letters), then you may enter the storage system name without quotation marks. If the logical storage system name is made up of mixed case characters or contains spaces, enclose it in quotation marks.

Note: The ResourcePak Base for z/OS Product Guide provides more information about the storage system naming facility.

The CONTROLLER parameter is available on the following commands:

- CLEANUP {EXTENT TRACK ON}
- CONFIG
- QUERY VDEVICE
- QUERY VOLume
- RESTORE VOLume
- SNAP VOLume
- STOP SNAP TO VOLume

Default value

None

COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASs] [STORageCLASs] [ALL])

The COPYsourceSMSclasses parameter determines whether SMS class values are to be used from the existing source dataset:

- DATACLASs Use dataclass.
- ManaGeMenTCLASs Use management class.
- STORageCLASs Use storage class.
- ALL Use all classes.

You may supply SMS class information about the action statement, or copy it from the source dataset.

The COPYsourceSMSclasses parameter is honored only while creating new datasets. If you reuse a dataset, the existing DATA, MANAGEMENT, and STORAGE class values are not affected. Any class value coded on the GLOBAL or SNAP DATASET statement overrides the ability to copy the class value from the source dataset (that is, DATACLASs parameter takes precedence over the COPYSMS(DATACLASs) parameter).
The COPYSMS parameter is not valid when using TF/Clone with alternate index datasets. This is because SMS does not record the class information when an alternate index dataset is created.

The COPYsourceSMSclasses parameter with the ManaGeMenTCLASs option has a matching site option, &CSMSMNGMT.

The COPYsourceSMSclasses parameter with the DATACLASs option has a matching site option, &CSMSDATA.

The COPYsourceSMSclasses parameter with the STORageCLASs option has a matching site option, &CSMSSTOR.

**Default value**

None

**Example**

COPYSMS(DATACLAS STORCLASS)

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 (that is, TimeFinder issues a VARY OFFLINE against the target volume).
- **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 (that is, TimeFinder issues a VARY ONLINE against the target volume).

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

The COPYVOLID parameter only applies to locally addressable volumes. TimeFinder ignores COPYVOLID 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 the SNAP VOLUME command:

- If a VTOC index and VVDS are present and active on the target volume, TimeFinder updates any records for the VTOC index and VVDS files to reflect the new names of these files. VTOC index names have the form SYS1.VTOCIX.volser and VVDS names have the form SYS1.VVDS.Vnnnnn. The volser portion of these names is the same as the volser of the target volume.
- If the volser begins with a numeric character, the default name for the VTOC index is SYS1.VTOCIX.Vnnnnn, where nnnnn is the last five characters of the volser of the target volume.
- TimeFinder updates the VTOC records for the VTOC index and VVDS, if present, to reflect the new names for these files with the same naming conventions as for the VTOC index updates.
- If you enter a SNAP VOLUME command with COPYVOLID(YES) in a JES3 environment, the target volume must be manually varied offline to JES3 after the SNAP VOLUME completes.
During the SNAP VOLUME operation with COPYVOLID(NO), the target volume is normally varied offline during the operation and varied online after the SNAP VOLUME completes.

You can link edit a user exit into TimeFinder to be invoked before the VARY ONLINE and VARY OFFLINE commands are issued. This exit is available for automating the JES3 operations.

**Example**

```
COPYV(YES)
```

**DATACLASs(classname)**

The DATACLASs parameter specifies the SMS data class to be assigned to the target dataset after TF/Clone dynamically allocates the target dataset.

`classname`

Specifies a locally defined data class to be assigned to the target dataset. Your storage administrator determines the valid data class names.

You must have SAF or equivalent authorization for the data class specified.

Local SMS ACS routines may place the target dataset in a data class other than that specified by this parameter. As with all SMS datasets, specifying data class is only a suggestion to SMS, and may or may not be accepted by SMS.

TF/Clone does not assign the source data class to a target dataset automatically unless you specify COPYSMS(DATACLAS). You must ensure that the correct data class is assigned to the target dataset by using the DATACLASS parameter or ACS selection.

If an existing target dataset is reused, the data class information associated with the target dataset is not changed.

The DATACLASs parameter has a matching site option, &DATACLAS.

**Default value**

None

**DataMoverName(ADRDSSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)**

Normally, the source and target dataset must reside within the same physical storage system for the operation to be performed. In some situations, this is not feasible. The DataMoverName parameter allows you to specify a datamover utility program that can actually copy the physical tracks.

- ADRDSSU
- FDRDSF

ADRDSSU (also DFDSS, DSS) is an IBM utility program which may be invoked to copy physical tracks between physical control units. The utility control statement used to invoke ADRDSSU is:
FDRDSF (also called FDR) is a utility program from INNOVATION which may be invoked to copy physical tracks between physical control units. The utility control statement used to invoke FDRDSF is:

COPY TYPE=DSF SELECT FROM( ) TO( ) VOL= NEWTOCYL= NEWTOTRK= NVOL=

Possible values are:

ADRDSU|DFDSS|DSS Specifies that ADRDSU is to be used to copy physical tracks between physical control units or in non-PowerMax/VMAX control units. ADRDSU is used to copy physical tracks within a PowerMax/VMAX system that TF/Clone is not able to handle. DFDSS can be used to perform a logical dataset copy for dataset types such as IMBED, REPLICATE and KEYRANGE.

COPYCYL Specifies that the internal copy utility COPYCYL is to be used. COPYCYL reads/writes a full cylinder at a time (fewer if necessary).

Note: PRESNAP(YES) parameter cannot be specified when COPYCYL is involved in processing.

COPYTRK Specifies that the internal copy utility COPYTRK is to be used. COPYTRK reads/writes three tracks at a time (fewer if necessary).

Note: PRESNAP(YES) parameter cannot be specified when COPYTRK is involved in processing.

FDRDSF|FDR Specifies that FDRDSF is to be used to copy physical tracks between physical control units or in non-PowerMax/VMAX control units. FDRDSF is used to copy physical tracks within a PowerMax/VMAX system that regular TimeFinder is not able to handle.

Note: For correct work of DFDSS and FDR datamovers, both source and target devices have to be ONLINE. PRESNAP processing makes the target devices NOT-READY to the channel, so PRESNAP(YES) parameter cannot be specified when DFDSS/FDR are involved in processing.

When using DataMoverName(FDRDSF), ensure that the job includes the SYSPRINT DD statement. Otherwise, message ESNP510E is issued with error code 402.

IDCAMS Specifies that IDCAMS may be used to perform a logical dataset copy for going between differing VSAM organizations, differing stripe counts, STRIPE=1 with differing track counts or volume counts, and going to/from extended format.

Note: IDCAMS is a secondary datamover, so it can be used with any other datamover as coded above. IDCAMS is invoked to copy the logical records from the source to the target if there is a compatibility problem, such as a different stripe count between source and target. The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

NONE (Default) Specifies that an error occurs if regular TF/Clone is not able to handle the requested copy operation.
The DataMoverName parameter applies only to locally addressable volumes. DataMoverName is ignored if you specify it on actions with the SYMDV#, LOCAL, or REMOTE parameters.

To duplicate datasets, an appropriate datamover must be selected. When the source and target datasets are both in the same storage system, TimeFinder may be used according to the operating environment level.

If the source and target datasets or volumes are both in the same RVA and the IBM SNAPSHOT software is available, it is automatically used.

When the source and target datasets or volumes are in separate storage systems, an appropriate datamover must be selected. DFDSS, FDRDSF, COPYCYL, or COPYTRK works.

If the source and target datasets or volumes are in storage systems not supported by TF/Clone or IBM SNAPSHOT, an appropriate datamover must be selected.

DATA_MOVER_Name and DATAmover are aliases of DataMoverName.

The DataMoverName parameter has a matching site option, &DATAMOVR.

**Example**

```plaintext
DATAmover(COPYCYL)
DataMoverName(DFDSS, IDCAMS)
```

**DATASET_CHANGED_indicator (SET | RESET | LEAVE)**

Specifies the value of the dataset-changed indicator (DS1DSCHA bit in the format-1/8 dataset control block) that a target dataset should have after a SNAP DATASET command:

- **LEAVE** (Default) Sets the DS1DSCHA bit to the source dataset change bit value.
- **RESET** Sets the DS1DSCHA bit to 0.
- **SET** DS1DSCHA bit to a 1.

DSCHI, DS1DSCHA, DSCHA are aliases of DATASET_CHANGED_indicator.

The DATASET_CHANGED_indicator has a matching site option, &DS1DSCHA.

**DEBUG (ON | OFF)**

The DEBUG parameter controls the logging of diagnostic messages:

- **OFF** (Default) Disable diagnostic message generation.
- **ON** Enable diagnostic message generation.

The DEBUG parameter enables or disables diagnostic message logging. It would normally be used at the request of Dell EMC Customer Support.

The information resulting from specifying DEBUG(ON) may be of use only to a Dell EMC Customer Support representative.

**Example**

```plaintext
DEBUG(ON)
```
DFDSS_ADMIN(Yes|No)

The DFDSS_ADMIN parameter determines whether the ADMINISTRATOR parameter is passed to DFDSS to avoid the RACF calls for each dataset. This can make the DFDSS processing faster.

Yes Specifies passing the ADMINISTRATOR parameter to DFDSS.
No  (Default) Specifies not passing the ADMINISTRATOR parameter to DFDSS.

DFDSS does require you to have certain RACF privileges for ADMINISTRATOR to be accepted.

Note: The IBM publication, Implementing ESS Copy Services with IBM eServer zSeries (SG24-5680) provides more information.

If ADMINISTRATOR(YES) is specified, DFDSS_ADMIN(YES) is implied.

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

DFDSS_CC(Yes|No)

The DFDSS_CC parameter determines whether ADRDSSU establishes a concurrent copy session while performing the track copy:

Yes  (Default) Directs ADRDSSU to use concurrent copy to protect tracks being copied.
No   Directs ADRDSSU not to use concurrent copy to protect tracks being copied.

A concurrent copy session allows a more point-in-time type of copy operation to occur. This optional parameter is valid only when the DATAMOVERNAME specifies ADRDSSU.

When you use DFDSS_CC(Y), the DFDSS_CC parameter has some implications when using an IBM RVA.

Note: The IBM publication, Implementing ESS Copy Services with IBM eServer zSeries (SG24-5680) provides more information.

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

Example

DFDSS_CC(NO)

DIFFerential(Yes|No)

The DIFFERENTIAL parameter determines whether the Differential Snap feature is used:

Yes Use Differential Snap.
No   (Default) Do not use Differential Snap.

The Differential Snap feature creates a relationship so that, after the initial snap, only changed tracks are moved for subsequent snaps of the same source/target volume pair. Always specify DIFFerential for a Differential Snap.

You must have purchased and installed the TF/Clone licensed feature code to perform full-volume snaps.
TimeFinder automatically supports the snapback operation by performing a SNAP VOLUME with DIFF(YES) in the opposite direction. There is no need to specify a RESTORE operation.

The DIFFERENTIAL parameter has a matching site option, &DIFF.

**Example**

```plaintext
DIF(NO)
```

**DIFFERENTIAL_DATASET (Yes | No)**

The DIFFERENTIAL_DATASET parameter enables or disables the Differential Dataset Snap feature:

- **Yes** Enables the Differential Dataset Snap feature.
- **No** *(Default)* Disables the Differential Dataset Snap feature.

With Differential Dataset Snap, a dataset’s contents are copied in their entirety when that dataset is snapped for the first time.

When you set DIFFERENTIAL_DATASET(YES), only the changed tracks are copied when the dataset is snapped again.

This feature is only effective if REPLACE(YES) and REUSE(YES) are also specified.

The DIFFERENTIAL_DATASET parameter has a matching site option, &DIFFDSN.

**EATTR (NO | OPT)**

EATTR is an IBM parameter that specifies whether the dataset can support extended attributes or not. These datasets must be allocated on an extended address volume (EAV).

Values include:

- **NO** Extended attributes are not allowed, and the dataset cannot reside in EAS space on EAV devices.
- **OPT** Extended attributes are allowed. The dataset may also reside in EAS space on EAV devices.

EATTR is also a site option.

**Default value**

- `'NO'` for non-VSAM files
- `'OPT'` for VSAM files

**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.

---

1. Available starting with Mainframe Enablers 8.2.
The EMUL_TYPE parameter is also available as a site option, &EMUL_TYPE.

ENQSCOPE (REQUEST | STEP)

The ENQSCOPE parameter determines when and for how long the source dataset ENQ is held:

REQUEST (Default) Specifies that, at the beginning of request, the source dataset ENQ is obtained. When the request is completed, the source dataset ENQ is released (DEQ).

STEP Specifies that all source dataset ENQ is obtained after the parse phase, but before any requests are processed. After ALL requests have completed, the source dataset ENQ is released (DEQ).

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

ENQWAIT (Yes | No)

The ENQWAIT parameter is used with HOSTCOPYMODE. If you specify HOSTCOPYMODE(NONE), ENQWAIT is ignored.

If you specify HOSTCOPYMODE(EXCLUSIVE) or HOSTCOPYMODE(SHARED), ENQWAIT determines the action to take if exclusive or shared access is not immediately available for a source dataset.

If you specify ENQWAIT(YES), the action waits until the source dataset becomes available. If you specify ENQWAIT(NO), the action continues or fails based upon the TOLERATEENQFAILURE parameter setting.

Values can be:

Yes (Default) Processing waits until the source dataset becomes available.

No Processing continues. The action may continue or fail based upon the TOLERATEENQFAILURE parameter setting.

The ENQWAIT parameter does not apply to datasets specified by INDDname or OUTDDname.

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

ERROR_CHECKING (NORMAL | REDUCED)

The ERROR_CHECKING parameter specifies special error handling.

NORMAL (Default) The extents are checked and an attempt to resolve all “protected and indirects” occurs before the establish.

Protected and indirects are the Dell EMC terms for the controlled relationship of tracks that is established between a source and a target. Source tracks are “protected” before being copied to a target’s “indirects”, or the tracks locations dedicated to receive the data.

NORMAL is how error checking has always worked. It is still recommended for a mixed SNAP DATASET and SNAP VOLUME environment.

REDUCED The checks before the establish are skipped and TimeFinder issues the establish.

REDUCED would be used in circumstances where you believe that there shouldn’t be any reason for the establish to fail.

You can abbreviate the ERROR_CHECKING parameter name as ERRCHK.

The ERROR_CHECKING parameter has a matching site option, &ERRCHK.

Table 2 on page 47 lists the site options and their possible values.
Normally, you would use the two error handling parameters, ERROR_CHECKING and ERROR_RECOVERY in the following combinations.

- ERROR_CHECKING(NORMAL) and ERROR_RECOVERY(NORMAL)
- ERROR_CHECKING(REDUCED) and ERROR_RECOVERY(ENHANCED)

Other combinations do not work well. Since ERROR_CHECKING(NORMAL) performs checking before the establish, it would be very difficult for ERROR_RECOVERY(ENHANCED) to help because the checking was already performed. ERROR_CHECKING(REDUCED) and ERROR_RECOVERY(NORMAL) would not perform before any checking, before the establish or when the establish fails.

**Example**

```plaintext
ERROR_CHECKING(REDUCED)
```

**ERRor_DISPosition** (DELETE | KEEP)

The ERROR_DISPOSITION parameter specifies what to do with the target datasets when a SNAP DATASET request fails:

- DELETE (Default) Delete the target datasets if a SNAP DATASET request fails.
- KEEP Keep the target datasets if a SNAP DATASET request fails.

The normal action is to delete any target datasets. An alternative is to keep the target datasets.

**Note:** The ERROR_DISPosition parameter can also be used in the shorter form, ERR_DISP.

The ERROR_DISPOSITION parameter has a matching site option, &ERRDISP.

**ERROR_RECOVERY** (NORMAL | ENHANCED)

The ERROR_RECOVERY parameter specifies how TimeFinder should handle recovery in an error situation:

- NORMAL (Default) If the establish fails, so does the request. NORMAL is how error checking has always worked in the past.
- ENHANCED With ENHANCED, if the establish fails, TimeFinder attempts to resolve protection and indirects.

Normally, you would use the two error handling parameters, ERROR_CHECKING and ERROR_RECOVERY in the following combinations.

- ERROR_CHECKING(NORMAL) and ERROR_RECOVERY(NORMAL)
- ERROR_CHECKING(REDUCED) and ERROR_RECOVERY(ENHANCED)

Other combinations do not work well. Since ERROR_CHECKING(NORMAL) performs checking before the establish, it would be very difficult for ERROR_RECOVERY(ENHANCED) to help because the checking was already performed. ERROR_CHECKING(REDUCED) and ERROR_RECOVERY(NORMAL) would not perform before any checking, before the establish or when the establish fails.

You can abbreviate the ERROR_RECOVERY parameter name as ERRREC.
The ERROR_RECOVERY parameter has a matching site option, &ERRREC.

**ESNP220 (ERROR | WARNING)**

Determines whether message ESNP220 is a warning or error message:

- **ERROR** *(Default)* Message is issued and processing stops.
- **WARNING** Message is issued as a warning and processing continues.

The following ESNP220 message involves dataset extents and has two different outcomes that can be set.

**SOURCE DATA SET HAS NO EXTENTS**

It can be a warning message, where the extent discovery that caused the message is identified and the processing is continued, or it can be set as an error condition where the processing is stopped.

**Example**

ESNP220 WARNING

**EXAMINE (Yes | No)**

The EXAMINE parameter causes TimeFinder to do an IDCAMS EXAMINE on the target VSAM dataset:

- **Yes** Directs TimeFinder to do an IDCAMS EXAMINE on the target VSAM dataset.
- **No** *(Default)* Directs TimeFinder not to do an IDCAMS EXAMINE on the target VSAM dataset.

**Note:** The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

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

**EXclude_PathGroupID(pathlist)**

Normal processing of SNAP VOLUME requests ensure that the target volume is not online (path group established) to any other LPAR or system. A parameter already exists (CHECK_ONLINE_PATH_STATUS) that allows the severity of the situation to be changed from an error to a warning.

The EXCLUDE_PATHGROUPID parameter allows certain LPARs or systems to be ignored. No error or warning message are issued if encountered.

**pathlist**

Specifies a list of one or more entries. Each entry is made up of the first 14 characters in a 22-character path group ID. (The remaining, right most, eight characters are the timestamp.)

**Note:** The SRDF Host Component for z/OS Product Guide provides more information about timestamps.

Each digit may be a valid hexadecimal character or a wildcard character. Valid wildcard characters are ‘*’ or ‘%’. Both mean that a single digit is masked.

You can specify up to 127 path group IDs.
EX_PGID is an alias of EXCLUDE_PATHGROUPID.

The EXCLUDE_PATHGROUPID parameter has a matching site option, &EXPATHGRP.

Default value

None

Example

If the complete path group ID is: 880002A75C2084C173D526
then, you would enter as a list entry: EX_PGID(880002A75C2084)

EXPlain(VOLUME_SELECTION(Yes|No))

As each potential volume is examined to determine whether it can be a candidate for dataset allocation, a line is written with an explanation. Messages ESNP0A0I or ESNP0A1I are issued.

Depending on the number of devices, this parameter can generate a lot of output.

This parameter has a matching site option, &EXPLAIN_VOL_SEL.

Default value

No

EXTENT_ALLOCATION(Yes[,CONSOLIDATE_VOLUME|,CONSOLIDATE_ALL]|No)

The EXTENT_ALLOCATION parameter specifies whether operations should use extent allocation for target datasets in snap operations:

CONSOLIDATE_ALL

Force extent allocation to consolidate the extents across all volumes. The number of volumes used may not match the source and the number and size of the extents on the targets may not have any relationship to the source. For multivolume datasets, the CONSOLIDATE_ALL option will not affect the total number of volumes (PRIME/CANDIDATE) available for the dataset to expand onto.

CONSOLIDATE_VOLUME

Force extent allocation to consolidate the extents on each volume. Each volume contains the same number of tracks, but the individual number and size of the extents on the target may not match the source.

No

(Default) Do not use extent allocation.

Yes

Use extent allocation.

To maximize the possibility of successful snaps of PowerMax/VMAX devices and IBM RVA devices, YES specifies that as long as appropriate candidate volumes are available, the target dataset should have the same number and size of extents as the source.

Some dataset types always use extent allocation. All dataset types are supported by this method.
Because extent allocation bypasses the normal SVC99 and IDCAMS allocation methods, internal SMS storage group resolution and eligible volume determination is provided. TimeFinder invokes the SMS ACS routines and exits. Then TimeFinder builds a list of candidate volumes using selected storage groups.

**Note:** The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

The EXTENT_ALLOCATION parameter has the following matching site options:

- &EXTALLOC = EXTENT_ALLOCATION(YES|NO)
- &CONSALL = EXTENT_ALLOCATION(YES, CONSOLIDATE_ALL)
- &CONSVOL = EXTENT_ALLOCATION(YES, CONSOLIDATE_VOL)

**Example**

EXTENT_ALLOCATION(YES, CONSOLIDATE_VOL)

EXTALLOC_EMC_ONLY(Yes | No)

EXTALLOC_EMC_ONLY controls whether only Dell EMC-manufactured devices are to be used as possible candidates for extent allocation.

- **Yes** Only use Dell EMC-manufactured devices as candidates for extent allocation.
- **No** *(Default)* Use any device as a candidates for extent allocation.

The EXTALLOC_EMC_ONLY parameter has a matching site option, &EMCONLY.

EXTENT_EXPAND(Yes | No, [ADDNEW(Yes | No)] [, SAMEVOL] [, NEWVOL])

The EXTENT_EXPAND parameter controls how extent allocation allocates a dataset. When allocating a dataset, extent allocation normally makes every attempt to create a new dataset that, extent-wise, looks the same. This means that a multivolume target dataset is created to look just like a multivolume source dataset, down to the size of each extent and the number of volumes.

Extent allocation also requires that the dataset be allocated from scratch. That means that the existing dataset must be deleted first.

The option values determine the method of extent allocation used:

- **ADDNEW** A value of YES allows new volumes to be added to the existing dataset. A value of NO does not allow new volumes to be added to an existing dataset.
- **NEWVOL** When the existing dataset cannot be expanded on the current volume, it is removed from the current volume and an attempt is made to create the dataset on a new candidate volume.
- **No** Extent allocation is not used to adjust the size of an existing dataset to match the size of the source dataset.
- **SAMEVOL** New extents must be found on the same volume.
- **Yes** Extent allocation is used to adjust the size of an existing dataset to match the size of the source dataset. This means that existing extent sizes may change. New extents may be created and existing extents may be removed.

The EXTENT_EXPAND parameter has several matching site options:

- &EXTADDNEW to specify the ADDNEW option value.
- &EXTPVOL to specify the SAMEVOL and NEWVOL option value.
- &EXTPXVOL to specify the SAMEVOL and NEWVOL option value.

**Default value**

None

<table>
<thead>
<tr>
<th>FBA (EXCLUDE</th>
<th>INCLUDE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The FBA parameter includes or excludes FBA devices from a QUERY VOLUME device list:</td>
<td></td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>Exclude FBA devices from QUERY VOLUME device list.</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>(Default) Include FBA devices on QUERY VOLUME device list.</td>
</tr>
</tbody>
</table>

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

**FLASH_SNAP (FLASHCOPY | SNAP)**

FLASH_SNAP determines whether FlashCopy or TimeFinder operations is used by default. Values are:

<table>
<thead>
<tr>
<th>FLASHCOPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlashCopy is used by default.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default) TimeFinder is used by default.</td>
</tr>
</tbody>
</table>

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

You should not change the value of this parameter unless you are directed to do so by Dell EMC.

**FORCE (Yes | No)**

The FORCE parameter establishes the FORCE parameter for all SNAP DATASET operations. It allows you to snap a dataset that was created as absolute track (ABSTR) or as unmovable (PSU, POU or DAU). TF/Clone makes no attempt to ensure that the target dataset is accessible. TF/Clone does not allocate absolute track locations.

Possible values are:

| Yes | Specifies that an unmovable or absolute track dataset can be snapped to a different physical track location on another device. This option is required to snap an unmovable or absolute track dataset. |
| No  | (Default) Specifies that an unmovable or absolute track dataset is not to be snapped. |

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

**FORCE_COMPLETION (Yes | No)**

The FORCE_COMPLETION parameter specifies that CLEANUP does not complete until all source extents and sessions on the device are completed. Any NOCOPY extents and sessions are changed to copy. After all of the extents and sessions are completed, the source extent track is also removed. Values can be:

| Yes | CLEANUP does not complete until all source extents and sessions on the device complete. |
| No  | (Default) CLEANUP may complete before all source extents and sessions on the device complete. |

You can abbreviate FORCE_COMPLETION as FORCE_CMP.
FORCE_COMPLETION is also available as a site option, &FORCECMP.

The prime use for FORCE_COMPLETION is for conversion of native extents.

The two formats of the extent track entries are mutually exclusive.

FREESPACE(Yes|No)

When a SNAP VOLUME request is processed, the FREESPACE parameter specifies whether to snap unallocated space. When you use the default NO, snapping unallocated space can only occur if the source volume is online and the z/OS VTOC services are available. After the snap is initiated for the complete volume, an internal STOP SNAP TO VOLUME is issued for all of the unallocated space. Possible values are:

- Yes  Specifies snapping the freespace.
- No   (Default) Specifies that after the snap has initiated, stop the snap to freespace areas on the volume.

**Note:** Under certain conditions, FREESPACE(YES) can overlay existing data on the target volume, but this is dependent on a number of variables, such as the timing and the size of the volume.

The FREESPACE parameter only applies to locally addressable volumes. FREESPACE is ignored if specified on actions with the SYMDV#, LOCAL or REMOTE parameters.

TimeFinder ignores the FREESPACE parameter with differential snap.

When SOFTLINK(YES) is set, the specified value of the FREESPACE parameter is ignored, and FREESPACE(YES) is assumed.

The FREESPACE parameter has a matching site option, &FREESPC.

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

The GROUP parameter 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.

**Note:** “DEFINE GROUP” on page 320 provides more information about the DEFINE GROUP command.

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 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.

**Default value**

None
The HOSTCOPYMODE parameter specifies whether and what type of disposition is used with dynamic allocations for all SNAP DATASET operations:

- **EXCLUSIVE** Specifies use of the dynamic allocations with disposition of OLD. Exclusive control offers better protection of data integrity than shared control. Choose this option for exclusive control over the source during a snap operation.
- **NONE** No disposition used.
- **SHARED** Specifies the use of dynamic allocations with a disposition of SHR. Choose this option to modify the source during a snap operation.

If you are attempting to snap a dataset previously opened within a Database Management System (DBMS) for which update activity has not been quiesced, you cannot obtain exclusive control because the database has not been closed or deallocated. In this case, choosing the SHARED option actually provides user-managed exclusive control.

Use the HOSTCOPYMODE(SHARED) or HOSTCOPYMODE(none) parameter if you know that the dataset is shared by another job because this option would use less overhead than using the TOLERATEENQFAILURE(YES) parameter.

HOSTCOPYMODE does not apply to datasets specified by INDDname or OUTDDname.

The HOSTCOPYMODE parameter has a matching site option, &HOSTCOPY.

**Default value**

None

The INVALIDATE_PDSE_buffers parameter causes or prevents the flushing of the PDSE buffers when a SNAP DATASET or SNAP VOLUME command is processed:

- **Yes** (Default) Causes the PDSE buffers to be flushed when a SNAP DATASET or SNAP VOLUME command is processed
- **No** Prevents the PDSE buffers from being flushed when a SNAP DATASET or SNAP VOLUME command is processed.

The situation in which you must flush the PDSE buffers needs further explanation. The PDSE buffers must be flushed if they are being cached, which only happens if the SMS parameters PDSE_BUFFER_BEYOND_CLOSE or PDSE1_BUFFER_BEYOND_CLOSE are set to YES.

This can cause unwanted results. If you do choose to leave the SMS parameters and INVALIDATE_PDSE_BUFFERS set to YES and one or more of the PDSEs is open, the flush fails and you receive one or more messages (ESNPX10W-ESNPX12W).

To avoid the flush and the possibility of an error, set the two SMS parameters (&SMSKSDS and &SMSPASSVOL) and INVALIDATE_PDSE_BUFFERS to NO.

---

**Note:** The current version of the IBM publication, *z/OS DFSMS Technical Update* (SG25-7435-00) provides more information about the SMS parameters PDSE_BUFFER_BEYOND_CLOSE and PDSE1_BUFFER_BEYOND_CLOSE.
The INVALIDATE_PDSE_BUFFERS parameter is also available as a site option, &INVALIDATE_PDSE.

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.

UNIT(cuu)

Specifies the unit address of the gatekeeper.

Note: The MVS 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.

DDNAME(ddname)

Identifies the DD statement that refers to the gatekeeper.

CONTROLLER

Optional. If using the LOCAL parameter, then it would be a 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 (xxxxxxxx-xxxxx) serial number. Or, you may specify a logical storage system name if you previously defined that name to ResourcePak Base.
If the logical storage system name is simple in format (single-word string, all uppercase 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.

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.

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

Cleanup of extent track contents (dataset level compared to cleanup of full device sessions) requires the UNIT (CCUU) or VOLUME parameter, and extent track cleanup (dataset level) is not performed when the SYMDV# is used.

**Default value**

None

**LOGINDYNAM(volume[, volume...])**

The LOGINDYNAM parameter specifies a list of volumes to be used for comparison purposes when selecting source datasets.

The criteria you want to apply to volumes in a LOGINDYNAM list before processing selection can take place are specified using the SELECTMULTI parameter.

**volume**

Volume specification.

**Default value**

None

**MESSages(DISplay|PROmpt|NONE|DETAIL)**

The MESSAGES parameter controls console messages before and after the activation of a snap volume:

**DETAIL** Displays console messages before and after, 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.

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

The MESSAGES parameter also works with SNAP DATASET to interface with automation systems.

**MSGs** is an alias of MESSages.

The MESSages parameter has a matching site option, &MESSAGE.

** ManaGeMenTCLASs(classname)**

The MANAGEMENTCLASS parameter establishes a default management class to be used for all new dataset allocations. This overrides the COPYSMS(MANAGEMENTCLASS) parameter.
classname

Specifies a logically defined management class to be assigned to the target dataset. Your storage administrator determines the valid management class names for the site.

You must have SAF or equivalent authorization for the management class specified.

Local SMS ACS routines can place the target dataset in a management class other than that specified by MANAGEMENTCLASS. As with all SMS datasets, specifying management class is only a suggestion to SMS. SMS may or may not accept it.

TF/Clone does not assign the source management class to a target dataset automatically unless you specify COPYSMS(MGMTCLAS). You must ensure that the correct management class is assigned to the target dataset by using the MANAGEMENTCLASS parameter or ACS selection.

If an existing target dataset is reused, the management class information is not changed.

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

Default

None

MIGrate([PURge(Yes|No)] [RECall(Yes|No)])

Normally a snap operation fails if the source or target dataset is migrated. The MIGRATE parameter allows the source dataset to be automatically recalled and any existing migrated target dataset to be purged:

- **PURge(No)** (Default) If the target dataset is migrated, then the snap operation fails.
- **PURge(Yes)** If the target dataset is migrated, then it is deleted and a new target dataset allocated.
- **RECall(IGNORE)** If the source dataset is migrated, the dataset is not recalled, an error is not generated, and the dataset is ignored.
- **RECall(NO)** (Default) If the source dataset is migrated, then the snap operation fails.
- **RECall(Yes)** If the source dataset is migrated, then the snap operation is suspended and a recall is performed for the source dataset. After the source dataset is recalled, the snap operation continues.

The RECALL subparameter applies only to the source dataset and the PURGE subparameter applies only to target datasets.

A recall operation may take several minutes to complete.

Internally, the equivalent of a HDELETE and HRECALL is performed.

The MIGRATE parameter has two matching site options:

- MIGRATE(PURGE) has &PURGE.
- MIGRATE(RECALL) has &RECALL.

Example

MIG(PUR(Y) REC(Y))
MODE (COPY | NOCOPY | NOCOPYRD | VSE)

The MODE parameter specifies when the background copy from source to target occurs. MODE is available on the following commands:

- CONFIG
- GLOBAL
- SNAP DATASET
- SNAP VOLUME

The MODE and BACKGROUNDCOPY parameters serve the same purpose. You may specify these parameters interchangeably; but, you cannot specify both at the same time. They are mutually exclusive.

COPY

(Default) Specifies that the source to target background copy should begin immediately after the snap is issued.

Use MODE(COPY) for:

- A snap to an R1 device (even if you also specify MODE(NOCOPY) or MODE(NOCOPYRD))
- A snap from an SRDF/A R2 device
- A differential snap.

Note: The COPY option is not valid for virtual devices.

NOCOPY

Specifies that the background copy task does not copy any tracks that are marked protected (NOCOPY). A read of the source does not cause the source track image to be copied to the target device. However, the source track image is copied when the track on the target is an indirect.

Note: The operating environment no longer distinguishes between NOCOPY and NOCOPYRD, as both have the same result, which is NOCOPYRD.

With MODE(NOCOPY), the source and target of the snap are available for processing after the snap is activated. Updates to the target remain intact as of the last update to the target.

When used with the CONFIG command, MODE(NOCOPY) allows you to change MODE(NOCOPY) dynamically to MODE(COPY) without requiring a resnap or a “stop snap” (STOP SNAP TO DATASET or STOP SNAP TO VOLUME) operation.

Otherwise, the snap relationship between the source and target remains until either:

- A “stop snap” is issued against the target
- All of the tracks on the source have been updated, creating a complete original source image on the target
– In a dataset snap relationship, the target dataset is deleted.

**Note:** The NOCOPY option is not valid for virtual devices.

**Restrictions**

The only restriction to MODE(NOCOPY) is that the source and target must be in the same storage system. The current limit on the number of active TimeFinder sessions for any one source (dataset or full volume) is four (4). This means that after a source dataset or volume has been snapped four times with the MODE(NOCOPY) option, it cannot be snapped again until one of the previous sessions completes or is stopped.

If all of the snapped tracks are not accessed, MODE(NOCOPY) snaps may never complete. To cause the MODE(NOCOPY) snap to normally complete, run the original TimeFinder job (JCL and control cards), adding PARM="GLOBAL MODECOPYFINISH" to the PGM=EMCSNAP execute statement. If you use STOP SNAP *, the target of a MODE(NOCOPY) snap is indeterminate because all the source tracks may not have been copied. If the target is accessed after a STOP SNAP * data checks results when referencing tracks that have not been copied.

**NOCOPYRD**

Specifies that the background copy occur 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. Read of the source or target track does not cause the track to be copied.

**Note:** The operating environment no longer distinguishes between NOCOPY and NOCOPYRD, as both have the same result, which is NOCOPYRD.

You can specify the NOCOPYRD keyword as:

– NOBACKGROUNDCOPYONREAD
– NOBGCOPYONREAD
– NOCOPYONREAD
– NOCOPYREAD
– NOCOPYRD

**Note:** The NOCOPYRD option is not valid for virtual devices.

**Advantages**

The main benefit of MODE(NOCOPYRD) comes when the system is under stress conditions. For example, if volumes are copied with TF/Clone and a consistent copy is produced, you can back up the copy to tape and give up the copy after it resides on tape. In this situation, NOCOPYRD may be desirable because the target device is being read, not written, and there is no long term desire to create a “hardened copy” of the data on the target device.

A “hardened” copy is one where the data contents exist on the actual device.
With predictive read ahead in the control unit, you can maximize cache for read-ahead track images, instead of holding track images that must be destaged (written) to the target device, as would happen with regular MODE(NOCOPY).

This cache advantage also benefits the source device. Regular production traffic on the source devices are not impacted by a potential cache shortage which occurs with the track images that must be destaged (written) to the target devices.

Cautions

One possible caution with MODE(NOCOPYRD) is in situations where you intend to retain the copy for any period of time and expect that copy to be a truly “hardened” copy.

- With NOCOPY, a read of the target device or a write of the source or target device causes the track image to be copied to the target device. Over time, most or all track images are copied to the target device.
- With NOCOPYRD, only write operations cause the track image to be copied to the target device. If a failure was to occur on a source device, the target device may not be accurate.

Another possible caution is in situations where you are using both SNAP DATASET and SNAP VOLUME. These two request types, when intermixed, regularly conflict, especially if you specify NOCOPY. There are several situations where copying track images is not allowed if the source or target track is involved in a NOCOPY relationship. This is currently handled in host software, typically by reading the indirect track that is marked NOCOPY.

With NOCOPY, a read of the track actually causes the track to be destaged so that it is no longer indirect. With NOCOPYRD, if a track is involved in a NOCOPYRD relationship and a SNAP DATASET or SNAP VOLUME request bumps into that track, the request fails. If SNAP VOLUME is used with NOCOPYRD and the device then used with SNAP DATASET as source or target, SNAP DATASET fails.

VSE

Note: The VSE parameter is only available for the SNAP VOLUME command.

With MODE(VSE), device allocations are shared for THIN FBA devices and requires both the source and target device to be THIN FBA.

MODE(VSE) works the same as MODE(NOCOPY) and specifies that the background copy task does not copy any tracks that are marked VSE for thin FBA devices. A read of the source does not cause the source track image to be copied to the target device. However, any read or write of the target causes the source track image to be written to the target device.

Restrictions

- The VSE parameter is supported only on PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876.
The source and target must be in the same storage system. Both devices must be THIN FBA devices. MODE(VSE) is limited to 32 sessions. No more than 16 VSE differential sessions may be established.

Note: The limits do not apply to PowerMaxOS 5978 and HYPERMAX OS 5977.

The persistent preallocation feature is not supported for VP Snap (formerly Clone VSE).

If there is a persistent preallocation associated with a device, a VP Snap (Clone VSE) session cannot be created using that device as a target.

If a device is the target of a VP Snap (Clone VSE) session, persistent preallocation cannot be used on that device.

If MODE(VSE) is used together with SOFTLINK(YES), the mode will be changed to NOCOPY.

MULTI_LINE_query(Yes|No)

The MULTI_LINE_query parameter specifies a new multiline query option that displays additional detail lines beyond the single summary line for each device:

Yes Display multiple lines of information for each device.

No (Default) Display only the single summary line of information for each device.

For example, some EMCSNAP commands operate differently depending on the SRDF type. The multiline query shows the SRDF type (A/S/PPRC/XRC) and whether it is an R1, R2, R11, or other device designation. Additionally, it shows if certain features are in use on the device, such as parallel clone, inhibit outboard copy, hold, and more.

In addition, each mirror position is shown along with its attributes (adaptive copy, synchronized or not, ready state, write state).

The single summary line displays information such as device number, CCUU, device attribute (STD, BCV, TDEV, and so on), CKD or FBA, number of cylinders, ready state, and primary RAID protection.

The multiple line query displays message ESNPP36I, which contains the following information for each device:

- Remote device type (R1, R11, R21, R2, R22 or blank)
- Parallel Clone status (PC or blank)
- Inhibit Outboard Copy status (IOC or blank)
- Hold status (HOLD or blank)
- PPRC/XRC status (PPRC or XRC)
- ECA status (ECA)
- Meta Setting (META-HEAD)
- For each mirror position, the following information is available:
  - NCNFG if not configured. R1, R2 or LCL if mirror is configured.
- For remote mirrors, Sync or Async indicator (-S or -A), Adaptive Copy indicator (-ADCOPY and /WPO, /DISK or /WP). RAGROUP value (RAG=(xx)).
- Ready status (RDY or NRDY)
- Read/write status (R/W or R-ONLY)

Aliases for this parameter includes MULTILINEquery and MLQ.

This parameter has a matching site option, &MLQ.

MULTI_VIRTUAL(Yes|No)

The MULTI_VIRTUAL parameter specifies the method of handling virtual devices (VDEVs):

Yes Allows for the newer method of allowing multiple virtual devices (up to 128) to share a single standard device session.

No (Default) Assigns the default method of allowing a single virtual device to share a maximum of 8 sessions.

The MULTI_VIRTUAL parameter is available with Enginuity 5773 and 5876. With Enginuity 5876, the multivirtual method is the only method used, so whether the MULTI_VIRTUAL parameter is set to NO or to YES, or whether it is used at all, the system always allows 128 virtual device sessions.

With PowerMaxOS 5978 and HYPERMAX OS 5977, the MULTI_VIRTUAL parameter is not supported. Whether set to YES or NO, the system allows up to 32 virtual device sessions for a single standard device. To overcome this limitation, use the SOFTlink parameter.

The two methods, default multivirtual (NO) and multivirtual (YES), are not allowed to be intermixed on a single standard device.

Once a session is created on a standard device, all subsequent sessions must be the same type. When a standard device has no virtual device relationships, either method can be used.

MULTI_VDEV is an alias of MULTI_VIRTUAL.

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

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, and underscore (_). Embedded spaces and dashes (-) are not allowed.

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 SNAP VOLUME command.

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

This parameter has a matching site option, &SNAPSHOT_NAME.

**NEWVOLID(volser)**

The NEWVOLID parameter specifies a new volser on the target.

`volser` specifies the new volser for the target.

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

**NOTIFYwhencomplete([GROUP(name)] [DATASET|JOB|STEP|SNAP])**

After all of the requested snap operations have started, the snap step ends. The actual copy operation continues within the storage system without host intervention. You can use the NOTIFYWHENCOMPLETE parameter to signal when the actual copy operation completes. The NOTIFYWHENCOMPLETE parameter requests asynchronous notification upon completion of TimeFinder operations:

**DATASET**

Issue a completion message for each dataset or volume as the background snap is completed.

The DATASET, JOB, STEP and SNAP subparameters are mutually exclusive.

**GROUP(name)**

You can use the GROUP name to supply identifying information in the completion message. You may specify this value alone or with the other parameter values.

`name` is a completion message that you want to include. `name` can be up to 44 characters in length. If `name` includes blanks, you must enclose `name` in double quotation marks.
The NOTIFYWHENCOMPLETE parameter applies only to locally addressable volumes. NOTIFYWHENCOMPLETE is ignored if you specify it with the SYMDV#, LOCAL, or REMOTE parameters.

The NOTIFYWHENCOMPLETE parameter has to poll the storage system to determine when the copy is complete. The poll is performed with a one-second wait (default) between checks. However, the overhead varies based on the number of extents and how long it takes the background copy to complete.

When the number of extents is large, you can add a parameter to the SCF INI file that allows a larger delay between polls, such as an interval of 60 seconds or more. The parameter is SCF.SNAP.NOTIFY_POLLTIME and the value is in seconds.

**Note:** The notify message is issued from the EMCSCF address space, and requires EMCSCF 5.1 or later.

Any extents copied by a datamover other than by TimeFinder are automatically considered complete after the datamover has finished copying the tracks.

You can use the NOTIFYWHENCOMPLETE parameter with the following commands:

- **GLOBAL**
- **SNAP VOLUME**
- **RESTORE VOLUME**
- **SNAP DATASET**

The NOTIFYWHENCOMPLETE parameter has a matching site option, &NTFYlvl.

**Default value**

None

**Example**

GLOBAL NOTIFY
GLOBAL NOTIFY(GROUP("TESTING NOTIFY"))
GLOBAL NOTIFY(GROUP(My_SNAP_IS_DONE) STEP)
SNAP DATASET(SOURCE(EMC.**) TARGET(EMCT.**) NOTIFICATION(SNAP (GROUP(COPY_MY_DATASETS) JOB)))
NOTREADY (EXCLUDE | INCLUDE)

The NOTREADY parameter includes or excludes devices that are not ready from a QUERY VOLUME device list:

EXCLUDE Exclude devices that are not ready from the device list.

INCLUDE (Default) Include devices that are not ready from the device list.

The NOTREADY parameter has a matching site option, &OPT_NOTREADY.

PARALLEL_CLONE(Yes|No|PREFERed|REQUIRED)

The PARALLEL_CLONE parameter is used to invoke the Simultaneous TF/Clone feature (if available). When conditions are met, a dual clone session is established between the source and target R2 devices, avoiding the secondary SRDF/S transmission of a copied dataset from the target R1 to the corresponding R2 device.

This feature ensures that disaster restartability is intact at all times.

Possible values are:

No Disables parallel cloning. Parallel Clone is not attempted, even if the devices have the potential to take advantage of the parallel clone function.

PREFERed Parallel clone is enabled, but if there is a reason why the device cannot execute the simultaneous TF/Clone microcode feature, then the request continues using non-parallel clone methods.

REQuired Parallel clone is enabled for the appropriate devices (for example, source and target are both R1 devices in a common storage system, and the matching R2 devices are also in a common storage system). If there is some reason why the request cannot be completed using parallel clone, then the request fails.

Yes Enables parallel cloning. Yes and PREFERed are aliases of each other, and provide identical functionality.

Requirements and restrictions

- Parallel cloning of FBA devices is not supported.
- The ACTIVATE CONSISTENT(YES) parameter is required. If omitted, the parallel clone operation is still performed and the following informational message is issued:

```
ESNPF371 PARALLEL_CLONE(YES) DETECTED, CONSISTENT(YES) ASSUMED.
```

- SRDF/S is required.
- PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876 is required on both side of the link.
- R2 source and target snap volumes are in the same storage system.
- The R2 source and target snap volumes cannot be larger than the R1 volumes. TimeFinder blocks this operation.
- Not supported:
  - Cascaded SRDF devices.
  - SRDF/Star environments.
  - Flashcopy
- Virtual Provisioning (VDEV)

- The following SRDF operations are blocked on Simultaneous TF/Clone (PARALLEL_CLONE) devices:
  - Delete and Half Delete
  - Swap and Half Swap
  - Move Group and Half Move Group

- The PARALLEL_CLONE parameter is incompatible with the SOFTLink parameter.

**Note:** The PARALLEL_CLONE parameter should not be confused with the Global PARALLEL parameter for multitasking or the PARALLEL SNAP solution. PARALLEL_CLONE invokes the Simultaneous TF/Clone feature with PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876.

This parameter is also allowed on the SNAP DATASET and the SNAP VOLUME statements, and can be set as a site option.

**Default value**

None

**Example**

PARALLEL_CLONE(YES)
SNAP VOLUME...
...
...
ACTIVATE CONSISTENT(YES) MESSAGES(DISPLAY)

PERSISTent (Yes | No)

The PERSISTENT parameter is only used on VDEV’s and determines whether a standard virtual restore or a persistent virtual restore is performed:

**Yes**

Perform persistent virtual restore.
With PERSISTent(YES), you do not have terminate any other VDEV session or virtual device assigned to the restore device in order to proceed with a RESTORE VOLUME.

**No** (Default)

Perform standard virtual restore.
With PERSISTent(NO), you must terminate any other VDEV session or virtual device assigned to the restore device in order to proceed with a RESTORE VOLUME.

The PERSISTENT parameter has a matching site option, &PERSIST.

**Example**

PERSISTENT(YES)

**POOL(poolname)**

The POOL parameter is only used on VDEVs and allows multiple pools to be selected, each with specific snap pool devices. When virtual devices are created, you can associate them with a particular pool.

**Note:** Pools are used with TF/Snap only. You do not use pools with TF/Clone.
The POOL parameter has a matching site option, &POOL.

**Default value**

None

**POSTSNAP (Yes | No)**

The POSTSNAP parameter indicates whether SNAP VOLUME post processing should be automatically performed after the ACTIVATE command is executed or as part of the SNAP VOLUME command processing:

- Yes: Perform SNAP VOLUME post processing automatically after the ACTIVATE command.
- No: (Default) Perform SNAP VOLUME post processing as part of SNAP VOLUME processing.

“Postsnap processing” on page 120 provides more information about post processing.

The POSTSNAP parameter only applies to regular input (after a //QCINPUT DD * JCL statement) SNAP VOLUME 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 SNAP VOLUME operation is initiated that is not group processing, both PRESNAP and POSTSNAP are automatically set to YES.

**Default value**

No

**PRECOPY (Yes | No)**

The SNAP VOLUME command causes three operations to be performed.

- The first is to create the snap session on the source device.
- The second is to protect the source and mark the target as indirect.
- The third is to activate and make it all effective.

The PRECOPY parameter allows the background copy to begin after the source and target have been marked, prior to the activate operation:

- Yes: (Default) Specifies to initiate the background copy prior to the activation operation.
- No: Specifies not to initiate background copying.

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

**PRESNAP (Yes | No)**

The PRESNAP parameter indicates whether SNAP VOLUME preprocessing should be automatically performed before the ACTIVATE command is executed or performed as part of the SNAP VOLUME command processing:

- Yes: Perform SNAP VOLUME preprocessing automatically before the ACTIVATE command.
- No: (Default) Perform SNAP VOLUME preprocessing as part of SNAP VOLUME processing.

“Presnap processing” on page 119 provides more information about preprocessing.
The PRESNAP parameter only applies to regular input (after a //QCINPUT DD * JCL statement) SNAP VOLUME 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 SNAP VOLUME operation is initiated that is not group processing, both PRESNAP and POSTSNAP are automatically set to YES.

**Default value**

No

R1FULLCOPYonly (Yes | No)

In the situation where you specify MODE(NOCOPY) and perform a snap to an R1 device, the data may never be copied to the R1 (because of NOCOPY) or be copied to the corresponding R2 device.

The R1FULLCOPYonly parameter is designed to handle this situation. Works with the MODE(NOCOPY) parameter as it applies to R1 devices. If R1FULLCOPYonly is enabled, TimeFinder ignores MODE(NOCOPY) when the target of the snap is an R1 device. Possible values are:

- **Yes** (Default) Ignore MODE(NOCOPY) when the target of the snap is an R1.
- **No** Perform MODE(NOCOPY) when the target of the snap is an R1.

The R1FULLCOPYonly parameter has a matching site option, &R1FULLCOPY.

**Example**

R1FULLCOPY (NO)

RAID (ALL | NONE | RAIDS | RAID1 | RAID5 | RAID6 | RAID10 | FTS)

The RAID parameter is used to restrict the list of devices to one or more types of RAID device. (You can specify multiple arguments.) The RAID parameter is a specification of RAID types required. For example, if you specify RAID(ALL) on the GLOBAL command and RAID(RAID1,RAID10) on a QUERY VOLUME command, your output is limited to RAID 1 and RAID 10 devices.

Possible values are:

- **ALL** (Default) List all types of RAID devices.
- **FTS** List FTS devices.
- **NONE** Do not list RAID devices.
- **RAID1** List RAID 1 devices.
- **RAID10** List RAID 10 devices.
- **RAID5** List RAID 5 devices.
- **RAID6** List RAID 6 devices.
- **RAIDS** List RAID S devices.
The READY parameter includes or excludes ready devices from a QUERY VOLUME device list:

**EXCLUDE**
Exclude ready devices from the QUERY VOLUME device list.

**INCLUDE**
(Default) Include ready devices on the QUERY VOLUME device list.

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

The READY parameter has a matching site option, &OPT_READY.

### READY (Yes | No)

The READY parameter specifies whether the target device is made ready to the host:

- **Yes** (Default) Specifies that the target device is made Ready to the host.
- **No** Specifies that 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 SNAP VOLUME with the READY parameter set (YES) or cleared (NO), the status of the target device may show AVAILB for up to 5 seconds.

**Example**

```
READY (N)
```

### RECALCULATE_FREESPACES (Yes | No)

The RECALCULATE_FREESPACES parameter determines whether freespace is recalculated only on effected volumes or on all candidate volumes after the current SNAP DATASET operation:

- **Yes** Recalculate freespace on all candidate volumes.
- **No** (Default) Recalculate specifics only on affected volumes.

Normally, after any allocation of a new dataset occurs, the free space amount is recalculated only on the affected volumes. If you specify RECALCULATE_FREESPACES(YES), the free space amount is recalculated on all candidate volumes.

This is especially useful when allocations are occurring in other jobs simultaneously with the execution of TimeFinder (for example: running multiple simultaneous TimeFinder jobs using the same target volumes).

The RECALCULATE_FREESPACES parameter as a site option, &RECALC_FREE.

**Example**

```
RECALCULATE_FREESPACES (YES)
```
REFVTOC (Yes | No)

The REFVTOC parameter enables or prohibits automatic running of ICKDSF for all SNAP VOLUME operations:

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 SNAP VOLUME processing completes. Otherwise, TimeFinder/Clone Mainframe Snap Facility 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.

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

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

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

Example

REFVTOC(Y)

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.

UNIT(cuu)

 Specifies the unit address of the gatekeeper.

Note: The MVS device number cannot be the unit address of a VDEV.

VOLUME(volser)

 Specifies the volser of the gatekeeper.
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 identifiers, separated by periods.

CONTROLLER

Optional. If using the REMOTE parameter, then it would be 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 uppercase 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.

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

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

Default value

None

Example

REMOTE(VOL(UMC001) RAGROUP(21) CONTROLLER(0001879-90171) )

REMOVE_REMOTE_extent_sessions(Yes|No)

The REMOVE_REMOTE_extent_sessions parameter allows or prohibits removal of any extent sessions found on a remote device that is being cleaned up:

Yes Allow removal of any extent session found on a remote device that is being cleaned up.

No (Default) Prohibit removal of any extent session found on a remote device that is being cleaned up.

Because the device involved in the extent session is remote, TimeFinder cannot determine whether the background activity for the extent session has completed. If the background activity for the extent session has completed, everything proceeds normally. If the background activity for the extent session has not completed, the target datasets is incomplete or corrupted when the session is removed.

The REMOVE_REMOTE_extent_sessions parameter has a matching site option, REMOVE_REMOTE.
You can issue RENAMEUnconditional for any snap dataset. It is a method to provide alternate naming conventions to components being copied. The \textit{oldnamemask} is used to match existing PATH or AIX names. The corresponding \textit{newnamemask} is used to transform the old name into a new name.

A maximum of 127 (\textit{oldnamemask}, \textit{newnamemask}) pairs may be specified.

\textbf{Note:} RENUNC is a valid alias for the RENAMEUnconditional parameter.

\textit{newnamemask}

Specifies a mask used to derive the new dataset name when the existing dataset name matches the corresponding \textit{oldnamemask} filtering criterion.

\textit{oldnamemask}

Specifies a mask to be used as a filtering criterion to check if it matches the dataset name.

\textit{pfx}

Specifies the prefix you want to use to replace the first-level qualifier of the dataset name. It is optional, but if specified, must be the first parameter in the list of sub-fields. The prefix is used only if the (\textit{oldnamemask}, \textit{newnamemask}) parameters are not specified or the \textit{oldnamemask} filters do not match the dataset name.

\textbf{Default value}

None

\textbf{REPLace (Yes | No)}

The REPLACE parameter establishes the REPLACE value for all operations. REPLACE specifies whether the source device is to overwrite data on an existing target device. 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.

\textbf{CAUTION}

Take care when using this parameter. Some datasets that appear to be empty actually contain data.
Values can be:

Yes Specifies that an existing target device with user data be overwritten. 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.

Depending upon the REUSE parameter, the existing target dataset may be scratched and reallocated (REPLACE(Y) REUSE(N)), or reused (REPLACE(Y) REUSE(Y)). Table 10 demonstrates the effect of the REPLACE and REUSE parameters:

Table 10 REPLACE and REUSE effects

<table>
<thead>
<tr>
<th>Target dataset</th>
<th>REPLACE</th>
<th>REUSE</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>New - does not exist</td>
<td>(Y) or (N)</td>
<td>(Y) or (N)</td>
<td>A new target dataset is created.</td>
</tr>
<tr>
<td>Exists</td>
<td>(N)</td>
<td>(N)</td>
<td>The action fails.</td>
</tr>
<tr>
<td>Exists</td>
<td>(N)</td>
<td>(Y)</td>
<td>Syntax error, REUSE(Y) is not allowed with REP(N).</td>
</tr>
<tr>
<td>Exists</td>
<td>(Y)</td>
<td>(N)</td>
<td>The existing target dataset is erased and a new target dataset is created.</td>
</tr>
<tr>
<td>Exists</td>
<td>(Y)</td>
<td>(Y)</td>
<td>The existing target dataset is reused. If it is not large enough, a new target dataset is created.</td>
</tr>
</tbody>
</table>

Note: If REPLACE(Y) and REUSE(Y) and TOLERATE_REUSE_FAILURE(Y) is specified, then initially an existing target dataset is reused. If for some reason, such as size or attributes, the target is not reusable, then the existing target dataset is erased and a new target dataset is created.

This optional parameter is valid only with the TARGET parameter.

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.

This REPLACE is valid only with the TARGET parameter.

The REPLACE parameter has a matching site option, &REPLACE.
REUSE(Yes|No[, WAIT])

The REUSE parameter specifies for all SNAP DATASET operations whether any existing target dataset is to be erased. It allows replacement of the contents of an existing target dataset without erasing and allocating it again.

Yes Specifies that an existing target dataset is not to be erased, but the existing allocation is to be reused.

No (Default) Specifies that the existing target dataset is to be erased.

WAIT The WAIT option allows the job to wait for the target dataset enqueue to become available before trying to delete it. If a wait time is specified in the TARGET_ENQ_DATASET_WAIT parameter, it is also used as the wait time for this parameter; otherwise REUSE(NO,WAIT) waits forever.

Specifying REUSE(YES) is valid only with REPLACE(YES).

If you specify REPLACE(YES) and REUSE(YES) when the existing target dataset is not large enough to hold the source dataset, TF/Clone tries to expand the existing target dataset until it is capable of holding the entire source dataset.

Note: If you specify REPLACE(YES) and REUSE(YES) and TOLERATE_REUSE_FAILURE(YES), then an existing target dataset is initially reused. If for some reason, such as size or attributes, the target to not reusable, the existing target dataset is erased and a new target dataset is created.

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

Example

REUSE(Y)

REUSE_AUTO_expand(Yes|No)

The REUSE parameter controls whether expanding an existing dataset is allowed:

Yes (Default) If the existing target dataset is not large enough, an attempt is made to expand it.

No If the existing target dataset is not large enough, no attempt is made to expand it.

Note: You can use RESUSE_AUTO_EXPAND(NO) with ALLOCATE_UNUSED_SPACE(NO) and TOLERATE_TRUNCATION(YES).

To take effect, REPLACE (YES) and REUSE(YES) must also be set.

Note: This parameter is used only for dataset copies, not for volume copies.

The REUSE_AUTO_expand parameter has a matching site option, &AUTOXPND.

SAVEDEV (EXCLUDE | INCLUDE)

The SAVEDEV parameter includes or excludes SAVEDEV devices from QUERY VOLUME device lists:

EXCLUDE Exclude SAVEDEV devices from QUERY VOLUME device list.

INCLUDE (Default) Include SAVEDEV devices on QUERY VOLUME device list.

The SAVEDEV parameter is also available as a site option, &OPT_SAVEDV.
**SELECTMULTI (ALL | ANY | FIRST)**

SELECTMULTI specifies the criteria you want to apply to volumes in a LOGINDYNAM list before processing selection can take place:

**ALL** *(Default)* All dataset volumes must be in the LOGINDYNAM list for the dataset to be selected for processing. If one volume is not in the list, then the dataset is not selected.

**ANY** Any of the dataset volumes must be found in the LOGINDYNAM list for the dataset to be selected for processing. If no volumes are found in the list, then the dataset is not selected.

**FIRST** Check the first source dataset volume. If that first source dataset volume is not found in the LOGINDYNAM list of volumes, that dataset is not selected.

Any volumes supplied to DFDSS’s INDYNAM are passed to the EMCSNAPI interface as if they were supplied by LOGINDYNAM with SELECTMULTI(ALL) specified.

**SESSION_LIST (Yes | No [, DETail | NODETail | DIFFerential])**

The SESSION_LIST query parameter indicates whether more detail is required about sessions active on a device. If you specify you want a list of active sessions (by selecting YES), you can then specify what kind of information you want about the sessions by choosing DETAIL, NODETAIL, or DIFFERENTIAL. Possible values are:

**DETail** Provide detail about the active sessions on the list.

**DIFFerential** The DIFFERENTIAL parameter allows you to add data to the QUERY volume report (messages ESNPP31I and ESNPP30I) about changed tracks on the source and target volumes. If you specify DIFFERENTIAL, you see the following additional fields in the report:

- **DIFF_CNT** = The total number of changed tracks on the source and target (determined by combining bit-masks before counting bits).
- **DIFF-SRC** = The total number of tracks changed on the source.
- **DIFF-TGT** = The total number of tracks changed on the target.

“QUERY VOLUME” on page 263 provides more information.

**No** Do not list the active sessions.

**NODETail** *(Default)* Do not provide detail about the active sessions on the list.

**Yes** *(Default)* List the active sessions.

---

**Note:** SESSIONLIST, SESS_LIST and SESSLIST are all valid short-forms for SESSION_LIST.

SESSION_LIST(Yes|No) has a matching site option, &SESSLIST.

SESSION_LIST with the DETail|NODETail parameter has a matching site option, &SESSDETL.

SESSION_LIST with the DIFFerential parameter has a matching site option, &SESSDIFF.

**SIZE (ALL | MOD1 | MOD2 | MOD3 | MOD9 | MOD27 | MOD54 | EAV | # | low-high)**

The SIZE parameter selects the devices to be listed based on the number of cylinders that a device has. You can specify multiple keywords in a single command. Separate each keyword with a space (as shown in the following example). You can also specify a range of values.
Possible values are:

- `#`  List devices of the specified number of cylinders.
- `ALL` *(Default)* List devices of all cylinder configurations.
- `EAV` List EAV (Extended Address Volume) devices.
- `low-high` List devices in the specified range (low to high) of cylinders.
- `MOD1` List devices of 1113 cylinders.
- `MOD2` List devices of 2226 cylinders.
- `MOD27` List devices of 32760 cylinders.
- `MOD3` List devices of 3339 cylinders.
- `MOD54` List devices of 65520 cylinders.
- `MOD9` List devices of 10017 cylinders.

**Example**

`QUERYVOLUME(SIZ(MOD9 MOD27))`

**SMS_PASS_volumes (Yes | No)**

The SMS_PASS_VOLUMES parameter allows you to change SMS processing so that you can supply volumes on a SNAP DATASET command that is passed to SVC99 and IDCAMS even through the dataset is an SMS-managed dataset. This allows the ACS routine to determine whether the supplied volumes is allowed or ignored.

**Note:** The ACS routines also determine if the UNITNAME or ESOTERIC is ignored.

Possible values are:

- `Yes` Allow SMS-managed datasets to be passed to SVC99 and IDCAMS, with a volume list, where the ACS routine can use or ignore them.

  **Note:** The *TimeFinder Utility for z/OS Product Guide* provides more information about IDCAMS.

- `No` *(Default)* Ignore any user-supplied volumes.

  This has no effect if the storage class is guaranteed space. With guaranteed space, the user-supplied volume list is always passed.

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

**SNAPSHOT_LIST (ALL | LINKED | NOT_LINKED | SNAPSHOT)**

The SNAPSHOT_LIST parameter is used to restrict the list of snapshots and links to one or more link status of snapshot:

- `ALL` *(Default)* List all links and snapshots.
- `LINKED` List links (target device is not `X'FFFFFFFF').
- `NOT_LINKED` List not linked snapshots (target device is `X'FFFFFFFF'` and there is no link with this snapshot).
  
  NLINKED is an alias of NOT_LINKED.
- `SNAPSHOT` List snapshots (target device is `X'FFFFFFFF').

---

1. Available starting with Mainframe Enablers 8.2.
The SNAPSHOT_LIST parameter is also available as a site option, &SNAPSHOT_LIST.

**SNAP_UNUSED_SPACE (Yes|No)**

The SNAP_UNUSED_SPACE parameter determines whether the tracks copied to the target dataset are only those tracks in the used portion of the source dataset or the entire allocation including both used and unused space:

- **Yes** *(Default)* Specifies copying all the tracks in the source dataset including both used and unused space.
- **No** Specifies only copying the tracks in the used portion of the source dataset.

This parameter only applies to sequential and standard partitioned datasets.

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

**SOFTlink (Yes|No)**

Determines whether softlinking is used:

- **Yes** Softlinking is used.
- **No** *(Default)* Softlinking is not used.

Determines whether softlinking is used.

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

The following table shows the actions that will be taken when SOFTlink (YES) is used in the SNAP VOLUME command.

**Table 11 Command actions when SOFTlink (YES) is specified**

<table>
<thead>
<tr>
<th>Command</th>
<th>Current status</th>
<th>Actions taken</th>
<th>End status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAP VOLUME</td>
<td>No relation between A and B</td>
<td>Create snapshot1 on A. Activate snapshot1 on A. Link snapshot1 to B.</td>
<td>The data at A has been replicated to B.</td>
</tr>
<tr>
<td>SNAP VOLUME (resnap)</td>
<td>Snapshot1 on A is linked to B.</td>
<td>Snapshot2 is created on A. Activate snapshot2 on A. Link snapshot2 to B. Terminate snapshot1 on A.</td>
<td>The original snapshot1 has been terminated (microcode automatically unlinks the existing link), and a new snapshot1 has been created activated and linked to B.</td>
</tr>
<tr>
<td>SNAP VOLUME (restore)</td>
<td>Snapshot1 on A is linked to B.</td>
<td>Create snapshot1 on B. Activate snapshot1 on B. Link snapshot1 to A. Snapshot1 on A is unlinked. Snapshot1 on A is terminated.</td>
<td>The data at B has been replicated to A.</td>
</tr>
<tr>
<td>SNAP VOLUME (cascading B-&gt;C)</td>
<td>Snapshot1 on A is linked to B.</td>
<td>Create snapshot1 on B. Activate snapshot1 on B. Link snapshot1 to C.</td>
<td>The data at B has been replicated to C.</td>
</tr>
</tbody>
</table>
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 182 describes use of the date and time variables.

The SOFTlink parameter is incompatible with the PARALLEL_CLONE parameter.

**SOURCE_VOLUME_LIST(vollist)**

The SOURCE_VOLUME_LIST parameter specifies the source volumes for SNAP DATASET and QUERY DATASET commands. You must have previously defined the source volume list name using a DEFINE SOURCE_VOLUME_LIST command.

vollist

The name of the defined volume list. The name can be up to 16 characters.

A single SNAP DATASET statement may reference cataloged datasets or datasets on offline volumes; but not both. In other words, if you use the SOURCE_VOLUME_LIST parameter in a SNAP DATASET command, TF/Clone selects only datasets from the volumes on the source volume list.

Note: “DEFINE SOURCE_VOLUME_LIST (TF/Clone)” on page 225 provides more information about how to create and use source volume lists.
SRDFA_CONSISTENT_RETRY(Yes|No|nn)

This parameter controls the retry attempts when SRDF/A is not consistent. The default value is 10. 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 switches 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 is also available as a site option, &SRDFA_RETRY.

Default value

10 (retry attempts)

SRDFA_R1_target(Yes|No|DATAMOVERName|PHYSical|INFormational)

The SRDFA_R1_TARGET parameter specifies how SRDF/A R1 devices are to be used if they are designated as target devices:

- DATAMOVERName: Allows SRDF/A R1 devices to be used as target devices as long as a datamover name is also specified. If you do not specify a datamover name, DATAMOVERNAME acts as if you specified No. If a device is not active on the link, it is treated like a non-SRDF device, so the datamover will not be used. If DATAMOVERNAME is used with the SRDFA_R1_TARGET parameter, the parameter DATAMOVERNAME(datamover utility) is also required.

- INFormational: Allows SRDF/A devices to be used as target devices and allow the operating environment to copy the track contents. An informational message is generated when R1 devices are targets.

- No (Default): Does not allow SRDF/A R1 devices to be used as target devices. Also generates an error message when an R1 device is the target.

- PHYSical: Allows SRDF/A devices to be used as targets. Forces an internal datamover (COPYCYL) to always be used to copy track contents to a targeted R1 device.

- Yes: Allows SRDF/A R1 devices to be used as target devices. A warning message is generated when an R1 device is the target.

The SRDFA_R1_TARGET parameter has a matching site option, &SRDFAR1.

TimeFinder in an SRDF environment

Whenever you use an SRDF/A R1 device as the target device, you receive messages to warn you that the data is not available on the R2 when the snap is first initiated. In fact, it could take some time for the data to actually be copied to the R2 device.

To allow an SRDF/A R1 device to be used as a target device, specify the SRDFA_R1_TARGET parameter indicating YES, DATAMOVER, or PHYSICAL.
On an SRDF/A device, the operating environment uses an internal copy mode to migrate the indirect tracks to the R1 device, and then copy and apply the tracks to the R2 device during normal SRDF/A cycles. Because it is difficult to know when the indirects are done copying, it is also hard to know which cycle may complete the copy to the R2. By default an error message is issued when the R1 device is an SRDF/A device.

**Note:** Protected and indirects are the Dell EMC terms for the controlled relationship of tracks that is established between a source and a target. Data on the source tracks is “protected” before being copied to a target’s “indirects”, or the tracks locations dedicated to receive the data.

If the data is critical in an outage situation, then you should use a physical datamover. This ensures that the data placement on the R2 device is complete. It is definitive that when the physical datamover is complete, the data is in a cycle to show up on the R2 (in SRDF/A mode).

**SRDFA R2 sync (WARNING | R1R2SYNC | DATAMOVER)**

The SRDFA_R2_SYNC parameter indicates the processing that should occur if TimeFinder detects that it can use the SRDF/A R2 device as the snap source in a two-storage system SRDF/A snap situation:

**DATAMOVER**  If you specify datamover name, then TimeFinder uses it to copy the data from the primary device, instead of snapping from the R2 device. If you do not specify a datamover name, the snap proceeds from the secondary device and a warning message is issued.

**R1R2SYNC**  The snap occurs from the R2 device, but a wait takes place. TimeFinder monitors the cycle values and waits for two complete cycles to pass to ensure that the contents of the primary device at the time the snap was invoked have propagated to the secondary device.

**WARNING**  (Default) A warning message is issued when a snap occurs from the secondary device and there is the possibility that the contents may not match the primary device at the time the snap is invoked.

This parameter only affects snaps that are not consistent. Consistent snaps always suspend cycle switching on the SRDF/A group.

The operation is a snap of the R1 source volume or dataset through the SRDF/A R2 device to a target in the same storage system where the R2 device resides.

With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, a TF/Snap off an active SRDF/A R2 device is supported with group and device level pacing set by SRDF HC commands. See the **SRDF Host Component for z/OS Product Guide** for more information.

The SRDFA R2 SYNC parameter has a matching site option, &SRDFAR2.
SRDFS_R1_target(Yes|No|DATAMOVERName|PHYSical|INFormationa)l)

The SRDFS_R1_TARGET parameter specifies how SRDF/S R1 devices are to be used if they are designated as target devices:

DATAMOVERName
Allows SRDF/S R1 devices to be used as target devices as long as a datamover name is also specified. If you do not specify a datamover name, DATAMOVERNAME acts as if you specified No.

If a device is not active on the link, it is treated like a non-SRDF device, so the datamover will not be used. If DATAMOVERNAME is used with the SRDFS_R1_target parameter, the parameter DATAMOVERNAME(datamover utility) is also required.

An informational message is generated that confirms the datamover is starting.

INFormationa)l
Allows SRDF/S devices to be used as target devices and allows the operating environment to be used to copy the track contents.

An informational message is generated when R1 devices are targets.

No
Prohibits SRDF/S R1 devices to be used as target devices.

Also generates an error message when an R1 device is the target.

PHYSical
Allows SRDF/S devices to be used as targets. Forces an internal datamover (COPYCYL) always to be used to copy track contents to a targeted R1 device.

Yes (Default) Allows SRDF/S R1 devices to be used as target devices.

A warning message is generated when an R1 device is the target.

The SRDFS_R1_TARGET parameter has a matching site option, &SRDFSR1.

TimeFinder in an SRDF environment

Whenever you use an SRDF/S R1 device as the target device, you receive messages that the data is not available on the R2 the instant the snap is initiated. In fact, it could take some time for the data to be copied to the R2 device.

On an SRDF/S device, the operating environment uses an internal copy mode to migrate the indirect tracks to the R1 device (independent of the source device). Then, the operating environment uses adaptive copy mode to migrate the copied tracks to the R2 device.

Because two background copies are occurring, the R1/R2 are normally equal, but may be out of sync for a very brief period of time. By default, a warning message is issued when the R1 device is a SRDF/S device.

If the data is critical is an outage situation, then a physical data move should be used, ensuring the data placement on the R2 device is complete. When the physical datamover is complete, the data is on the R2 (in SRDF/S mode).

STORAGECLASS(classname)

The STORAGECLASS parameter sets the SMS storage class for a newly allocated target dataset.
**classname**

Specifies the locally defined list of storage attributes required for the target dataset. Your storage administrator determines the valid storage class names for the site.

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

SAF or equivalent authorization is required if you are requesting the source class.

Local SMS ACS routines may place the target dataset in a storage class other than that specified by this parameter. As with all SMS datasets, specifying storage class is only a suggestion. SMS may or may not accept it.

TF/Clone does not assign the source storage class to a target dataset automatically unless you specify COPYSMS(STGCLASS). You must ensure that the correct storage class is assigned to the target dataset by using the STORAGECLASS parameter or ACS selection.

If an existing target dataset is reused, the storage class information is not changed.

**Default value**

None

**Example**

`STORCLASS(SITESSCL)`

**SYMDV# (symdv#)**

SYMDV# identifies the device number or a range of device numbers in the destination storage system for the command-specified operation. This is a device in a local storage system if the LOCAL parameter is used to identify the storage system. This is a device number in a remote storage system if the REMOTE parameter and RAGROUP subparameter is used to identify the storage system.

You can specify a single PowerMax/VMAX device number:

`SYMDV# (symdv#)`

You can also specify a range of device numbers. You can write 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# (lowsymdv#-highsymdv#)`

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

  `SYMDV# (lowsymdv#:highsymdv#)`

- Specify the starting device number in the range and a count value (in parentheses) that indicates how many additional devices there are between that number and the highest numbered device in the range. 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.)
The total number of devices in the range (that is, the count value) cannot exceed 256.

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

**TDEV (EXClude | INClude)**

This TDEV parameter determines whether thin devices are to be included in reports generated by the QUERY VOLUME command:

- **EXClude**  Exclude thin devices on QUERY VOLUME reports.
- **INClude**  (Default) Include thin devices on QUERY VOLUME reports.

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

**TDEV_RECLAIM (Yes | No)**

This TDEV parameter can initiate a reclaim of the target device after a full device clone operation to prevent any “unallocated but assigned” tracks on the source device to be copied to the target device.

- **Yes**  (Default) Initiate a reclaim of unallocated but assigned tracks.
- **No**  Copy without reclaiming unallocated but assigned tracks.

**Example**

```
TDEV_RECLAIM(NO)
```

**TERMINATE_SESSION_when_complete (Yes | No)**

The TERMINATE_SESSION_WHEN_COMPLETE parameter allows a full-volume session snap to terminate automatically after the copy is complete. This eliminates the need to remove the session by running a CLEANUP on the source device.

**Note:** “CLEANUP [EXTENT TRACK ON]” on page 217 provides more information about the CLEANUP command.

Possible values are:

- **Yes**  (Default) The full-volume snap session is terminated automatically when the background copy is complete.
- **No**  The full-volume snap session is not automatically removed. A CLEANUP request is necessary to remove the session after the background copy is complete.

The TERMINATE_SESSION_WHEN_COMPLETE parameter has a matching site option, &TERMSESS.

**TIMEOUT (nnn)**

The TIMEOUT parameter determines the maximum time ECA is active during consistent SNAP VOLUME operations.

**Note:** Enginuity Consistency Assist (ECA) is a feature of the operating environment. ECA (often called RDF-ECA, a part of SRDF consistency) provides an enterprise solution for ensuring dependent write consistency in SRDF/S configurations with more than one SRDF group. ECA requires that you have the

nnn

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

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

**Default value**

15 (seconds)

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

THINPOOL (EXCLUDE|INCLUDE)

Used to include or exclude thin pool log devices from a QUERY VOLUME device list.

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

**Default value**

INCLUDE

TOLERATE_REUSE_FAILURE(Yes|No)

The TOLERATE_REUSE_FAILURE parameter specifies whether to continue the snap operation if the target dataset is not reusable by scratching and reallocating the dataset:

<table>
<thead>
<tr>
<th>Yes</th>
<th>Continue the snap operation if the target dataset is not reusable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td><em>(Default)</em> Do not continue the snap operation if the target dataset is not reusable.</td>
</tr>
</tbody>
</table>

**Note:** If you specify REPLACE(YES) and REUSE(YES) and TOLERATE_REUSE_FAILURE(YES), then an existing target dataset is initially reused. If for some reason, such as size or attributes, the target is not reusable, the existing target dataset is erased and a new target dataset is created.

The TOLERATE_REUSE_FAILURE parameter has a matching site option, &REUSFAIL.

**Example**

TOL_REUS_F (YES)
TOLERATEALLOCATIONFailure(Yes|No)

The TOLERATEALLOCATIONFailure parameter specifies whether to override normal multiple target allocation failure procedure to allow execution to continue on successfully allocated targets:

- **Yes** (Default) Allows execution to continue even when some target datasets were not successfully allocated.
- **No** Specifies that execution is to stop and to erase any successfully allocated target datasets when any target datasets are not successfully allocated.

This parameter is useful where the target dataset name is wildcarded.

When multiple targets are allocated due to wildcarding, the normal procedure stops the action and deletes any successfully allocated targets as soon as any one allocation fails. Specifying TOLERATEALLOCATIONFailure overrides the normal procedures and allows the allocation and subsequent copy to continue on the successfully allocated targets.

When several target datasets are being allocated, it may be desirable for those which are successfully allocated to be copied even when some of the target dataset allocations fail.

The TOLERATEALLOCATIONFailure parameter has a matching site option, &ALLOFAIL.

TOLERATE_COPY_Failure(Yes|No)

The TOLERATECOPYFailure parameter specifies whether to override normal multiple extent copy procedure to allow execution to continue on successfully allocated targets:

- **Yes** Allows all existing target datasets to remain allocated after a copy failure.
- **No** (Default) Specifies that all allocated target datasets are deleted if a copy failure occurs.

When a copy process fails for any extent or group of extents, the normal procedure stops the action and deletes any successfully allocated datasets. Specifying the TOLERATECOPYFailure parameter overrides the normal procedures to prevent deletion of successfully allocated datasets and to continue with the copy.

The TOLERATECOPYFailure parameter prevents the target datasets from being deleted after a copy failure.

When several target datasets are being allocated, it may be desirable for those which are successfully allocated to be copied even when some of the target dataset allocations fail.

The TOLERATECOPYFailure parameter has a matching site option, &COPYFAIL.

TOLERATEENQFailure(Yes|No)

The TOLERATEENQFailure parameter allows you to snap datasets when exclusive serialization control over them cannot be obtained:

- **Yes** Enables the snapping of a dataset that is unavailable at the exclusive serialization level.
- **No** (Default) Specifies that the dataset is not to be snapped if it is unavailable at the exclusive serialization level.
Integrity of the dataset cannot be assured. You must use the TOLERATEENQFAILURE parameter, if you specify HOSTCOPYMODE(Excl), to snap a dataset that is unavailable at the exclusive serialization level.

This parameter also is used for snapping a volume that must remain allocated during the snap operation. An example is a volume in use by a database management system (DBMS).

The TOLERATEENQFAILURE parameter has a matching site option, &ENQFAIL.

TOLERATETRUNCATION(Yes | No)

The TOLERATETRUNCATION parameter set to YES allows a dataset to be truncated and a warning message to be issued if the target extent size is smaller than the source. TimeFinder truncates the dataset only if it cannot allocate more space. Truncation may cause loss of data.

If the TOLERATETRUNCATION parameter is defaulted, or set to NO, the snap fails if the target cannot be extended.

Because VSAM, PDSE, striped, or extended addressability datasets never allow truncation, this option has no effect on them. Values can be:

- Yes Enables the truncation of a data snapped to a smaller dataset.
- No (Default) Specifies that the dataset is not to be truncated if the target is smaller than the source dataset and the snap fails.

If the target dataset cannot be allocated as large as the source dataset (x37), TF/Clone allows the snap operation but truncates the data being snapped to the size of the target dataset.

This parameter is only active when the target dataset requires more space than its primary allocation.

The TOLERATETRUNCATION parameter has a matching site option, &TRUNC.

TOLERATEVSAMENQFailure(Yes | No)

If a VSAMENQMODE cannot be satisfied, the TOLERATEVSAMENQFailure parameter determines what happens:

- Yes Specifies that a warning message is issued and processing continues.
- No (Default) Specifies that an error message is issued and processing stops.

Note: Specifying TOLERATEVSAMENQFAILURE(YES) allows copy operations to proceed when datasets may be open for update by another job at the same time. This could compromise the integrity of the copied data.

If TOLERATEVSAMENQFAILURE (NO) is specified (or defaulted), an error message is issued and processing of the request terminates. If TOLERATEVSAMENQFAILURE (YES) is specified, a warning message is issued but processing of the request continues.

The TOLERATEVSAMENQFAILURE parameter has a matching site option, &VSAMFAIL.
TRACE(ON | OFF)

The TRACE parameter allows you to control whether trace messages are written to the message file:

ON  (Default) Enables trace records to be generated.
OFF  Disables trace records from being generated.

Note: This option should only be set on at the request of a Dell EMC Customer Support representative. Some of the information resulting from TRACE(ON) may be of use only to a Dell EMC Customer Support representative.

Example

TRACE(ON)

UNIT(cuu)

UNIT specifies the unit-address(es) of a device(s) on which the command-specified operation is to be performed. The unit value is a CCUU value.

You can specify a single unit:

UNIT(cuu)

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

- Specify the lowest addressed unit in the range and the highest addressed unit 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 units there are between that number and the highest unit in the range. The count value includes the lowest numbered unit in the range and the highest numbered unit in the range. (For example, to specify addresses between 10 and 13, enter 4 as the count.)
  The total number of units in the range (that is, the count value) cannot exceed 256.
  UNIT(cuu(count))

VARY_OFFline(AUTO | NEVER)

The VARY_OFFLINE parameter specifies whether the target device must be offline before the request processing begins:

AUTO  (Default) Use VARY OFFLINE when appropriate.
NEVER  Do not vary devices offline.

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.
When VARY_OFFLINE is set to AUTO and the device is already in the required state, TimeFinder takes no action. When VARY_OFFLINE is set to NEVER, TimeFinder also takes no action, but leaves the device in the existing state.

The VARY_OFFLINE parameter has a matching site option, &VARYOFF.

VARY_ONLINE(AUTO|Yes|No)

The VARY_ONLINE parameter specifies whether the target device must be online after the requested processing completes:

AUTO (Default) Use VARY ONLINE when appropriate.
No   Do not vary devices online.
Yes  Use VARY ONLINE all the time whatever the current 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.

When VARY_ONLINE is set to AUTO and the device is already in the required state, TimeFinder takes no action. When VARY_ONLINE is set to YES, TimeFinder always varies the device online regardless of the device’s current state. When VARY_ONLINE is set to NO, TimeFinder takes no action, but leaves the device in the existing state.

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.

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.

VCLOSE is applied only to online volumes.

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

Note: You must have purchased and installed the TF/Clone Licensed Feature Code to perform full-volume snaps.

The VCLOSE parameter has a matching site option, &VCLOSE.
VDevice(Excluded | Include)

The VDEVICE parameter excludes or includes virtual devices on QUERY VOLUME lists:

Excluded

INCLUDE (Default) Includes virtual devices from QUERY VOLUME lists.

The VDEVICE parameter is only available if you purchase the TF/Snap Licensed Feature Code.

The VDEVICE parameter has a matching site option, &OPT_VDEV.

Example

GLOBAL (VDEV (EXCLUDE))

VDEFWAIT(Yes | No)

Multiple jobs executing TF/Snap should not perform operations on the same VDEV at the same time. It causes confusion. The VDEFWAIT parameter indicates what should happen if TF/Snap attempts to perform an operation against a VDEV that already has another job operating against it.

If you allow the default (VDEFWAIT(NO)), an error message, ESNPT30E is issued that indicates that VDEV is in use. If you specify VDEFWAIT(YES), the additional TF/Snap job waits until the first TF/Snap job finishes with the VDEV before proceeding.

This behavior may cause unwanted actions on the VDEV, depending on the order of the jobs executing.

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

Default value

No

VERIFY(Yes | No | NEVER)

The VERIFY parameter causes TimeFinder to do an IDCAMS VERIFY on the target VSAM dataset:

NEVER

If you specify VERIFY(NEVER), no verification is performed, neither for open nor for non-open datasets.

VERIFY(NEVER) overrides VERIFY_OPEN_SOURCE(YES).

No

Directs TimeFinder not to perform an IDCAMS VERIFY on the target VSAM dataset.

Note: VERIFY_OPEN_SOURCE(YES) overrides the VERIFY(NO) parameter so that the verification is performed.

Yes

(Default) Directs TimeFinder to perform an IDCAMS VERIFY on the target VSAM dataset.

The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

The VERIFY parameter has a matching site option, &VERIFY.
VERIFY_OPEN_SOURCE([Yes|No])

The VERIFY_OPEN_SOURCE parameter determines whether TimeFinder issues a warning message (ESNPB21) and performs an IDCAMS VERIFY whenever an attempt is made to copy an open VSAM file. The purpose of this verification is to attempt to reset the VSAM OPEN indicator for those files that were not really open but had a VSAM OPEN indicator left set from a previous operation.

Possible values are:

Yes  Specifies that the message is issued and an IDCAMS VERIFY is performed whenever an attempt is made to copy an open VSAM file.

Note: VERIFY_OPEN_SOURCE(YES) overrides the VERIFY(NO) parameter so that the verification is performed. VERIFY(NEVER) overrides VERIFY_OPEN_SOURCE(YES) to disable verification for both open and non-open datasets.

No   (Default) Specifies that no message is issued and no IDCAMS VERIFY is performed whenever an attempt is made to copy an open VSAM file.

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

The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

VSAMENQMODE([SHAREd|EXClusive|NONE])

The VSAMENQMODE parameter identifies the type of VSAM ENQ testing to perform:

SHAREd   Specifies that an ENQ is issued with the SHR attribute. SHR is an alias of SHAREd.

EXClusive Specifies that an ENQ is issued with the EXClusive attribute.

NONE     (Default) Specifies that no testing of the SYSVSAM ENQ is performed.

If you specify VSAMENQMODE (SHAREd), then TF/Clone issues an ENQ with the SHR attribute. If the ENQ is satisfied, then processing continues normally. After the request is processed, TF/Clone issues a DEQ to release the resource. If the ENQ cannot be satisfied, the TOLERATEVSAMENQFAILURE parameter determines what happens.

If you specify TOLERATEVSAMENQFAILURE (NO) (or default it), TF/Clone writes an error message and terminates processing of the request. If you specify TOLERATEVSAMENQFAILURE (YES), TF/Clone writes a warning message and continues processing the request.

If you specify VSAMENQMODE (EXClusive), an ENQ is issued with the EXC attribute. If the ENQ is satisfied, processing continues normally. After the request is processed, then TF/Clone issues a DEQ to release the resource. If the ENQ cannot be satisfied, the TOLERATEVSAMENQFAILURE parameter determines what happens.

Note: Table 3 on page 120 provides additional information.

The VSAMENQMODE parameter has a matching site options, &VSAMENQ.
VOLUME(volser)

VOLUME specifies the volser of the volume on which the command-specified operation is to be performed.

**Note:** VOLUME(volser) allows only a single device.

WAITFORCOMPLETION([Yes|No|hh:mm:ss][,MeSsaGes][,R1R2SYNC][TIMEOUT(INFormational|WARNing|ERRor)])

The WAITFORCOMPLETION parameter specifies for all SNAP DATASET, SNAP VOLUME, and RESTORE VOLUME operations, whether TimeFinder is to wait for the copy operations to complete before terminating:

**Yes**
This causes TimeFinder to wait for the copy operations to complete before terminating.

**Note:** Under PowerMaxOS 5978 and HYPERMAX OS 5977, WAITFORCOMPLETION(YES) is ignored.

**No** *(Default)* This causes TimeFinder to terminate without waiting for copy operations to complete.

**R1R2SYNC**
Wait for the snap to an R1 to complete and for the R1 to complete synchronization with its partner R2.

**Note:** R1R2SYNC is not supported with HYPERMAX OS 5977 and PowerMaxOS 5978.000 to 5978.354.

**hh:mm:ss**
Wait for a specific time limit represented by hh:mm:ss, after which the system polling is stopped and TF/Clone reports an error condition and terminates the snap of the dataset.

**MeSSaGes**
Displays an ongoing status message while waiting for the copy operation to complete.

**TIMEOUT**(INFormational|WARNing|ERRor)
Specify the type of error message to be issued.

Default value is INFormational.

> 20000  Wait 60 seconds before next check.
> 10000  Wait 30 seconds before next check.
> 5000   Wait 15 seconds before next check.
> 1000   Wait 5 seconds before next check.
> 500    Wait 2 seconds before next check.
> 200    Wait 1 second before next check.
> 100    Wait 1/2 second before next check.
Otherwise wait 1/10 second before next check.

**Example**

WAITFORCOMPLETION(Y,MSG)
At program termination, wait for the copy to complete within the storage system. A status message is written each time the storage system is checked, identifying the number of tracks remaining to be processed.

**WAIT_FOR_DEFINITION** *(Yes|No)*

Determines whether the STOP SNAP TO VOLUME command will wait for all of the target tracks to be defined before unlinking the target device:

- **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.

This parameter requires PowerMaxOS 5978 or HYPERMAX OS 5977.

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.

Aliases for **WAIT_FOR_DEFINITION** include **WAIT_FOR_D** and **WFD**.

The **WAIT_FOR_DEFINITION** parameter has a matching site option, **&WFDEF**.

**WAIT_FOR_PRECOPY_PASS1** *(Yes|No)*

The **WAIT_FOR_PRECOPY_PASS1** parameter determines for SNAP VOLUME and ACTIVATE GROUP requests whether any precopy activity must have completed one pass of the whole volume before the appropriate ACTIVATE occurs:

- **Yes** Wait for the precopy to complete one pass before activating the devices.
- **No** *(Default)* Do not wait for the precopy to complete one pass. Activate the devices.

The **WAIT_FOR_PRECOPY_PASS1** parameter has a matching site options, **&WAIT_PRECOPY**.

**WAITforsession** *(Yes|No|hh:mm:ss)*

The **WAITforsession** parameter controls whether TimeFinder is to wait for available sessions on the source device. Each time a dataset is snapped, a PowerMax/VMAX session is required.

Each source device is allowed multiple sessions for processing datasets.

**Note:** “TimeFinder and protection sessions” on page 38 provides details on protection session limits.
Many datasets may share each session, but if a single dataset is snapped multiple times, each snap of that dataset requires its own session. After a snap of a dataset completes, you can reuse that session. Also, if the volume is snapped twice, then a single dataset may only be snapped two additional times before the sessions are exhausted.

Possible values are:

- **hh:mm:ss** If all sessions are in use, TF/Clone waits for the requested time period (limit) specified by `hh:mm:ss`. If a session completes within the time period, TF/Clone continues processing normally. If the time period expires and all four sessions are still in use, TF/Clone reports an error condition and terminates the snap of the dataset.
- **No** *(Default)* If all sessions are in use, TF/Clone reports an error condition and terminates the snap of this dataset.
- **Yes** If all sessions are in use, TF/Clone waits for one to finish and then continues processing normally.

The WAITFORSESSION parameter has a matching site option, &WAIT.

**WHEN_SAVEDEV_FULL (READY | NOTREADY)**

The WHEN_SAVEDEV_FULL parameter determines the state of a virtual device that encounters a snap pool device full condition. Possible values are:

- **READY** *(Default)* Track is marked invalid but device remains available.
- **NOTREADY** Track is marked invalid and the device is made not ready.

The WHEN_SAVEDEV_FULL parameter has a matching site option, &SAVEFULL.

**Example**

```
WHEN_SAVEDEV_FULL (NOTREADY)
```
ACTIVATE

The ACTIVATE command determines when the preceding SNAP VOLUME or SNAP DATASET actions are to take place. ACTIVATE optionally specifies whether the SNAP actions are to be performed using Enginuity Consistency Assist (ECA) to form consistent point-in-time volume snaps.

For a SNAP DATASET with ACTIVATE and CONSISTENT, ECA and PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876 are required.

ACTIVATE applies to SNAP VOLUME and SNAP DATASET commands preceding it in the input stream, but after any previous ACTIVATE command.

SNAP VOLUME
The SNAP VOLUME may use:
- Physical target volumes
- Virtual devices\(^1\)
- Mixed target references
- Remote volumes

For example:

SNAP VOLUME (SOURCE(VOLUME(VOL001)))
SNAP VOLUME (SOURCE(UNIT(AA10)) TARGET(UNIT(BA00)))
ACTIVATE(CONSISTENT(YES))

SNAP VOLUME (SOURCE(VOLUME(VOL002)) TARGET(UNIT(C100)))
SNAP VOLUME (SOURCE(VOLUME(VOL003)) TARGET(UNIT(C101)))
SNAP VOLUME (SOURCE(VOLUME(VOL004)) TARGET(UNIT(C102)))
SNAP VOLUME (SOURCE(VOLUME(VOL005)) TARGET(UNIT(C103)))
ACTIVATE

The first ACTIVATE requests a CONSISTENT snap using ECA of volume VOL001 and the volume at address AA10. Note that the second snap is a physical snap to the target address BA00.

The second ACTIVATE refers to the snap of volumes VOL002, VOL003, VOL004, and VOL005 to the respective targets.

ACTIVATE allows you to specify when a group of SNAP VOLUMEs occurs and whether to use ECA to form a consistent snap operation.

ACTIVATE allows you to specify when a group of SNAP VOLUMEs occurs and whether to use ECA to form a consistent snap operation.

The MESSAGE(DISPLAY) parameter provides more control to automate controls to outside applications such as quiescing a DB2 database in coordination with snap commands.

SNAP DATASET
With SNAP DATASET commands, the ECA mechanism is at the volume level and access to other datasets is affected while the ECA window is active. Dependent-write consistency is provided across a group of target datasets.

WARNING

With SNAP DATASET and CONSISTENT(YES) parameter, only inter-dataset dependent write consistency is provided. Intra-dataset (metadata) consistency is not guaranteed. Users must ensure that metadata changes, such as additional extents, DO NOT occur during consistent dataset snap processing.

1. If you have installed the TF/Snap licensed feature code.
Syntax

ACTIVATE [(optional_parameters)]

Where optional_parameters are as follows:

[ACTIVATE_SUBTASK#(nnn)]

[CONSISTENT (Yes | No)]

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

[MESsages (DISplay | PROMpt | NONE | DETAIL)]

[POSTSNAP (Yes | No)]

[PRESNAP (Yes | No)]

[SRDFA_CONSISTENT_RETRY (Yes | No | nn)]  

[TIMEOUT (nnn)]

Note: 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.

Optional parameters

ACTIVATE_SUBTASK#(nnn)

See “ACTIVATE_SUBTASK#(nnn)” on page 153.

Note: There is an associated site option and global command parameter.

CONSISTENT (Yes | No)

See “CONSISTENT (Yes | No)” on page 159.

For the duration of the ACTIVATE command, CONSISTENT overrides any value set by the GLOBAL command CONSISTENT parameter or by the &CONSIST site option.

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

See “GROUP (grpname[, grpname,...])” on page 173.

MESsages (DISplay | PROMpt | NONE | DETAIL)

See “MESsages (DISplay | PROMpt | NONE | DETAIL)” on page 176.

For the duration of the ACTIVATE command, MESSAGES overrides any value set by the GLOBAL command MESSAGES parameter or by the &MESSAGE site option.

POSTSNAP (Yes | No)

See “POSTSNAP (Yes | No)” on page 187.

This parameter may be only used if GROUP is also specified.

See “ACTIVATE_SUBTASK#” on page 54 for a description of the site option that can help to minimize the ECA window when multiple storage systems and subsequent syccalls are required.
PRESNAP (Yes | No)
   See “PRESNAP(Yes|No)” on page 187.
   This parameter may only be used if GROUP is also specified.

SRDFA_CONSISTENT_RETRY (Yes | No | nn)
   See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 199.

TIMEOUT (nnn)
   See “TIMEOUT(nnn)” on page 203.

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

   Example

   ACTIVATE (TIMEOUT(10))
CLEANUP [EXTENT TRACK ON]

The CLEANUP command checks each extent track on the indicated volume to determine whether it is complete. CLEANUP then removes each completed extent in the extent track. If all individual extents within a session are completed, CLEANUP also frees the session.

CLEANUP can support both local and remote operations with full-volume cleanups. With PowerMaxOS 5978, HYPERMAX OS 5977 and Enginuity 5876, CLEANUP supports cleanups on remote dataset extents.

The CLEANUP command must be executed against source devices. The following show two different ways of cleaning up a range of devices:

```plaintext
CLEANUP EXTENT TRACK ON UNIT (0C00-0C1F)
CLEANUP EXTENT TRACK ON UNIT (0C20-0C20)
```

**Note:** Use REMOTE and LOCAL with SYMDV# parameters only with full-volume sessions. Extent sessions only work if you omit the REMOTE and LOCAL parameters. As a result, datasets cannot be snapped remotely and CLEANUP does not affect remote dataset extents or sessions.

Starting with Mainframe Enablers 8.3, when terminating a hardlinked snapshot with the help of CLEANUP EXTENT command with the FORCE_COMPLETION(YES) parameter specified, the default behavior is to wait for the target tracks to be defined.

**Syntax**

```
CLEANUP [EXTENT TRacK ON]
VOLUME(volser)|UNIT(cuu)|SYMDV#(symdv#))| [optional_parameters]
```

Where **optional_parameters** are as follows:

- [AUTOmatic_CLEANUP_R2(Yes|No)]
- [CLEANup_DIFFerential(Yes|No)]
- [CONTROLLER([xxxxxxx-]xxxxx|name)]
- [FORCE_COMPLETION(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)])]
- [REMOVE_REMOTE_extent_sessions(Yes|No)]
Note: 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.

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

Note: The GROUP parameter is an alternative to VOLUME, UNIT, or SYMDV# parameters and cannot be used together in the same CLEANUP command.

Required parameters

SYMDV# (symdv#)
See “SYMDV#(symdv#)” on page 202.

UNIT (cuu)
See “UNIT(cuu)” on page 207.

VOLUME (volser)
VOLUME specifies the volser of the volume on which the command-specified operation is to be performed.

Note: VOLUME(volser) allows only a single device.

Optional parameters

AUTOMATIC_CLEANUP_R2 (Yes | No)
See “AUTOMATIC_CLEANUP_R2(Yes|No)” on page 154.

CLEANUP_DIFFERENTIAL (Yes | No)
See “CLEANUP_DIFFERENTIAL(Yes|No)” on page 158.
For the duration of the CLEANUP command, CLEANUP_DIFFERENTIAL overrides any value set by the GLOBAL command CLEANUP_DIFFERENTIAL parameter or by the &CLEANDIFF site option.

CONTROLLER ([xxxxxxx-]xxxxx | name)
See “CONTROLLER([xxxxxxx-]xxxxx|name)” on page 159.

FORCE_COMPLETION (Yes | No)
See “FORCE_COMPLETION(Yes|No)” on page 172.
For the duration of the CLEANUP command, FORCE_COMPLETION overrides any value set by the GLOBAL command FORCE_COMPLETION parameter or by the &FORCECMP site option.
GROUP(grpname[, grpname,...])

See “GROUP(grpname,grpname,...)” on page 173.

**Note:** The GROUP parameter is an alternative to VOLUME, UNIT, or SYMDV# parameters and cannot be used together in the same CLEANUP command.

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 175.

Cleanup of extent track contents (dataset level compared to cleanup of full device sessions) requires the UNIT(cuu) or VOLUME parameter, and extent track cleanup (dataset level) is not performed when the SYMDV# is used.

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 190.

CLEANUP supports only full volume remote sessions. Therefore, the REMOTE parameter only works with full-volume sessions. REMOTE does not work with remote extent sessions.

REMOVE_REMOTE_extent_sessions(Yes|No)

See “REMOVE_REMOTE_extent_sessions(Yes|No)” on page 191.

For the duration of the CLEANUP command, REMOVE_REMOTE_extent_sessions overrides any value set by the GLOBAL command REMOVE_REMOTE_extent_sessions parameter or by REMOVE_REMOTE site option.

**Example**

CLEANUP EXTENT TRACK ON VOLUME (USER00)

Or, for a remote volume:

CLEANUP ( SYMDV# (032E) REMOTE(VOLUME(UMC001) RAGROUP(21) - CONTROLLER(0001879-90171) )

Traditional TimeFinder commands 219
CONFIG (TF/Clone)

The CONFIG command specifies various TimeFinder/Clone Mainframe Snap Facility settings for the target devices.

**Note:** You can use this command only if you install the TF/Clone licensed feature code. The *Mainframe Enablers Installation and Customization Guide* provides more information.

**Syntax**

```
CONFIG (Target (Volume(volser) | UNIT(cuu) | SYMDV#(symdv#)) | [optional_parameters])

Where optional_parameters are as follows:

[ALLOW_REPLICATION(Yes|No)]
[CONTROLLER([xxxxxxx-]xxxxx|name)]
[EXPIration(days)]
[GROUP(grpname[, grpname,..])]  
[LOCAL({UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name]) |
VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name])] |
DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name]) |
CONTROLLER([xxxxxxx-]xxxxx|name)]
})]

[MODE(COPY|NOCOPY|NOCOPYRD)]
[NAME(snapshot_name)]
[READY(Yes|No)]
[RELEASE(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)])])

[SNAPSHOTID(id)]
```

**Note:** CNFG is an alias of CONFIG.
Note: 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.

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

Note: The NAME(snapshot_name) and EXPIration(days) parameters are required when configuring a softlinked snapshot.

Required parameters

```
TaRGet(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#))
```

The TARGET parameter specifies the device(s) on which the command-specified operation is to be performed.

Note: You can only use this parameter if you install the TF/Clone licensed feature code.

```
VOLUME(volser)
```

VOLUME specifies the volser of the volume.

Note: VOLUME(volser) allows only a single device.

```
UNIT(cuu)
```

See “UNIT(cuu)” on page 207.

Note: If you specify a range of UCBs, you cannot use some of the optional parameters, such as NEWVOLID and VOLUME. In the case of SNAP VOLUME, you must specify the same number of UCBs in the SOURCE and TARGET parameters.

```
SYMDV#(symdv#)
```

See “SYMDV#(symdv#)” on page 202.

Optional parameters

```
ALLOW_REPLICATION(Yes|No)
```

This parameter is used to set or reset the FlashCopy inhibit outboard bit for devices. When this bit is set to NO, the device cannot be used in any local or remote replication.

Note: Aliases for ALLOW_REPLICATION include ALLOW_FLASHCOPY, ALLOW_FC and ALLOW_COPY.
When the parameter is set to NO, the following message is issued:

ESNP982E TARGET DEVICE HAS "INHIBIT OUTBOARD COPY" SET, PREVENTING MICROCODE COPIES

Default value

None

Example

```
CONFIG (GROUP(SNP8510) -
  MODE(COPY) -
  READY(YES) -
  RELEASE(YES) -
  ALLOW_REPLICATION(YES)
)
```

CONTROLLER([xxxxxxx-]xxxxx|name)

See “CONTROLLER([xxxxxxx-]xxxxx|name)” on page 159.

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

EXPIration(days)

A "time-to-live" value may be associated with the snapshot by using the EXPIRATION parameter.

Note: The EXPIration(days) parameter is required when configuring a softlinked snapshot.

The expiration value is specified as number of days from 0-3600 decimal.

The expiration time specified is relative to the command execution time, not the snapshot creation time. Specifying a value of 0 will remove the expiration time and the snapshot will never expire.

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

See “GROUP(grpname[,grpname,...])” on page 173.

Examples

You can use the GROUP parameter to specify the devices; for example:

```
CONFIG (GROUP(SNP7350) -
  MODE(COPY) -
  READY(YES) -
  RELEASE(YES) -
)
```

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 175.
The LOCAL parameter is only needed and can only be used if you use the SYMDV# parameter.

**MODE (COPY | NOCOPY | NOCOPYRD)**

See “MODE(COPY|NOCOPY|NOCOPYRD)VSE)” on page 178.

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

**Note:** For sessions created with MODE(VSE), the MODE parameter is ignored and cannot affect those sessions.

**NAME(snapshot_name)**

See “NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%%])” on page 182.

The NAME parameter is ignored when specified with the following parameters:

- MODE(COPY)

**READY (Yes | No)**

See “READY(Yes|No)” on page 189.

**Default value**

YES

**Example**

READY(Y)

**RELEASE (Yes | No)**

The RELEASE parameter specifies whether a hold is placed on the target device:

- Yes  Specifies that the target device, which has been held by a RELEASE(N) command, is made available for operations.
- No  Specifies that the target device is not available for operations.

**Default value**

There is no default value for this parameter.

**Example**

RELEASE(Y)

**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 190.

The REMOTE parameter is only needed and can only be used if you use the SYMDV# parameter.
Default value
None

Example
REMOTE(VOL(UMC001) RAGROUP(21) CONTROLLER(90171))

SNAPSHOTID(id)
The ID of the snapshot to be processed.
DEFINE SOURCE_VOLUME_LIST (TF/Clone)

You can use DEFINE SOURCE_VOLUME_LIST to create a list of offline devices. When you use the DEFINE SOURCE_VOLUME_LIST, you assign a name (of up to 16 characters) to the list. Then, you specify the devices you want to include in the list.

After you create the list, you can specify that list (as an argument to the SOURCE_VOLUME_LIST parameter) to the QUERY DATASET and SNAP DATASET commands.

You must use DEFINE SOURCE_VOLUME_LIST to define a source volume list before you use it with QUERY DATASET and SNAP DATASET. The source volume list you specify is not stored. Therefore, you must supply it (through a new DEFINE SOURCE_VOLUME_LIST) every time you use it in QUERY DATASET and SNAP DATASET.

Syntax

DEFINE SOURCE_VOLUME_LIST sourcevollist
  (UNIT(cuu)|VOLUME(volser))

Required parameters

sourcevollist
   The list name. The name can be a text string of up through 16 characters.

UNIT(cuu)
   See “UNIT(cuu)” on page 207.

VOLUME(volser)
   The VOLUME parameter specifies a volume label or a mask for matching volume labels.
   If the volser contains a hyphen, enclose it in single quotes, for example: VOL(‘vol-ser’)

Example

The following example defines a source volume list (BAPVOLS) with offline volumes, then uses that list to snap the volumes.

* * DEFINE *
* DEFINE SOURCE_VOLUME_LIST BAPVOLS ( -
  UNIT(6EF0) -
  UNIT(6EF6-6EF7) -
  VOL(U6A230) -
  VOL(U6A23*) -
  )
* *
* SNAP *
SNAP_DATASET (SOURCE(HLQ.TESTING.TWOVOL) -
TARGET(HLQ.TESTING.NEWTWO) -
HOSTCOPYMODE(NONE) -
SOURCE_VOLUME_LIST (BAPVOLS) -
REPLACE(Y) -
REUSE(N) -
VOL(U6A231,U6A230) -
)

1. With Mainframe Enablers 8.1 and later.
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 as follows:

Table 12 Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVATE_SUBTASK# (nnn)</td>
<td>153, 233</td>
</tr>
<tr>
<td>ADMINISTRATOR (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>ALLOCATE_UNUSED_SPACE (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>ALLOCATION_SEQUENCE (DATASET</td>
<td>NONE</td>
</tr>
<tr>
<td>ALLOW_FBA_META (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>AUTOMATIC_ACTivate (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>AUTOMATIC_CLEANup (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>AUTOMATIC_CLEANUP_R2 (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>AUTOMATIC_DEALLOC (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>AUTOMATIC_RELEASE_hold (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>BACKGROUNDCOPY (Yes</td>
<td>No</td>
</tr>
<tr>
<td>BCVOnly (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>BUILD_VTOCIX (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>CACHE_FULL_SYM (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>CATalog (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>CHECKBCVholdstatus (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>CHECKONLINEpathstatus (Yes</td>
<td>No</td>
</tr>
<tr>
<td>CHECK_POOL_usable (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>CKD (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>CLEANup_DIFFerential (Yes</td>
<td>No)</td>
</tr>
</tbody>
</table>
Table 12  Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLLAPSE_dataset_extents(VSAM</td>
<td>NONVSAM</td>
</tr>
<tr>
<td>CONDITIONVolume(ALL</td>
<td>DUMP</td>
</tr>
<tr>
<td>CONSISTENT(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>COPYValid(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASs] [STORageCLASs] [ALL])</td>
<td>160, 237</td>
</tr>
<tr>
<td>DATACLASs(classname)</td>
<td>162, 237</td>
</tr>
<tr>
<td>DaTaMoverNaMe(ADRDSSU</td>
<td>COPYCYL</td>
</tr>
<tr>
<td>DATASET_CHANGED_indicator(SET</td>
<td>RESET</td>
</tr>
<tr>
<td>DEBUG(ALL</td>
<td>EXTRA</td>
</tr>
<tr>
<td>DEBUG_EXTENTS(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_ADMIN(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_CC(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_OPTimize(n)</td>
<td>239</td>
</tr>
<tr>
<td>Differential(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DIFFERENTIAL_DATASET(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EATTR(NO</td>
<td>OPT)</td>
</tr>
<tr>
<td>EMUL_TYPE(ALL</td>
<td>HARDLINK</td>
</tr>
<tr>
<td>ENQSCOPE(REQUEST</td>
<td>STEP)</td>
</tr>
<tr>
<td>ENQWAIT(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>ERROR_CHECKING(NORMAL</td>
<td>REDUCED)</td>
</tr>
<tr>
<td>ERRORDisposition(DELETE</td>
<td>KEEP)</td>
</tr>
<tr>
<td>ERROR_RECovery(NORMAL</td>
<td>ENHANCED)</td>
</tr>
<tr>
<td>ESNP119(WARNING</td>
<td>ERROR)</td>
</tr>
<tr>
<td>ESNP220(ERROR</td>
<td>WARNING)</td>
</tr>
<tr>
<td>EXAMINE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EXclude_PathGroupID(pathlist)</td>
<td>169, 241</td>
</tr>
<tr>
<td>EXPLAIN(VOLUME_SELECTION(Yes</td>
<td>No))</td>
</tr>
<tr>
<td>EXTENT_ALLOCAtion(Yes[,CONSOLIDATE_VOLUME</td>
<td>,CONSOLIDATE_ALL]</td>
</tr>
</tbody>
</table>
### Table 12 Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTALLOC_EMC_ONLY (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EXTENT_EXPAND (Yes</td>
<td>No, [ADDNEW (Yes</td>
</tr>
<tr>
<td>FBA (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>FLASH_SNAP (FLASHCOPY</td>
<td>SNAP)</td>
</tr>
<tr>
<td>FORCE (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>FORCE_COMPLETION (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>FREESPACE (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>GROUP_DATASET_NAME (‘dataset_name’)</td>
<td>242</td>
</tr>
<tr>
<td>GROUP_DEVICE_ready_state (AUTO</td>
<td>NEVER)</td>
</tr>
<tr>
<td>GROUP_EMCQCAPI_VERIFY (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>HostcopyMODE (SHARED</td>
<td>EXCLUSIVE</td>
</tr>
<tr>
<td>INVALIDATE_PDSE_buffers (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>LIST ([NO]STATEMENTS)</td>
<td>[NO]HISTORY)</td>
</tr>
<tr>
<td>LOGINDYMAN (volume[, volume...])</td>
<td>176, 244</td>
</tr>
<tr>
<td>ManageMENCLASs (classname)</td>
<td>176, 244</td>
</tr>
<tr>
<td>MAXIMUM_ADDRDSSU_address_spaces (number)</td>
<td>244</td>
</tr>
<tr>
<td>MAXIMUM_SUBTASKS (number1, number2)</td>
<td>245</td>
</tr>
<tr>
<td>MAXRC (return_code_value)</td>
<td>245</td>
</tr>
<tr>
<td>MESSAGES (DISPLAY</td>
<td>PROMPT</td>
</tr>
<tr>
<td>MIGRATE ([PURGE (Yes</td>
<td>No)] [RECALL (Yes</td>
</tr>
<tr>
<td>MODECOPYFINISH</td>
<td>247</td>
</tr>
<tr>
<td>MODE (COPY</td>
<td>NOCOPY</td>
</tr>
<tr>
<td>MULTI_VIRTual (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>MULTI_LINE_query (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>NAME (snapshot_name [%date [4</td>
<td>6</td>
</tr>
<tr>
<td>NOTIFYwhencomplete ([GROUP (name)] [DATASET</td>
<td>JOB</td>
</tr>
<tr>
<td>NOTREADY (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>PARallel (Yes</td>
<td>No)</td>
</tr>
</tbody>
</table>
### Table 12  Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLLAPSE_dataset_extents(VSAM</td>
<td>NONVSAM</td>
</tr>
<tr>
<td>CONDITIONVolume(ALL</td>
<td>DUMP</td>
</tr>
<tr>
<td>CONSISTENT(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>COPYValid(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>COPYsourceSMSclasses([DATACLASSs] [ManageMenTCLASSs] [STORAGECLASSs] [ALL])</td>
<td>160, 237</td>
</tr>
<tr>
<td>DATACLASSs(classname)</td>
<td>162, 237</td>
</tr>
<tr>
<td>DaTaMoverNaMe(ADRDSSU</td>
<td>COPYCYL</td>
</tr>
<tr>
<td>DATASET_CHANGED_indicator(SET</td>
<td>RESET</td>
</tr>
<tr>
<td>DEBUG(ALL</td>
<td>EXTRA</td>
</tr>
<tr>
<td>DEBUG_EXTENTS(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_ADMIN(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_CC(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DFDSS_Optimize(n)</td>
<td>239</td>
</tr>
<tr>
<td>DIFFerential(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>DIFFERENTIAL_DATASET(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EATTR(NO</td>
<td>OPT)</td>
</tr>
<tr>
<td>EMUL_TYPE(ALL</td>
<td>HARDLINK</td>
</tr>
<tr>
<td>ENQSCOPE(REQUEST</td>
<td>STEP)</td>
</tr>
<tr>
<td>ENQWAIT(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>ERROR_Checking(NORMAL</td>
<td>REDUCED)</td>
</tr>
<tr>
<td>ERROR_DISPosition(DELETE</td>
<td>KEEP)</td>
</tr>
<tr>
<td>ERROR_RECovery(NORMAL</td>
<td>ENHANCED)</td>
</tr>
<tr>
<td>ESNP119(WARNING</td>
<td>ERROR)</td>
</tr>
<tr>
<td>ESNP220(ERROR</td>
<td>WARNING)</td>
</tr>
<tr>
<td>EXAMINE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EXclude_PathGroupID(pathlist)</td>
<td>169, 241</td>
</tr>
<tr>
<td>EXPlain(VOLUME_SELECTION(Yes</td>
<td>No))</td>
</tr>
<tr>
<td>EXTENT_ALLOCa tion(Yes[,CONSOLIDATE_VOLUME</td>
<td>,CONSOLIDATE_ALL]</td>
</tr>
</tbody>
</table>
### Table 12  Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTALLOC_EMC_ONLY (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>EXTENT_EXPAND (Yes</td>
<td>No, [ADDNEW (Yes</td>
</tr>
<tr>
<td>FBA (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>FLASH_SNAP (FLASHCOPY</td>
<td>SNAP)</td>
</tr>
<tr>
<td>FORCE (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>FORCE_COMPLETION (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>FREESPACE (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>GROUP_DATASET_NAME (‘dataset_name’)</td>
<td>242</td>
</tr>
<tr>
<td>GROUP_DEVICE.ready_state (AUTO</td>
<td>NEVER)</td>
</tr>
<tr>
<td>GROUP_EMCQCAPI_VERIFY (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>HostcopyMODE (SHAReD</td>
<td>EXclusive</td>
</tr>
<tr>
<td>INVALIDATE_PDSE_buffers (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>LIST ([NO]Statements) ([NO]HISTORY)</td>
<td>175, 244</td>
</tr>
<tr>
<td>LOGINDYMAN (volume [, volume ...])</td>
<td>176, 244</td>
</tr>
<tr>
<td>ManageMenTCLASS (classname)</td>
<td>176, 244</td>
</tr>
<tr>
<td>MAXIMUM_ADDRDSSU_address_spaces (number)</td>
<td>244</td>
</tr>
<tr>
<td>MAXIMUM_SUBTASKS (number1, number2)</td>
<td>245</td>
</tr>
<tr>
<td>MAXRC (return_code_value)</td>
<td>245</td>
</tr>
<tr>
<td>MESSages (DISplay</td>
<td>PROMpt</td>
</tr>
<tr>
<td>MIGrate ([PURge (Yes</td>
<td>No)] [RECall (Yes</td>
</tr>
<tr>
<td>MODECOPYFINISH</td>
<td>247</td>
</tr>
<tr>
<td>MODE (COPY</td>
<td>NOCOPY</td>
</tr>
<tr>
<td>MULTI_VIRTual (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>MULTI_LINE_query (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>NAME (snapshot_name [%date [4</td>
<td>6</td>
</tr>
<tr>
<td>NOTIFYwhencomplete ([GROUP (name)] [DATASET</td>
<td>JOB</td>
</tr>
<tr>
<td>NOTREADY (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>PARallel (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>PARALLEL_CLONE(Yes</td>
<td>No</td>
</tr>
<tr>
<td>PERSISTent (Yes</td>
<td>No)</td>
</tr>
<tr>
<td>POOL(poolname)</td>
<td>186, 248</td>
</tr>
<tr>
<td>PRECOPY(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>PREPARE_FOR_SNAP(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>R1FULLCOPYonly(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>RAID(ALL</td>
<td>NONE</td>
</tr>
<tr>
<td>Ready (EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>RECALCULATE_FREESPACE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>RENAMEUnconditional(pfx)</td>
<td>192, 250</td>
</tr>
<tr>
<td>RENAMEUnconditional((pfx) (oldnamemask,newnamemask)...)</td>
<td></td>
</tr>
<tr>
<td>RENAMEUnconditional((oldnamemask,newnamemask)...)</td>
<td></td>
</tr>
<tr>
<td>REFVTOC(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>REMOVE_REMOTE_extent_sessions(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>REPLace(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>RESERVE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>REUSE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>REUSE_AUTO_expand(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>SAVEDEV(EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>SELECTMULTI (ALL</td>
<td>ANY</td>
</tr>
<tr>
<td>SESSION_LIST(Yes</td>
<td>No[,DETail</td>
</tr>
<tr>
<td>SIZE(ALL</td>
<td>MOD1</td>
</tr>
<tr>
<td>SMS_PASS_volumes(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>SNAPSHOT_LIST(ALL</td>
<td>LINKED</td>
</tr>
<tr>
<td>SNAP_UNUSED_SPACE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>SOFTlink(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>SRDFA_CONSISTENT_RETRY(Yes</td>
<td>No</td>
</tr>
<tr>
<td>SRDFA_R1_target(Yes</td>
<td>No</td>
</tr>
<tr>
<td>SRDFA_R2_sync(WARNING</td>
<td>R1R2SYNC</td>
</tr>
</tbody>
</table>
### Table 12  Global parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Pagea</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRDFS_R1_target(Yes</td>
<td>No)DATAMOVERNaMe</td>
</tr>
<tr>
<td>STORageCLASs(classname)</td>
<td>201, 252</td>
</tr>
<tr>
<td>STORED_LOG_SIZE(size)</td>
<td>253</td>
</tr>
<tr>
<td>TDEV(EXCLude</td>
<td>INCLude)</td>
</tr>
<tr>
<td>TDEV_RECLAIM(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TERMINATE_SESSION_when_complete(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>THINPOOL(INCLUDE</td>
<td>EXCLUDE)</td>
</tr>
<tr>
<td>TIMEOUT(0</td>
<td>nnn)</td>
</tr>
<tr>
<td>TOLerate_REUSe_Failure(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLerateALLocationFailure(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLerate_COPY_Failure(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLERATE_DATACLASS_COMPACTION_MISMATCH(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLERATE_DATACLASS_EXTENDED_MISMATCH(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLerateENQFailure(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLerateTRUNCation(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TOLerateVSAMENQFailure(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>TYPRUN(NORUN</td>
<td>RUN</td>
</tr>
<tr>
<td>VALIDATE_RANGE({LOCAL</td>
<td>REMOTE}({AUTO</td>
</tr>
<tr>
<td>VARY_OFFline(AUTO</td>
<td>NEVER)</td>
</tr>
<tr>
<td>VARY_ONline(AUTO</td>
<td>Yes</td>
</tr>
<tr>
<td>VCLOSE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>VDEVICE(EXCLUDE</td>
<td>INCLUDE)</td>
</tr>
<tr>
<td>VDEVWAIT(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>VERIFY(Yes</td>
<td>No</td>
</tr>
<tr>
<td>VERIFY_OPEN_SOURCE(Yes</td>
<td>No)</td>
</tr>
<tr>
<td>VSAmenqMODE(SHARED</td>
<td>EXClusive</td>
</tr>
<tr>
<td>WAITFORCOMPLETION([Yes</td>
<td>No]hh:mm:ss[,MesSsaGes][,R1R2SYNC][,TIMEOUT(INFormational</td>
</tr>
<tr>
<td>WAIT_FOR_Definition(Yes</td>
<td>No)</td>
</tr>
</tbody>
</table>
Optional parameters

**ACTIVATE_SUBTASK# (nnn)**

See “ACTIVATE_SUBTASK#(nnn)” on page 153.

The ACTIVATE_SUBTASK# parameter has a matching site option.

**ADMINISTRATOR (Yes | No)**

See “ADMINISTRATOR(Yes|No)” on page 153.

This parameter sets a global value for the following commands:

- DFDSS (ADRDSSU)
- SNAP DATASET
- SNAP VOLUME

**ALLOCATE_UNUSED_SPACE (Yes | No)**

See “ALLOCATE_UNUSED_SPACE(Yes|No)” on page 153.

This parameter sets a global value for the following command:

- SNAP DATASET

**ALLOCATION_SEQUENCE (DATASET | NONE | SIZE)**

See “ALLOCATION_SEQUENCE(DATASET|NONE|SIZE)” on page 154.

This parameter sets a global value for the following command:

- SNAP DATASET

**ALLOW_FBA_META (Yes | No)**

Allows or prohibits execution of commands against FBA meta devices:

- Yes: FBA meta devices are allowed.
- No: 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.

AUTOMATIC_ACTIVATE(Yes|No)

The AUTOMATIC_ACTIVATE parameter allows or disallows automatic performance of an ACTIVATE when there are two or more SNAP VOLUME requests in the input stream and no ACTIVATE has been supplied by the user:

- **Yes** (Default) Multiple SNAP VOLUME requests are processed when there are two or more SNAP VOLUME commands in the input stream and no ACTIVATE was supplied.
- **No** Multiple SNAP VOLUME requests are not processed when there are two or more SNAP VOLUME commands in the input stream without the presence of an ACTIVATE command.

**Note:** The AUTOMATIC_ACTIVATE parameter cannot be used for group processing.

TimeFinder ignores AUTOMATIC_ACTIVATE for any SNAP VOLUME requests that specify a group name.

TimeFinder ignores AUTOMATIC_ACTIVATE for any SNAP VOLUME requests that specify a virtual device (VDEV).

The activate provided by AUTOMATIC_ACTIVATE is not consistent. For the activate to be consistent, you must either:

- Specify the consistent ACTIVATE command with the CONSISTENT parameter.
- Specify the GLOBAL command with the CONSISTENT parameter.

The AUTOMATIC_ACTIVATE parameter has a matching site option, &AUTOACTIVATE.

This parameter sets a global value for the following command:

- SNAP VOLUME

AUTOMATIC_CLEANup(Yes|No)

See “AUTOMATIC_CLEANup(Yes|No)” on page 154.

This parameter sets a global value for the following command:

- RESTORE VOLUME

AUTOmatic_CLEANUP_R2|CLEANUP_R2(Yes|No)

See “AUTOmatic_CLEANUP_R2(Yes|No)” on page 154.

This parameter sets a global value for the following command:

- CLEANUP

AUTOMATIC_DEALLOC(Yes|No)

See “AUTOMATIC_DEALLOC(Yes|No)” on page 154.

This parameter sets a global value for the following command:

- SNAP VOLUME

1. Available starting with Mainframe Enablers 8.2.
AUTOMATIC_RELEASE_hold(Yes | No)
   See “AUTOMATIC_RELEASE_hold(Yes|No)” on page 155.
   This parameter sets a global value for the following command:
   SNAP VOLUME

BACKGROUNDCOPY (Yes | No | NOCOPYRD | VSE)
   See “BACKGROUNDCOPY(Yes|No|NOCOPYRD|VSE)” on page 155.
   This parameter sets a global value for the following commands:
   - SNAP DATASET
   - SNAP VOLUME

BCVOnly(Yes | No)
   See “BCVOnly(Yes|No)” on page 156.
   This parameter sets a global value for the following command:
   - SNAP DATASET

BUILD_VTOCIX (Yes | No)
   See “BUILD_VTOCIX(Yes|No)” on page 156.
   This parameter sets a global value for the following command:
   - SNAP DATASET

CACHE_FULL_SYM(Yes | No)
   This parameter improves job completion times for jobs with high device counts.
   For Mainframe Enablers 8.1, 8.2, and 8.3, the default value is YES. For Mainframe
   Enablers 8.4 and later, the default value is NO.

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

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

   To obtain the latest device status/session information in each QUERY VOLUME
   command in a job step, use CACHE_FULL_SYM(NO).

   When CACHE_FULL_SYM(NO) is used, device status and session information is
   retrieved for each QUERY VOLUME command and is not cached. This ensures that
   the information is up-to-date in case device status (READY/NOT READY) or
   sessions were changed between two QUERY VOLUME requests within the same
   job step.

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

CATalog (Yes | No)
   See “CATalog(Yes|No)” on page 156.

---
1. Available starting with Mainframe Enablers 8.1.
This parameter sets a global value for the following command:

- SNAP DATASET

CHECKBCVHoldstatus (Yes | No)

See “CHECKBCVHoldstatus(Yes|No)” on page 157.

This parameter sets a global value for the following commands:

- SNAP VOLUME
- RESTORE VOLUME

CHECKONLINEPathstatus (Yes | No | NEVER)

See “CHECKONLINEPathstatus(Yes|No|NEVER)” on page 157.

This parameter sets a global value for the following commands:

- SNAP VOLUME
- STOP SNAP TO VOLUME
- RESTORE VOLUME

CHECK_POOL_usable (Yes | No)

See “CHECK_POOL_usable(Yes|No)” on page 157.

This parameter sets a global value for the following command:

- SNAP VOLUME

CKD (EXCLUDE | INCLUDE)

See “CKD(EXCLUDE|INCLUDE)” on page 157.

This parameter sets a global value for the following command:

- QUERY VOLUME

CLEANup_DIFFerential (Yes | No)

See “CLEANup_DIFFerential(Yes|No)” on page 158.

This parameter sets a global value for the following command:

- CLEANUP [EXTENT TRACK ON]

COLLAPSE_dataset_extents (VSAM | NONVSAM | VSAM, NONVSAM)

See “COLLAPSE_dataset_extents(VSAM|NONVSAM|VSAM,NONVSAM)” on page 158.

This parameter sets a global value for the following command:

- SNAP DATASET

CONDitionVOLUME (ALL | Label | DUMP)

See “CONDitionVOLUME(ALL|Label|DUMP)” on page 158.

This parameter sets a global value for the following commands:

- SNAP VOLUME
- RESTORE VOLUME
CONSISTENT(Yes|No)

See “CONSISTENT(Yes|No)” on page 159.

This parameter sets a global value for the following command:

- SNAP DATASET

COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASS] [STORageCLASS] [ALL])

See “COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASS] [STORageCLASS] [ALL])” on page 160.

This parameter sets a global value for the following command:

- SNAP DATASET

COPYVolid(Yes|No)

See “COPYVolid(Yes|No)” on page 161.

This parameter sets a global value for the following commands:

- SNAP VOLUME
- RESTORE VOLUME

DATACLASs(classname)

See “DATACLASs(classname)” on page 162.

This parameter sets a global value for the following command:

- SNAP DATASET

DaTaMoverNaMe(ADR|SSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)

See “DaTaMoverNaMe(ADR|SSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)” on page 162.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

DATASET_CHANged_indicator(SET|RESET|LEAVE)

See “DATASET_CHANged_indicator(SET|RESET|LEAVE)” on page 164.

This parameter sets a global value for the following commands:

- SNAP DATASET
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.

- **EXTRA**
  - Produce all possible TRACE and DEBUG information (more complete than ALL).

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

- **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.

- **SDUMP**
  - When an abend occurs, an SDUMP (SVC DUMP) is automatically taken.

**Note:** If you do not include DEBUG, no debugging is performed.

The DEBUG parameter with SDUMP has a matching site option, &DEBUG_SDUMP=YES|NO.

The DEBUG parameter with ERROR has a matching site option, &DEBUG_ERROR. Table 2 on page 47 lists the site options and their parameters.

DEBUG sets the type of debugging actions that are to be performed by default. You can control DEBUG and TRACE default actions on specific commands through the DEBUG(ON|OFF) and TRACE(ON|OFF) parameters.

For example, consider the three cases shown in Table 13.

**Table 13 Effect of GLOBAL DEBUG**

<table>
<thead>
<tr>
<th>Case</th>
<th>DEBUG parameter</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GLOBAL DEBUG(ALL)</td>
<td>Sets run to produce the TRACE and DEBUG information needed for most situations.</td>
</tr>
<tr>
<td></td>
<td>SNAP DATASET .... DEBUG(OFF)</td>
<td>For SNAP DATASET action, sets DEBUG(OFF) TRACE(ON).</td>
</tr>
<tr>
<td></td>
<td>SNAP VOLUME .... TRACE(OFF)</td>
<td>For SNAP VOLUME action, sets DEBUG(ON) TRACE(ON) TRACE is set to ON because DEBUG includes trace.</td>
</tr>
<tr>
<td>2</td>
<td>GLOBAL DEBUG(TRACE)</td>
<td>Sets run to produce normal TRACE output.</td>
</tr>
<tr>
<td></td>
<td>SNAP DATASET .... DEBUG(OFF)</td>
<td>For SNAP DATASET action, sets DEBUG(OFF) TRACE(ON).</td>
</tr>
<tr>
<td></td>
<td>SNAP VOLUME .... TRACE(OFF)</td>
<td>For SNAP VOLUME action, sets DEBUG(ON) TRACE(ON) TRACE is set to ON because DEBUG includes trace.</td>
</tr>
<tr>
<td>3</td>
<td>GLOBAL DEBUG(EXTRA)</td>
<td>Sets run to produce all possible TRACE and DEBUG information.</td>
</tr>
<tr>
<td></td>
<td>SNAP DATASET .... DEBUG(OFF)</td>
<td>For SNAP DATASET action, sets DEBUG(OFF) TRACE(ON) and produces extra information.</td>
</tr>
<tr>
<td></td>
<td>SNAP VOLUME .... TRACE(OFF)</td>
<td>For SNAP VOLUME action, sets DEBUG(ON) TRACE(ON) and produces extra information. TRACE is set to ON because DEBUG includes trace.</td>
</tr>
</tbody>
</table>

This parameter sets a global value for the following commands:
SNAP DATASET

SNAP VOLUME

RESTORE VOLUME

**Default value**

None

**DEBUG_EXTENTS (Yes | No)**

DEBUG_EXTENTS controls whether the EXTENTS program, when invoked, writes debug information to the console log:

- **Yes** Causes EXTENTS to write debug information to the console log on startup.
- **No** *(Default)* Prohibits EXTENTS from writing debug information to the console log on startup.

This parameter is not normally required. It should only be used when requested by Dell EMC.

This parameter sets a global value for the following command:

- EXTENTS program

**DFDSS_ADMIN (Yes | No)**

See “DFDSS_ADMIN (Yes | No)” on page 165.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

**DFDSS_CC (Yes | No)**

See “DFDSS_CC (Yes | No)” on page 165.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

**DFDSS_OPTimize (n)**

The DFDSS_OPTIMIZE parameter specifies the OPTIMIZE value to be used when DFDSS is specified as the DATAMOVERNAME.

Acceptable values for n are 1 through 4. The default value is 4.

The DFDSS_OPTIMIZE parameter has a matching site option, &DFDSS_OP.

This parameter sets a global value for the following command:

- DFDSS datamover program

**DIFFerential (Yes | No)**

See “DIFFerential (Yes | No)” on page 165.

This parameter sets a global value for the following command:

- SNAP VOLUME
DIFFERENTIAL_DATASET (Yes | No)
See “DIFFERENTIAL_DATASET(Yes|No)” on page 166.
This parameter sets a global value for the following command:
- SNAP DATASET

EATTR (NO | OPT)
See “EATTR(NO|OPT)” on page 166.
This parameter sets a global value for the following command:
- SNAP DATASET

EMUL_TYPE (ALL | HARDLINK | SNAPVX)
See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 166.
This parameter sets a global value for the following command:
- QUERY VOLUME

ENQSCOPE (REQUEST | STEP)
See “ENQSCOPE(REQuest|STEP)” on page 167.
This parameter sets a global value for the following command:
- SNAP DATASET

ENQWAIT (Yes | No)
See “ENQWAIT(Yes|No)” on page 167.
This parameter sets a global value for the following command:
- SNAP DATASET

ERROR_CHecking | ERRCHK (NORMal | REDUCED)
See “ERROR_CHecking(NORMal|REDUCED)” on page 167.
This parameter sets a global value for the following commands:
- SNAP DATASET
- SNAP VOLUME

ERRor_DISPosition (DELETE | KEEP)
See “ERRor_DISPosition(DELETE|KEEP)” on page 168.
This parameter sets a global value for the following command:
- SNAP DATASET

ERROR_RECovery (NORMal | ENHanced)
See “ERROR_RECovery(NORMal|ENHanced)” on page 168.
This parameter sets a global value for the following commands:
- SNAP DATASET
- SNAP VOLUME
ESNP119 (WARNING | ERROR)
Determines whether message ESNP119 is a warning or error message:

- ERROR Message is issued and processing stops.
- WARNING (Default) Message is issued as a warning and processing continues.

The following ESNP119 message involves a request for a consistent copy and has two different outcomes that can be set.

CONSISTENT COPY ATTEMPTED, BUT TIMEOUT OCCURRED OR UNSUPPORTED DEVICE, COPY NOT CONSISTENT

It can be a warning message, where the attempt that caused the message is identified and the processing is continued, or it can be set as an error condition where the processing is stopped.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

ESNP220 (ERROR | WARNING)
See “ESNP220(ERROR|WARNING)” on page 169.

This parameter sets a global value for the following command:

- SNAP DATASET

EXAMINE (Yes | No)
See “EXAMINE(Yes|No)” on page 169.

This parameter sets a global value for the following command:

- SNAP DATASET

EXclude_PathGroupID(pathlist)
See “EXclude_PathGroupID(pathlist)” on page 169.

This parameter sets a global value for the following commands:

- SNAP VOLUME
- RESTORE VOLUME

EXPlain(VOLUME_SELECTION(Yes | No))
See “EXPlain(VOLUME_SELECTION(Yes|No))” on page 170.

This parameter sets a global value for the following command:

- SNAP DATASET

EXTENT_ALLOCAtion(Yes[,CONSOLIDATE_VOLUME|,CONSOLIDATE_ALL]|No)
See “EXTENT_ALLOCAtion(Yes[,CONSOLIDATE_VOLUME|,CONSOLIDATE_ALL]|No)” on page 170.

This parameter sets a global value for the following command:

- SNAP DATASET
EXTALLOC_EMC_ONLY(Yes|No)

See “EXTALLOC_EMC_ONLY(Yes|No)” on page 171.

This parameter sets a global value for the following command:

- SNAP DATASET

EXTENT_EXPAND(Yes|No, [ADDNEW(Yes|No) [,SAMEVOL] [,NEWVOL]] )

See “EXTENT_EXPAND(Yes|No,[ADDNEW(Yes|No)][,SAMEVOL][,NEWVOL])” on page 171.

This parameter sets a global value for the following command:

- SNAP DATASET

FBA (EXCLUDE | INCLUDE)

See “FBA(EXCLUDE|INCLUDE)” on page 172.

This parameter sets a global value for the following command:

- QUERY VOLUME

FLASH_SNAP (FLASHCOPY | SNAP)

See “FLASH_SNAP(FLASHCOPY|SNAP)” on page 172.

This parameter sets a global value for the following commands:

- QUERY VOLUME
- SNAP DATASET

FORCE (Yes|No)

See “FORCE(Yes|No)” on page 172.

This parameter sets a global value for the following command:

- SNAP DATASET

FORCE_COMPLETION (Yes|No)

See “FORCE_COMPLETION(Yes|No)” on page 172.

This parameter sets a global value for the following command:

- CLEANUP

FREESPACE (Yes|No)

See “FREESPACE(Yes|No)” on page 173.

This parameter sets a global value for the following command:

- SNAP VOLUME

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. This must be a partitioned dataset or partitioned dataset extended and you must have read/write access to the dataset. Any group references are resolved into this dataset, along with the status information for actions executed for a group.
If this parameter omitted, then TimeFinder uses a DDNAME of EMCGROUP as the “working” group dataset.

GROUP_DSName is an alias of GROUP_DATset_name.

The GROUP_DATASET_NAME parameter has a matching site option, &GROUP_DSNAME.

This parameter sets a global value for the following command:

- SNAP VOLUME

Default value

None

GROUP_DEVICE_ready_state(AUTO|NEVER)

The GROUP_DEVICE_READY_STATE parameter controls the access, or readiness, to the devices on a channel. This determines when devices on the channel are available for an operation and when they are not:

AUTO (Default) Use the standard group processing option. Make the devices not ready on the channel during PRESNAP processing and ready on the channel when POSTSNAP is performed.

NEVER Do not change the readiness of the devices on the channel.

With Mainframe Enablers 8.1 and later, when issuing SNAP VOLUME by group with GROUP_DEVICE_READY_STATE(NEVER), the target devices are made ready/not ready depending on the value of the READY parameter. If the READY parameter is not specified explicitly, the device state is not changed.

The GROUP_DEVICE_READY_STATE parameter can be used in group processing only.

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

You can abbreviate the “GROUP” in the GROUP_DEVICE_READY_STATE parameter name as “GRP.”

This parameter sets a global value for the following command:

- SNAP VOLUME

Example

GRP_DEV(NEVER)

GROUP_EMCCAPI_VERIFY(Yes|No)

SNAP VOLUME processing ensures that the source and target volumes are completely suitable for use before requesting that the storage system establish a relationship between the two devices. The GROUP_EMCCAPI_VERIFY parameter allows you to avoid this additional overhead.

This can be a major benefit when a group is being executed on a periodic basis, and no other usage of the target devices occurs that might disrupt the device status.

The parameter values are:

Yes (Default) Avoid the additional suitability check. This eliminates some overhead, but raises the chance that the request fails.

No Do not avoid the additional suitability check.
The GROUP_EMCQCAPI_VERIFY parameter has a matching site option, &GROUP_EMCQCAPI_VERIFY.

This parameter sets a global value for the following command:

- SNAP VOLUME

**Example**

GROUP_EMCQCAPI_VERIFY(NO)

**HostcoPYMODE(SHaRed|EXClusive|NONE)**

See “HostcoPYMODE(SHaRed|EXClusive|NONE)” on page 174.

This parameter sets a global value for the following command:

- SNAP DATASET

**INVALIDATE_PDSEBuffers(Yes|No)**

See “INVALIDATE_PDSEBuffers(Yes|No)” on page 174.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

**HostcoPYMODE(SHaRed|EXClusive|NONE)**

See “HostcoPYMODE(SHaRed|EXClusive|NONE)” on page 174.

This parameter sets a global value for the following command:

- SNAP DATASET

**LIST([[NO]STatements][[NO]HIStory])**

See “LIST([[NO]STatements][[NO]HIStory])” on page 175.

This parameter sets a global value for the following command:

- QUERY GROUP

**LOGINDYNAM(volume[, volume...])**

See “LOGINDYNAM(volume[, volume...])” on page 176.

This parameter sets a global value for the following command:

- SNAP DATASET

**ManaGeMenTCLASs(classname)**

See “ManaGeMenTCLASs(classname)” on page 176.

This parameter sets a global value for the following command:

- SNAP DATASET

**MAXIMUM_ADRDSSU_address_spaces(number)**

TimeFinder automatically uses multiple address spaces when you request multitasking and TimeFinder encounters SNAP VOLUME requests. The MAXIMUM_ADRDSSU_ADDRESS_SPACES parameter determines the limit on address spaces that ADRDSSU (ADRXMAIA) spawns when used as a datamover.
number

Specifies the limit on address spaces. The minimum value you can specify is one (1) and the normal maximum value you can specify is 15.

**Note:** Contact Dell EMC if you want to use a larger maximum value than 15.

MAXDSSU is an alias of MAXIMUM_ADRDSSU_ADDRESS_SPACES.

The MAXIMUM_ADRDSSU_ADDRESS_SPACES parameter has a matching site option, &MAXDSSU.

This parameter sets a global value for the following command:

- ADRDSSU (ADRMAIA) - this is a datamover program

**Default value**

10

**Example**

MAXDSSU(5)

MAXIMUM_SUBTASKS(number1, number2)

The MAXIMUM_SUBTASKS parameter establishes an absolute maximum number of subtasks that can be attached and used. TimeFinder automatically limits the number of subtasks based on the requests specified and the low and high region available. TimeFinder never exceeds the limits specified in this parameter.

**number1**

The limit to the number of individual requests that can be processed simultaneously.

The minimum value you can specify is two (2). The maximum value you can specify is 9999.

**number2**

The limit to the number of individual activities that can be performed within a single request, typically as the result of wildcarding.

The minimum value you can specify is two (2). The maximum value you can specify is 9999.

MAXTASKs is an alias of MAXIMUM_SUBTASKS.

MAXIMUM_SUBTASKS (number1...) has a matching site option, MAXTASK2.

MAXIMUM_SUBTASKS (number2...) has a matching site option, MAXTASKR.

**Default value**

99 (number1)
999 (number2)

**Example**

MAXTASK(10,10)

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

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

```
return_code_value
```

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

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

This parameter sets a global value for the following:

- All TimeFinder commands

**Default value**

4

**Example**

Here are a few examples:

Message ESNPxxxI - severity is 0

If MAXRC(4), then additional commands is executed (0 is not greater than 4).

Message ESNPxxxW - severity is 4

If MAXRC(4), then additional commands is executed (4 is not greater than 4).

Message ESNPxxxE - severity is 8

If MAXRC(4), then additional commands is NOT executed (8 is greater than 4).

**MESSages(DISplay|PROmpt|NONE|DETAIL)**

See “MESSages(DISplay|PROmpt|NONE|DETAIL)” on page 176.

This parameter sets a global value for the following command:

- ACTIVATE

**MIGrate([PURge(Yes|No)] [RECall(Yes|No)])**

See “MIGrate([PURge(Yes|No)] [RECall(Yes|No)])” on page 177.

This parameter sets a global value for the following command:

- SNAP DATASET
MODECOPYFINISH

MODECOPYFINISH resolves outstanding indirected tracks. MODECOPYFINISH applies to both SNAP DATASET and SNAP VOLUME and is an alias of the PREPARE_FOR_SNAP parameter.

By default, MODECOPYFINISH is not used. If you do not want to use MODECOPYFINISH, do not specify the parameter.

If you want MODECOPYFINISH, code the GLOBAL command as follows:

GLOBAL MAXRC(0) MODECOPYFINISH

**Note:** The MODECOPYFINISH parameter applies only to locally addressable volumes. MODECOPYFINISH is ignored if specified on actions with the SYMDV#, LOCAL, or REMOTE parameters.

This parameter sets a global value for the following commands:
- SNAP DATASET
- SNAP VOLUME

**MODE(COPY|NOCOPY|NOCOPYRD)**

See “MODE(COPY|NOCOPY|NOCOPYRD|VSE)” on page 178.

This parameter sets a global value for the following commands:
- CONFIG
- SNAP DATASET
- SNAP VOLUME

**MULTI_LINE_query(Yes|No)**

See “MULTI_LINE_query(Yes|No)” on page 181.

This parameter sets a global value for the following command:
- QUERY VOLUME

**MULTI_VIRTual(Yes|No)**

See “MULTI_VIRTual(Yes|No)” on page 182.

This parameter sets a global value for the following commands:
- SNAP VOLUME

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

See “NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])” on page 182.

**NOTIFYwhencomplete([GROUP(name)][DATASET|JOB|STEP|SNAP])**

See “NOTIFYwhencomplete([GROUP(name)][DATASET|JOB|STEP|SNAP])” on page 183.

This parameter sets a global value for the following commands:
- SNAP DATASET
- SNAP VOLUME
- **RESTORE VOLUME**

**NOTREADY (EXCLUDE | INCLUDE)**

See “NOTREADY(EXCLUDE|INCLUDE)” on page 185.

This parameter sets a global value for the following command:

- **QUERY VOLUME**

**PARALLEL(Yes | No)**

The PARALLEL parameter enables or disables multitasking:

- *Yes* Enables multitasking.
- *No* (Default) Disables multitasking.

PAR is an alias of PARALLEL.

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

This parameter sets a global value to enable or disable multitasking.

**Example**

```
PARALLEL(YES)
```

**PARALLEL_CLONE(Yes | No | PREFERred | REQuired)**

See “PARALLEL_CLONE(Yes|No|PREFERred|REQuired)” on page 185.

**PERSISTent (Yes | No)**

See “PERSISTent(Yes|No)” on page 186.

This parameter sets a global value for the following command:

- **RESTORE VOLUME**

**POOL(poolname)**

See “POOL(poolname)” on page 186.

This parameter sets a global value for the following command:

- **SNAP VOLUME**

**PRECOPY (Yes | No)**

See “PRECOPY(Yes|No)” on page 187.

This parameter sets a global value for the following command:

- **SNAP VOLUME**

**PREPARE_FOR_SNAP (Yes | No)**

Volume or dataset snaps require that you perform validations and setup work before you issue a SNAP VOLUME or SNAP DATASET command. For example, for volume snaps, any tracks remaining to be copied from a prior snap must be completed. For dataset snaps, the REUSE parameter validates the targets of the snap.

In many cases, this setup work represents a significant portion of the total elapsed time of the snap job. The actual SNAP VOLUME and SNAP DATASET commands are very fast.
The PREPARE_FOR_SNAP parameter separates some of the preparatory work from the actual snap. Running a PowerMax/VMAX snap job with PREPARE_FOR_SNAP at a noncritical time and then running the same snap job without PREPARE_FOR_SNAP in the critical batch path of the workload may provide reductions in the elapsed time of the second execution of snap and positively affect the critical batch path.

The first execution performs some of the validation and setup work but not the actual snap. The second execution performs the final validation and setup work, and then issues the snap.

The PREPARE_FOR_SNAP parameter allows you to bypass the actual snap action and instead, perform and report on all validation, resolve outstanding indirected tracks, and other preparatory work.

**Note:** The target dataset must already exist for this parameter to be used.

PREPARE_FOR_SNAP is *not designed* to work for a new snap, volume or dataset. It is designed so that if a situation occurs where a previous snap is not completed, all the relationships and snap status can be completed without initiating any new work, allowing the next snap operation to proceed without waiting for the previous snap to complete.

The PREPARE_FOR_SNAP parameter applies to both SNAP DATASET and SNAP VOLUME. Possible values are:

- **Yes** Bypass the actual snap action.
- **No** (*Default*) Do not bypass the actual snap action.

The PREPARE_FOR_SNAP parameter has a matching site option, &PREPARE.

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

MODECOPYFINISH is an alias for this parameter.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

**R1FULLCOPYonly** *(Yes|No)*

See “**R1FULLCOPYonly** *(Yes|No)*” on page 188.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

**RAID** *(ALL|NONE|RAIDS|RAID1|RAID5|RAID6|RAID10|FTS)*

See “**RAID** *(ALL|NONE|RAIDS|RAID1|RAID5|RAID6|RAID10|FTS)*” on page 188.

This parameter sets a global value for the following command:

- QUERY VOLUME
ReaDY (EXCLUDE | INCLUDE)

See “ReaDY(EXCLUDE|INCLUDE)” on page 189.

This parameter sets a global value for the following command:

- QUERY VOLUME

RECALCULATE_FREESPACE (Yes | No)

See “RECALCULATE_FREESPACE(Yes|No)” on page 189.

This parameter sets a global value for the following command:

- SNAP DATASET

RENAMEUnconditional(pfx) | RENAMEUnconditional((pfx)(oldnamemask,newnamemask)...)

See “RENAMEUnconditional(pfx) | RENAMEUnconditional((pfx)(oldnamemask,newnamemask)...)” on page 192.

REFVTOC (Yes | No)

See “REFVTOC(Yes|No)” on page 190.

This parameter sets a global value for the following command:

- QUERY VOLUME

REMOVE_REMOTE_extent_sessions (Yes | No)

See “REMOVE_REMOTE_extent_sessions(Yes|No)” on page 191.

This parameter sets a global value for the following command:

- CLEANUP

REPLace (Yes | No)

See “REPLace(Yes|No)” on page 192. The REPLACE parameter establishes the REPLACE value for all operations.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

RESERVE (Yes | No)

If RESERVE(YES) is specified, then the source and target volumes are enqueued and reserved so that VTOC changes cannot take place during the validate function.

If RESERVE(NO) is specified, then the source and target volumes is not enqueued and it is possible for VTOC changes to take place, causing a missed compare during validation.

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

This parameter sets a global value for the following commands:

- SNAP VOLUME
Default value

YES

REUSE (Yes | No)
See “REUSE(Yes|No[,WAIT])” on page 194.
This parameter sets a global value for the following command:
- SNAP DATASET

REUSE_AUTO_expand (Yes | No)
See “REUSE_AUTO_expand(Yes|No)” on page 194.
This parameter sets a global value for the following command:
- SNAP DATASET

SAVEDEV (EXCLUDE | INCLUDE)
See “SAVEDEV(EXCLUDE|INCLUDE)” on page 194.
This parameter sets a global value for the following command:
- QUERY VOLUME

SELECTMULTI (ALL | ANY | FIRST)
See “SELECTMULTI(ALL|ANY|FIRST)” on page 195.
This parameter sets a global value for the following command:
- SNAP DATASET

SESSION_LIST (Yes | No [, DETail | NODEtail | DIFFerential])
See “SESSION_LIST(Yes|No[,DETail|NODEtail,DIFFerential])” on page 195.
This parameter sets a global value for the following command:
- QUERY VOLUME

SIZE (ALL | MOD1 | MOD2 | MOD3 | MOD9 | MOD27 | MOD54 | EAV | # | low-high)
See “SIZE(ALL|MOD1|MOD2|MOD3|MOD9|MOD27|MOD54|EAV|#|low-high)” on page 195.
This parameter sets a global value for the following command:
- QUERY VOLUME

SMS_PASS_volumes (Yes | No)
See “SMS_PASS_volumes(Yes|No)” on page 196.
This parameter sets a global value for the following command:
- SNAP DATASET

SNAPSHOT_LIST (ALL | LINKED | NOT_LINKED | SNAPSHOT)
See “SNAPSHOT_LIST(ALL|LINKED|NOT_LINKED|SNAPSHOT)” on page 196.
This parameter sets a global value for the following command:
- QUERY VOLUME
SNAP_UNUSED_SPACE(Yes|No)

See “SNAP_UNUSED_SPACE(Yes|No)” on page 197.

This parameter sets a global value for the following command:

- SNAP DATASET

SOFTlink(Yes|No)

See “SOFTlink(Yes|No)” on page 197.

SRDFA_CONSISTENT_RETRY(Yes|No|nn)

See “SRDFA_CONSISTENT_RETRY(Yes|No|nn)” on page 199.

This parameter sets a global value for the following command:

- ACTIVATE

  Default value

  10 (retry attempts)

SRDFA_R1_target(Yes|No|DATAMOVERName|Physical|Informational)

See “SRDFA_R1_target(Yes|No|DATAMOVERName|Physical|Informational)” on page 199.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

SRDFA_R2_sync(WARNING|R1R2SYNC|DATAMOVER)

See “SRDFA_R2_sync(WARNING|R1R2SYNC|DATAMOVER)” on page 200.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

SRDFS_R1_target(Yes|No|DATAMOVERName|Physical|Informational)

See “SRDFS_R1_target(Yes|No|DATAMOVERName|Physical|Informational)” on page 201.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

STORAGECLASSES(classname)

See “STORAGECLASSES(classname)” on page 201.

This parameter sets a global value for the following command:

- SNAP DATASET
STORED_LOG_SIZE(size)

The STORED_LOG_SIZE parameter is used with GLOBAL_DEBUG(ERROR) and
GLOBAL_DEBUG(EMCQCAPI_INLINE_TRACE) to set the number of output
debug lines that are stored in memory. If an error occurs, the stored debug lines are
written to the output listing file.

size

Specifies the number of output debug lines to be stored. The value can be an
integer from 0 (zero) to a very large number, over a billion.

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

This parameter sets a global value for the following command:

- GLOBAL (only)

Default value

None

Example

GLOBAL DEBUG(ERROR,EMCQCAPI_INLINE_TRACE)
GLOBAL STORED_LOG_SIZE(25000)

TDEV(EXClude|INClude)

See “TDEV(EXClude|INClude)” on page 203.

This parameter sets a global value for the following command:

- QUERY VOLUME

TDEV_RECLAIM(Yes|No)

See “TDEV_RECLAIM(Yes|No)” on page 203.

This parameter sets a global value for the following command:

- SNAP VOLUME

TERMINATE_SESSION_when_complete(Yes|No)

See “TERMINATE_SESSION_when_complete(Yes|No)” on page 203.

This parameter sets a global value for the following command:

- SNAP VOLUME

THINPOOL(INCLUDE|EXCLUDE)

See “THINPOOL(EXCLUDE|INCLUDE)” on page 204.

This parameter sets a global value for the following command:

- QUERY VOLUME

TIMEOUT(nnn|0)

See “TIMEOUT(nnn)” on page 203.

This parameter sets a global value for the following command:

- ACTIVATE
TOLERATE_REUSE_FAILURE(Yes|No)
See “TOLERATE_REUSE_FAILURE(Yes|No)” on page 204.
This parameter sets a global value for the following command:
- SNAP DATASET

TOLERATE_ALLOCATION_FAILURE(Yes|No)
See “TOLERATE_ALLOCATION_FAILURE(Yes|No)” on page 205.
This parameter sets a global value for the following command:
- SNAP DATASET

TOLERATE_COPY_FAILURE(Yes|No)
See “TOLERATE_COPY_FAILURE(Yes|No)” on page 205.
This parameter sets a global value for the following command:
- SNAP DATASET

TOLERATE_DATACLASS_COMPACTION_MISMATCH(Yes|No)
The default value of NO checks to ensure that the source dataset compaction type matches the target data class compaction type. If they do not match, an error occurs.
If the value is changed to YES, then the check is not made and it is possible to copy a non-compact dataset to a compact data class, or a compact dataset to a non-compact data class.
This parameter sets a global value for the following command:
- SNAP DATASET

Default value
No

TOLERATE_DATACLASS_EXTENDED_MISMATCH(Yes|No)
The default value of NO checks to ensure that the source dataset extended type matches the target data class extended type. If they do not match, an error occurs.
If the value is changed to YES, the check is not made and it is possible to copy a non-extended dataset to a extended data class, or an extended dataset to a non-extended data class.
This parameter sets a global value for the following command:
- SNAP DATASET

Default value
No

TOLERATE_ENQ_FAILURE(Yes|No)
See “TOLERATE_ENQ_FAILURE(Yes|No)” on page 205.
This parameter sets a global value for the following command:
- RESTORE VOLUME
- SNAP DATASET
- SNAP VOLUME

TOLERATE_TRUNCATION(Yes|No)

See “TOLERATE_TRUNCATION(Yes|No)” on page 206.

This parameter sets a global value for the following command:

- SNAP DATASET

TOLERATE_VSAM_ENQ_FAILURE(Yes|No)

See “TOLERATE_VSAM_ENQ_FAILURE(Yes|No)” on page 206.

This parameter sets a global value for the following command:

- SNAP DATASET

TYPRUN (NORUN | RUN | SCAN)

The TYPRUN parameter determines the type of command processing. Possible values are:

- NORUN Specifies parsing all commands and identifying the datasets that are going to be processed, but not actually doing any work.
- RUN (Default) Specifies fully processing all commands.
- SCAN Specifies stopping command processing after all commands have been parsed. In other words, performing syntax checking and then stopping.

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

Default value

RUN

VALIDATE_RANGE({LOCAL|REMOTE}({AUTO|IGNORE}))

In normal SNAP VOLUME processing, each request is handled as independent and unrelated to any other requests. This means that each SNAP VOLUME request is processed as if there were no other statements present in the input stream. Each device (source or target) is validated individually and perhaps repeatedly (for example, a source volume may be used in multiple requests).

Depending on the keyword you use, the VALIDATE_RANGE parameters can change the validation processing.

VALIDATE_RANGE(LOCAL... can change validation processing for local (source and target) devices.

VALIDATE_RANGE(REMOTE... can change validation processing for remote (source and target) devices.
Parameter values can be:

**AUTO** *(Default)* Changes device validation processing. Information about the device that is obtained from the storage system includes information about many devices instead of a single device. This additional information is cached and is used if additional requests refer to any of these devices. This processing is automatically chosen for three situations:
- SYMDV# is specified and uses a range.
- UNIT is specified and uses a range.
- A group is processed.

**LOCAL** Changes device validation. Information about local devices is obtained from the local storage system. This additional information is cached and is used if additional requests refer to any of these devices. This processing is automatically chosen for three situations:
- SYMDV# is specified and uses a range.
- UNIT is specified and uses a range.
- A group is processed.

**REMOTE** Changes device validation. Information about remote devices is obtained from the remote storage system. This additional information is cached and is used if additional requests refer to any of these devices. This processing is automatically chosen for three situations:
- SYMDV# is specified and uses a range.
- UNIT is specified and uses a range.
- A group is processed.

**IGNORE** Does not change device validation.

VALIDATE_RANGE(LOCAL...) and VALIDATE_RANGE(REMOTE...) have matching site options:
- &VALRANGE_LOCAL
- &VALRANGE_REMOTE

This parameter sets a global value for the following command:
- SNAP VOLUME

**VARY_OFFline**(AUTO|NEVER)
See “VARY_OFFline(AUTO|NEVER)” on page 207.

This parameter sets a global value for the following commands:
- RESTORE VOLUME
- SNAP VOLUME

**VARY_ONline**(AUTO|Yes|No)
See “VARY_ONline(AUTO|Yes|No)” on page 208.

This parameter sets a global value for the following commands:
- RESTORE VOLUME
- SNAP VOLUME

**VCLOSE**(Yes|No)
See “VCLOSE(Yes|No)” on page 208.
This parameter sets a global value for the following commands:

- RESTORE VOLUME
- SNAP VOLUME

VDEVice(EXCLUDE|INCLUDE)

See “VDEVice(EXCLUDE|INCLUDE)” on page 209.

This parameter sets a global value for the following commands:

- ACTIVATE
- QUERY VOLUME

VDEVWAIT(Yes|No)

See “VDEVWAIT(Yes|No)” on page 209.

This parameter sets a global value for the following command:

- SNAP VOLUME

VERIFY(Yes|No|NEVER)

See “VERIFY(Yes|No|NEVER)” on page 209.

This parameter sets a global value for the following command:

- SNAP DATASET

VERIFY_OPEN_SOURCE(Yes|No)

See “VERIFY_OPEN_SOURCE(Yes|No)” on page 210.

This parameter sets a global value for the following command:

- SNAP DATASET

VSaMENQMODE(SHAREd|EXClusive|NONE)

See “VSaMENQMODE(SHAREd|EXClusive|NONE)” on page 210.

This parameter sets a global value for the following command:

- SNAP DATASET

WAITFORCOMPLETION([Yes|No|hh:mm:ss]
[,MeSsaGes][,R1R2SYNC]
[TIMEOUT(INFormational|WARNing|ERRor)])

See “WAITFORCOMPLETION([Yes|No|hh:mm:ss]
[,MeSsaGes][,R1R2SYNC]
[TIMEOUT(INFormational|WARNing|ERRor)])” on page 211.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME
- RESTORE VOLUME

WAIT_FOR_Definition(Yes|No)

See “WAIT_FOR_Definition(Yes|No)” on page 212.
WAIT_FOR_PRECOPY_PASS1(Yes | No)

See “WAIT_FOR_PRECOPY_PASS1(Yes|No)” on page 212.

This parameter sets a global value for the following commands:

- ACTIVATE
- SNAP VOLUME

WAITforsession(Yes | No | hh:mm:ss)

See “WAITforsession(Yes|No|hh:mm:ss)” on page 212.

This parameter sets a global value for the following commands:

- SNAP DATASET
- SNAP VOLUME

WHEN_SAVEDEV_FULL(READY | NOTREADY)

See “WHEN_SAVEDEV_FULL(READY|NOTREADY)” on page 213.

This parameter sets a global value for the following command:

- SNAP VOLUME
QUERY DATASET (TF/Clone)

The QUERY DATASET command returns dataset status information.

Syntax

```
QUERY DataSet
(
SOURCE(dataset) | INDDname(ddname)
[optional_parameters]
)
```

Where `optional_parameters` are as follows:

```
[SOURCE_VOLUME_LIST(vollist)]
```

Required parameters

```
INDDname(ddname)
```

The INDDname parameter specifies a DD statement already allocated to the source dataset.

```
ddname
```

The DD statement.

Default value

None

Example

```
INDD(DD1)
```

```
SOURCE(dataset)
```

The SOURCE parameter specifies the dataset name for which you want status information.

```
dataset
```

The name of the dataset. The dataset name can be masked or wildcarded.

Default value

None

Example

```
SOURCE(MY.DATASET)
```

Optional parameters

```
SOURCE_VOLUME_LIST(vollist)
```

See “SOURCE_VOLUME_LIST(vollist)” on page 198.
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

The following example is output from a QUERY GLOBAL command:

```
ESNPW20I --- EMCSNAPO --- VER n.n.n --- SIZE 494 --- DATE/TIME mn/nn/nn 14.53 ---
ESNPW21I SITE SETTING GLOBAL OVERRIDE
ESNPW22I ADMINISTRATOR N N
ESNPW22I ALLOCATE_UNUSED_SPACE Y Y
ESNPW22I ALLOCATION_SEQUENCE D D
ESNPW22I ALLOCATION_UNITNAME SYSALLDA -N/A-
ESNPW22I ALLOW SYMDV# Y -N/A-
ESNPW22I AUTOMATIC_CLEANUP Y Y
ESNPW22I AUTOMATIC_DEALLOC Y Y
ESNPW22I AUTOMATIC_RELEASE N N
ESNPW22I BCVONLY N N
ESNPW22I BUILD_VTOCIX N N
```

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 175.

For the duration of the current QUERY GROUP command, the value of LIST
overrides any value set by the GLOBAL command LIST parameter.
**QUERY VDEVICE (TF/Snap)**

The QUERY VDEVICE command returns information about the status of virtual devices in one or more storage system(s). The QUERY VDEVICE command without any parameters returns information about all the storage systems addressable by the host.

When issued against a storage system with PowerMaxOS 5978 or HYPERMAX OS 5977, QUERY VDEVICE returns all thin devices. This causes the query to run longer and produce more output, as compared to Enginuity 5876 and 5773.

---

**Note:** This command is only available if you purchase the TF/Snap Licensed Feature Code.

---

**Syntax**

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

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

---

**Optional parameters**

1. `CONTROLLER([xxxxxxx-]xxxxx|name)`
   - See “`CONTROLLER([xxxxxxx-]xxxxx|name)" on page 159.`

2. `LOCAL({
UNIT(cuu) [CONTROLER([xxxxxxx-]xxxxx|name])
VOLUME(volser) [CONTROLER([xxxxxxx-]xxxxx|name])]
DDNAME(ddname) [CONTROLER([xxxxxxx-]xxxxx|name])
CONTROLER([xxxxxxx-]xxxxx|name)
})`
   - See “`LOCAL([ UNIT(cuu) [CONTROLER([xxxxxxx-]xxxxx|name)])
VOLUME(volser) [CONTROLER([xxxxxxx-]xxxxx|name])]
DDNAME(ddname) [CONTROLER([xxxxxxx-]xxxxx|name)]) CONTROLER([xxxxxxx-]xxxxx|name)])" on page 175.

3. `REMOTE (RAGROUP(nn.nn.nn.nn)
{UNIT(cuu) [CONTROLER([xxxxxxx-]xxxxx|name])]
VOLUME (volser) [CONTROLER([xxxxxxx-]xxxxx|name])]
DDNAME(ddname) [CONTROLER([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 190.
**QUERY VOLUME**

The QUERY VOLUME command shows the status of devices in one or more storage system(s) using various filters.

**Syntax**

For Mainframe Enablers 8.4, 8.3, and 8.2:

```
QUERY VOLUME
  (UNIT(cuu) | VOLUME(volser) |
   | CONTROLLER([xxxxxxx-]xxxxx|name)
   | GROUP(grpname[, grpname,...])
  [optional_parameters])
```

For Mainframe Enablers 8.0 and 8.1:

```
QUERY VOLUME
  (UNIT(cuu) | VOLUME(volser) | SCFGROUP(gnsgrp) |
   | CONTROLLER([xxxxxxx-]xxxxx|name)
   | GROUP(grpname[, grpname,...])
  [optional_parameters])
```

Where **optional_parameters** are as follows:

- [CKD (EXCLUDE | INCLUDE) ]
- [CcUU (ccuu | low-high | low:high | ccuu(count))]
- [DEVice (symdv# | low-high | low:high | symdv#(count) | ALL)]
- [DISPLAY_CUU (Yes | No)]
- [EMUL_TYPE (ALL | HARDLINK | SNAPVX)]
- [FBA (EXCLUDE | INCLUDE)]
- [FLASH_SNAP (FLASHCOPY | SNAP)]
- [MULTI_LINE_query (Yes | No)]
- [NotReaDY (EXCLUDE | INCLUDE)]
- [RAID (ALL | NONE | RAIDS | RAID1 | RAID5 | RAID6 | RAID10 | FTS)]
- [ReaDY (EXCLUDE | INCLUDE)]

- [LOCAL (UNIT (cuu) | CONTROLLER([xxxxxxx-]xxxxx|name)]
- [VOLUME (volser) | CONTROLLER([xxxxxxx-]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)])

SAVEDEV(EXCLUDE|INCLUDE]
SESSION_LIST(Yes|No[,DETAIL|,NODETAIL|,DIFFERENTIAL])]
SIZE(ALL|MOD1|MOD2|MOD3|MOD9|MOD27|MOD54|EAV|#|low-high])
SNAPSHOT_LIST(ALL|LINKED|NOT_LINKED|SNAPSHOT)
TDEV(EXCLUDE|INCLUDE]}
THINPOOL(EXCLUDE|INCLUDE]}
VDEVice(EXCLUDE|INCLUDE]}

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

**Note:** 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.

**Note:** If the CcUU or DEVice parameters are not set, information is only returned for the one VOLUME, UNIT, or CONTROLLER specified.

**Required parameters**

**CONTROLER([xxxxxxxx-]xxxxx|name)**
Specifies the storage system to be queried.

See “CONTROLER([xxxxxxxx-]xxxxx|name)” on page 159.

**GROUP(grpname[,grpname,...])**
This parameter allows you to display all the devices in a group together without having to build the query manually.

See “GROUP(grpname[,grpname,...])” on page 173.

Each storage system you reference has just the devices in the specified group displayed. Both the source and target devices are listed.

If you use the GROUP parameter, then you should not use the UNIT, SCFGROUP, CONTROLLER, LOCAL, REMOTE, CcUU, or DEVICE parameters.

If you use the GROUP parameter, you can use the CKD, FBA, SAVEDEV, VDEV, READY, NOTREADY, RAID, SESSION_LIST, and SIZE parameters to tailor the results.
SCFGROUP(gnsgrp)

SCFGROUP(gnsgrp) identifies the storage systems to be listed for QUERY VOLUME.

Starting with Mainframe Enablers 8.3 PTF SN83091, the QUERY VOLUME command with the SCFGROUP parameter displays all devices on each storage system that were added to the GNS group specified on the SCFGROUP parameter.

Starting with Mainframe Enablers 8.3 PTF SN83091, the SCFGROUP parameter cannot be used together with the following keywords:

- UNIT(cuu)
- Volume(volser)
- CONTROLLER([xxxxxxx-]xxxxx|name)
- GROUP(grpname[, grpname, ...])
- CcUu(ccuu|low-high|low:high|ccuu(count))
- Device(symdv#|low-high|low:high|symdv#(count)|ALL)
- 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)])

- UNIT(cuu)
  UNIT(cuu) specifies the unit address of the volume to be queried.

- Example
  UNIT(A099)

1. SCFGROUP is a required parameter with Mainframe Enablers 8.4, 8.3, and 8.2. However, it is optional with Mainframe Enablers 8.0 and 8.1.

Note: The ResourcePak Base for z/OS Product Guide provides more information.
VOLUME(volser)

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

If the volser contains a hyphen, enclose it in single quotes, for example:

VOL('vol-ser')

Default value
None

Example
VOL(VOL000)

Optional parameters

CCUU(ccuu|low-high|low:high|ccuu(count))

The CCUU parameter specifies using the z/OS CCUU to define the devices to be queried. This limits the QUERY VOLUME output to those devices that match the CCUU specification. If you omit this parameter, then TimeFinder does not check the CCUU value and may show all devices.

You can specify a single CCUU:

CUU(ccuu)

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

- Specify the lowest CCUU in the range and the highest CCUU in the range separated by a dash:
  
  CUU(low-high)

- Specify the lowest CCUU in the range and the highest CCUU in the range separated by a colon:
  
  CUU(low:high)

- Specify the starting CCUU in the range and a count value (in parentheses) that indicates how many additional devices there are between that CCUU and the end of the range. The count value includes the lowest and the highest CCUUs.
  
  The total number of CCUUs in the range (that is, the count value) cannot exceed 256.

  CUU(ccuu(count))

CKD(EXCLUDE|INCLUDE)

See “CKD(EXCLUDE|INCLUDE)” on page 157.

For the duration of the current QUERY VOLUME command, the value of CKD overrides any value set by the GLOBAL command CKD parameter or by the &OPT_CKD site option.

1. With Mainframe Enablers 8.1 and later.
DEVice(symdv# | low-high | low:high | symdv#(count) | ALL)

The DEVICE parameter defines the devices to query by using the internal PowerMax/VMAX device numbers. The resulting output includes only the devices specified.

You can specify a single PowerMax/VMAX device number:

DEVice(symdv#)

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

- Specify the lowest device in the range and the highest device in the range separated by a dash:
  DEVice(low-high)

- Specify the lowest device in the range and the highest device in the range separated by a colon:
  DEVice(low:high)

- 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.

  The total number of devices in the range (that is, the count value) cannot exceed 256.

  DEVice(symdv#(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 view all devices.

**Default value**

None

DISPLAY_CUU(Yes | No)

Determines whether to include CUU information in QUERY VOLUME output:

- Yes Display CUUs for all devices in the query range
- No (Default) Do not display CUUs

The query command can take a long time to complete when querying a large range of devices. Much of this time is spent obtaining CUU information for each device in the query range. DISPLAY_CUU(NO) bypasses this process and speeds up the command significantly.
The DISPLAY_CUU parameter is automatically set to YES when the QUERY VOLUME command is issued by CUU.

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

See “EMUL_TYPE(ALL|HARDLINK|SNAPVX)” on page 166.

For the duration of the current QUERY VOLUME command, the value of EMUL_TYPE overrides any value set by the GLOBAL command EMUL_TYPE parameter or by the &EMUL_TYPE site option.

**FBA(EXCLUDE|INCLUDE)**

See “FBA(EXCLUDE|INCLUDE)” on page 172.

For the duration of the current QUERY VOLUME command, the value of FBA overrides any value set by the GLOBAL command FBA parameter or by the &OPT_FBA site option.

**FLASH_SNAP(FLASHCOPY|SNAP)**

See “FLASH_SNAP(FLASHCOPY|SNAP)” on page 172.

For the duration of the current QUERY VOLUME command, the value of FLASH_SNAP overrides any value set by the GLOBAL command FLASH_SNAP parameter or by the &FLASH_SNAP site option.

**IMPORTANT**

Do not change the value of this parameter unless directed to do so by Dell EMC.

**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 175.

**MULTI_LINE_query(Yes|No)**

See “MULTI_LINE_query(Yes|No)” on page 181.

**NotReaDY(EXCLUDE|INCLUDE)**

See “NOTREADY.EXCLUDE|INCLUDE)” on page 185.

For the duration of the current QUERY VOLUME command, the value of NOTREADY overrides any value set by the GLOBAL command NOTREADY parameter or by the &OPT_NOTREADY site option.

**RAID(ALL|NONE|RAIDS|RAID1|RAID5|RAID6|RAID10|FTS)**

See “RAID(ALL|NONE|RAIDS|RAID1|RAID5|RAID6|RAID10|FTS)” on page 188.

For the duration of the current QUERY VOLUME command, the value of RAID overrides any value set by the GLOBAL command RAID parameter or by the &OPT_RAID site option.
READY (EXCLUDE | INCLUDE)
  See “READY(EXCLUDE|INCLUDE)” on page 189.
  For the duration of the current QUERY VOLUME command, the value of READY overrides any value set by the GLOBAL command READY parameter or by the &OPT_READY site option.

REMOTE (RAGROUP(nn.nn.nn.nn)
  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 190.

Example
  QUERY VOLUME (-
    REMOTE (VOL(U6A230) RAGROUP(06) CONTROLLER(0001879-90132) ) )

SAVEDEV (EXCLUDE | INCLUDE)
  See “SAVEDEV(EXCLUDE|INCLUDE)” on page 194
  For the duration of the current QUERY VOLUME command, the value of SAVEDEV overrides any value set by the GLOBAL command SAVEDEV parameter or by the &OPT_SAVEDEV site option.

SESSION_LIST (Yes|No[,DETail|,NODETail|,DIFFerential])
  See “SESSION_LIST(Yes|No[,DETail|,NODETail|,DIFFerential)” on page 195.
  For the duration of the current QUERY VOLUME command, the value of SESSION_LIST overrides any value set by the GLOBAL command SESSION_LIST parameter or by the &SESSDETL, &SESSDIFF, and &SESSLIST site options.

SIZE (ALL|MOD1|MOD2|MOD3|MOD9|MOD27|MOD54|EAV|#|low-high)
  See “SIZE(ALL|MOD1|MOD2|MOD3|MOD9|MOD27|MOD54|EAV|#|low-high)” on page 195.
  For the duration of the current QUERY VOLUME command, the value of SIZE overrides any value set by the GLOBAL command SIZE parameter.

SNAPSHOT_LIST (ALL|LINKED|NOT_LINKED|SNAPSHOT)1
  See “SNAPSHOT_LIST(ALL|LINKED|NOT_LINKED|SNAPSHOT)” on page 196.
  For the duration of the current QUERY VOLUME command, the value of SNAPSHOT_LIST overrides any value set by the GLOBAL command SNAPSHOT_LIST parameter or by the &SNAPSHOT_LIST site option.

TDEV (EXClude|INClude)
  See “TDEV(EXClude|INClude)” on page 203.

1. Available starting with Mainframe Enablers 8.2.
Command Reference

For the duration of the current QUERY VOLUME command, the value of TDEV overrides any value set by the GLOBAL command TDEV parameter or by &OPT_TDEV site option.

**THINPOOL** (EXCLUDE | **INCLUDE**)

See “**THINPOOL(EXCLUDE|INCLUDE)**” on page 204.

**VDEVice** (EXCLUDE | **INCLUDE**)

See “**VDEVice(EXCLUDE|INCLUDE)**” on page 209.

For the duration of the current QUERY VOLUME command, the value of VDEVICE overrides any value set by the GLOBAL command VDEVICE parameter or by the &OPT_VDEV site option.

**Example**

The following example\(^1\) shows the multi-line query output:

```
EMCP001I QUERY VOLUME (DEVICE(1000) UNIT(8000) MLQ( NO))
ESNP504I UNIT 8000 WAS REQUESTED, FOUND OFFLINE
ESNP017I COMMAND PARSE COMPLETE
ESNP040I PROCESSING REQUESTS
ESNP160I PROCESSING FOR STATEMENT #1 BEGINNING, QUERY VOLUME REQUEST
ESNP165I PROCESSING CONTROLLER S/N 0001968-01232 - MICROCODE LEVEL - 5977
ESNP163I 00001000(N/A ) TDVS CKD-0000001113 RDY SNAP SRC
ESNP11A1I TOTAL NUMBER OF SNAPSHOTS RETURNED: 00000004
ESNP11AI SRC CUU VOLSER TGT CUU VOLSER NAME YYDDD/HH:MM:SS STATUS ... #1 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
```

Note: For field explanation, see the corresponding message ID in the *Mainframe Enablers Message Guide*.

The following example\(^2\) shows the output of a QUERY VOLUME command with the SESSION_LIST(YES,DETAIL,DIFF) parameter:

```
ESNP160I PROCESSING FOR STATEMENT #2 BEGINNING, QUERY VOLUME REQUEST
ESNP111I QUERY BY CUU WAS ISSUED - DISPLAY_CUU CHANGED TO YES
ESNP165I PROCESSING CONTROLLER S/N 0001926-01076 (UTD1) - MICROCODE LEVEL - 5876
ESNP163I 00000CB6(5056) SYMD76 STD CKD-0000010017 RDY RAID/10 SNAP SRC
ESNP11A1I SRC CUU VOLSER TOT CUU VOLSER NAME YYDDD/HH:MM:SS STATUS ACT
ESNP11AI 00001000(N/A ) FFFFFFF(N/A ) BGJ_UZB........161251325000006.. 16125/13:25:18 CREATE SOFT Y
ESNP11A1 00001000(N/A ) FFFFFFF(N/A ) BGJ_UZB........16125154300011.. 16125/15:43:31 CREATE SOFT N
ESNP11A1 00001000(N/A ) FFFFFFF(N/A ) BGJ_UZB........16125172500012.. 16125/17:31:39 CREATE SOFT Y
ESNP11A1 00001000(N/A ) FFFFFFF(N/A ) BGJ_UZB........16125174000013.. 16125/17:40:42 CREATE SOFT N
ESNP11AI 1 DEVICES LISTED
ESNP33I 8227 DEVICES SKIPPED, OUT OF DEVICE RANGE
ESNP161I PROCESSING FOR STATEMENT #1 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
```

The following example\(^3\) shows the output of a QUERY VOLUME command with the SESSION_LIST(DETAIL) parameter:

```
ESNP160I PROCESSING FOR STATEMENT #2 BEGINNING, QUERY VOLUME REQUEST
ESNP111I QUERY BY CUU WAS ISSUED - DISPLAY_CUU CHANGED TO YES
ESNP165I PROCESSING CONTROLLER S/N 0001926-01076 (UTD1) - MICROCODE LEVEL - 5876
ESNP163I 00000CB6(5056) SYMD76 STD CKD-0000010017 RDY RAID/10 SNAP SRC
ESNP11A1I SESSION TARGET TRACKCNT PROT-TRK PRECOPY # BGCPY DIFF PC DIFF-CNT DIFF-SRC DIFF-TGT
ESNP33I 25FD 00000CB8(5058) 150255 0 150255 YES YES NO 0 0
ESNP33I 04PC
```

The following example\(^3\) shows the output of a QUERY VOLUME command with the SESSION_LIST(DETAIL) parameter:

1. For Mainframe Enablers 8.4 and later.
2. For Mainframe Enablers 8.3.
**Note:** For field explanation, see the corresponding message ID in the Mainframe Enablers Message Guide.

The following example shows the output of a QUERY VOLUME command with the MULTI_LINE_query(Yes) parameter:

```
EMCP001I QUERY VOLUME { -
EMCP001I   DEVICE(35)-
EMCP001I   UNIT(6200) -
EMCP001I   MLQ(YES) -
EMCP001I   SESSION_LIST(DETAIL) -
EMCP001I   )
ESNP050I UNIT 6200 WAS REQUESTED, FOUND OFFLINE
ESNP017I COMMAND PARSE COMPLETE
ESNP040I PROCESSING REQUESTS
ESNP160I PROCESSING FOR STATEMENT #3 BEGINNING, QUERY VOLUME REQUEST
ESNP165I PROCESSING CONTROLLER S/N 0001970-00825 - MICROCODE LEVEL - 5977
ESNP163I 00000035(N/A ) TDVS CKD-0000001113 RDY SNAP SRC NO INVALID TRKS
ESNP036I M1-LCL,RY/R/W M2-NCNFG M3-NCNFG M4-NCNFG
ESNP11A1I
ESNP37I   SESSION#-TYPE    TARGET     TRACKCNT PROT-TRK PRECOPY # BOCPY DIFF PC
ESNP31I   65FD-NATIVE EXTENT SNAP 00000036(N/A ) 10 10 0 NO NO NO
ESNP11A1I
ESNP11A1I
ESNP11A1I
ESNP11A1I
ESNP11A1I
ESNP11A1I
ESNP32I  1 DEVICES LISTED
ESNP34I TOTALS: 10 TRACKS PROTECTED, AND 10 TRACKS INDIRECT
ESNP33I  1255 DEVICES SKIPPED, OUT OF DEVICE RANGE
ESNP161I PROCESSING FOR STATEMENT #3 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
```

3. Starting with Mainframe Enablers 8.4.
Command Reference

ESNP1AAI  TOTAL TRACKS TO BE COPIED IN REQUEST:  0000000000148814
ESNP1AAI  TOTAL SRC MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST:  0000000000000000
ESNP1AAI  TOTAL TGT MODIFIED TRACKS (UNIQUE+SHARED) IN REQUEST:  0000000000000000

Note: For field explanation, see the corresponding message ID in the Mainframe Enablers Message Guide. For explanation of snapshot-related fields in message ESNP1AAI, see the description of the QUERY SNAPSHOT command output in the TimeFinder SnapVX and zDP Product Guide.
RESTORE VOLUME (TF/Snap)

The RESTORE VOLUME command restores the contents of a virtual device (VDEV) to a standard (STD or BCV) volume. The term “standard” refers to both STD and BCV volumes, but not virtual devices.

There are three types of RESTORE VOLUME operations.

◆ From a VDEV to a BCV that has been SPLIT from the original standard that had a relationship with the virtual device.

This only applies to an original TF/Mirror split. It does not apply to a TF/Mirror Clone Emulation (CE) split. When clone emulation is involved, the CE session needs to be removed, using a DELINC command in TF/Mirror against the BCV. In addition to any CE sessions, any dataset extent or clone sessions must be removed before a RESTORE can proceed.

Note: This restore type is not available with PowerMaxOS 5978 and HYPERMAX OS 5977.

◆ From a VDEV to a different standard device

Note: This restore type is not available with PowerMaxOS 5978 and HYPERMAX OS 5977.

◆ From a VDEV to the original standard device (SNAP back)

For a TF/Clone or TF/Clone Emulation (CE), the VDEV restore to a standard device can proceed to completion when the clone is fully copied and in a split state. The clone session can remain in place for future full or differential operations, with the following exception; until the standard or VDEV relationship is removed, neither differential nor full BCV RESTORE for clone emulation or “snap back” for TF/Clone are allowed. PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876 is required for this VDEV restore action.

The restore is accomplished by a background copy task that copies indirect tracks to the restore device. The virtual device being restored is then removed and returns to the available pool of virtual devices for future use, if you are performing a VDEV to Standard RESTORE, and the PERSISTENT parameter is set to YES, then the Standard/VDEV relationship remains, and the virtual device is not returned to the virtual device pool.

If you are using a virtual restore (PERSISTENT parameter set to NO), you must terminate any other VDEV session or any virtual device assigned to the restore device in order to proceed with a RESTORE VOLUME.

For example:

```
RESTORE VOLUME (VDEV(UNIT(6FC0)) TO (UNIT(6C10)) - NEWVOLID(U6A010) REPLACE(YES))
```
If you are using a persistent restore (PERSISTENT parameter set to YES), you do not have to terminate any other VDEV session or any virtual device assigned to the restore device in order to proceed with a RESTORE VOLUME.

**Note:** This command is available only if you purchase the TF/Snap Licensed Feature Code.

### Syntax

```
RESTORE VOLUME
  (VDEVICE(VOLUME(volser)|UNIT(cuu)|SYMDV#(symdv#))
  TO(VOLUME(volser)|UNIT(cuu)|SYMDV#(symdv#))
  [optional_parameters]
)
```

Where *optional_parameters* are as follows:

- [AUTOMATIC_CLEANUP(Yes|No)]
- [CHECKBCVholdstatus(Yes|No)]
- [CHECKONLINEpathstatus(Yes|No|NEVER)]
- [CONDITIONVOLUME(ALL|LaBeL|DUMP)]
- [CONTROLLER([xxxxxxxx-]xxxxx|name)]
- [COPYVolid(Yes|No)]
- [DEBUG(ON|OFF)]
- [EXclude_PathGroupID(pathlist)]
- [INDDname(ddname)]
- [LOCAL({UNIT(cuu) CONTROLLER([xxxxxxxx-]xxxxx|name)] VOLUME(volser) CONTROLLER([xxxxxxxx-]xxxxx|name)] DDNAME(ddname) CONTROLLER([xxxxxxxx-]xxxxx|name)] CONTROLER([xxxxxxxx-]xxxxx|name)])
- [NEWVOLID(volser)]
- [NOTIFYwhencomplete([[GROUP(name)] [DATASET|JOB|STEP|SNAP]])]
- [OUTDDname(ddname)]
- [PERSISTent(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)]))]
[TOLERateENQFailure(Yes|No)]
[TRACE(ON|OFF)]
[VARY_OFFline(AUTO|NEVER)]
[VARY_ONline(AUTO|Yes|No)]
[VCLOSE(Yes|No)]
[WAITFORCOMPLETION([Yes|No|hh:mm:ss[,MeSsaGes],R1R2SYNC[,TIMEOUT(Informational|WARNing|ERRor)])]

**Note:** 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.

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

**Required parameters**

TO(VOLUME(volser)|UNIT(cuu,[cuu])|SYMDV#(symdv#))

The TO parameter identifies the target volume of the restore.

VOLUME(volser)

Identifies the target volume volser.

UNIT(cuu[,cuu])

Identifies the target volume UNIT address or range of addresses.

SYMDV#(symdv#)

Specifies the PowerMax/VMAX device number(s) in the remote storage storage system. If SYMDV# is specified, then the UNIT and VOLUME, and INDDNAME and OUTDDNAME parameters are not allowed.

**IMPORTANT**

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.

**Default value**

None

**Example**

UNIT(AA0F)

VDEVice(VOLUME(volser)|UNIT(cuu)|SYMDV#(symdv#))

**Note:** You can use this parameter only if you install the TF/Snap Licensed Feature Code.
The VDEVICE parameter identifies the virtual device to use for the restore operation.

**VOLume** *(volser)*

Specifies the volser of the virtual device.

**UNIT** *(cuu[, cuu])*  

Specifies the unit address of the virtual device or range of devices.

**SYMDV#** *(symdv#)*  

Specifies the remote PowerMax/VMAX device number(s) to be used as the target device(s). If SYMDV# is specified, then the UNIT and VOLUME, and INDDNAME and OUTDDNAME parameters are not allowed.

**Default value**

None

**Example**

UNIT(C100)

**Optional parameters**

**AUTOMATIC_CLEANUP** *(Yes|No)*  

See “AUTOMATIC_CLEANUP(Yes|No)” on page 154.

For the duration of the current RESTORE VOLUME command, the value of AUTOMATIC_CLEANUP overrides any value set by the GLOBAL command AUTOMATIC_CLEANUP parameter or by the &AUTOCLN site option.

**CHECKBCVholdstatus** *(Yes|No)*  

See “CHECKBCVholdstatus(Yes|No)” on page 157.

For the duration of the current RESTORE VOLUME command, the value of CHECKBCVHOLDSTATUS overrides any value set by the GLOBAL command CHECKBCVHOLDSTATUS parameter or by the &CHECKBCV site option.

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

See “CHECKONLINEpathstatus(Yes|No|NEVER)” on page 157.

For the duration of the current RESTORE VOLUME command, the value of CHECKONLINEpathstatus overrides any value set by the GLOBAL command CHECKONLINEpathstatus parameter or by the &CHKONLIN site option.

**CONDITIONVOLUME** *(ALL|LaBeL|DUMP)*  

See “CONDITIONVOLUME(ALL|LaBeL|DUMP)” on page 158.

For the duration of the current RESTORE VOLUME command, the value of CONDITIONVOLUME overrides any value set by the GLOBAL command CONDITIONVOLUME parameter or by the &CONDVOL site option.

**CONTROLLER** *( [xxxxxxx-]xxxxxx | name)*  

See “CONTROLLER([xxxxxxx-]xxxxxx|name)” on page 159.

The CONTROLLER parameter is only needed and can only be used if you use the SYMDV# parameter.
COPYVOLID(Yes|No)

See “COPYVOLID(Yes|No)” on page 161.

For the duration of the current RESTORE VOLUME command, the value of COPYVOLID overrides any value set by the GLOBAL command COPYVOLID parameter or by the &COPYVOL site option.

DEBUG(ON|OFF)

See “DEBUG(ON|OFF)” on page 164.

EXclude_PathGroupID(pathlist)

See “EXclude_PathGroupID(pathlist)” on page 169.

For the duration of the current RESTORE VOLUME command, the value of EXCLUDE_PATHGROUPID overrides any value set by the GLOBAL command EXCLUDE_PATHGROUPID parameter or by the &EXPATHGRP site option.

INDDname(ddname)

The INDDNAME parameter refers to a DD statement already allocated to the virtual volume to be restored.

ddname

The DD statement that refers to the virtual volume to be restored.

Default value

None

Example

INDD(INVOL)

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 175.

The LOCAL parameter is only needed and can only be used if you use the SYMDV# parameter.

NEWVOLID(volser)

See “NEWVOLID(volser)” on page 183.

NOTIFYwhencomplete({[GROUP(name)][DATASET|JOB|STEP|SNAP]})

See “NOTIFYwhencomplete({[GROUP(name)][DATASET|JOB|STEP|SNAP]})” on page 183.

For the duration of the current RESTORE VOLUME command, the value of NOTIFYWHENCOMPLETE overrides any value set by the GLOBAL command NOTIFYWHENCOMPLETE parameter or by the &NTFYLVI site option.
OUTDDName(ddname)

The OUTDDNAME parameter refers to a DD statement already allocated to the target volume of the restore.

ddname

Identifies the DD statement that refers to the target volume of the restore.

Default value
None

Example
OUTDD(OUTVOL)

PERSISTent(Yes|No)

See “PERSISTent(Yes|No)” on page 186.

For the duration of the current RESTORE VOLUME command, the value of PERSISTENT overrides any value set by the GLOBAL command PERSISTENT parameter or by the &PERSIST 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 190.

The REMOTE parameter is only needed and can only be used if you use the SYMDV# parameter.

Example

RESTORE VOLUME (TO (SYMDV# (008C)) VDEV (SYMDV# (021E)) - REMOTE(VOL(U6A230) RAGROUP(06) CONTROLLER(0001879-90132)) )

REPLac(e(Yes|No)

See “REPLac(e(Yes|No)” on page 192.

For the duration of the current RESTORE VOLUME command, the value of REPLACE overrides any value set by the GLOBAL command REPLACE parameter or by the &REPLACE site option.

TOLERateENQFailure(Yes|No)

See “TOLERateENQFailure(Yes|No)” on page 205.

For the duration of the current RESTORE VOLUME command, the value of TOLERATEENQFAILURE overrides any value set by the GLOBAL command TOLERATEENQFAILURE parameter or by the &ENQFAIL site option.

TRACE(ON|OFF)

See “TRACE(ON|OFF)” on page 207.
VARY_OFFline(\texttt{AUTO}|\texttt{NEVER})

See “VARY_OFFline(AUTO|NEVER)” on page 207.

For the duration of the current RESTORE VOLUME command, the value of VARY_OFFLINE overrides any value set by the GLOBAL command VARY_OFFLINE parameter or by the &VARYOFF site option.

VARY_ONline(\texttt{AUTO}|\texttt{Yes}|\texttt{No})

See “VARY_ONline(AUTO|Yes|No)” on page 208.

For the duration of the current RESTORE VOLUME command, the value of VARY_ONLINE overrides any value set by the GLOBAL command VARY_ONLINE parameter or by the &VARYON site option.

VCLOSE(\texttt{Yes}|\texttt{No})

See “VCLOSE(Yes|No)” on page 208.

For the duration of the current RESTORE VOLUME command, the value of VCLOSE overrides any value set by the GLOBAL command VCLOSE parameter or by the &VCLOSE site option.

WAITFORCOMPLETION([\texttt{Yes}|\texttt{No}|\texttt{hh:mm:ss}] [,\texttt{MesSageS}][,\texttt{R1R2SYNC}] [\texttt{TIMEOUT(INformational|WARNing|ERRor)}])

See “WAITFORCOMPLETION([\texttt{Yes}|\texttt{No}|\texttt{hh:mm:ss}] [,\texttt{MesSageS}][,\texttt{R1R2SYNC}] [\texttt{TIMEOUT(INformational|WARNing|ERRor)}])” on page 211.
SNAP DATASET (TF/Clone)

Use the SNAP DATASET command to create a copy of the specified dataset. Source and target devices must be the identical models. That is, you can snap from a 3390 device to another 3390 device, but you cannot snap from a 3390 to a 3380 device.

Note: "Performing a SNAP DATASET copy" on page 122 provides more information about SNAP DATASET operations.

Syntax

SNAP DataSet

(SOURCE(dsname) | INDDname(ddname))
TARGET(dsname) | OUTDDname(ddname)
[optional_parameters]
)

Where optional_parameters are as follows:

[ADMINISTRATOR(Yes|No)]
[ALLOCATE_UNUSED_SPACE(Yes|No)]
[ALLOCATION_SEQUENCE(DATASET|NONE|SIZE)]
[BACKGROUND COPY(Yes|No|NOCOPYRD)]
[BCVGROUP(groupname)]
[BCVOnly(Yes|No)]
[BUILD_VTOCIX(Yes|No)]
[BY(DSORG=|EQ|NE|NQ[(BDAM|EXCP|HFS|ISAM|PAM|PDS|PDSE|SAM|VSAM[])|]
    DATACLAS=|EQ|NE|NQ[(classname…[])]|]
    MGMTCLAS=|EQ|NE|NQ[(classname…[])]|]
    STORCLAS=|EQ|NE|NQ[(classname…[])]]
[CATalog(Yes|No)]
[COLLAPSE_dataset_extents(VSAM|NONVSAM|VSAM,NONVSAM)]
COPYsourceSMSclasses([[DATACLASs] [ManageMenTCLAss]
    [STORageCLASs] [ALL]])
[DATACLASs(classname)]
[DataMoverNaMe(ADDRSSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|
    FDRDSF|IDCAMS|NONE)]
[DATASET_CHANGED_indicator(SET|RESET|LEAVE)]
[DEBUG(OFF|ON)]
[DFDSS_ADMIN(Yes|No)]
[DFDSS_CC(Yes|No)]
[DIFFERENTIAL_DATASET(Yes|No)]
[EATTR(NO|OPT)]
[ENQSCOPE(REQUEST|STEP)]
[ENQWAIT(Yes|No)]
[ERROR_CHECKING(NORMAL|REDUCED)]
[ERROR_DISPOSITION(DELETE|KEEP)]
[ERROR_RECOVERY(NORMAL|ENHANCED)]
[ESNP220(ERROR|WARNING)]
[ESOTERIC(esoteric_name)]
[EXAMINE(Yes|No)]
[EXCLUDE(exclude_list)]
[EXPLAIN(VOLUME_SELECTION(Yes|No))]
### Command Reference


**[SMS_PASS_volumes]**:

Yes | No

**[SNAP_UNUSED_SPACE]**:

Yes | No

**[SOURCE_VOLUME_LIST]**:

vollist

**[SPHERE]**:

Yes | No

**[SRDFA_R1_target]**:

Yes | No | DATAMOVERName | Physical | Informational

**[SRDFA_R2_sync]**:

WARNING | DATAMOVER | R1R2SYNC

**[SRDFS_R1_target]**:

Yes | No | DATAMOVERName | Physical | Informational

**[STORAGECLASS]**:

classname

**[TARGET_ENQ_dataset_wait]**:

Yes | No | hh:mm:ss

**[TOLERATE_REUSE_FAILURE]**:

Yes | No

**[TOLERATE_ALLOCATION_FAILURE]**:

Yes | No

**[TOLERATE_COPY_FAILURE]**:

Yes | No

**[TOLERATE_ENQ_FAILURE]**:

Yes | No

**[TOLERATE_TRUNCATION]**:

Yes | No

**[TOLERATE_VSAM_ENQ_FAILURE]**:

Yes | No

**[TRACE]**:

ON | OFF

**[UNITNAME]**:

unitname

**[VERIFY]**:

Yes | No | NEVER

**[VERIFY_OPEN_SOURCE]**:

Yes | No

**[VOLUME]**:

volser

**[VOLUME_COUNT]**:

volumecount

**[VSAMENQMODE]**:

NONE | SHARED | EXCLUSIVE

**[WAITFORCOMPLETION]**:

[Yes | No | R1R2SYNC | hh:mm:ss]

[,,Message|,,R1R2SYNC],[,TIMEOUT(Informational|WARNING|ERROR)]

**[WAITforsession]**:

Yes | No | hh:mm:ss

---

**Note**: 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.

---

**Note**: The TARGET parameter is optional when RENAMEUnconditional is specified.

### Required parameters

**INDDname** *(ddname)*

You must specify INDDname or the SOURCE parameter. INDDname refers to a DD statement already allocated to the source dataset to be snapped.

**ddname**

Identifies the DD statement referring to the dataset to be snapped. The dataset may be any supported dataset type.
The INDDname parameter is the only way to specify a source dataset that is not cataloged.

Concatenated DD statements are not supported.

The ENQWAIT and HOSTCOPYMODE parameters are ignored with the INDDname parameter.

Default value
None

Example
INDD(INFILE)

OUTDDname (ddname)

The OUTDDname parameter refers to a DD statement already allocated to the target dataset that is reused. You must specify this parameter or the TARGET parameter.

ddname

Identifies the DD statement that refers to the dataset to be reused. The dataset may be any supported dataset type, but must match the source dataset type.

You should not attempt to snap a dataset onto itself.

If you use the OUTDDname parameter, the dataset is automatically reused. REPLACE(YES) and REUSE(YES) are assumed, and do not need to be specified.

Concatenated DD statements are not supported.

Default value
None

Example
OUTDD(OUTFILE)

SOURCE(dsname)

The SOURCE parameter specifies the name of the dataset to be snapped. You must specify SOURCE or the INDDname parameter.

To use TF/Clone, both the source and target datasets must resolve to compatible devices within the same storage system. All extents for the source dataset must reside on devices that are currently online and accessible.

The source device can be a PowerMax/VMAX standard device or a business continuance volume (BCV).

The dataset is located by using the standard catalog search sequence. JOBCAT and or STEPCAT statements are not supported. The source dataset name must be different from the target dataset name.

dsname

Specifies the dataset name of the source dataset. It can contain from one to 44 alphanumeric or national ($ @ #) characters, and two special characters (- or {}).
When special characters are used, the name must be surrounded by quotation marks and the special characters cannot be used as the first character of the name.

You can use wildcard characters in the source dataset name:

- `%` = a single character wildcard.
- `*` = a number of characters, up to the next period.
- `**` = a number of characters, including periods.

Wildcard characters (`*`, `%`) do not need to be enclosed in quotation marks.

You must have SAF READ or equivalent authorization for the source dataset.

**Default value**

None

**Example**

The following example identifies the source by standard dataset name:

```
SOURCE (PAYROLL.EMPLOYEE.MASTER)
```

```
TARGET (dsname)
```

**Note:** You can only use this parameter if you install the TF/Clone Licensed Feature Code.

The TARGET parameter specifies the name of the target dataset. You must specify this parameter or the OUTDDname parameter.

**dsname**

Specifies the dataset name of the target dataset. It can contain from one to 44 alphanumeric or national ($@#$) characters, and two special characters (- or {).}

When special characters are used, the name must be surrounded by quotation marks and the special characters cannot be used as the first character of the name.

The value can be:

You can use wildcard characters in the target dataset name:

- `%` = a single character wildcard.
- `*` = a number of characters, up to the next period.
- `**` = a number of characters, including periods.

Wildcard characters (`*`, `%`) do not need to be enclosed in quotation marks.

The source and target datasets may be on the same storage system, on different storage system, or even other compatible storage devices.
Normally, the source and target dataset must reside within the same physical storage system for the snap operation to be performed by the storage system. In some situations, this is not feasible. The DataMoverName parameter allows for a utility program to be specified to be invoked and actually copy the physical tracks.

For new non-VSAM target allocations, TF/Clone copies the LSTAR, TRBAL, BLKSIZE, DSORG, RECFM, LRECL, RKP and KEYLEN attributes from the source dataset to the target dataset. For existing non-VSAM target allocations which is reused, TF/Clone copies the LSTAR, TRBAL, BLKSIZE, RECFM, LRECL, RKP and KEYLEN attributes from the source dataset to the target dataset. The DSORG of the target dataset must match that of the source dataset, or the snap fails.

During the target dataset allocation process for VSAM clusters, if a new index or data component name can be made by appending the appropriate suffix to the cluster name, then TF/Clone builds the new name. Otherwise, TF/Clone does not generate the new name. Instead, TF/Clone passes the request to IDCAMS, which uses IBM rules for component name generation.

Note: The TimeFinder Utility for z/OS Product Guide provides more information about IDCAMS.

After the new dataset is successfully allocated, TF/Clone obtains the new component names from the catalog.

Note: You can find an explanation of VSAM component naming in the appropriate IBM Access Method Service manual.

SAF ALTER authorization is required for the target dataset. To ensure that the requestor has access to the target, TF/Clone opens the dataset for output.

You cannot specify the same dataset name on both the SOURCE and TARGET parameters.

DB2 linear datasets have a specific naming convention. The second level qualifier denotes whether it is the cluster or the data portion of the dataset. TF/Clone can only be used against the cluster. The component of the cluster can not be specified.

The following two examples use TF/Clone to duplicate DB2 linear datasets. Both of the options require the use of wildcarding that is available in TF/Clone.

For the following source dataset:

```
SOURCE(EMCDB2.DSNDBC.STORE.TSSTORA.I00001.A001)
```

- Option 1: Use a wildcard to denote the single unique level in the target dataset name. If the single level wildcard specification is used, it must be the second level qualifier:

  ```
  TARGET(BCVDB2.*.STORE.TSSTORA.I0001.A001)
  ```

- Option 2: Use a wildcard to denote the entire target dataset name after the specification of a unique high level qualifier (HLQ):

  ```
  TARGET(BCVDB2.**)
  ```
Default value
None

Example
The following example identifies the target by standard dataset name:

```
TARGET(PAYROLL.EMPLOYEE.SNAP)
```

Optional parameters

**ADMINISTRATOR (Yes|No)**

See “ADMINISTRATOR(Yes|No)” on page 153.

For the duration of the current SNAP DATASET command, the value of ADMINISTRATOR overrides any value set by the GLOBAL command ADMINISTRATOR parameter or by the &ADMIN site option.

**ALLOCATE UNUSED SPACE (Yes|No)**

See “ALLOCATE UNUSED SPACE(Yes|No)” on page 153.

For the duration of the current SNAP DATASET command, the value of ALLOCATE UNUSED SPACE overrides any value set by the GLOBAL command ALLOCATE UNUSED SPACE parameter or by the &ALUNUSED site option.

**ALLOCATION_SEQUENCE (DATASET|NONE|SIZE)**

See “ALLOCATION_SEQUENCE(DATASET|NONE|SIZE)” on page 154.

For the duration of the current SNAP DATASET command, the value of ALLOCATION_SEQUENCE overrides any value set by the GLOBAL command ALLOCATION_SEQUENCE parameter or by the &ALLOSEQ site option.

**BACKGROUND COPY (Yes|No|NOCOPYRD)**

See “BACKGROUND COPY(Yes|No|NOCOPYRD|VSE)” on page 155.

For the duration of the current SNAP DATASET command, the value of BACKGROUND COPY overrides any value set by the GLOBAL command BACKGROUND COPY parameter or by the &BACKGRND site option.

**BCVGROUP (groupName)**

The BCVGROUP parameter allows a group of BCV volumes to be indirectly referenced. The BCVGROUP input file contains a list of valid BCVGROUPs. This list is searched for a matching BCVGROUP name. All volumes referenced by the BCVGROUP are added to the SNAP DATASET volume candidate list. The BCVGROUP you specify can reference no more than 60 volumes.

**Note:** The group specified for BCVGROUP must contain devices with volsers.

**groupName**

Specifies a name that represents a BCV group.

If you choose BCVONLY(NO) (the default option), you can include standard volumes in the BCVGROUP as well.

**Default value**
Examples

BCVGROUP (IMGROUP1)

The following is an example of a definition of a BCVGROUP in the JCL.

```jcl
//BCVGROUP DD *
BCVGROUP IMGROUP1 VOL(VOL001 VOL002 VOL003)
/*
```

BCVOnly(Yes|No)

See “BCVOnly(Yes|No)” on page 156.

For the duration of the current SNAP DATASET command, the value of BCVONLY overrides any value set by the GLOBAL command BCVONLY parameter or by the &BCVONLY site option.

BUILD_VTOCIX(Yes|No)

See “BUILD_VTOCIX(Yes|No)” on page 156.

For the duration of the current SNAP DATASET command, the value of BUILD_VTOCIX overrides any value set by the GLOBAL command BUILD_VTOCIX parameter or by the &VTOCIX site option.

BY

```jcl
BY(DSORG=|EQ|NE|NQ[(|BDAM|EXCP|HFS|ISAM|PAM|PDS|PDSE|SAM|VSAM[)]) |
    DATACLAS=|EQ|NE|NQ[|classname...[]] |
    MGMTCLAS=|EQ|NE|NQ[|classname...[]] |
    STORCLAS=|EQ|NE|NQ[|classname...[]] |
```

The BY parameter determines how datasets are selected:

- **DATACLAS**: Selection based on membership in the specified SMS data class.
- **DSORG**: Selection based on dataset allocation type. This can be a comma delimited list of dataset allocation types. EXCP = Exclude datasets types that are not supported, such as HFS.
- **MGMTCLAS**: Selection based on membership in the specified SMS management class.
- **STORCLAS**: Selection based on membership in the specified SMS storage class. Valid values are:
  - **EQ**—Equal
  - **NE**—Not equal
  - **NQ**—Not equal

Default value

None

Example

```jcl
BY(DSORG=EQ(PDS PDSE))
```

CATalog(Yes|No)

See “CATalog(Yes|No)” on page 156.
For the duration of the current SNAP DATASET command, the value of CATALOG overrides any value set by the GLOBAL command CATALOG parameter or by the &CATALOG site option.

`COLLAPSE_dataset_extents(VSAM|NONVSAM|VSAM,NONVSAM)`

See “`COLLAPSE_dataset_extents(VSAM|NONVSAM|VSAM,NONVSAM)`” on page 158.

For the duration of the current SNAP DATASET command, the value of COLLAPSE_DATASET_EXTENTS overrides any value set by the GLOBAL command COLLAPSE_DATASET_EXTENTS parameter or by the &COLLAPSE site option.

`COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASs] [STORageCLASs] [ALL])`

See “`COPYsourceSMSclasses([DATACLASs] [ManaGeMenTCLASs] [STORageCLASs] [ALL])`” on page 160.

For the duration of the current SNAP DATASET command, the value of COPYSOURCESMSCLASSES overrides any value set by the GLOBAL command COPYSOURCESMSCLASSES parameter.

`DATACLASs(classname)`

See “`DATACLASs(classname)`” on page 162.

For the duration of the current SNAP DATASET command, the value of DATACLASS overrides any value set by the GLOBAL command DATACLASS parameter or by the &DATACLAS site option.

`DataMoverNaMe(ADRDSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)`

See “`DataMoverNaMe(ADRDSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|IDCAMS|NONE)`” on page 162.

For the duration of the current SNAP DATASET command, the value of DATAMOVERNAME overrides any value set by the GLOBAL command DATAMOVERNAME parameter or by the &DATAMOVR site option.

`DATASET_CHANGED_indicator(SET|RESET|LEAVE)`

See “`DATASET_CHANGED_indicator(SET|RESET|LEAVE)`” on page 164.

For the duration of the current SNAP DATASET command, the value of DATASET_CHANGED_indicator overrides any value set by the GLOBAL command DATASET_CHANGED_indicator parameter or by the &DS1DSCHA site option.

`DEBUG(OFF|ON)`

See “`DEBUG(OFF|ON)`” on page 164.

`DFDSS_ADMIN(Yes|No)`

See “`DFDSS_ADMIN(Yes|No)`” on page 165.

For the duration of the current SNAP DATASET command, the value of DFDSS_ADMIN overrides any value set by the GLOBAL command DFDSS_ADMIN parameter or by the &DFDSS_ADMIN site option.
DFDSS_CC (Yes | No)
   See “DFDSS_CC(Yes|No)” on page 165.
   For the duration of the current SNAP DATASET command, the value of DFDSS_CC overrides any value set by the GLOBAL command DFDSS_CC parameter or by the &DFDSS_CC site option.

DIFFERENTIAL_DATASET (Yes | No)
   See “DIFFERENTIAL_DATASET(Yes|No)” on page 166.
   For the duration of the current SNAP DATASET command, the value of DIFFERENTIAL_DATASET overrides any value set by the GLOBAL command DIFFERENTIAL_DATASET parameter or by the &DIFFDSN site option.

EATTR (NO | OPT)
   See “EATTR(NO|OPT)” on page 166.

ENQSCOPE (REQuest | STEP)
   See “ENQSCOPE(REquest|STEP)” on page 167.
   For the duration of the current SNAP DATASET command, the value of ENQSCOPE overrides any value set by the GLOBAL command ENQSCOPE parameter or by the &ENQSCOPE site option.

ENQWAIT (Yes | No)
   See “ENQWAIT(Yes|No)” on page 167.
   For the duration of the current SNAP DATASET command, the value of ENQWAIT overrides any value set by the GLOBAL command ENQWAIT parameter or by the &ENQWAIT site option.

ERROR_CHecking (NORmal | REDUCED)
   See “ERROR_CHecking(NORmal|REDUCED)” on page 167.
   For the duration of the current SNAP DATASET command, the value of ERROR_CHECKING overrides any value set by the GLOBAL command ERROR_CHECKING parameter or by the &ERRCHK site option.

ERRor_DISPosition (DELete | KEEP)
   See “ERRor_DISPosition(DElete|KEEP)” on page 168.
   For the duration of the current SNAP DATASET command, the value of ERROR_DISPOSITION overrides any value set by the GLOBAL command ERROR_DISPOSITION parameter or by the &EFFDISP site option.

ERROR_RECovery (NORmal | ENHanced)
   See “ERROR_RECovery(NORmal|ENHanced)” on page 168.
   For the duration of the current SNAP DATASET command, the value of ERROR_RECOVERY overrides any value set by the GLOBAL command ERROR_RECOVERY parameter or by the &ERRREC site option.

ESNP220 (ERROR | WARNING)
   See “ESNP220(ERROR|WARNING)” on page 169.
ESOteric(esoteric_name)

The ESOTERIC parameter specifies the device group name of the DASD devices onto which the source dataset is snapped.

esoteric_name

The device group name.

The ESOteric parameter may be ignored for SMS targets. When the ESOteric is passed to ACS routines, the routine determines if the ESOteric is ignored.

The ESOteric and UNITname parameters are aliases of each other and are mutually exclusive. If you use ESOteric, you cannot use UNITname.

Default value

None

Examples

ESO(DASD)

EXAMINE (Yes | No)

See “EXAMINE(Yes|No)” on page 169.

For the duration of the current SNAP DATASET command, the value of EXAMINE overrides any value set by the GLOBAL command EXAMINE parameter or by the &EXAMINE site option.

EXCLUDE(exclude_list)

The EXCLUDE parameter prevents datasets that are already selected from being snapped. If the SOURCE parameter causes several datasets to be selected, the EXCLUDE parameter may be used to eliminate some of them from the selection list. You can specify a comma-delimited list of up to 127 names or masks.

exclude_list

Specifies a list of up to 127 names or masks of datasets.

Default value

None

Examples

SNAP DATASET (SOURCE(EMC.**.ASM)
EXCLUDE(EMC.TEST*.**) TARGET (BACKUP.**.ASM)
VOL (EMCBCV))

Selects all datasets with the high level index EMC and the third index of ASM. All datasets where the second index begins with TEST are not snapped.

SNAP DATASET (SOURCE(EMC.**.ASM)
EXCLUDE(EMC.MASK*.**, EMC.DATASET1, EMC.DATASET2, EMC.OTHER.**) TARGET (BACKUP.**.TEXT)
VOL (EMCBCV))

Selects all datasets with the high level index EMC and the third index of ASM. All datasets where the second index begins with MASK, EMC.DATASET1, EMC.DATASET2, and all datasets where the second level index is OTHER are not snapped.
EXPlain(VOLUME_SELECTION(Yes|No))
See “EXPlain(VOLUME_SELECTION(Yes|No))” on page 170.

EXTENT_ALLOCATION(Yes[,CONSOLIDATE_Volume|CONSOLIDATE_ALL]|No)
See “EXTENT_ALLOCATION(Yes[,CONSOLIDATE_Volume|CONSOLIDATE_ALL]|No)” on page 170.
For the duration of the current SNAP DATASET command, the value of EXTENT_ALLOCATION overrides any value set by the GLOBAL command EXTENT_ALLOCATION parameter or by the following matching site options:
- &EXTALLOC = EXTENT_ALLOCATION(YES|NO)
- &CONSOALL = EXTENT_ALLOCATION(YES, CONSOLIDATE_ALL)
- &CONSVOL = EXTENT_ALLOCATION(YES, CONSOLIDATE_VOL)

EXTALLOC_EMC_ONLY(Yes|No)
See “EXTALLOC_EMC_ONLY(Yes|No)” on page 171.
For the duration of the current SNAP DATASET command, the value of EXTALLOC_EMC_ONLY overrides any value set by the GLOBAL command EXTALLOC_EMC_ONLY parameter or by the &EMCCOPY site option.

EXTENT_EXPAND(Yes|No,[ADDNEW(Yes|No)],[SAMEVOL],[NEWVOL])]
See “EXTENT_EXPAND(Yes|No,[ADDNEW(Yes|No)],[SAMEVOL],[NEWVOL])” on page 171.
For the duration of the current SNAP DATASET command, the value of EXTENT_EXPAND overrides any value set by the GLOBAL command EXTENT_EXPAND parameter or by the following site options:
- &EXTADDNEW
- &EXTXPVOL
- &EXTXPAND

FLASH_SNAP(FLASHCOPY|SNAP)
See “FLASH_SNAP(FLASHCOPY|SNAP)” on page 172.
For the duration of the current SNAP DATASET command, the value of FLASH_SNAP overrides any value set by the GLOBAL command FLASH_SNAP parameter or by the &FLASH_SNAP site option.

IMPORTANT
Do not change the value of this parameter unless directed to do so by Dell EMC.

FORCE(Yes|No)
See “FORCE(Yes|No)” on page 172.
For the duration of the current SNAP DATASET command, the value of FORCE overrides any value set by the GLOBAL command FORCE parameter or by the &FORCE site option.
HostcoPYMODesyReD|EXClusive|NONE)
See “HostcoPYMODesyReD|EXClusive|NONE)” on page 174.
For the duration of the current SNAP DATASET command, the value of
HOSTCOPYMODE overrides any value set by the GLOBAL command
HOSTCOPYMODE parameter or by the &HOSTCOPY site option.

INVALIDATE_PDSE_buffers(Yes|No)
See “INVALIDATE_PDSE_buffers(Yes|No)” on page 174.
For the duration of the current SNAP DATASET command, the value of
INVALIDATE_PDSE_BUFFERS overrides any value set by the GLOBAL command
INVALIDATE_PDSE_BUFFERS parameter or by INVALIDATE_PDSE site option.

LOGINDYNAM(volume[,volume...])
See “LOGINDYNAM(volume[,volume...])” on page 176.
For the duration of the current SNAP DATASET command, the value of
LOGINDYNAM overrides any value set by the GLOBAL command LOGINDYNAM
parameter.

ManaGeMenTCLASSs(classname)
See “ManaGeMenTCLASSs(classname)” on page 176.
For the duration of the current SNAP DATASET command, the value of
MANAGEMENTCLASS overrides any value set by the GLOBAL command
MANAGEMENTCLASS parameter or by the &MGMTCLAS site option.

MIGrate([PURge(Yes|No)] [RECall(Yes|No|IGNORE)])
See “MIGrate([PURge(Yes|No)] [RECall(Yes|No|IGNORE)])” on page 177.
For the duration of the current SNAP DATASET command, the value of MIGRATE
overrides any value set by the GLOBAL command MIGRATE parameter and the
following site options:
- &PURGE (MIGRATE PURGE)
- &RECALL (MIGRATE RECALL)

MODE(COPY|NOCOPY|NOCOPYRD)
See “MODE(COPY|NOCOPY|NOCOPYRD|VSE)” on page 178.
For the duration of the current SNAP DATASET command, the value of MODE
overrides any value set by the GLOBAL command MODE parameter.

NOTIFYwhencomplete([[GROUP(name)][DATASET|JOB|STEP|SNAP]])
See “NOTIFYwhencomplete([[GROUP(name)][DATASET|JOB|STEP|SNAP]])” on
page 183.
For the duration of the current SNAP DATASET command, the value of
NOTIFYWHENCOMPLETE overrides any value set by the GLOBAL command
NOTIFYWHENCOMPLETE parameter or by the &NTFYlvl site option.

OUTDDname(ddname)
The OUTDDNAME parameter refers to a DD statement already allocated to the
target volume to be snapped.
ddname

Identifies the DD statement that refers to the volume to be snapped. The volume may be any supported dataset type.

PARALLEL_CLONE(Yes|No|PREFERred|REQUIRED)

See “PARALLEL_CLONE(Yes|No|PREFERred|REQUIRED)” on page 185.

R1FULLCOPYONLY(Yes|No)

See “R1FULLCOPYONLY(Yes|No)” on page 188.

For the duration of the current SNAP DATASET command, the value of R1FULLCOPYONLY overrides any value set by the GLOBAL command R1FULLCOPYONLY parameter or by the &R1FULLCOPY site option.

RECALCULATE_FREESPACE(Yes|No)

See “RECALCULATE_FREESPACE(Yes|No)” on page 189.

For the duration of the current SNAP DATASET command, the value of RECALCULATE_FREESPACE overrides any value set by the GLOBAL command RECALCULATE_FREESPACE parameter or by the &RECALC_FREE site option.

RENAMEUnconditional(pfx) | RENAMEUnconditional((pfx) (oldnamemask,newnamemask)...)| RENAMEUnconditional((oldnamemask,newnamemask)...)

See “RENAMEUnconditional(pfx)| RENAMEUnconditional((pfx)(oldnamemask,newnamemask)...)| RENAMEUnconditional((oldnamemask,newnamemask)...)” on page 192.

Note: The TARGET parameter is optional when RENAMEUnconditional is specified.

RELate(dsname)

The RELATE parameter is used when snapping an alternate index dataset. By default, the target dataset is related to the same base cluster to which the source dataset is related. You can change this base cluster name by using the RELATE parameter. If the target dataset is being reused, this parameter is ignored.

dsname

Specifies the name of the base cluster that the new target alternate index is related.

The RELATE and SPHERE parameters are mutually exclusive. The RELATE parameter is used with alternate indexes, whereas the SPHERE parameter is used with primary clusters.

The RELATE parameter may be used only with a single source dataset. The SOURCE and TARGET parameters may not be wildcarded.

Default value

None

Example

REL(EMC.BASE,MASTER)
REPLace(Yes | No)
See “REPLace(Yes|No)” on page 192.
For the duration of the current SNAP DATASET command, the value of REPLACE overrides any value set by the GLOBAL command REPLACE parameter or by the &REPLACE site option.

REUSE(Yes | No [,WAIT])
See “REUSE(Yes|No[,WAIT])” on page 194.
For the duration of the current SNAP DATASET command, the value of REUSE overrides any value set by the GLOBAL command REUSE parameter or by the &REUSE site option.

REUSE_AUTO_expand(Yes | No)
See “REUSE_AUTO_expand(Yes|No)” on page 194.
For the duration of the current SNAP DATASET command, the value of REUSE_AUTO_EXPAND overrides any value set by the GLOBAL command REUSE_AUTO_EXPAND parameter or by the &AUTOXPND site option.

SCFGROUP(gnsgrp)
The SCFGROUP parameter specifies an SCF group that is made up of a target volume candidate list to which the dataset is to be snapped. This optional parameter is only valid with the TARGET parameter.

  gnsgrp
  The name of the Group Name Services (GNS) group. The name can contain up to 65 characters. If the name includes any special characters (including spaces), enclose the name in single quotes.

  The name must be predefined to ResourcePak Base.

  Note: The ResourcePak Base for z/OS Product Guide provides more information.

SELECTMULTI (ALL | ANY | FIRST)
See “SELECTMULTI(ALL|ANY|FIRST)” on page 195.
For the duration of the current SNAP DATASET command, the value of SELECTMULTI overrides any value set by the GLOBAL command SELECTMULTI parameter.

SMS_PASS_volumes(Yes | No)
See “SMS_PASS_volumes(Yes|No)” on page 196.
For the duration of the current SNAP DATASET command, the value of SMS_PASS_VOLUMES overrides any value set by the GLOBAL command SMS_PASS_VOLUMES parameter or by the &SMSPASSVOL site option.

SNAP_UNUSED_SPACE(Yes | No)
See “SNAP_UNUSED_SPACE(Yes|No)” on page 197.
For the duration of the current SNAP DATASET command, the value of SNAP_UNUSED_SPACE overrides any value set by the GLOBAL command SNAP_UNUSED_SPACE parameter or by the &SNUNUSED site option.

**SOURCE_VOLUME_LIST(volist)**

See “SOURCE_VOLUME_LIST(volist)” on page 198.

**SPHERE(Yes|No)**

The SPHERE parameter specifies that, for any VSAM cluster copied, all associated AIX clusters and paths are to be copied. Individual names of sphere components do not need to be specified. The base cluster name must be specified through the SOURCE or INDDname parameters.

Values can be:
- **Yes** All associated AIX clusters and PATHs are to be copied.
- **No** Only the selected base cluster is copied.

Because you can copy multiple components without a common naming structure in a single action, you can use the RENAMEUNCONDITIONAL parameter to provide the naming convention used for the copied PATH and AIX names.

DB2 linear datasets have a specific naming convention. The second level qualifier denotes whether it is the cluster or the data portion of the dataset. TF/Clone can only be used against the cluster. A component of the cluster can not be specified.

SPHERE can be used with wildcarded sources and targets.

**Default value**

None

**Example**

```plaintext
SPHERE(YES)
```

**SRDFA_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)**

See “SRDFA_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)” on page 199.

For the duration of the current SNAP DATASET command, the value of SRDFA_R1_TARGET overrides any value set by the GLOBAL command SRDFA_R1_TARGET parameter or by the &SRDFAR1 site option.

**SRDFA_R2_sync(WARNING|DATAMOVER|R1R2SYNC)**

See “SRDFA_R2_sync(WARNING|DATAMOVER|R1R2SYNC)” on page 200.

For the duration of the current SNAP DATASET command, the value of SRDFA_R2_SYNC overrides any value set by the GLOBAL command SRDFA_R2_SYNC parameter or by the &SRDFAR2 site option.

**SRDFS_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)**

See “SRDFS_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)” on page 201.

For the duration of the current SNAP DATASET command, the value of SRDFS_R1_TARGET overrides any value set by the GLOBAL command SRDFS_R1TARGET parameter or by the &SRDFS_R1 site option.
STORageCLASs(classname)

See “STORageCLASs(classname)” on page 201.

TARGET_ENQ_dataset_wait(Yes|No|hh:mm:ss)

The TARGET_ENQ_DATASET_WAIT parameter allows you to wait for a target dataset enqueue to become available:

- YES means waiting forever.
- NO (default) means no waiting.
- The hh:mm:ss option allows you to wait the specified amount of time before failing the job.

The aliases for this parameter are TNQDSWAIT and TNQDSW.

This parameter has a matching site option, &TARGET_WAIT.

TOLerate_REUSe_Failure(Yes|No)

See “TOLerate_REUSe_Failure(Yes|No)” on page 204.

For the duration of the current SNAP DATASET command, the value of TOLERATE_REUSE_FAILURE overrides any value set by the GLOBAL command TOLERATE_REUSE_FAILURE parameter or by the &REUSFAIL site option.

TOLerateALLOcationFailure(Yes|No)

See “TOLerateALLOcationFailure(Yes|No)” on page 205.

For the duration of the current SNAP DATASET command, the value of TOLERATE_ALLOCATION_FAILURE overrides any value set by the GLOBAL command TOLERATE_ALLOCATION_FAILURE parameter or by the &ALLOFAIL site option.

TOLerate_COPY_Failure(Yes|No)

See “TOLerate_COPY_Failure(Yes|No)” on page 205.

For the duration of the current SNAP DATASET command, the value of TOLERATE_COPY_FAILURE overrides any value set by the GLOBAL command TOLERATE_COPY_FAILURE parameter or by the &COPYFAIL site option.

TOLerateENQFailure(Yes|No)

See “TOLerateENQFailure(Yes|No)” on page 205.

For the duration of the current SNAP DATASET command, the value of TOLERATE_ENQ_FAILURE overrides any value set by the GLOBAL command TOLERATE_ENQ_FAILURE parameter or by the &ENQFAIL site option.

TOLerateTRUNCation(Yes|No)

See “TOLerateTRUNCation(Yes|No)” on page 206.

For the duration of the current SNAP DATASET command, the value of TOLERATE_TRUNCATION overrides any value set by the GLOBAL command TOLERATE_TRUNCATION parameter or by the &TRUNC site option.

TOLerateVSAMENQFailure(Yes|No)

See “TOLerateVSAMENQFailure(Yes|No)” on page 206.
For the duration of the current SETSNAP command, the value of TOLERATEVSAMENQFAILURE overrides any value set by the GLOBAL command TOLERATEVSAMENQFAILURE parameter or by the &VSAMFAIL site option.

**TRACE(ON|OFF)**

See “TRACE(ON|OFF)” on page 207.

**UNITName(unitname)**

The UNITName parameter specifies the device group name of the DASD devices onto which the source dataset is snapped. This optional parameter is only valid with the TARGET parameter.

*unitname*

Specifies the name of the generic locally defined group.

The UNITName parameter may be ignored for SMS targets. When the UNITName is passed to ACS routines, the routine determines if UNITName is ignored.

The ESOteric and UNITName parameters are aliases of each other and are mutually exclusive. If UNITName is used, ESOteric cannot be used.

**Default value**

None

**Example**

UNITN(DASD)

**VERIFY(Yes|No|NEVER)**

See “VERIFY(Yes|No|NEVER)” on page 209.

For the duration of the current SETSNAP command, the value of VERIFY overrides any value set by the GLOBAL command VERIFY parameter or by the &VERIFY site option.

**VERIFY_OPEN_SOURCE(Yes|No)**

See “VERIFY_OPEN_SOURCE(Yes|No)” on page 210.

For the duration of the current SETSNAP command, the value of VERIFY_OPEN_SOURCE overrides any value set by the GLOBAL command VERIFY_OPEN_SOURCE parameter or by the &VERIFY_OPEN_SOURCE site option.

**VOLUME(volser)**

The VOLUME parameter specifies the target volume candidate list to which the dataset is to be snapped. This optional parameter is only valid with the TARGET parameter.

*volser*

Specifies the volume serial number of a device that is online to the host attempting the snap. You can specify up to 59 volumes. Multiple volsers are separated by a space.
Guaranteed space is required, if this parameter is used in an SMS environment, or SMS could ignore the request for a specific target volume.

- The snap operation terminates when TF/Clone makes a request to obtain an extent and:
  - An allocation cannot be satisfied on the specified volume.
  - The VOLUME parameter does not resolve to an eligible device (the specified device has an invalid device type, is offline, or is unknown).

If the source dataset is SMS-controlled, the allocation may be resolved to another volume that is not in the VOLUME list. This would occur if none of the volumes in the VOLUME list were eligible, but there were eligible volumes in the SMS storage group.

See “Snapping to GSPACE datasets” on page 138 for more information about snaps to guaranteed space volumes.

**Default value**

None

**Example**

```
VOL(BKUP44)
VOL(BKUP44 BKUP45 BKUP46)
```

**VOLUMECount**(volume_count)

The VOLUMECount parameter specifies the maximum number of volumes on which a new target dataset can be allocated. This optional parameter is only valid with the TARGET parameter.

`volume_count`

Specifies the maximum number of volumes of which the target dataset is allocated. The `volume_count` must be a decimal number from 1 to 59 without quotes.

The rules of precedence are:

- If VOLUMECount is coded on the request statement, it is used with no override.
- If a data class is detected for the dataset, the volume count is used from that data class.
- If COPYSMS(DATACLAS) is specified and the source dataset has a data class, the volume count is used from that data class.
- If a data class is coded on the request statement, the volume count is used from that data class.
- If no data class is selected, coded, or implied, the source volume count (including candidates) is used.

If you do not specify the VOLUMECount parameter, TF/Clone uses, by default, the same number of volumes for the target as the source uses.

The VOLUMECount parameter is ignored for striped datasets.
In an SMS environment, when the VOLUMECOUNT parameter specifies a number greater than the actual number of source volumes, TF/Clone assigns candidate volumes to the target datasets.

**Default value**

Use the same number of volumes for the target as the number used for the source.

**Example**

```
VOLCNT(4)
```

**VSAMENQMODE (NONE|SHARED|EXClusive)**

See “VSaMENQMODE(SHAREd|EXClusive|NONE)” on page 210.

For the duration of the current SET DATASET command, the value of VSAMENQMODE overrides any value set by the GLOBAL command VSAMENQMODE parameter or by the &VSAMENQ site option.

**WAITFORCOMPLETION([Yes|No]|hh:mm:ss[,MeSSaGes][,R1R2SYNC][,TIMEOUT(INFormational|WARNing|ERRor)])**

See “WAITFORCOMPLETION([Yes|No]|hh:mm:ss[,MeSSaGes][,R1R2SYNC][TIMEOUT(INFormational|WARNing|ERRor)])” on page 211.

For the duration of the current SNAP DATASET command, the value of WAITFORCOMPLETION overrides any value set by the GLOBAL command WAITFORCOMPLETION parameter or by the &CMPLT site option.

**WAITforsession(Yes|No|hh:mm:ss)**

See “WAITforsession(Yes|No|hh:mm:ss)” on page 212.

For the duration of the current SNAP DATASET command, the value of WAITFORSESSION overrides any value set by the GLOBAL command WAITFORSESSION parameter or by the &WAIT site option.
SNAP VOLUME

The SNAP VOLUME command duplicates a single volume to another volume. You can snap only between devices of the same device type and model.

**Note:** The SNAP VOLUME command is not native for PowerMaxOS 5978 and HYPERMAX OS 5977, and may not be supported in future levels of the operating environment.

“Performing a SNAP VOLUME copy” on page 101 and “Performing a SNAP VOLUME using virtual devices” on page 111 provide more information about SNAP VOLUME operations.

When running SNAP VOLUME in GROUP processing with the SOFTLINK parameter set to YES, TF/Clone creates a softlinked snapshot but does not LINK it to any target.

Syntax

```
SNAP VOLume
(
SOURce(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#)) | 
SOURCE_VDEV(UNIT(cuu) | VOLume(volser) | SYMDV#(symdv#)) | 
INDDname(ddname) 
TaRget(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#)) | 
TARGET_VDEV(VOLume(volser) | UNIT(device[s]) | SYMDV#(symdv#)) | 
OUTDDname(ddname) | 
VDEVice(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#)) 
) 
```

Where optional_parameters are as follows:

- [ADMINISTRATOR(Yes|No)]
- [AUTO_BIND_thin_device(Yes|No)]
- [AUTOMATIC_DEALLOC(Yes|No)]
- [AUTOMATIC_RELEASE_hold(Yes|No)]
- [BACKGROUNDCOPY(Yes|No|NOCOPYRD)]
- [BCVOnly(Yes|No)]
- [CHECKBCVholdstatus(Yes|No)]
- [CHECKONLINEpathstatus(Yes|No|NEVER)]
- [Check_POOL_usable(Yes|No)]
[CONDitionVOLume(ALL|LaBeL|DUMP)]
[CONTROLLER([xxxxxxx-]xxxxx|name)]
[COPYVoid(Yes|No)]
[DataMoverNaMe(ADDRSSU|CO PYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDF|NONE)]
[DEBUG(ON|OFF)]
[DFDSS_ADMIN(Yes|No)]
[DFDSS_CC(Yes|No)]
[DIFferential(Yes|No)]
[ERROR_CHecking(NORMal|REDUCED)]
[ERROR_RECovery(NORMal|ENHanced)]
[EXclude_PathGroupID(pathlist)]
[FREESPACE(Yes|No)]
[GROUP(grpname[, grpname, ...])] 
[INDDname(ddname)]
[INVALIDATE_PDSE_buffers(Yes|No)]
[LOCAL({
    UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
    VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
    DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
    CONTROLLER([xxxxxxx-]xxxxx|name) })]
[MODE(COPY|NOCOPY|NOCOPYRD|VSE)]
[MULTI_VIRTual(Yes|No)]
[NAME(snapshot_name[%date[4|6|8]%][%time[4|6]%])]
[NEWVOLID(volser)]
[NOTIFYwhencomplete([GROUP(name)] [DATASET|JOB|STEP|SNAP])]
[OUTDDname(ddname)]
[PARALLEL_CLONE(Yes|No|PREFerred|REQuired)]
[POOL(poolname)]
[POSTSNAP(Yes|No)]
[PRECOPY(Yes|No)]
[PRESNAP(Yes|No)]
[R1FULLCOPYonly(Yes|No)]
[READY(Yes|No)]
[REFVTOC(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)])}]
[REPLACE(Yes|No)]
[SOFTlink(Yes|No)]
[SRDFA_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)]
[SRDFA_R2_sync(WARNING|DATAMOVER|R1R2SYNC)]
[SRDFS_R1_target(Yes|No|DATAMOVERName|PHysical|INformational)]
[TDEV_RECLAIM(Yes|No)]
[TERMINATE_SESSION_when_complete(Yes|No)]
[TOLerateENQFailure(Yes|No)]
[TRACE(ON|OFF)]
[VARY_OFFline(AUTO|NEVER)]
[VARY_ONline(AUTO|Yes|No)]
[VCLOSE(Yes|No)]
[VDEVWAIT(Yes|No)]
[WAITFORCOMPLETION([Yes|No|hh:mm:ss]
[,MeSSaGes][,R1R2SYNC][,TIMEOUT(INFormational|WARNing|ERRor)])
[WAIT_FOR_PRECOPY_PASS1(Yes|No)]
[WAITforsession(Yes|No|hh:mm:ss])
[WHEN_SAVEDEV_FULL(READY|NOTREADY)]

Note: 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.

Note: GROUP, PRESNAP or POSTSNAP parameters are not allowed in SNAP VOLUME statements that are stored within a group. These parameters are allowed when the SNAP VOLUME statements occur in regular input, such as after a //QCINPUT DD * JCL statement.

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

Required parameters

INDDname(ddname)

You must specify the SOURCE parameter or the INDDname parameter. You may use either, but not both. The INDDname parameter refers to a DD statement already allocated to the source volume to be snapped.

ddname

Identifies the DD statement that refers to the volume to be snapped. The volume may be any supported dataset type.
If you use the SOURCE or INDDNAME parameters, then you must also use the TARGET or OUTDDNAME parameters. However, if you use the GROUP parameter, do not use the SOURCE or TARGET parameters.

If you use TARGET subparameter SYMDV#, you cannot use INDDname.

Default value

None

Example

INDD(INVOL)

OUTDDname (ddname)

The OUTDDNAME parameter refers to a DD statement already allocated to the target volume to be snapped. You must specify this parameter, the TARGET parameter, or the VDEVice parameter.

ddname

Identifies the DD statement that refers to the volume to be snapped. The volume may be any supported dataset type.

If you use the SOURCE or INDDNAME parameters, then you must also use the TARGET or OUTDDNAME parameter. However, if you use the GROUP parameter, do not use the SOURCE or TARGET parameters.

If you specify the TARGET subparameter SYMDV#, then you cannot specify OUTDDname.

Default value

None

Example

OUTDD(OUTVOL)

SOUrce(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#))

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 snap.

If the volser contains a hyphen, enclose it in single quotes, for example: VOL('vol-ser')

UNIT(cuu)

The unit address of one or more devices that are online to the host attempting the snap. See “UNIT(cuu)” on page 207.

SYMDV#(symdv#)

1. With Mainframe Enablers 8.1 and later.
See “SYMDV#(symdv#)” on page 202.

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

If you use SOURCE, you must also use TARGET. However, if you use the GROUP parameter, do not use the SOURCE or TARGET parameters.

The number of devices specified in the SOURCE parameter must be the same as the number of devices specified in the TARGET parameter.

**Default value**

None

**Example**

```plaintext
SOU(VOLUME(STDVOL))
```

```plaintext
SOURCE_VDEV(UNIT(cuu)|VOLUME(volser)|SYMDV#(symdv#))
```

**Note:** You can use this parameter for virtual volume operations only if you have installed the TF/Snap licensed feature code.

The SOURCE_VDEV parameter determines whether a virtual device is used as the source device of a “duplicate snap” operation. This would mean creating a point-in-time copy of a virtual device that is already participating in a previously activated Snap Session.

**Note:** Duplicate snaps are not supported with PowerMaxOS 5978 and HYPERMAX OS 5977.

By specifying a virtual device as both the source and target of a SNAP VOLUME statement, the source VDEV is duplicated to the target VDEV and the target VDEV is based on the same original standard device.

**UNIT(cuu)**

Specifies using the virtual device(s) addressed as a PowerMax/VMAX device number. See “UNIT(cuu)” on page 207.

**VOLUME(volser)**

Specifies to use the virtual device labeled volser.

If the volser contains a hyphen, enclose it in single quotes, for example:

```plaintext
VOL('vol-ser')
```

**SYMDV#(symdv#)**

Specifies the device number to be used as the source device. See “SYMDV#(symdv#)” on page 202.

---

1. With Mainframe Enablers 8.1 and later.
If SYMDV# is specified, then the UNIT and VOLUME, and INDDNAME and OUTDDNAME parameters are not allowed. However, you must use the LOCAL, REMOTE, or CONTROLLER parameters when you specify SYMDV#.

A duplicate VDEV is counted as part of the 128 multivirtual limit off of a source device.

There is a maximum of 2 duplicate VDEVs per source in an “established and inactive” state at any time. Once activated, up to 2 more can be established.

Termination or re-snap of the original VDEV session is not allowed with an inactive duplicate VDEV.

Default

None

Example

SNAP VOLUME (SOURCE_VDEV VOLUME(scrdvdev) TARGET_VDEV VOLUME(tgtvdev))

TARGET(VOLUME(volser)|UNIT(cuu)|SYMDV#(symdv#))|
TARGET_VDEV(VOLUME(volser)|UNIT(device[s])|SYMDV#(symdv#))

Note: You can only use this parameter for full-volume snaps if you install the TF/Clone Licensed Feature Code.

The TARGET parameter defines the target of the snap. With Enginuity 5876, the target device can also be a virtual device that participates in a “duplicate snap” operation.

Note: Duplicate snaps are not supported with PowerMaxOS 5978 and HYPERMAX OS 5977.

With PowerMaxOS 5978 and HYPERMAX OS 5977, the system always allows 32 virtual device sessions with the TARGET_VDEV parameter. To overcome this limitation, use the SOFTlink parameter.

TARGET_VDEV is an alias for TARGET.

VOLUME(volser)

Specifies the volume serial number of a device that is online to the host attempting the snap.

If the volser contains a hyphen, enclose it in single quotes, for example: VOL('vol-ser')

Note: TARGET parameters that apply to a group must use the UNIT subparameter instead of the VOLUME subparameter.

1. With Mainframe Enablers 8.1 and later.
UNIT(cuu)

Specifies the unit address of a device that is online to the host attempting the snap. See “UNIT(cuu)” on page 207.

SYMDV#(symdv#)

Identifies the internal PowerMax/VMAX device number in the target storage for the snap operation. See “SYMDV#(symdv#)” on page 202.

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#.

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

If you use the TARGET or OUTDDNAME parameters, then you must also use the SOURCE or INDDNAME parameters. However, if you use the GROUP parameter, do not use the SOURCE or TARGET parameters.

The number of devices specified in the TARGET parameter must be the same as the number of devices specified in the SOURCE parameter.

Default value

None

Example

TRG(VOLUME(TGTVOL))

VDEVice(UNIT(cuu)|VOLume(volser)| SYMDV#(symdv#))

Note: You can use this parameter for virtual volume operations only if you have installed the TF/Snap licensed feature code.

The VDEVICE parameter determines whether a virtual device is used for the snap.

UNIT(cuu)

Specifies using the virtual device(s) addressed as a PowerMax/VMAX device number. See “UNIT(cuu)” on page 207.

VOLume(volser)

Specifies to use the virtual device labeled volser.

If the volser contains a hyphen, enclose it in single quotes, for example: VOL('vol-ser')1.

SYMDV#(symdv#)

Specifies the device number to be used as the target device. See “SYMDV#(symdv#)” on page 202.

1. With Mainframe Enablers 8.1 and later.
If SYMDV# is specified, then the UNIT and VOLUME, and INDDNAME and OUTDDNAME parameters are not allowed. However, you must use the LOCAL, REMOTE, or CONTROLLER parameters when you specify SYMDV#.

Default
None

Optional parameters

ADMINISTRATOR(Yes|No)
See “ADMINISTRATOR(Yes|No)” on page 153.
For the duration of the current SNAP VOLUME command, the value of ADMINISTRATOR overrides any value set by the GLOBAL command ADMINISTRATOR parameter or by the &ADMIN site option.

AUTO_BIND_thin_device(Yes|No)
AUTO_BIND_THIN_DEVICE(YES), along with the required pool name, binds a thin device to a requested pool prior to the SNAP VOLUME processing. This parameter is used when an unbound thin device is referenced as a target device in a SNAP VOLUME statement.

Note: The AUTO_BIND parameter is only compatible with Enginuity 5876 and 5773.
The AUTO_BIND_thin_device parameter has a matching site option, &AUTO_BIND_TDEV.

AUTOMATIC_DEALLOC(Yes|No)
See “AUTOMATIC_DEALLOC(Yes|No)” on page 154.
For the duration of the current SNAP VOLUME command, the value of AUTOMATIC_DEALLOC overrides any value set by the GLOBAL command AUTOMATIC_DEALLOC parameter or by the &AUTODEAL site option.

AUTOMATIC_RELEASE_hold(Yes|No)
See “AUTOMATIC_RELEASE_hold(Yes|No)” on page 155.
For the duration of the current SNAP VOLUME command, the value of AUTOMATIC_RELEASE_HOLD overrides any value set by the GLOBAL command AUTOMATIC_RELEASE_HOLD parameter or by the &AUTORLSE site option.

BACKGROUNDCOPY(Yes|No|NOCOPYRD)
See “BACKGROUNDCOPY(Yes|No|NOCOPYRD|VSE)” on page 155.
For the duration of the current SNAP VOLUME command, the value of BACKGROUNDCOPY overrides any value set by the GLOBAL command BACKGROUNDCOPY parameter or by the &BACKGRND site option.

BCVOnly(Yes|No)
See “BCVOnly(Yes|No)” on page 156.
For the duration of the current SNAP VOLUME command, the value of BCVONLY overrides any value set by the GLOBAL command BCVONLY parameter or by the &BCVONLY site option.
CHECKBCVholdstatus(Yes|No)

See “CHECKBCVholdstatus(Yes|No)” on page 157.

For the duration of the current SNAP VOLUME command, the value of CHECKBCVHOLDSTATUS overrides any value set by the GLOBAL command CHECKBCVHOLDSTATUS parameter or by the &CHECKBCV site option.

CHECKONLINEpathstatus(Yes|No|NEVER)

See “CHECKONLINEpathstatus(Yes|No|NEVER)” on page 157.

For the duration of the current SNAP VOLUME command, the value of CHECKONLINEPATHSTATUS overrides any value set by the GLOBAL command CHECKONLINEPATHSTATUS parameter or by the &CHKONLIN site option.

CHECK_Pool_usable(Yes|No)

See “CHECK_Pool_usable(Yes|No)” on page 157.

CONDITIONVOLUME(ALL|LaBeL|DUMP)

See “CONDITIONVOLUME(ALL|LaBeL|DUMP)” on page 158.

For the duration of the current SNAP VOLUME command, the value of CONDITIONVOLUME overrides any value set by the GLOBAL command CONDITIONVOLUME parameter or by the &CONDVOL site option.

CONTROLLER([xxxxxxxx-]xxxxx|name)

See “CONTROLLER([xxxxxxxx-]xxxxx|name)” on page 159.

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

COPYVolid(Yes|No)

See “COPYVolid(Yes|No)” on page 161.

For the duration of the current SNAP VOLUME command, the value of COPYVOLID overrides any value set by the GLOBAL command COPYVOLID parameter or by the &COPYVOL site option.

DataMoverName(ADRDSSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|NONE)

See “DataMoverName(ADRDSSU|COPYCYL|COPYTRK|DFDSS|DSS|FDR|FDRDSF|NONE)” on page 162.

For the duration of the current SNAP VOLUME command, the value of DATAMOVERNAME overrides any value set by the GLOBAL command DATAMOVERNAME parameter or by the &DATAMOVR site option.

DEBUG(ON|OFF)

See “DEBUG(ON|OFF)” on page 164.

DFDSS_ADMIN(Yes|No)

See “DFDSS_ADMIN(Yes|No)” on page 165.

For the duration of the current SNAP VOLUME command, the value of DFDSS_ADMIN overrides any value set by the GLOBAL command DFDSS_ADMIN parameter or by the &DFDSS_ADMIN site option.
DFDSS_CC (Yes | No)
See “DFDSS_CC(Yes|No)” on page 165.
For the duration of the current SNAP VOLUME command, the value of DFDSS_CC overrides any value set by the GLOBAL command DFDSS_CC parameter or by the &DFDSS_CC site option.

DIFFerential (Yes | No)
See “DIFFerential(Yes|No)” on page 165.
For the duration of the current SNAP VOLUME command, the value of DIFFERENTIAL overrides any value set by the GLOBAL command DIFFERENTIAL parameter or by the &DIFFDSN site option.

ERROR_CHecking (NORmal | REDUCED)
See “ERROR_CHecking(NORmal|REDUCED)” on page 167.
For the duration of the current SNAP VOLUME command, the value of ERROR_CHECKING overrides any value set by the GLOBAL command ERROR_CHECKING parameter or by the &ERRCHK site option.

ERROR_RECovery | ERRREC (NORmal | ENHanced)
See “ERROR_RECovery(NORmal|ENHanced)” on page 168.
For the duration of the current SNAP VOLUME command, the value of ERROR_RECOVERY overrides any value set by the GLOBAL command ERROR_RECOVERY parameter or by the &ERRREC site option.

EXclude_PathGroupID (pathlist)
See “EXclude_PathGroupID(pathlist)” on page 169.
For the duration of the current SNAP VOLUME command, the value of EXCLUDE_PATHGROUPID overrides any value set by the GLOBAL command EXCLUDE_PATHGROUPID parameter or by the &EXPATHGRP site option.

FREESPACE (Yes | No)
See “FREESPACE(Yes|No)” on page 173.
For the duration of the current SNAP VOLUME command, the value of FREESPACE overrides any value set by the GLOBAL command FREESPACE parameter or by the &FREESPC site option.

GROUP (grpname[, grpname, ...])
See “GROUP(grpname[,grpname,...])” on page 173.
If you use the GROUP parameter, do not use the SOURCE or TARGET parameters.

INVALIDATE_PDSE_buffers (Yes | No)
See “INVALIDATE_PDSE_buffers(Yes|No)” on page 174.
For the duration of the current SNAP VOLUME command, the value of INVALIDATE_PDSE_buffers overrides any value set by the GLOBAL command INVALIDATE_PDSE_buffers parameter or by INVALIDATE_PDSE site option.

LOCAL ({
UNIT (cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |
VOLUME (volser) [CONTROLLER([xxxxxxx-]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 175.

The LOCAL parameter is only needed and can only be used if you use the SYMDV# parameter.

MODE(COPY|NOCOPY|NOCOPYRD|VSE)

See “MODE(COPY|NOCOPY|NOCOPYRD|VSE)” on page 178.
For the duration of the current SNAP VOLUME command, the value of MODE
overrides any value set by the GLOBAL command MODE parameter.

MULTI_VIRTual(Yes|No)

See “MULTI_VIRTual(Yes|No)” on page 182.

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

See “NAME(snapshot_name[%date[4|6|8%][%time[4|6]%])” on page 182.

NEWVOLID(volser)

See “NEWVOLID(volser)” on page 183.

NOTIFYwhencomplete([GROUP(name)][DATASET|JOB|STEP|SNAP]])

See “NOTIFYwhencomplete([GROUP(name)][DATASET|JOB|STEP|SNAP]])” on page 183.

For the duration of the current SNAP VOLUME command, the value of
NOTIFYWHENCOMPLETE overrides any value set by the GLOBAL command
NOTIFYWHENCOMPLETE parameter or by the &NTFYVL parameter.

PARALLEL_CLONE(Yes|No|PREFERed|REQuired)

See “PARALLEL_CLONE(Yes|No|PREFERed|REQuired)” on page 185.

POOL(poolname)

See “POOL(poolname)” on page 186.

For the duration of the current SNAP VOLUME command, the value of POOL
overrides any value set by the GLOBAL command POOL parameter or by the
&POOL site option.

POSTSNAP(Yes|No)

See “POSTSNAP(Yes|No)” on page 187.

PRECOPY(Yes|No)

See “PRECOPY(Yes|No)” on page 187.

For the duration of the current SNAP VOLUME command, the value of PRECOPY
overrides any value set by the GLOBAL command PRECOPY parameter or by the
&PRECOPY site option.

PRESNAP(Yes|No)
See “PRESNAP(Yes|No)” on page 187.

PRESNAP may only be used if GROUP is also specified.

R1FULLCOPYonly(Yes|No)

See “R1FULLCOPYonly(Yes|No)” on page 188.

For the duration of the current SNAP VOLUME command, the value of R1FULLCOPYONLY overrides any value set by the GLOBAL command R1FULLCOPYONLY parameter or by the &R1FULLCOPY site option.

READY(Yes|No)

See “READY(Yes|No)” on page 189.

REFVTOC(Yes|No)

See “REFVTOC(Yes|No)” on page 190.

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 190.

The REMOTE parameter is only needed and can only be used if you use the SYMDV# parameter.

REPLace(Yes|No)

See “REPLace(Yes|No)” on page 192.

If the existing device contains user data and you leave REPLACE with the default value of NO, the snap operation terminates. If you specify REPLACE(YES) and the target volume is not empty, old data on that volume is overwritten.

For the duration of the current SNAP VOLUME command, the value of REPLACE overrides any value set by the GLOBAL command REPLACE parameter or by the &REPLACE site option.

SOFTlink(Yes|No)

See “SOFTlink(Yes|No)” on page 197.

SRDFA_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormational)

See “SRDFA_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormational)” on page 199.

For the duration of the current SNAP VOLUME command, the value of SRDFA_R1_TARGET overrides any value set by the GLOBAL command SRDFA_R1_TARGET parameter or by the &SRDFAR1 site option.

SRDFA_R2_sync(WARNING|DATAMOVER|R1R2SYNC)

See “SRDFA_R2_sync(WARNING|DATAMOVER|R1R2SYNC)” on page 200.
For the duration of the current SNAP VOLUME command, the value of SRDFA_R2_SYNC overrides any value set by the GLOBAL command SRDFA_R2_SYNC parameter or by the &SRDFAR2 site option.

SRDFS_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormationaI)

See “SRDFS_R1_target(Yes|No|DATAMOVERNaMe|PHYsical|INFormationaI)” on page 201.

For the duration of the current SNAP VOLUME command, the value of SRDFS_R1_TARGET overrides any value set by the GLOBAL command SRDFS_R1_TARGET parameter or by the &SRDFS_R1 site option.

TDEV_RECLAIM(Yes|No)

See “TDEV_RECLAIM(Yes|No)” on page 203.

TERMINATE_SESSION_when_complete(Yes|No)

See “TERMINATE_SESSION_when_complete(Yes|No)” on page 203.

For the duration of the current SNAP VOLUME command, the value of TERMINATE_SESSION_WHEN_COMPLETE overrides any value set by the GLOBAL command TERMINATE_SESSION_WHEN_COMPLETE parameter or by the &TERMSESS site option.

TOLerateENQFailure(Yes|No)

See “TOLerateENQFailure(Yes|No)” on page 205.

For the duration of the current SNAP VOLUME command, the value of TOLERATEENQFAILURE overrides any value set by the GLOBAL command TOLERATEENQFAILURE parameter or by the &ENQFAIL site option.

TRACE(ON|OFF)

See “TRACE(ON|OFF)” on page 207.

VARY_Offline(AUTO|NEVER)

See “VARY_Offline(AUTO|NEVER)” on page 207.

For the duration of the current SNAP VOLUME command, the value of VARY_OFFLINE overrides any value set by the GLOBAL command VARY_OFFLINE parameter or by the &VARYOFF site option.

VARY_Online(AUTO|Yes|No)

See “VARY_Online(AUTO|Yes|No)” on page 208.

For the duration of the current SNAP VOLUME command, the value of VARY_ONLINE overrides any value set by the GLOBAL command VARY_ONLINE parameter or by the &VARYON site option.

VCLOSE(Yes|No)

See “VCLOSE(Yes|No)” on page 208.

For the duration of the current SNAP VOLUME command, the value of VCLOSE overrides any value set by the GLOBAL command VCLOSE parameter or by the &VCLOSE site option.

VDEVWAIT(Yes|No)

See “VDEVWAIT(Yes|No)” on page 209.
For the duration of the current SNAP VOLUME command, the value of VDEVWAIT overrides any value set by the GLOBAL command VDEVWAIT parameter or by the &VDEVWAIT site option.

\texttt{WAITFORCOMPLETION([Yes|No|hh:mm:ss] [,MeSsaGes] [,R1R2SYNC] [TIMEOUT(INformational|WARNing|ERRor)])}

See “\texttt{WAITFORCOMPLETION([Yes|No|hh:mm:ss] [,MeSsaGes] [,R1R2SYNC] [TIMEOUT(INformational|WARNing|ERRor)])}” on page 211.

\textbf{Note:} Under PowerMaxOS 5978 and HYPERMAX OS 5977, \texttt{WAITFORCOMPLETION(YES)} is ignored.

For the duration of the current SNAP VOLUME command, the value of WAITFORCOMPLETION overrides any value set by the GLOBAL command WAITFORCOMPLETION parameter or by the &CMPLTMSG site option.

\texttt{WAIT\_FOR\_PRECOPY\_PASS1(Yes|No)}

See “\texttt{WAIT\_FOR\_PRECOPY\_PASS1(Yes|No)}” on page 212.

For the duration of the current SNAP VOLUME command, the value of \texttt{WAIT\_FOR\_PRECOPY\_PASS1} overrides any value set by the GLOBAL command \texttt{WAIT\_FOR\_PRECOPY\_PASS1} parameter or by the &WAIT\_PRECOPY site option.

\texttt{WAIT\_for\_session(Yes|No|hh:mm:ss)}

See “\texttt{WAIT\_for\_session(Yes|No|hh:mm:ss)}” on page 212.

For the duration of the current SNAP VOLUME command, the value of WAIT\_for\_session overrides any value set by the GLOBAL command WAIT\_for\_session parameter or by the &WAIT site option.

\texttt{WHEN\_SAVEDEV\_FULL(READY|NOTREADY)}

See “\texttt{WHEN\_SAVEDEV\_FULL(READY|NOTREADY)}” on page 213.

For the duration of the current SNAP VOLUME command, the value of WHEN\_SAVEDEV\_FULL overrides any value set by the GLOBAL command WHEN\_SAVEDEV\_FULL parameter or by the &SAVEFULL site option.
STOP SNAP TO DATASET (TF/Clone)

The STOP SNAP TO DATASET command stops the specified target dataset.

Syntax

```
STOP SNAP TO DATASET
  (OUTDDname(ddname) | TARGET(dsname) [SCRATCHdataset(Yes | No)])
```

Required parameters

OUTDDname(ddname)

The OUTDDNAME parameter specifies the target dataset.

**ddname**

The DD name of the target dataset.

**Default value**

None

**Example**

None

TARGET(dsname)

**Note:** You can use this parameter only if you install the TF/Clone Licensed Feature Code.

Specifies the target dataset by name. The value can be:

**dsname**

The dsname of the target dataset.

No wildcard name patterns are allowed with this command.

**Default value**

None

**Example**

None

Optional parameters

SCRATCHdataset(Yes | No)

The SCRATCHDATASET parameter is valid only when used with the TARGET parameter. SCRATCHdataset erases the dataset specified in TARGET(dsname), even if the dataset specified is the source of a dataset snap:

**Yes**

Erases the scratch dataset identified by the TARGET(dsname) parameter.

**No** (Default)

Does not erase the scratch dataset identified by the TARGET(dsname) parameter.

You can substitute ON for YES and OFF for NO.
STOP SNAP TO VOLUME

The STOP SNAP TO VOLUME command stops the specified target volume. When you issue a STOP SNAP TO VOLUME, you might need to reinitialize the target volume before you can use it again.

Note: The STOP SNAP TO VOLUME command stops the sessions as data remains on the target. For thin target devices, STOP SNAP TO VOLUME does not free the SRP pool used tracks. Run the FREE command described in the *TimeFinder SnapVX and zDP Product Guide* to free the target device SRP pool space if necessary.

Syntax

```
STOP SNAP TO VOLUME

(OUTDDname(ddname) | TaRGet(VOLUME(volser) | UNIT(cuu) | SYMDV#(symdv#)) | VDEVice(VOLUME(volser) | UNIT(cuu) | SYMDV#(symdv#)) | GROUP(grpname[, grpname,...])

[optional_parameters]
)
```

Where optional_parameters are as follows:

- [AUTO_UNBIND_thin_device(Yes|No)]
- [CHECKBCVholdstatus(Yes|No)]
- [CONTROLLER([xxxxxxx-]xxxxx|name)])
- [LOCAL({
  UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)]) | VOLUME(volser) [CONTROLLER([xxxxxxx-]xxxxx|name)]) | DDNAME(ddname) [CONTROLLER([xxxxxxx-]xxxxx|name)]) | CONTROLLER([xxxxxxx-]xxxxx|name)])
- [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)])})
- [WAIT_FOR_Definition(Yes|No)]

Note: 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.

1. Starting with Mainframe Enablers 8.4.
Note: Only one of the following can be present: CONTROLLER, LOCAL, or REMOTE. These parameters are mutually exclusive.

Note: The OUTDDNAME, TARGET, VDEV, and GROUP parameters are required, but mutually exclusive. That is, you need to specify one of these parameters, but you cannot specify any of the other three in the same statement.

Required parameters

OUTDDname *(ddname)*

The OUTDDNAME parameter specifies the target volume.

*ddname*

Specifies the DDname of the target volume.

Target *(VOLUME.volser)|UNIT.cuu|SYMDV#.symdv#)*

Note: You can use this parameter only if you install the TF/Clone licensed feature code.

The TARGET parameter defines the target of the snap.

VOLUME *(volser)*

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

UNIT *(cuu)*

The unit address(es) of one or more devices that is online to the host attempting the snap. See “UNIT(cuu)” on page 207.

SYMDV#.symdv#

The internal PowerMax/VMAX device number(s) in the target storage system for the snap operation. See “SYMDV#.symdv#” on page 202.

If you specify SYMDV#, then the UNIT and VOLUME parameters are not allowed. You must, however, include the LOCAL REMOTE, or CONTROLLER parameter.

The OUTDDNAME, TARGET, VDEV, and GROUP parameters are required, but mutually exclusive. That is, you need to specify one of these parameters, but you cannot specify any of the other three in the same statement.

Default

None

Example

None
VDEVice(VOLume(volser) | UNIT(cuu) | SYMDV#(symdv#))

**Note:** This parameter is only available if you install the TF/Snap licensed feature code.

Identifies which virtual device to use as the target.

VOLume(volser)

The volser of the virtual device.

UNIT(cuu)

The unit addresses of one or more virtual devices.

You can specify a single device or a range of devices.

SYMDV#(symdv#)

The internal PowerMax/VMAX device number(s) in the target storage system for the SNAP operation. See “SYMDV#(symdv#)” on page 202.

If SYMDV# is specified, then the UNIT and VOLUME parameters are not allowed. You must include the LOCAL, REMOTE, or CONTROLLER parameter.

The OUTDDNAME, TARGET, VDEV, and GROUP parameters are required, but mutually exclusive. That is, you need to specify one of these parameters, but you cannot specify any of the other three in the same statement.

**Default**

None

**Example 1**

```
//STOPDD DD DISP=SHR,UNIT=3390,VOL=SER=MV3493

* ENTER STOP COMMAND
*
STOP SNAP TO VOLUME (OUTDDNAME (STOPDD))
STOP SNAP TO VOLUME (TARGET (VOLUME (MV3494 )))
STOP SNAP TO VOLUME (TARGET (UNIT (1200))))
STOP SNAP TO VOLUME (TARGET (VOLSER(MV0088) UNIT(1300))))
STOP SNAP TO VOLUME (VDEV (UNIT(A000)))
```

**Example 2**

```
STOP SNAP TO VOLUME (LOCAL (UNIT(8520) RAGROUP(90) - CONTROLLER (0001949-01031) ) - TARGET (SYMDV# (04D0-04D 1))))
```

**Example 3**

This example issues a STOP SNAP to two volumes in a REMOTE storage system not defined in the SCF of the user.

```
* STOP SNAP TO VOLUME ( REMOTE (UNIT(7800) RAGROUP(90) - CONTROLLER (0001949-01031) ) - TARGET (SYMDV# (04D0-04D1 ) ) )
* ```
GROUP(grpname[, grpname,...])

See “GROUP(grpname[, grpname,...])” on page 173.

The OUTDDNAME, TARGET, VDEV, and GROUP parameters are required, but mutually exclusive. That is, you need to specify one of these parameters, but you cannot specify any of the other three in the same statement.

Optional parameters

AUTO_UNBIND_thin_device(Yes|No)

During a STOP SNAP request against a thin device, AUTO_UNBIND_THIN_DEVICE(YES) unbinds the device from any pool after the sessions have been terminated.

Note: The AUTO_UNBIND parameter is only compatible with Enginuity 5876 and 5773.

The AUTO_UNBIND_thin_device parameter has a matching site option, &AUTO_UNBIND_TDEV.

CHECKBCVholdstatus(Yes|No)

See “CHECKBCVholdstatus(Yes|No)” on page 157.

CONTROLLER( [xxxxxxx-]xxxxx | name)

See “CONTROLLER([xxxxxxx-]xxxxx|name)” on page 159.

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

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 175.

The LOCAL parameter is only needed and can only be used if you use the SYMDV# parameter.

NAME(snapshot_name)

The NAME parameter is used to indicate that you want to stop a softlinked snapshot. Using the NAME parameter has the effect of unlink terminating the specified snapshot. Only use the NAME(snapshot_name) parameter if the snapshot was created with the SOFTLINK(YES) option set, or if you are certain that the specified snapshot is softlinked.

REMOTE (RAGROUP(nn.nn.nn.nn) 
{UNIT(cuu) [CONTROLLER([xxxxxxx-]xxxxx|name)] |

1. Starting with Mainframe Enablers 8.4.
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 190.

The REMOTE parameter is only needed and can only be used if you use the
SYMDV# parameter.

WAIT For Definition (Yes|No)

See “WAIT_For Definition(Yes|No)” on page 212.
Grouping commands (TF/Snap)

DEFINE GROUP

Allows you to define a group of SNAP VOLUME and GLOBAL statements which are then stored into a PDS or PDS/E file.

SNAP VOLUME statements are mandatory in DEFINE GROUP while GLOBAL statements are optional.

**Note:** Statements within the DEFINE GROUP definition cannot include a VOLUME(volser) parameter for a target.

Syntax

```
DEFINE GROUP  grpname  [(optional_parameters)]
```

Where *optional_parameters* are as follows.

[DESCRIPTION ('descriptive_text')]

[FORCE(Yes|No)]

[REPLACE(Yes|No)]

**Note:** If a parameter has YES and NO keywords, you can substitute ON for YES and OFF for 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.

**Default value**

None

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. “FORCE(Yes|No)” on page 172 provides more information.

REPLACE (Yes | No)

The REPLACE parameter indicates 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 192.
DELETE GROUP

Allows you to delete an existing group (that was defined with DEFINE GROUP). The deletion removes the definition from the PDS in which it is stored.

Note that you cannot edit a previously defined group to change its contents. Instead, you must:

1. Delete the group.
2. Redefine the group with DEFINE/REPLACE and a new set of SNAP VOLUME or GLOBAL commands.

Syntax

```
DELETE GROUP grpname [(FORCE(Yes|No))]  
```

Note: If a parameter has YES and NO keywords, you can substitute ON for YES and OFF for 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 command FORCE parameter or by the &FORCE site option. “FORCE(Yes|No)” on page 172 provides more information.

END GROUP

The END GROUP command completes the definition of a group. You must enter an END GROUP after you finish entering the SNAP VOLUME and GLOBAL statements that define the group.

Syntax

```
END GROUP
```
CHAPTER 6
Messages and Error Codes

This chapter describes messages you may receive from TimeFinder/Mirror and provides a comprehensive list of error codes that can be issued by TimeFinder Clone Mainframe Snap Facility.

- TimeFinder/Mirror messages ................................................................. 324
- User abend codes ............................................................................... 325
- DOIO error codes .............................................................................. 325
- PowerMax/VMAX interface error codes ................................................ 326
TimeFinder/Mirror messages

If you are using TimeFinder/Clone Mainframe Snap Facility with PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876, or with Enginuity 5773 and earlier with various RAID operations, you may receive messages from TimeFinder/Mirror.

These messages can include:

- BCVM006E ESTABLISH failed on BCV xxxx, reason code yy
- BCVM009E SPLIT failed on BCV xxxx, reason code yy
- BCVM011E RE-ESTABLISH failed on BCV xxxx, reason code yy
- BCVM013E RESTORE failed on BCV xxxx, reason code yy

When TimeFinder/Mirror ESTABLISH, SPLIT, RE-ESTABLISH, and RESTORE operations are invoked with PowerMaxOS 5978, HYPERMAX OS 5977 or Enginuity 5876, or with Enginuity 5773 and earlier on clone emulation operations, TimeFinder/Clone Mainframe Snap Facility accomplishes the requested function. This is confirmed by the BCVM140I message, as shown in the following example:

    BCVM140I COMMAND PROCESSED VIA TF/CLONE EMULATION

TimeFinder/Mirror reason code conversion

Whenever a TimeFinder/Mirror error occurs during the Snap operation, one of the usual TimeFinder/Mirror error messages is generated (as stated above), followed by message BCVM144I.

Message BCVM144I shows the original TimeFinder Mirror hexadecimal reason code (yy) converted into a decimal number (xxx) that translates to a specific EQCAxxxxE message identifier.

    BCVM144I - REFER TO EQCAxxxxE JOBLOG MESSAGE

See the EQCA message description in this manual for an explanation of the error.

**WARNING**

If the BCVM144I message is not contained in the JOBSTEP LOG, an error occurred where TimeFinder has not been able to generate a message. In this case, contact Dell EMC Customer Support and save as much information a possible to help resolve the issue.

**Note:** You can find descriptions of these messages in the *Mainframe Enablers Message Guide*.

Example

The following is an example of one of those error conditions:

    BCVM004I RE-ESTABLISH REMOTE BCV SYMDEV 07A0 THROUGH 93C8
    BCVM140I COMMAND PROCESSED VIA TF/CLONE EMULATION
    BCVM011E RE-ESTABLISH FAILED ON BCV 07A0, REASON CODE 78
    BCVM144I - REFER TO EQCA120E JOBLOG MESSAGE
    BCVM108E BCV 07A0 LOCK FREE FAILED, RC 001F, RSNC 0000001C
    BCVM108E BCV 07A0 LOCK FREE FAILED, RC 001F, RSNC 0000001C
    BCVM047I ALL CONTROL STATEMENTS PROCESSED, HIGHEST RC 8
Note the reason code of 78 in the BCVM011E message. The reason code of 78 is the hexadecimal equivalent of decimal 120, as in the TimeFinder message EQCA120E. This is confirmed in the next message:

**BCVM144I - REFER TO EQCA120E JOBLOG MESSAGE**

Using this information, you can then find:

**EQCA120E DEVICE nnnn FAILED TO GO READY, RC: xx R0: xx R1: xx**

**Note:** The TimeFinder/Mirror message BCVM108E also gives specific information about this particular problem, but BCVM108E is only displayed when there is relevant extra information available.

**User abend codes**

The following user abend codes are issued in the event an error occurs before the message system is initialized.

**Table 14 Abend codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0001</td>
<td>The SYSOUT DDNAME was not specified in the JCL.</td>
</tr>
<tr>
<td>U0002</td>
<td>The OPEN request failed for the SYSOUT DDNAME.</td>
</tr>
<tr>
<td>U0806</td>
<td>SCF Address space not found.</td>
</tr>
</tbody>
</table>

**DOIO error codes**

**Figure 11 shows the format for DOIO error codes.**

![DOIO error code format](image-url)
PowerMax/VMAX interface error codes

The following tables contain reason codes related to TimeFinder operations. Table 15 contains return and reason codes for TimeFinder. Table 16 on page 328 contains error codes for the EXTENTS program:

Table 15 PowerMax/VMAX interface error codes (page 1 of 2)

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1701</td>
<td>The CRC is invalid in an individual extent in the extent track. A request to obtain the extents track pointer failed because the device number is invalid.</td>
</tr>
<tr>
<td>1702</td>
<td>The destination device number in an extent entry is not a valid destination device. A request to obtain the extents track pointer failed because the device count is invalid.</td>
</tr>
<tr>
<td>1703</td>
<td>The destination device number in an extent entry is not a valid BCV device.</td>
</tr>
<tr>
<td>1704</td>
<td>The beginning CCBH in an extent entry is not valid for the source device.</td>
</tr>
<tr>
<td>1705</td>
<td>The beginning CCBH in an extent entry is not valid for the target device.</td>
</tr>
<tr>
<td>1706</td>
<td>The flag setting in an extent entry is not valid or the feature is not available at the installed operating environment level.</td>
</tr>
<tr>
<td>1707</td>
<td>The number of tracks to copy is not valid. Either the source CCBH plus the number of tracks to copy exceeds the capacity of the source device, or the target CCBH plus the number of tracks to copy exceeds the capacity of the target device.</td>
</tr>
<tr>
<td>1708</td>
<td>Protection is not on for the selected extent entry.</td>
</tr>
<tr>
<td>1709</td>
<td>The session ID in an extent entry is not valid.</td>
</tr>
<tr>
<td>1710</td>
<td>A timeout occurred.</td>
</tr>
<tr>
<td>1711</td>
<td>The request to remove a session failed because the session ID has not been established.</td>
</tr>
<tr>
<td>1712</td>
<td>The request to remove a session failed because the session selected is not a SNAP session.</td>
</tr>
<tr>
<td>1713</td>
<td>The request to remove a session has failed.</td>
</tr>
<tr>
<td>1721</td>
<td>The request to identify the extent track failed. The CCBH is invalid for the device.</td>
</tr>
<tr>
<td>1722</td>
<td>The request to identify the extent track failed. The device is a BCV device.</td>
</tr>
<tr>
<td>1723</td>
<td>The request to identify the extent track failed. The extent track is not in a Perma-Cache slot.</td>
</tr>
<tr>
<td>1724</td>
<td>The request to identify the extent track failed. The extent track is not in cache.</td>
</tr>
<tr>
<td>1725</td>
<td>The request to identify the extent track failed. The PowerMax/VMAX number is invalid.</td>
</tr>
<tr>
<td>1726</td>
<td>The request to identify the extent track failed. The extent track has no record one.</td>
</tr>
<tr>
<td>1727</td>
<td>Snap to destination R2 device is disabled.</td>
</tr>
<tr>
<td>1731</td>
<td>The request to establish or remove an extent failed. The extent track address is invalid.</td>
</tr>
<tr>
<td>1732</td>
<td>The request to establish or remove an extent failed. The extent track is not in a Perma-Cache slot.</td>
</tr>
</tbody>
</table>
Table 15  PowerMax/VMAX interface error codes (page 2 of 2)

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1733</td>
<td>The request to establish or remove an extent failed. The destination device in an extent entry is not a BCV device.</td>
</tr>
<tr>
<td>1735</td>
<td>The request to establish or remove an extent failed. The extent track slot is invalid.</td>
</tr>
<tr>
<td>1736</td>
<td>The request to establish or remove an extent failed. The start extent entry is invalid.</td>
</tr>
<tr>
<td>1737</td>
<td>The request to establish or remove an extent failed. The end extent entry is invalid.</td>
</tr>
<tr>
<td>1738</td>
<td>The request to establish or remove an extent failed. The number of extents to process exceeds the maximum permissible.</td>
</tr>
<tr>
<td>1739</td>
<td>The request to establish or remove an extent failed. An invalid extent entry was detected.</td>
</tr>
<tr>
<td>173A</td>
<td>The request to establish or remove an extent failed. The GST queue is full.</td>
</tr>
<tr>
<td>173B</td>
<td>The request to establish or remove an extent failed. More than one target device is specified in a single request.</td>
</tr>
<tr>
<td>173C</td>
<td>The request to establish or remove an extent failed. Unable to obtain the lock for the destination device.</td>
</tr>
<tr>
<td>173D</td>
<td>The request to establish or remove an extent failed. The destination device is not ready.</td>
</tr>
<tr>
<td>173E</td>
<td>The request to establish or remove an extent failed. The destination and the source devices are not of the same emulation type.</td>
</tr>
<tr>
<td>1741</td>
<td>Device is not a BCV.</td>
</tr>
<tr>
<td>1742</td>
<td>Device already set or released.</td>
</tr>
<tr>
<td>1743</td>
<td>BCV device is established.</td>
</tr>
<tr>
<td>1744</td>
<td>BCV has active TimeFinder sessions.</td>
</tr>
<tr>
<td>1746</td>
<td>Device has active concurrent copy session.</td>
</tr>
<tr>
<td>1751</td>
<td>The request to remove protection failed. The command was issued to a non-BCV device.</td>
</tr>
<tr>
<td>1752</td>
<td>The request to remove protection failed. The extent indicated is not a valid extent.</td>
</tr>
<tr>
<td>1761</td>
<td>The request to establish an extent failed. The maximum of 16 sessions has been exceeded.</td>
</tr>
<tr>
<td>1779</td>
<td>Displayed when an establish occurs against a thin pool with insufficient available tracks.</td>
</tr>
<tr>
<td>1792</td>
<td>DA error.</td>
</tr>
<tr>
<td>1799</td>
<td>The request has failed. Cannot lock extents track.</td>
</tr>
<tr>
<td>179A</td>
<td>The request has failed. GST (Global Special Task) call failed.</td>
</tr>
<tr>
<td>17FF</td>
<td>DA failure/timeout on syscall.</td>
</tr>
</tbody>
</table>
### Table 16 EXTENTS error codes

<table>
<thead>
<tr>
<th>R15</th>
<th>Reason</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Successful</td>
</tr>
<tr>
<td>4 or 8</td>
<td>1</td>
<td>Illegal mask specified</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>No matching datasets found</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Unknown function code</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Overflow, unable to return all matching datasets.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>UCB not found</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Dataset not found on volume</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>DSCB not type 1 or type 4</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Tracks allocated = 0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Extents not available due to HSM migrate</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>Caller not APF authorized</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>Extents program logic error</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>Entry type not GDG base</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>Entry name and catalog name match</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>Volume serial not supplied</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>DEVTYPE indicates unsupported device type geometry</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>DEVTYPE failed</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Catalog management return code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Locate failed - the reason code contains the locate return code</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Dataset is not catalogued</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>EXTWA Failed validation</td>
</tr>
</tbody>
</table>
This appendix discusses DFDSS command support through EMCDSSU.

- Introduction ................................................................. 330
- Installation considerations .............................................. 330
- EMCDSSU parameters .................................................. 331
Introduction

TimeFinder supports the invoking of TimeFinder processing within a storage system in response to a DFDSS COPY DATASET and COPY FULL statements for dataset or volume copies.

This support is provided through the module, EMCDSSU, which handles the SYSIN and SYSPRINT files for the DFDSS syntax.

EMCDSSU accepts DFDSS supported syntax and processes COPY statements by invoking TimeFinder to create copies of datasets or volumes with a storage system.

There are four categories of EMCDSSU COPY parameters:

- Fully supported.
- Partially supported; that is, supported for one action and not another or for one subparameter and not more than one.
- Not supported.
- Ignored.

For a command to be processed by EMCDSSU, all of the parameters on that command must be either supported or ignored.

- If a parameter is supported, EMCDSSU processes the command and that parameter.
- If a parameter is one that is ignored, then EMCDSSU processes the command, but does not process that parameter.
- If a parameter is one that is not supported, then the command with that parameter is not processed by EMCDSSU. Instead, it is passed to the module ADRDSSU for processing.

Installation considerations

You can install EMCDSSU as a replacement for ADRDSSU. In this case, you must rename ADRDSSU to IBMDSSU and rename EMCDSSU to ADRDSSU.

**Note:** IBMDSSU is the only valid rename or alias for ADRDSSU.

Keep in mind that this approach has additional considerations for the application of future maintenance to the IBM ADRDSSU module.
### EMCDSSU parameters

*Table 17* lists the EMCDSSU parameters and their categories. A Yes in a column indicates that the parameter is in that category. If a parameter is partially supported, the supported column lists the supported form(s) and the unsupported column lists the unsupported forms.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fully supported?</th>
<th>Partially supported?</th>
<th>Not supported?</th>
<th>Ignored?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMINISTRATOR</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALLDATA</td>
<td>ALLDATA(+)</td>
<td>Yes</td>
<td>ALLDATA(dsn)</td>
<td></td>
</tr>
<tr>
<td>ALLEXCP</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>AUTORELBLOCKADDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYPASSACS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANCELERROR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATALOG</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGCREATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHECKVTOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICSVRBACKUP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONCURRENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPYVOLID</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPVOLUME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATASET</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBUG</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>DELETE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMPCONDITIONING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNALLOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCLUDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FASTREPLICATION</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>FCFREEZE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCNOCOPY</td>
<td>COPY VOLUME</td>
<td>Yes</td>
<td>COPY DATASET</td>
<td></td>
</tr>
<tr>
<td>FCTOPPRCPRimary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILTERDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORCE</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>FORCECP</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Parameter</td>
<td>Fully supported?</td>
<td>Partially supported?</td>
<td>Not supported?</td>
<td>Ignored?</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>----------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>FREESPACE</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>FULL</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCAT</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>INCLUDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDDNAME</td>
<td>COPY VOLUME</td>
<td>Yes</td>
<td></td>
<td>COPY DATASET</td>
</tr>
<tr>
<td>INDDNAME COPY VOLUME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDDNAME COPY VOLUME (single volume)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGINDNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGINDYNAM</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>MAKEMULTI</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>MENTITY</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOPACKING</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>NOTIFYCONCURRENT</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>NULLMGMTCLAS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>NULLSTORCLAS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ONLYINCAT</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>OUTDDNAME</td>
<td>COPY VOLUME</td>
<td>Yes</td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTDDNAME COPY VOLUME</td>
<td></td>
<td></td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTDDNAME COPY VOLUME (single volume)</td>
<td></td>
<td></td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTDYNAM</td>
<td>COPY VOLUME</td>
<td>Yes</td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTDYNAM COPY VOLUME</td>
<td></td>
<td></td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTDYNAM COPY VOLUME (single volume)</td>
<td></td>
<td></td>
<td>COPY VOLUME (multiple volumes)</td>
<td></td>
</tr>
<tr>
<td>OUTTRACKS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PASSWORD</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PERCENTUTILIZED</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PROCESS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PURGE</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READIOPACING</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>REBLOCK</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>RECATALOG</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>RELBLOCKADDRESS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>RENAMEUNCONDITIONALY</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
Table 17  EMCDSSU parameters  (page 3 of 3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fully supported?</th>
<th>Partially supported?</th>
<th>Not supported?</th>
<th>Ignored?</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPLACE</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACEUNCONDITIONAL</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>SELECTMULTI</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARE</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPHERE</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STORCLAS</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STORGRP</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TGTALLOC</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TGTGDS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TOLERATE</td>
<td>TOLERATE</td>
<td>TOLERATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(ENGFAILURE)</td>
<td>(IOERROR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKS</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TTRADDRESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNCATALOG</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>VOLCOUNT</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>WAIT</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>WRITECHECK</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
DFDSS COPY Command Support
APPENDIX B
SMF Record Layout
(Mainframe Enablers 8.4, 8.3, and 8.2)

This appendix shows the SMF record layout for Mainframe Enablers 8.3 and 8.2.

- SMF record layout ................................................................. 336
- SMF record sub-sections and TimeFinder actions.............................. 354
SMF record layout

The volume and variety of information in the SMF records enables sites to produce many types of analysis and summary reports. By keeping historical SMF data and determining trends, an installation can evaluate changes in the configuration, workload, or job scheduling procedures. Similarly, an installation can use SMF data to determine where system resources are wasted because of problems, such as inefficient operational procedures or poor programming conventions.

Note: See the IBM Manual *z/OS MVS System Management Facilities (SMF)*, section *Using SMF Macros*, subsection *SMFWTM—Writing SMF Records* for more information about the value and use of the SMF record.

SAMPLIB members ESNASMFJ and ESNAWSMF are JCL and REXX examples to use for processing TF/Clone Mainframe Snap Facility SMF records.

### Table 18  SNSMFRCD - STANDARD SMF RECORD HEADER

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SNSMFLEN</td>
<td>2</td>
<td>Binary</td>
<td>SMF record length (maximum size of 32,756). This field and the next field (total of four bytes) form the record descriptor word (RDW). The first two bytes (this field) must contain the logical record length including the RDW.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SNSMFSEG</td>
<td>2</td>
<td>Binary</td>
<td>Segment descriptor provided by SMF. Initialize with zeros.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SNSMFSYS</td>
<td>1</td>
<td>Binary</td>
<td>System indicator. Bit meanings are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bits 0, 1 and 2 are reserved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX'1E' - MVS/SP version 4 and later. Bits 3, 4, 5, and 6 are on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX'0E' - MVS/SP version 3. Bits 4, 5, and 6 are on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX'06' - MVS/SP version 2. Bits 5 and 6 are on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX'02' - VS2. Bit 6 is on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bit 7 is reserved.</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>SNSMFRC#</td>
<td>1</td>
<td>Binary</td>
<td>SMF record number. Can be set using the SMFRID site option.</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>SNSMFTME</td>
<td>4</td>
<td>Binary</td>
<td>Time in hundredths of seconds.</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>SNSMFDATE</td>
<td>4</td>
<td>Packed</td>
<td>Date in packed decimal format OCYLYDDDF, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OC - is the century (00= 1900-1999, 01= 2000-2099)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YY - is the last two digits of the year</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DDD - is the day of the year</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F - allows for unpacking.</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SNSMFSID</td>
<td>4</td>
<td>EBCDIC</td>
<td>System identification (SID).</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>SNSMFSKY</td>
<td>1</td>
<td>Binary</td>
<td>Subtype = 0.</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>SNSMFBAD</td>
<td>1</td>
<td>Binary</td>
<td>Indicates whether the SMF record is incomplete. An SMF record is incomplete if this field contains 1.</td>
</tr>
</tbody>
</table>
### Table 18  SNSMFRCRD - STANDARD SMF RECORD HEADER

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>14</td>
<td>SNSMFSG#</td>
<td>2</td>
<td>Binary</td>
<td>Number of segments in a record.</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td></td>
<td>10</td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFHDR_LEN</td>
<td></td>
<td></td>
<td>Equate 32 (HEX'0020') = length of the SMF record header.</td>
</tr>
</tbody>
</table>

### Table 19  SNSMFCMN - COMMON SEGMENT PORTION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SCMSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SCMSGID#</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### Table 20  SNSMFID - IDENTIFICATION SEGMENT

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SIDSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFID_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SIDSGID#</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFID#</td>
<td></td>
<td></td>
<td>1 = Identification segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td>Equate</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SIDJOBNM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Job name.</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SIDPROCS</td>
<td>8</td>
<td>EBCDIC</td>
<td>Proc step name.</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>SIDSTPNM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Step name.</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>SIDPGMNM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Program name.</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>SIDVERLV</td>
<td>6</td>
<td>EBCDIC</td>
<td>Program version in the VVLLRR format, where: VV - version; LL - level; RR - revision</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFID_LEN</td>
<td></td>
<td></td>
<td>Equate 42 (HEX'002A') = length of the identification segment.</td>
</tr>
</tbody>
</table>

### Table 21  SNSMFREQ - REQUEST INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SRQSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFREQ_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SRQSGID#</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFREQ#</td>
<td></td>
<td></td>
<td>2 = Request Information segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td>Equate</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
### Table 21 SNSMFREQ - REQUEST INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>SRQTYPE</td>
<td>2</td>
<td>Binary</td>
<td>Request type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#DSN</td>
<td></td>
<td>Equate</td>
<td>HEX'0001' = COPY DATASET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#VOL</td>
<td></td>
<td>Equate</td>
<td>HEX'0002' = COPY VOLUME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#RST</td>
<td></td>
<td>Equate</td>
<td>HEX'0003' = DESTROY EXTENT TRACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#CLN</td>
<td></td>
<td>Equate</td>
<td>HEX'0004' = CLEANUP EXTENT TRACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#DBG</td>
<td></td>
<td>Equate</td>
<td>HEX'0005' = DEBUG DATASET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#STD</td>
<td></td>
<td>Equate</td>
<td>HEX'0006' = STOP SNAP TO DATASET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#STV</td>
<td></td>
<td>Equate</td>
<td>HEX'0007' = STOP SNAP TO VOLUME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#RVD</td>
<td></td>
<td>Equate</td>
<td>HEX'0008' = RESTORE VIRTUAL DEVICE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#ACT</td>
<td></td>
<td>Equate</td>
<td>HEX'0009' = ACTIVATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#QDS</td>
<td></td>
<td>Equate</td>
<td>HEX'000A' = QUERY DATASET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#QSD</td>
<td></td>
<td>Equate</td>
<td>HEX'000B' = QUERY SAVEDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#QVD</td>
<td></td>
<td>Equate</td>
<td>HEX'000C' = QUERY VDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#CFG</td>
<td></td>
<td>Equate</td>
<td>HEX'000D' = CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#QVL</td>
<td></td>
<td>Equate</td>
<td>HEX'000E' = QUERY VOLUME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#SER</td>
<td></td>
<td>Equate</td>
<td>HEX'000F' = SERIAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#PAR</td>
<td></td>
<td>Equate</td>
<td>HEX'0010' = PARALLEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#CPL</td>
<td></td>
<td>Equate</td>
<td>HEX'0011' = CONFIGPOOL (SEE SRQSACTN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#QGR</td>
<td></td>
<td>Equate</td>
<td>HEX'0012' = QUERY GROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#FGR</td>
<td></td>
<td>Equate</td>
<td>HEX'0013' = DEFINE GROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#EGR</td>
<td></td>
<td>Equate</td>
<td>HEX'0014' = END GROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#DGR</td>
<td></td>
<td>Equate</td>
<td>HEX'0015' = DELETE GROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#GRP</td>
<td></td>
<td>Equate</td>
<td>HEX'0016' = EXTERNAL GRP RQST (SEE SRQSACTN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#GQL</td>
<td></td>
<td>Equate</td>
<td>HEX'0017' = QUERY GLOBAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#CDS</td>
<td></td>
<td>Equate</td>
<td>HEX'0018' = COMPARE DATASET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQTYPE#CVL</td>
<td></td>
<td>Equate</td>
<td>HEX'0019' = COMPARE VOLUME</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>SRQSACTN</td>
<td>2</td>
<td>Binary</td>
<td>Sub-action type. This is only valid when SRQTYPE is SRQTYPE#CPL or SRQTYPE#GRP. If the SRQTYPE contains SRQTYPE#CPL, then use the equates below. If the SRQTYPE contains SRQTYPE#GRP, use the equates defined for the SRQTYPE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_ADD</td>
<td></td>
<td>Equate</td>
<td>HEX'0001' = ADD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_CREATE</td>
<td></td>
<td>Equate</td>
<td>HEX'0002' = CREATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_DELETE</td>
<td></td>
<td>Equate</td>
<td>HEX'0003' = DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_DISABLE</td>
<td></td>
<td>Equate</td>
<td>HEX'0004' = DISABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_DISPLAY</td>
<td></td>
<td>Equate</td>
<td>HEX'0005' = DISPLAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_ENABLE</td>
<td></td>
<td>Equate</td>
<td>HEX'0006' = ENABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_REMOVE</td>
<td></td>
<td>Equate</td>
<td>HEX'0007' = REMOVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_DRAIN</td>
<td></td>
<td>Equate</td>
<td>HEX'0008' = DRAIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRQSACTN_UNDRAIN</td>
<td></td>
<td>Equate</td>
<td>HEX'0009' = UNDRAIN</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>SRQSTMT#</td>
<td>4</td>
<td>Binary</td>
<td>Statement number.</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SRQRC</td>
<td>2</td>
<td>Binary</td>
<td>Highest return code.</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SRQRE</td>
<td>2</td>
<td>Binary</td>
<td>Associated reason code.</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>SRQSTIME</td>
<td>8</td>
<td>Binary + Packed</td>
<td>Start time. The first 4 bytes store the time in hundredths of seconds, the next 4 bytes store the date in packed decimal format (OCYYDDDF).</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>SRQETIME</td>
<td>8</td>
<td>Binary + Packed</td>
<td>End time. The first 4 bytes store the time in hundredths of seconds, the next 4 bytes store the date in packed decimal format (OCYYDDDF).</td>
</tr>
</tbody>
</table>
### Table 21 SNSMFREQ - REQUEST INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>20</td>
<td>SRQPSTIM</td>
<td>8</td>
<td>Binary</td>
<td>Parse start time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>SRQPETIM</td>
<td>8</td>
<td>Binary</td>
<td>Parse end time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>SRQESTIM</td>
<td>8</td>
<td>Binary</td>
<td>Execution start time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
<tr>
<td>56</td>
<td>38</td>
<td>SRQSETIM</td>
<td>8</td>
<td>Binary</td>
<td>Execution suspend time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>SRQRSTIM</td>
<td>8</td>
<td>Binary</td>
<td>Execution resume time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
<tr>
<td>72</td>
<td>48</td>
<td>SRQEETIM</td>
<td>8</td>
<td>Binary</td>
<td>Execution end time. This is an unsigned 64-bit fixed-point number where bit 51= 1 microsecond (TOD contents).</td>
</tr>
</tbody>
</table>

**SNSMFREQ_LEN**

Equate 80 (HEX'0050') = length of the Request Information segment.

### Table 22 SNSMOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SOPSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMOPT_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SOPSIGID</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 3. 3 = Options Information segment ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFOP#</td>
<td></td>
<td>Equate</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Reserved.</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SOPFLG01</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_WILD</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - SOURCE IS WILD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_WILD</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - TARGET IS WILD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXCLUDE_HERE</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - EXCLUDE IS PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RELATE_HERE</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - RELATE WAS DERIVED FROM SOURCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RELATE_MADE</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - SPHERE (YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SPHERE</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VDEV(UNIT()/VOL()) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXECUTED</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - REQUEST WAS EXECUTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ERRDISP_KEEP</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - ERRDISP(KEEP)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>SOPFLG02</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PERMINDIRECT</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - PERMANENTINDIRECT(Y) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NO_BACKGRND</td>
<td></td>
<td>Equate</td>
<td>HEX'40' -BACKGROUNDCOPY(N) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_WAIT4COMPLET</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - WAITFORCOMPLETION(MSG) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_UCODE_FULL</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - SNAP &quot;FULL&quot; VOLUME REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SYMM_CYL</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - SYMMETRIX_CYLINDER(Y) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VIRTUAL</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VIRTUAL_DEVICE(Y) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VDEV(UNIT()/VOL()) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MIG_IGNORE</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - MIGRATE(RECALL-IGNORE) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_R1R2SYNC</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - WAITFORCOMPLETION(R1R2SYNC) SPECIFIED</td>
</tr>
</tbody>
</table>
### Table 22 SNSMFOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>SOPFLG03</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NEED_REFVTOC</td>
<td></td>
<td></td>
<td>Equate HEX'80' - REFVTOC NEEDED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CSMS_DATA</td>
<td></td>
<td></td>
<td>Equate HEX'40' - COPYSMS(DATACLAS) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CSMS_MGMT</td>
<td></td>
<td></td>
<td>Equate HEX'20' - COPYSMS(MGMTCLASS) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CSMS_STG</td>
<td></td>
<td></td>
<td>Equate HEX'10' - COPYSMS(STGCLASS) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MIG_PURGE</td>
<td></td>
<td></td>
<td>Equate HEX'08' - MIGRATE(PURGE) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MIG_RECALL</td>
<td></td>
<td></td>
<td>Equate HEX'04' - MIGRATE(RECALL-YES) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VERIFY</td>
<td></td>
<td></td>
<td>Equate HEX'02' - VERIFY(YES) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXAMINE</td>
<td></td>
<td></td>
<td>Equate HEX'01' - EXAMINE(YES) SPECIFIED</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>SOPFLG04</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CONSIST</td>
<td></td>
<td></td>
<td>Equate HEX'80' - CONSISTENT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VIBBLD</td>
<td></td>
<td></td>
<td>Equate HEX'40' - BUILD_VTOCIX(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VALIDATE</td>
<td></td>
<td></td>
<td>Equate HEX'20' - VALIDATE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VSAMFAIL</td>
<td></td>
<td></td>
<td>Equate HEX'10' - TOLERATE VSAMENG FAILURE (YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CHK_BCVHOLD</td>
<td></td>
<td></td>
<td>Equate HEX'08' - CHECKBCVHOLDSTATUS(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SCRATCH</td>
<td></td>
<td></td>
<td>Equate HEX'04' - SCRATCH(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VCLOSE</td>
<td></td>
<td></td>
<td>Equate HEX'02' - VCLOSE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DFDSS_ADMIN</td>
<td></td>
<td></td>
<td>Equate HEX'01' - DFDSS_ADMIN(YES)</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>SOPFLG05</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 5 (DDNAME settings)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_DSNAME</td>
<td></td>
<td></td>
<td>Equate HEX'80' - SOURCE DSNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_DDNAME</td>
<td></td>
<td></td>
<td>Equate HEX'40' - SOURCE DDNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_DSNAME</td>
<td></td>
<td></td>
<td>Equate HEX'20' - TARGET DSNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_DDNAME</td>
<td></td>
<td></td>
<td>Equate HEX'10' - TARGET DDNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_DD_JCL</td>
<td></td>
<td></td>
<td>Equate HEX'08' - SOURCE DDNAME ORIGINALLY PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_DD_JCL</td>
<td></td>
<td></td>
<td>Equate HEX'04' - TARGET DDNAME ORIGINALLY PRESENT</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>SOPFLG06</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REPLACE</td>
<td></td>
<td></td>
<td>Equate HEX'80' - REPLACE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REUSE</td>
<td></td>
<td></td>
<td>Equate HEX'40' - REUSE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_COPYVOLID</td>
<td></td>
<td></td>
<td>Equate HEX'20' - COPYVOLID(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_FORCE</td>
<td></td>
<td></td>
<td>Equate HEX'10' - FORCE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ENQFAIL</td>
<td></td>
<td></td>
<td>Equate HEX'08' - TOLERATE ENQ FAILURE (YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ALLOFAIL</td>
<td></td>
<td></td>
<td>Equate HEX'04' - TOLERATE ALLOCATION FAILURE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_COPYFAIL</td>
<td></td>
<td></td>
<td>Equate HEX'02' - TOLERATE COPY FAILURE (YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TRUNC</td>
<td></td>
<td></td>
<td>Equate HEX'01' - TOLERATE TRUNCATION (YES)</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>SOPFLG07</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CATALOG</td>
<td></td>
<td></td>
<td>Equate HEX'80' - CATALOG(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ENGWAIT</td>
<td></td>
<td></td>
<td>Equate HEX'40' - ENGWAIT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REFVTOC</td>
<td></td>
<td></td>
<td>Equate HEX'20' - REFVTOC(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CONDVOL_ALL</td>
<td></td>
<td></td>
<td>Equate HEX'10' - CONDITIONVOLUME(ALL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_BCVONLY</td>
<td></td>
<td></td>
<td>Equate HEX'08' - BCVONLY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CHKONLIN</td>
<td></td>
<td></td>
<td>Equate HEX'04' - CHECKONLINEPATHSTATUS(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DFDSS_CC</td>
<td></td>
<td></td>
<td>Equate HEX'02' - DFDSS_CC(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RECALC_FREE</td>
<td></td>
<td></td>
<td>Equate HEX'01' - RECALCULATE_FREESPACE(YES)</td>
</tr>
</tbody>
</table>
### Table 22: SNSMOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>B</td>
<td>SOPFLG08</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRCDSORG_VS</td>
<td></td>
<td>Equate</td>
<td>HEX‘80’ - INDSORG (VS) PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGDSORG_PS</td>
<td></td>
<td>Equate</td>
<td>HEX‘40’ - OUTDSORG (PS) PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SPACECYL</td>
<td></td>
<td>Equate</td>
<td>HEX‘20’ - NONVSAMSPACE (CYL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SPACETRK</td>
<td></td>
<td>Equate</td>
<td>HEX‘10’ - NONVSAMSPACE (TRK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DATACYL</td>
<td></td>
<td>Equate</td>
<td>HEX‘08’ - DATASPACE (CYL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DATATRK</td>
<td></td>
<td>Equate</td>
<td>HEX‘04’ - DATASPACE (TRK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_INDEXCYL</td>
<td></td>
<td>Equate</td>
<td>HEX‘02’ - INDEXSPACE (CYL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_INDEXTRK</td>
<td></td>
<td>Equate</td>
<td>HEX‘01’ - INDEXSPACE (TRK)</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SOPFLG09</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NOTIFY_DATASET</td>
<td></td>
<td>Equate</td>
<td>HEX‘80’ - NOTIFY(DATASET)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NOTIFY_JOB</td>
<td></td>
<td>Equate</td>
<td>HEX‘40’ - NOTIFY(JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NOTIFY_SNAP</td>
<td></td>
<td>Equate</td>
<td>HEX‘20’ - NOTIFY(SNAP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NOTIFY_STEP</td>
<td></td>
<td>Equate</td>
<td>HEX‘10’ - NOTIFY(STEP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VARYOFF</td>
<td></td>
<td>Equate</td>
<td>HEX‘08’ - VARY_OFFLINE(NEVER)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_FREESPC</td>
<td></td>
<td>Equate</td>
<td>HEX‘04’ - FREESPACE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ORDER_NAME</td>
<td></td>
<td>Equate</td>
<td>HEX‘02’ - ORDER(NAME)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ORDER_SIZE</td>
<td></td>
<td>Equate</td>
<td>HEX‘01’ - ORDER(SIZE)</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>SOPFLG10</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTALLOC_YES</td>
<td></td>
<td>Equate</td>
<td>HEX‘80’ - EXTENT_ALLOCATION(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTALLOC_NO</td>
<td></td>
<td>Equate</td>
<td>HEX‘40’ - EXTENT_ALLOCATION(NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTXPAND</td>
<td></td>
<td>Equate</td>
<td>HEX‘20’ - EXTENT_EXPAND(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DIFF_VOL</td>
<td></td>
<td>Equate</td>
<td>HEX‘02’ - DIFFERENTIAL(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DISBAND</td>
<td></td>
<td>Equate</td>
<td>HEX‘01’ - DISBAND(YES)</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SOPFLG11</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONE_EMCSNAP</td>
<td></td>
<td>Equate</td>
<td>HEX‘80’ - EMCSNAP USED FOR COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONE_EMCCOPY</td>
<td></td>
<td>Equate</td>
<td>HEX‘40’ - EMCCOPY USED FOR COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONE_IBMMSNAP</td>
<td></td>
<td>Equate</td>
<td>HEX‘20’ - IBMMSNAP USED FOR COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONE_IBMFLUSH</td>
<td></td>
<td>Equate</td>
<td>HEX‘10’ - IBM FLASH COPY USED FOR COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONEUTILITY</td>
<td></td>
<td>Equate</td>
<td>HEX‘08’ - UTILITY PROGRAM USED FOR COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DONE_EMCFLASH</td>
<td></td>
<td>Equate</td>
<td>HEX‘04’ - EMMCFLASH USED FOR COPY</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>SOPFLG12</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_COPY_ERROR</td>
<td></td>
<td>Equate</td>
<td>HEX‘80’ - COPY FAILED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REUSEFAIL</td>
<td></td>
<td>Equate</td>
<td>HEX‘40’ - TOLERATE REUSE FAILURE (YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CHKONL_NEVER</td>
<td></td>
<td>Equate</td>
<td>HEX‘20’ - CHECKONLINEPATHSTATUS(NEVER)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MSG_DISPLAY</td>
<td></td>
<td>Equate</td>
<td>HEX‘10’ - MESSAGE(DISPLAY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MSG_PROMPT</td>
<td></td>
<td>Equate</td>
<td>HEX‘08’ - MESSAGE(PROMPT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_QRY_CNTL</td>
<td></td>
<td>Equate</td>
<td>HEX‘04’ - CONTROLLER(#) SUPPLIED FOR QUERY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CONDVOL_DUMP</td>
<td></td>
<td>Equate</td>
<td>HEX‘02’ - CONDITIONVOLUME(DUMP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTO_EXPAND</td>
<td></td>
<td>Equate</td>
<td>HEX‘01’ - REUSE_AUTO_EXPAND(YES)</td>
</tr>
</tbody>
</table>
### SMF Record Layout (Mainframe Enablers 8.4, 8.3, and 8.2)

#### Table 22  SNSMFOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>10</td>
<td>SOPFLG13</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VARYON</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - VARY_ONLINE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VARYON_AUTO</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - VARY_ONLINE(AUTO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VDEVFREE</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - VDEV(FREE) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ADMIN</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - ADMINISTRATOR(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ALLO_UNUSED</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - ALLOCATE_UNUSED_SPACE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SNAP_UNUSED</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - SNAP_UNUSED_SPACE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RELEASE_YES</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - RELEASE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RELEASE_NO</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - RELEASE(NO)</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>SOPFLG14</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_READY_YES</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - READY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_READY_NO</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - READY(NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SAVEDEV_NR</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - SAVEDEV_FULL(NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MOVR_ACTIVE</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - DATAMOVER ACTIVE FOR REQUEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PARALLEL_REQ</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - PARALLEL_REQUESTED</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>SOPFLG15</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PERSIST</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - RESTORE PERSISTENT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTORLSE</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - AUTOMATIC_RELEASE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTODEAL</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - AUTOMATIC_DEALLOC(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RIFULLCOPY</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - RIFULLCOPYONLY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTOCLN</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - AUTOMATIC_CLEANUP(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DIFF_DSN</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - DIFFERENTIAL_DATASET(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MODE_COPY</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - MODE(COPY) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MODE_NOCOPY</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - MODE(NOCOPY) REQUESTED</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>SOPFLG16</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VOL_LEVEL</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - VOLUME LEVEL REQUEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DSN_LEVEL</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - DATASET_LEVEL REQUEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CTRL_LEVEL</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - CONTROLLER LEVEL REQUEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_COLLAPSE_NV</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - COLLAPSE_DATASET_EXTENTS(NV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_COLLAPSE_VS</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - COLLAPSE_DATASET_EXTENTS(VS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PRECOPY</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - PRECOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DUMPTRK</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - DUMP_TRACK_CONTENTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MGBRENABLE</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - MEMBERSTATE(ENABLE)</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>SOPFLG17</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_SYMDV</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - SOURCE SYMDV# PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_SYMDV</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - TARGET SYMDV# PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TERMSESS</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - TERMINATE_SESSION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SESSLIST</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - SESSION_LIST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RMT_DDNAME</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - REMOTE DDNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RMT_DD_JCL</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - REMOTE DDNAME ORIGINAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LCL_DDNAME</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - LOCAL DDNAME PRESENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LCL_DD_JCL</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - LOCAL DDNAME ORIGINAL</td>
</tr>
<tr>
<td>Offset (DEC)</td>
<td>Offset (HEX)</td>
<td>Name</td>
<td>Length</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-----------------------</td>
<td>--------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>SOPFLG18</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DEV_NM</td>
<td></td>
<td></td>
<td>HEX'80' - DEV(N:M) SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFA_WARN</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - SRDFA_R2_SYNC(WARNING)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFA_R1R2SYNC</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - SRDFA_R2_SYNC(R1R2SYNC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_ALLOW</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - SRDFS_R1 ALLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_DMOVR</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - SRDFS_R1 DMOVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_PHYSC</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - SRDFS_R1 PHYSIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_WARN_R1</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - WARNING MESSAGE ISSUED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_WARN_R1</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - WARNING MESSAGE ISSUED</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>SOPFLG19</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_CKD</td>
<td></td>
<td></td>
<td>HEX'80' - QUERY VOLUME CKD OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_FBA</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - QUERY VOLUME FBA OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_NOTREADY</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - QUERY VOLUME NOTREADY OPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_READY</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - QUERY VOLUME READY OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_SAVEDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - QUERY VOLUME SAVEDEV OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_VDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - QUERY VOLUME VDEV OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SESSDETL</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - SESSION_LIST W/DATADETAIL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_SIZE</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - QUERY VOLUME SIZE OPTION</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>SOPFLG20</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VEROPEN</td>
<td></td>
<td></td>
<td>HEX'80' - VERIFY_OPEN_SOURCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ENQ_BAD</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - REMEMBER - ENQ FAILURE OCCURRED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VSAMENQ_BAD</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - REMEMBER - VSAMENQ FAILURE OCCURRED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CLEANDIFF</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - CLEANUP_DIFFERENTIAL(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_UCODE_FULL_NOWAY</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - USE EXTENT UCODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRCLIST</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - SOURCE_VOLUME_LIST USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SMS_PASS</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - SMS_PASS_VOLUMES(YES)</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>SOPFLG21</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFAR1_ALLOW</td>
<td></td>
<td></td>
<td>HEX'80' - ALLOW SRDFAR1 FOR TARGET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFAR1_DMOVR</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - ALLOW SRDFAR1 W/DATAMOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFAR1_PHYSC</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - ALLOW SRDFAR1 W/DATAMOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_ALLOW</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - ALLOW SRDFS_R1 FOR TARGET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_DMOVR</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - ALLOW SRDFS_R1 W/DATAMOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_R1_PHYSC</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - ALLOW SRDFS_R1 W/DATAMOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_WARN_R1</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - WARNING MESSAGE ISSUED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFS_WARN_R1</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - WARNING MESSAGE ISSUED</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>SOPFLG22</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_ALL</td>
<td></td>
<td></td>
<td>HEX'80' - QUERY VOLUME RAID(ALL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_NONE</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - QUERY VOLUME RAID(NONE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_S</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - QUERY VOLUME RAID(RAIDS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_1</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - QUERY VOLUME RAID(RAID1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_5</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - QUERY VOLUME RAID(RAID5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_10</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - QUERY VOLUME RAID(RAID10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_RD_6</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - QUERY VOLUME RAID(RAID6)</td>
</tr>
<tr>
<td>Offset (DEC)</td>
<td>Offset (HEX)</td>
<td>Name</td>
<td>Length</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>--------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>SOPFLG23</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SELM_FIRST</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - SELECTMULTI(FIRST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SELM_ANY</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - SELECTMULTI(ANY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SELM_ALL</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - SELECTMULTI(ALL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LISTSTMT</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - LIST(STATEMENTS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LISTHIST</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - LIST(HISTORY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_GROUP</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - PART OF GROUP INVOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POSTSNAP</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - POSTSNAP(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PRESNAP</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - PRESNAP(YES)</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>SOPFLG24</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTPVOL_ADDNEW</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - EXTXPAND(ADDNEW)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_IGNORE_RDF</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - IGNORE_RDF(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EMCONLY</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - EMC_ONLY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LDMF_DIVERT</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - LDMF_DIVERT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFARI_INFO</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - SRDFA-R1 SEVERITY=I (RC=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRDFSR1_INFO</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - SRDFA-R1 SEVERITY=I (RC=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CCUU_NM</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - CCUU(N:M) SPECIFIED</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>SOPFLG25</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_USESVL4LG</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - USE_SVL_FOR_LOGINDYNAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_USESVL4SV</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - USE_SVL_FOR_SOURCE_VOLUME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTALLOC_CONSALL</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - EXTALLOC(CONSALL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTALLOC_CONSVOL</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - EXTALLOC(CONSVOL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTALLOC_TRKALIGN</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - ALIGN_TRACKS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VDEWWAIT</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VDEWWAIT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SESSDIFF</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - SESSION_LIST W/DIFFERENTIAL</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>SOPFLG26</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_GRPAPIVER</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - GROUP_EMCQCAPI_VERIFY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_GRPDEVRY</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - GROUP_DEVICE_READY_STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_UNIT_RANGE</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - 1ST OF UNIT RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_UNIT_NRANGE</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - NOT 1ST OF UNIT RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SYMDV_RANGE</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - 1ST OF SYMDV RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SYMDV_NRANGE</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - NOT 1ST OF SYMDV RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SVOL_PRESENT</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - SOURCE VOLUME SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TVOL_PRESENT</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - TARGET VOLUME SPECIFIED</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>SOPFLG27</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MODE_NOCOPYRD</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - MODE(NOCOPYRD) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NO_BACKGNDRD</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - BACKGROUNDCOPY(NOCOPYRD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PREPARE</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - PREPARE_FOR_SNAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SKIP</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - SKIP EXECUTION OF THIS REQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ERRCHK_REduced</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - REDUCED ERROR CHECKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ERRREC_ENHANCED</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - ENHANCED ERROR RECOVERY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CTRLNAME</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - CONTROLLER NAME SUPPLIED</td>
</tr>
<tr>
<td>Offset (DEC)</td>
<td>Offset (HEX)</td>
<td>Name</td>
<td>Length</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>-----------------------------------</td>
<td>--------</td>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>SOPFLG28</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CG_IGNORE</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - NO CONGROUP CHECKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CGSAME</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - CONGROUP SRC/TGT SAME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CG_ANY</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - CONGROUP SRC/TGT ANY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CG_TARGET</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - CONGROUP TGT REQUIRED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CG_WARNING</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - CONGROUP SRC/TGT WARNING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CG_NONE</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - CONGROUP NONE REQUIRED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PRECOPY_WAIT</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - WAITFORPRECOPYPASS1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_FORCECMP</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - FORCE_COMPLETION</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>SOPFLG29</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LOCAL</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - LOCAL KEYWORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REMOTE</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - REMOTE KEYWORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LDMF_CLUSTER</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - LDMF_COMPONENT(CLUSTER)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LDMF_DATA</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - LDMF_COMPONENT(DATA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_LDMF_INDEX</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - LDMF_COMPONENT(INDEX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXALLOC_MRGEXIT</td>
<td></td>
<td>Equate</td>
<td>QU HEX'04' - EXALLOC(MRGEXIT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXALLOC_XTNTBNDRY</td>
<td></td>
<td>Equate</td>
<td>EQU HEX'02' - EXALLOC(XTNTBNDRY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VERIFY_NEVER</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - VERIFY(NEVER)</td>
</tr>
<tr>
<td>33</td>
<td>21</td>
<td>SOPFLG30</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRC_SYMDV#_OK</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - SOURCE SYMDV# SECURITY OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TGT_SYMDV#_OK</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - TARGET SYMDV# SECURITY OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_INVALIDATE_PDSE</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - INVALIDATE PDSE BUFFER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REMOVE_REMOTE</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - REMOVEREMOTE_ EXTENT_SESSIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_TDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - QUERY VOLUME TDEV OPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXAMINE_INDEX</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - EXAMINE(INDEXTEST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXAMINE_DATA</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - EXAMINE(DATATEST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXAMINE_NEVER</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - EXAMINE(NEVER)</td>
</tr>
<tr>
<td>34</td>
<td>22</td>
<td>SOPFLG31</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MULTIVDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - MULTI_VIRTUAL(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EATTR_NO</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - EATTR(NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EATTR_OPT</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - EATTR(OPT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VALVOLID</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - VALIDATE(VOLID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VALBWD</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - VALIDATE(BACKWARD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_IOMODE_TRK</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VALIDATE(IOMODE(TRACK))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_IOMODE_BLK</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - VALIDATE(IOMODE(BLOCK))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_IOMODE_RDM</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - VALIDATE(IOMODE(RANDOM))</td>
</tr>
<tr>
<td>35</td>
<td>23</td>
<td>SOPFLG32</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VALUNUSE</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - VALIDATE(UNUSED)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POOLUSE</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - CHECK_POOL_USAGE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_OPT_THINPOOL</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - QUERY VOLUME THINPOOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_XXX_ESNP231W</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - MAKE ESNP231E INTO ESNP231W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CMPRSV</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - RESERVE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SRCVDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - VDEV(Unit(VOL()) REQUESTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ALLOWFC_YES</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - ALLOW_FLASHCOPY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP.AllowFC_NO</td>
<td></td>
<td>Equate</td>
<td>HEX'01' - ALLOW_FLASHCOPY(NO)</td>
</tr>
</tbody>
</table>
### Table 22 SNSMFOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>24</td>
<td>SOPFLG33</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PCLONE</td>
<td></td>
<td>Equate</td>
<td>HEX’80’  - PARALLEL_CLONE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_NOUCODE</td>
<td></td>
<td>Equate</td>
<td>HEX’40’  - DATAMOVER(NOUCODE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PCLONE_REQ</td>
<td></td>
<td>Equate</td>
<td>HEX’20’  - PARALLEL_CLONE(REQUIRED)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_PCLONE_PREF</td>
<td></td>
<td>Equate</td>
<td>HEX’10’  - PARALLEL_CLONE(PREFERRED)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ACTIVATE_OFF</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - DO NOT PERFORM ACTIVATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MLQ</td>
<td></td>
<td>Equate</td>
<td>HEX’02’  - MULTI_LINE_QUERY(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RPTDIFF</td>
<td></td>
<td>Equate</td>
<td>HEX’01’  - REPORT(DIFFERENTIAL(YES))</td>
</tr>
<tr>
<td>37</td>
<td>25</td>
<td>SOPFLG34</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TOL_COMPACT</td>
<td></td>
<td>Equate</td>
<td>HEX’80’  - TOLERATE_DATACLASS_COMPACT_MISMATCH(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MODELDS_WARN</td>
<td></td>
<td>Equate</td>
<td>HEX’40’  - ESNP220(WARNING)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TOL_EXTENDED</td>
<td></td>
<td>Equate</td>
<td>HEX’20’  - TOLERATE_DATACLASS_EXTENDED_MISMATCH(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXPLAIN_VOL</td>
<td></td>
<td>Equate</td>
<td>HEX’10’  - EXPLAIN(VOLUME_SELECTION)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CLEAN_R2</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - AUTOCLEANR2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CRC_COMPAREA</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - CRC_COMPARE(ALWAYS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CRC_COMPAREN</td>
<td></td>
<td>Equate</td>
<td>HEX’02’  - CRC_COMPARE(NEVER)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ESNP119</td>
<td></td>
<td>Equate</td>
<td>HEX’01’  - ESNP119 OFF=WARN, ON=ERR</td>
</tr>
<tr>
<td>38</td>
<td>26</td>
<td>SOPFLG35</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_MODE_VSE</td>
<td></td>
<td>Equate</td>
<td>HEX’80’  - MODE(VSE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTOBIND</td>
<td></td>
<td>Equate</td>
<td>HEX’20’  - AUTO_BIND_TDEV(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTOUNBIND</td>
<td></td>
<td>Equate</td>
<td>HEX’10’  - AUTO_UNBIND_TDEV(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POOL_VDEV</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - QCRPOOL IS VDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POOL_TDEV</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - QCRPOOL IS TDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POOL2_VDEV</td>
<td></td>
<td>Equate</td>
<td>HEX’02’  - QCRPOOL2 IS VDEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_POOL2_TDEV</td>
<td></td>
<td>Equate</td>
<td>HEX’01’  - QCRPOOL2 IS TDEV</td>
</tr>
<tr>
<td>39</td>
<td>27</td>
<td>SOPFLG36</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_BIND_PRE</td>
<td></td>
<td>Equate</td>
<td>HEX’80’  - AUTO_BIND(PREALLOCATE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_BIND_PER</td>
<td></td>
<td>Equate</td>
<td>HEX’40’  - AUTO_BIND(PERSISTENT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_RDF_TRAK_CONSIST_N0</td>
<td></td>
<td>Equate</td>
<td>HEX’20’  - RDF TRACK CONSISTENCY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_VER_VOL_LAB</td>
<td></td>
<td>Equate</td>
<td>HEX’10’  - VERIFY_VOLUME_LABEL(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SOFTLINK</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - SOFTLINK(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TERMALL</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - TERMINATE_ALL(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_AUTOCREATE</td>
<td></td>
<td>Equate</td>
<td>HEX’02’  - AUTOCREATE(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_FREEUNLK</td>
<td></td>
<td>Equate</td>
<td>HEX’01’  - FREE(YES)</td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>SOPFLG37</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 37 - not used</td>
</tr>
<tr>
<td>41</td>
<td>29</td>
<td>SOPFLG38</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 38 - not used</td>
</tr>
<tr>
<td>42</td>
<td>2A</td>
<td>SOPFLG39</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_QSN_SORT</td>
<td></td>
<td>Equate</td>
<td>HEX’40’  - SORT(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SECURE</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - SECURE(YES)</td>
</tr>
<tr>
<td>43</td>
<td>2B</td>
<td>SOPFLG40</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DSIDSCHA_SET</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - DSIDSCHA(SET)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DSIDSCHA_RESET</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - DSIDSCHA(RESET)</td>
</tr>
<tr>
<td>44</td>
<td>2C</td>
<td>SOPFLG41</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_REL_TGT_HOLD_VALUE</td>
<td></td>
<td>Equate</td>
<td>HEX’08’  - RELEASE_LINK_TGT_HOLD(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_SET_TGT_HOLD_VALUE</td>
<td></td>
<td>Equate</td>
<td>HEX’04’  - SET_LINK_TGT_HOLD(YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_CHK_TGT_HOLD_VALUE</td>
<td></td>
<td>Equate</td>
<td>HEX’02’  - CHECK_LINK_TGT_HOLD(YES)</td>
</tr>
</tbody>
</table>
## Table 22  SNSMFOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>2D</td>
<td>SOPFLG42</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 42 - not used</td>
</tr>
<tr>
<td>46</td>
<td>2E</td>
<td>SOPFLG43</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 43 - not used</td>
</tr>
<tr>
<td>47</td>
<td>2F</td>
<td>SOPFLG44</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 44 - not used</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>SOPFLG45</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 45 - not used</td>
</tr>
<tr>
<td>49</td>
<td>31</td>
<td>SOPFLG46</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 46 - not used</td>
</tr>
<tr>
<td>50</td>
<td>32</td>
<td>SOPFLG47</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 47 - not used</td>
</tr>
<tr>
<td>51</td>
<td>33</td>
<td>SOPFLG48</td>
<td>1</td>
<td>Binary</td>
<td>Flag byte 48 - not used</td>
</tr>
<tr>
<td>52</td>
<td>34</td>
<td>SOPDEBUG</td>
<td>1</td>
<td>Binary</td>
<td>Debug flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DEBUG</td>
<td></td>
<td>Equate</td>
<td>HEX'80' - DEBUG(ON)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_TRACE</td>
<td></td>
<td>Equate</td>
<td>HEX'40' - TRACE(ON)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ERROR</td>
<td></td>
<td>Equate</td>
<td>HEX'20' - DEBUG(ERROR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_EXTRA</td>
<td></td>
<td>Equate</td>
<td>HEX'10' - DEBUG(EXTRA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_QCAPI</td>
<td></td>
<td>Equate</td>
<td>HEX'08' - DEBUG(QCAPI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_ALLOC</td>
<td></td>
<td>Equate</td>
<td>HEX'04' - DEBUG(ALLOC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP_DBGSKIP</td>
<td></td>
<td>Equate</td>
<td>HEX'02' - DEBUG(SKIP)</td>
</tr>
<tr>
<td>53</td>
<td>35</td>
<td>SOPMOVER</td>
<td>8</td>
<td>EBCDIC</td>
<td>Data mover name.</td>
</tr>
<tr>
<td>61</td>
<td>3D</td>
<td>SOPMOVETY</td>
<td>2</td>
<td>Binary</td>
<td>Data mover keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_NONE</td>
<td></td>
<td>Equate</td>
<td>HEX'0001' = NO DATA MOVER SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_EMCCOPY</td>
<td></td>
<td>Equate</td>
<td>HEX'0002' = EMCCOPY SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_COPYTRK</td>
<td></td>
<td>Equate</td>
<td>HEX'0003' = COPYTRK SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_COPYCYL</td>
<td></td>
<td>Equate</td>
<td>HEX'0004' = COPYCYL SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_DFDSS</td>
<td></td>
<td>Equate</td>
<td>HEX'0005' = DFDSS SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_FDRDSF</td>
<td></td>
<td>Equate</td>
<td>HEX'0006' = FDRDSF SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETY_IDCAM</td>
<td></td>
<td>Equate</td>
<td>HEX'0007' = IDCAM SPECIFIED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPMOVETYUTILITY</td>
<td></td>
<td>Equate</td>
<td>HEX'0063' = GENERIC UTILITY SPECIFIED</td>
</tr>
<tr>
<td>63</td>
<td>3F</td>
<td>SOPWAIT</td>
<td>4</td>
<td>Binary</td>
<td>WAITFORSESSION time for API:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 - No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1 - Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Other value - indicates time in seconds.</td>
</tr>
<tr>
<td>67</td>
<td>43</td>
<td>SPOCMPLT</td>
<td>4</td>
<td>Binary</td>
<td>WAITFORCOMPLETION time:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 - No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1 - Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Other value - indicates time in seconds.</td>
</tr>
<tr>
<td>71</td>
<td>47</td>
<td>SOPVALF</td>
<td>4</td>
<td>Binary</td>
<td>VALIDATE(FIRST(##))</td>
</tr>
<tr>
<td>75</td>
<td>4B</td>
<td>SOPVALL</td>
<td>4</td>
<td>Binary</td>
<td>VALIDATE(LAST(##))</td>
</tr>
<tr>
<td>79</td>
<td>4F</td>
<td>SOPVALLM</td>
<td>4</td>
<td>Binary</td>
<td>VALIDATE(LIMIT(##))</td>
</tr>
<tr>
<td>83</td>
<td>53</td>
<td>SOPVOLCT</td>
<td>2</td>
<td>Binary</td>
<td>VOLUMECOUNT(##)</td>
</tr>
<tr>
<td>85</td>
<td>55</td>
<td>SOPRELAT</td>
<td>44</td>
<td>EBCDIC</td>
<td>RELATE(dname)</td>
</tr>
<tr>
<td>129</td>
<td>81</td>
<td>SOPBCVGP</td>
<td>8</td>
<td>EBCDIC</td>
<td>BCVGROUP(groupname)</td>
</tr>
<tr>
<td>137</td>
<td>89</td>
<td>SOPUNITN</td>
<td>8</td>
<td>EBCDIC</td>
<td>UNITNAME(unitname)</td>
</tr>
</tbody>
</table>
### Table 22  SNSMFOPT - OPTIONS INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>145</td>
<td>91</td>
<td>SOPDATCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>DATACLASS(classname)</td>
</tr>
<tr>
<td>153</td>
<td>99</td>
<td>SOPSTGCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>STORAGECLASS(classname)</td>
</tr>
<tr>
<td>161</td>
<td>A1</td>
<td>SOPMGTCI</td>
<td>8</td>
<td>EBCDIC</td>
<td>MANAGEMENTCLASS(classname)</td>
</tr>
<tr>
<td>169</td>
<td>A9</td>
<td>SOPSTGRP</td>
<td>8</td>
<td>EBCDIC</td>
<td>STORAGE GROUP</td>
</tr>
<tr>
<td>177</td>
<td>B1</td>
<td>SOPENQ</td>
<td>1</td>
<td>EBCDIC</td>
<td>HOSTCOPYMODE(Shared</td>
</tr>
<tr>
<td>178</td>
<td>B2</td>
<td>SOPPDFSS_OPT</td>
<td>1</td>
<td>Binary</td>
<td>DFDSS_OPTIMIZE(n)</td>
</tr>
<tr>
<td>179</td>
<td>B3</td>
<td>SOPVENQ</td>
<td>1</td>
<td>EBCDIC</td>
<td>VSAMENQMODE(Shared</td>
</tr>
<tr>
<td>180</td>
<td>B4</td>
<td>SOPNTGRP</td>
<td>44</td>
<td>EBCDIC</td>
<td>NOTIFYWHENCOMPLETE(GROUP(name))</td>
</tr>
<tr>
<td>224</td>
<td>E0</td>
<td>SOPNEWVL</td>
<td>6</td>
<td>EBCDIC</td>
<td>NEWVOLID(volser)</td>
</tr>
<tr>
<td>230</td>
<td>E6</td>
<td>SOPSCFGP</td>
<td>66</td>
<td>EBCDIC</td>
<td>SCFGROUP(gnsgrp)</td>
</tr>
<tr>
<td>296</td>
<td>128</td>
<td>SOPGRP_NM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Snap GROUP NAME</td>
</tr>
<tr>
<td>304</td>
<td>130</td>
<td>SOPCMPENQ</td>
<td>1</td>
<td>EBCDIC</td>
<td>COMPARE_MODE(Shared</td>
</tr>
<tr>
<td>305</td>
<td>131</td>
<td>SOP_ACT_SUBTASK#</td>
<td>1</td>
<td>Binary</td>
<td>ACTIVATE_SUBTASK#(nnn)</td>
</tr>
<tr>
<td>306</td>
<td>132</td>
<td>SOPTIMEOUT</td>
<td>4</td>
<td>Binary</td>
<td>TIMEOUT(nnn)</td>
</tr>
<tr>
<td>310</td>
<td>136</td>
<td>SOPPOOL</td>
<td>12</td>
<td>EBCDIC</td>
<td>POOL(poolname)</td>
</tr>
<tr>
<td>322</td>
<td>142</td>
<td>SOPPOOL2</td>
<td>12</td>
<td>EBCDIC</td>
<td>SAVEDEV POOL2 NAME</td>
</tr>
<tr>
<td>334</td>
<td>14E</td>
<td>SOPCPLTY</td>
<td>2</td>
<td>Binary</td>
<td>CONFIGPOOL TYPE - no longer supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOPCPLTY_SAVEDEV</td>
<td></td>
<td>Equate</td>
<td>HEX'0101' - no longer supported</td>
</tr>
<tr>
<td>336</td>
<td>150</td>
<td>SOPSRVCL</td>
<td>16</td>
<td>EBCDIC</td>
<td>SOURCE_VOLUME_LIST(vollist)</td>
</tr>
<tr>
<td>352</td>
<td>160</td>
<td>SOPDESC</td>
<td>64</td>
<td>EBCDIC</td>
<td>DESCRIPTION('descriptive_text')</td>
</tr>
<tr>
<td>416</td>
<td>1A0</td>
<td>SOPCNTLR12</td>
<td>12</td>
<td>EBCDIC</td>
<td>QUERY BY CONTROLLER</td>
</tr>
<tr>
<td>428</td>
<td>1AC</td>
<td>SOPSNAPSHOT_NAME</td>
<td>32</td>
<td>EBCDIC</td>
<td>NAME(snapshot_name)</td>
</tr>
<tr>
<td>460</td>
<td>1CC</td>
<td>SOPSNAPSHOT_NEWNAME</td>
<td>32</td>
<td>EBCDIC</td>
<td>NEWNAME(new_snapshot_name)</td>
</tr>
<tr>
<td>492</td>
<td>1EC</td>
<td>SOP_EXPIRE</td>
<td>4</td>
<td>Binary</td>
<td>Snapshot expiration time in 1/2 seconds.</td>
</tr>
<tr>
<td>496</td>
<td>1F0</td>
<td>SOP_SNAPSHOT_ID</td>
<td>4</td>
<td>Binary</td>
<td>SNAPSHOTID(snapshot_id)</td>
</tr>
<tr>
<td>500</td>
<td>1F4</td>
<td>SOP_TARGETWAIT</td>
<td>4</td>
<td>Binary</td>
<td>TARGET_ENQ_DATASET_WAIT time:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 - No;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1 - Yes;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Other value - indicates time in seconds.</td>
</tr>
<tr>
<td>504</td>
<td>1F8</td>
<td>SOP_REUSEWAIT</td>
<td>4</td>
<td>Binary</td>
<td>WAIT TIME FOR REUSE DELETE</td>
</tr>
<tr>
<td>508</td>
<td>1FC</td>
<td>SOP_SRPPERCENT</td>
<td>4</td>
<td>Binary</td>
<td>SRP_PERCENT(###)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFOPT_LEN</td>
<td></td>
<td>Equate</td>
<td>512 (HEX'0200') = length of the Options Information segment.</td>
</tr>
</tbody>
</table>

**Note:** SNSMFOPT_LEN is the length of the Options Information segment. SNSMFOPT_LEN is set to 512 characters (HEX'0200').
### Table 23 SNSMFSRC - SOURCE DATASET INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SSDSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFSRC_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SSDSGID</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFSD#</td>
<td></td>
<td>Equate</td>
<td>Equate 4 = Source Dataset Information segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>SSDCATNM</td>
<td>1</td>
<td>EBCDIC</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SSDDSNAM</td>
<td>44</td>
<td>EBCDIC</td>
<td>Source dataset name.</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>SSSDDNAM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source DDNAME.</td>
</tr>
<tr>
<td>56</td>
<td>38</td>
<td>SSDCATNM</td>
<td>44</td>
<td>EBCDIC</td>
<td>Source catalog name.</td>
</tr>
<tr>
<td>100</td>
<td>64</td>
<td>SSDDATCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source data class.</td>
</tr>
<tr>
<td>108</td>
<td>6C</td>
<td>SSDMGMTCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source management class.</td>
</tr>
<tr>
<td>116</td>
<td>74</td>
<td>SSDSTGCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source storage class.</td>
</tr>
<tr>
<td>124</td>
<td>7C</td>
<td>SSDTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Number of tracks in the source dataset.</td>
</tr>
<tr>
<td>128</td>
<td>80</td>
<td>SSDUSED#</td>
<td>4</td>
<td>Binary</td>
<td>Number of used tracks in the source dataset.</td>
</tr>
<tr>
<td>132</td>
<td>84</td>
<td>SSSDHARBA</td>
<td>4</td>
<td>Binary</td>
<td>VSAM High Allocated RBA (HARBA).</td>
</tr>
<tr>
<td>136</td>
<td>88</td>
<td>SSSDHURBA</td>
<td>4</td>
<td>Binary</td>
<td>VSAM High Used RBA (HURBA).</td>
</tr>
<tr>
<td>140</td>
<td>8C</td>
<td>SSSDHKRBA</td>
<td>4</td>
<td>Binary</td>
<td>VSAM High Key RBA (HKRBA).</td>
</tr>
<tr>
<td>144</td>
<td>90</td>
<td>SSDXTN#</td>
<td>2</td>
<td>Binary</td>
<td>Source number of extents.</td>
</tr>
<tr>
<td>146</td>
<td>92</td>
<td>SSDTYPE</td>
<td>1</td>
<td>EBCDIC</td>
<td>Dataset type. Possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'A' - Non-VSAM dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'B' - GDG base</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'C' - Cluster</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'D' - Data component</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'G' - Alternate index</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'I' - Index component</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'R' - Path</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'X' - Alias name</td>
</tr>
<tr>
<td>147</td>
<td>93</td>
<td>SSDSTRP</td>
<td>1</td>
<td>Binary</td>
<td>Source stripe count.</td>
</tr>
<tr>
<td>148</td>
<td>94</td>
<td>SSSDDSNG</td>
<td>2</td>
<td>Binary</td>
<td>Dataset organization (DSORG).</td>
</tr>
<tr>
<td>150</td>
<td>96</td>
<td>SSDLRCL</td>
<td>2</td>
<td>Binary</td>
<td>Record length (LRECL).</td>
</tr>
<tr>
<td>152</td>
<td>98</td>
<td>SSSDBLKZ</td>
<td>2</td>
<td>Binary</td>
<td>Block size (BLKSIZE).</td>
</tr>
<tr>
<td>154</td>
<td>9A</td>
<td>SSSDRFM</td>
<td>1</td>
<td>Binary</td>
<td>Record format (RECFM).</td>
</tr>
<tr>
<td>155</td>
<td>9B</td>
<td>SSD_CG</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source ConGroup name 1.</td>
</tr>
<tr>
<td>163</td>
<td>A3</td>
<td>SSD_CG2</td>
<td>8</td>
<td>EBCDIC</td>
<td>Source ConGroup name 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFSRC_LEN</td>
<td></td>
<td></td>
<td>171 (HEX'00AB') = length of the Source Dataset Information segment.</td>
</tr>
</tbody>
</table>
### Table 24 SNSMFTGT - TARGET DATASET INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STDSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFTGT_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>STDSGID SNSMFTD#</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 5. 5 = Target Dataset Information segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td>EBCDIC</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>STDDSNAME</td>
<td>44</td>
<td>EBCDIC</td>
<td>Target dataset name.</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>STDDDNAM</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target DDNAME.</td>
</tr>
<tr>
<td>56</td>
<td>38</td>
<td>STDCATNM</td>
<td>44</td>
<td>EBCDIC</td>
<td>Target catalog name.</td>
</tr>
<tr>
<td>100</td>
<td>64</td>
<td>STDDATCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target data class.</td>
</tr>
<tr>
<td>108</td>
<td>6C</td>
<td>STDMGTCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target management class.</td>
</tr>
<tr>
<td>116</td>
<td>74</td>
<td>STDSTGCL</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target storage class.</td>
</tr>
<tr>
<td>124</td>
<td>7C</td>
<td>STDTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Number of tracks in the target dataset.</td>
</tr>
<tr>
<td>128</td>
<td>80</td>
<td>STDXTN#</td>
<td>2</td>
<td>Binary</td>
<td>Target number of extents.</td>
</tr>
<tr>
<td>130</td>
<td>82</td>
<td>STDSTRP</td>
<td>1</td>
<td>Binary</td>
<td>Target stripe count.</td>
</tr>
<tr>
<td>131</td>
<td>83</td>
<td>STDDSG</td>
<td>2</td>
<td>Binary</td>
<td>Dataset organization (DSORG).</td>
</tr>
<tr>
<td>133</td>
<td>85</td>
<td>STDLRCL</td>
<td>2</td>
<td>Binary</td>
<td>Record length (LRECL).</td>
</tr>
<tr>
<td>135</td>
<td>87</td>
<td>STDBLKZ</td>
<td>2</td>
<td>Binary</td>
<td>Block size (BLKSIZE).</td>
</tr>
<tr>
<td>137</td>
<td>89</td>
<td>STDRFM</td>
<td>1</td>
<td>Binary</td>
<td>Record format (RECFM).</td>
</tr>
<tr>
<td>138</td>
<td>8A</td>
<td>STD_CG</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target ConGroup name 1.</td>
</tr>
<tr>
<td>146</td>
<td>92</td>
<td>STD_CG2</td>
<td>8</td>
<td>EBCDIC</td>
<td>Target ConGroup name 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFTGT_LEN</td>
<td></td>
<td></td>
<td>154 (HEX'009A') = length of the Target Dataset Information segment.</td>
</tr>
</tbody>
</table>

### Table 25 SNSMFGTK - GATEKEEPER INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SGKSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFGTK_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SGKSGID SNSMFGK#</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 6. 6 = Gatekeeper Information segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SGKVOL</td>
<td>6</td>
<td>EBCDIC</td>
<td>Specified gatekeeper volume.</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>SGKUNIT</td>
<td>2</td>
<td>Binary</td>
<td>Specified gatekeeper unit address.</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SGKDDN</td>
<td>8</td>
<td>EBCDIC</td>
<td>Specified gatekeeper DDNAME.</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>SGKMHOP</td>
<td>8</td>
<td>Binary</td>
<td>Specified gatekeeper hop list. All HEX'FF' bytes mean local gatekeeper.</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>SGKAUCB@</td>
<td>4</td>
<td>Binary</td>
<td>Gatekeeper UCB address.</td>
</tr>
</tbody>
</table>
### Table 25 SNSMFGTK - GATEKEEPER INFORMATION

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>20</td>
<td>SGKAVAL</td>
<td>6</td>
<td>EBCDIC</td>
<td>Actual gatekeeper volume.</td>
</tr>
<tr>
<td>38</td>
<td>26</td>
<td>SGKAUNIT</td>
<td>2</td>
<td>Binary</td>
<td>Actual gatekeeper unit address.</td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>SGKMHOP4</td>
<td>16</td>
<td>Binary</td>
<td>Specified gatekeeper hop list (4 hops, each 4 bytes). All HEX'FF' bytes mean</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>local gatekeeper.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFGTK_LEN</td>
<td></td>
<td></td>
<td>56 (HEX'0038') = length of the Gatekeeper Information segment.</td>
</tr>
</tbody>
</table>

### Table 26 SNSMFSXL - SOURCE EXTENT LIST

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SSXSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFSXL_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SSXSGID</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 7. 7 = Source Extent List segment ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFSX#</td>
<td></td>
<td>Equate</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SSXTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Number of tracks in the extent.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>SSXUCB@</td>
<td>4</td>
<td>Binary</td>
<td>UCB address.</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SSXVOL</td>
<td>6</td>
<td>EBCDIC</td>
<td>VOLSER for the current extent.</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>SSXCCUU</td>
<td>2</td>
<td>Binary</td>
<td>MVS CCUU.</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>SSXBCHH</td>
<td>4</td>
<td>Binary</td>
<td>Beginning CCHH of the extent.</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>SSXECCHH</td>
<td>4</td>
<td>Binary</td>
<td>Ending CCHH of the extent.</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>SSXXTNT#</td>
<td>2</td>
<td>Binary</td>
<td>Extent number (relative to 1).</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>SSXSTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Starting track number (relative to 0).</td>
</tr>
<tr>
<td>34</td>
<td>22</td>
<td>SSXVOL#</td>
<td>1</td>
<td>Binary</td>
<td>Volume number (relative to 1) or stripe number.</td>
</tr>
<tr>
<td>35</td>
<td>23</td>
<td>SSXSER#</td>
<td>12</td>
<td>EBCDIC</td>
<td>Storage system serial number.</td>
</tr>
<tr>
<td>47</td>
<td>2F</td>
<td>SSXDEV#</td>
<td>4</td>
<td>Binary</td>
<td>Internal PowerMax/VMAX device number.</td>
</tr>
<tr>
<td>51</td>
<td>33</td>
<td>SSXMCODE</td>
<td>2</td>
<td>Binary</td>
<td>Operating environment level (5077 for instance).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFSXL_LEN</td>
<td></td>
<td></td>
<td>53 (HEX'0035') = length of the Source Extent List segment.</td>
</tr>
</tbody>
</table>

### Table 27 SNSMFTXL - TARGET EXTENT LIST

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STXSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFTXL_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>STXSGID</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 8. 8 = Target Extent List segment ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFTX#</td>
<td></td>
<td>Equate</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
### Table 27 SNSMFTXL - TARGET EXTENT LIST

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>STXTTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Number of tracks in the extent.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>STXUCB@</td>
<td>4</td>
<td>Binary</td>
<td>UCB address.</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>STXVOL</td>
<td>6</td>
<td>EBCDIC</td>
<td>VOLSER for the current extent.</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>STXCCUU</td>
<td>2</td>
<td>Binary</td>
<td>MVS CCUU.</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>STXBCCHH</td>
<td>4</td>
<td>Binary</td>
<td>Beginning CCHH of the extent.</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>STXECCHH</td>
<td>4</td>
<td>Binary</td>
<td>Ending CCHH of the extent.</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>STXXTNT#</td>
<td>2</td>
<td>Binary</td>
<td>Extent number (relative to 1).</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>STXSTRK#</td>
<td>4</td>
<td>Binary</td>
<td>Starting track number (relative to 0).</td>
</tr>
<tr>
<td>34</td>
<td>22</td>
<td>STXVOL#</td>
<td>1</td>
<td>Binary</td>
<td>Volume number (relative to 1) or stripe number.</td>
</tr>
<tr>
<td>35</td>
<td>23</td>
<td>STXSER#</td>
<td>12</td>
<td>EBCDIC</td>
<td>Storage system serial number.</td>
</tr>
<tr>
<td>47</td>
<td>2F</td>
<td>STXDEV#</td>
<td>4</td>
<td>Binary</td>
<td>Internal PowerMax/VMAX device number.</td>
</tr>
<tr>
<td>51</td>
<td>33</td>
<td>STXMCODE</td>
<td>2</td>
<td>Binary</td>
<td>Operating environment level (5077 for instance).</td>
</tr>
</tbody>
</table>

SNSMFTXL_LEN = 53 (HEX'0035') = length of the Target Extent List segment.

### Table 28 SNSMFCXL - COPY EXTENT LIST

<table>
<thead>
<tr>
<th>Offset (DEC)</th>
<th>Offset (HEX)</th>
<th>Name</th>
<th>Length</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>SCXSGLEN</td>
<td>2</td>
<td>Binary</td>
<td>Segment length = SNSMFCXL_LEN.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SCXSGID</td>
<td>1</td>
<td>Binary</td>
<td>Segment ID = 9. 9 = Copy Extent List segment ID.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td>Binary</td>
<td>Reserved.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>SCXSUCB@</td>
<td>4</td>
<td>Binary</td>
<td>Source UCB address.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>SCXSCCUU</td>
<td>2</td>
<td>Binary</td>
<td>Source UCB CCUU.</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>SCXSYMD</td>
<td>4</td>
<td>Binary</td>
<td>Source internal PowerMax/VMAX device number.</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SCXTUCB@</td>
<td>4</td>
<td>Binary</td>
<td>Target UCB address.</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>SCXTCCUU</td>
<td>2</td>
<td>Binary</td>
<td>Target UCB CCUU.</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>SCXTSYMMD</td>
<td>4</td>
<td>Binary</td>
<td>Target internal PowerMax/VMAX device number.</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>SCXMHOP</td>
<td>8</td>
<td>Binary</td>
<td>Remote hop list. All HEX'FF' bytes mean local gatekeeper.</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>SCXASSTR</td>
<td>4</td>
<td>Binary</td>
<td>Source relative track address (CCCCCCCH).</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>SCXATSTR</td>
<td>4</td>
<td>Binary</td>
<td>Target relative track address (CCCCCCCH).</td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>SCXSSTR</td>
<td>4</td>
<td>Binary</td>
<td>Source absolute track address (CCCCccCH).</td>
</tr>
<tr>
<td>44</td>
<td>2C</td>
<td>SCXTSTR</td>
<td>4</td>
<td>Binary</td>
<td>Target absolute track address (CCCCccCH).</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>SCX#TRK</td>
<td>4</td>
<td>Binary</td>
<td>Number of tracks to copy.</td>
</tr>
<tr>
<td>Offset (DEC)</td>
<td>Offset (HEX)</td>
<td>Name</td>
<td>Length</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>--------</td>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>52</td>
<td>34</td>
<td>SCXCPYTY</td>
<td>1</td>
<td>Binary</td>
<td>Copy type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_SKIP</td>
<td></td>
<td>Equate</td>
<td>HEX'00' - SKIP COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_MCODE</td>
<td></td>
<td>Equate</td>
<td>HEX'0A' - USE MICROCODE TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_EMCFFLASH</td>
<td></td>
<td>Equate</td>
<td>HEX'14' - USE FLASH MICROCODE TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_EMCCOPY</td>
<td></td>
<td>Equate</td>
<td>HEX'1E' - USE EMCCOPY TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_IBMSNAP</td>
<td></td>
<td>Equate</td>
<td>HEX'28' - USE ANTRQST-SNAPSHOT TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_IBMFLASH2</td>
<td></td>
<td>Equate</td>
<td>HEX'32' - USE ANTRQST-FLASHCOPY V2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_IBMFLASH</td>
<td></td>
<td>Equate</td>
<td>HEX'3C' - USE ANTRQST-FLASHCOPY TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTY_COPYCYL</td>
<td></td>
<td>Equate</td>
<td>HEX'46' - USE COPYCYL TO COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCXCPYTYUTILITY</td>
<td></td>
<td>Equate</td>
<td>HEX'50' - USE UTILITY TO COPY</td>
</tr>
<tr>
<td>53</td>
<td>35</td>
<td>SCXSXTN#</td>
<td>2</td>
<td>Binary</td>
<td>Source extent number.</td>
</tr>
<tr>
<td>55</td>
<td>37</td>
<td>SCXTXTN#</td>
<td>2</td>
<td>Binary</td>
<td>Target extent number.</td>
</tr>
<tr>
<td>57</td>
<td>39</td>
<td>SCXMHOP4</td>
<td>16</td>
<td>Binary</td>
<td>Remote hop list (4 hops, each 4 bytes). All HEX'FF' bytes mean local gatekeeper.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNSMFCXL_LEN</td>
<td></td>
<td></td>
<td>73 (HEX'0049') = length of the Copy Extent List segment.</td>
</tr>
</tbody>
</table>
SMF record sub-sections and TimeFinder actions

The following tables identify which sub-sections of the SMF record are present for each TimeFinder action. In determining segment length values, check for the values embedded in the record content.

Table 29 SMF Record sub-sections and TimeFinder actions

<table>
<thead>
<tr>
<th>CLEANUP</th>
<th>ACTIVATE</th>
<th>QUERY VDEV</th>
<th>RESTORE VIRTUAL DEVICE</th>
<th>DESTROY EXTENT TRACK</th>
<th>QUERY DATASET</th>
<th>SNAP DATASET</th>
<th>STOP SNAP TO DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
</tr>
<tr>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
</tr>
<tr>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
</tr>
<tr>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
</tr>
<tr>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
</tr>
<tr>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
<td>SCMSGLEN=508</td>
</tr>
<tr>
<td>SCMSGID#=5</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
</tr>
<tr>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=171</td>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=171</td>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=171</td>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=171</td>
</tr>
<tr>
<td>SCMSGID#=6</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
</tr>
<tr>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
</tr>
<tr>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
</tr>
<tr>
<td>SCMSGLEN=56</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
<td>SCMSGLEN=73</td>
</tr>
</tbody>
</table>

Table 30 SMF Record sub-sections and TimeFinder actions

<table>
<thead>
<tr>
<th>STOP SNAP TO VOLUME</th>
<th>CONFIG</th>
<th>QUERY VOLUME</th>
<th>CLEANUP EXTENT TRACK</th>
<th>ACTIVATE</th>
<th>QUERY VDEV</th>
<th>RESTORE VIRTUAL DEVICE</th>
<th>DESTROY EXTENT TRACK</th>
<th>QUERY DATASET</th>
<th>SNAP DATASET</th>
<th>STOP SNAP TO DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
<td>SCMSGID#=1</td>
</tr>
<tr>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
</tr>
<tr>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
<td>SCMSGID#=2</td>
</tr>
<tr>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
</tr>
<tr>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
<td>SCMSGID#=3</td>
</tr>
<tr>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
</tr>
<tr>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
<td>SCMSGID#=4</td>
</tr>
<tr>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
</tr>
<tr>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
<td>SCMSGID#=5</td>
</tr>
<tr>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
</tr>
<tr>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
<td>SCMSGID#=6</td>
</tr>
<tr>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
</tr>
<tr>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
<td>SCMSGID#=7</td>
</tr>
<tr>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
</tr>
<tr>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
<td>SCMSGID#=8</td>
</tr>
<tr>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
</tr>
<tr>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
<td>SCMSGID#=9</td>
</tr>
<tr>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
</tr>
</tbody>
</table>

Note: There may be multiple type 7, 8, or 9 segments.
APPENDIX C
SMF Record Layout
(Mainframe Enablers 8.0)

This appendix shows the SMF record layout, which is contained in the macro SNSMFRCD and is a distributable macro, for Mainframe Enablers 8.0.

- SMF record layout ................................................................. 356
- SMF record sub-sections and TimeFinder actions.......................... 366
SMF record layout

The volume and variety of information in the SMF records enables sites to produce many types of analysis and summary reports. By keeping historical SMF data and determining trends, an installation can evaluate changes in the configuration, workload, or job scheduling procedures. Similarly, an installation can use SMF data to determine where system resources are wasted because of problems, such as inefficient operational procedures or poor programming conventions.

Note: See the IBM Manual z/OS MVS System Management Facilities (SMF), section Using SMF Macros, subsection SMFWTM—Writing SMF Records for more information about the value and use of the SMF record.

```
MACRO ,
  SNSMFRCD ,
  SNSMFRCD DSECT ,
  *
  * STANDARD SMF RECORD HEADER
  *
  SNSMFLEN DS H       RECORD LENGTH
  SNSMFSEG DS H       SEGMENT DESCRIPTOR (ALWAYS 0)
  SNSMFSYS DS XL1     SYSTEM INDICATOR
  SNSMFRC# DS XL1     SMF RECORD NUMBER
  SNSMFTME DS XL4     TIME IN 100THS OF A SECOND
  SNSMFUTE DS XL4     DATE IN PACK DECIMAL - 0CYYDDDF
  SNSMFSID DS XL4     SYSTEM ID
  DS 0F                ALIGN TO WORD BOUNDARY
  *
  * COMMON SNAP SMF RECORD PREFIX
  *
  SNSMFSKY DS XL1     SUBTYPE
  SNSMFBAD DS XL1     RECORD IS INCOMPLETE OR BAD
  SNSMFS# DS H        SEGMENT COUNT
  DS 5H
  SNSMFHDR_LEN EQU *-SNSMFRCD       LENGTH OF RECORD HEADER
  *
  * COMMON SEGMENT PORTION (1ST 4 BYTES)
  *
  SNSMFCMN DSECT ,
  SCSMGLEN DS H       SEGMENT LENGTH
  SCSMSGID# DS XL1     SEGMENT ID
  DS XL1
  *
  * IDENTIFICATION SEGMENT
  *
  SNSMFID DSECT ,
  SIDSLEN DS H        SEGMENT LENGTH
  SIDSID DS XL1       SEGMENT ID
  SNSMFID# EQU 1       . 1 = SEGMENT ID FOR IDENTIFICATION
  DS XL1
  SIDJOBNM DS CL8     JOBNAME FROM
  SIDPROCS DS CL8     PROC STEP NAME JOB
  SIDTSTPMN DS CL8    STEPTNAME TIOT
  SIDPGNMN DS CL8     PROGRAM NAME
  SIDVRLLV DS CL6     VVLLRR
  SNSMFID_LEN EQU *-SNSMFID       LENGTH OF SNSMFID
  *
  * REQUEST INFORMATION
  *
  SNSMFRQ DSECT ,
  SRQSLEN DS H        SEGMENT LENGTH
  SRQSGID DS XL1      SEGMENT ID
```
SNSMFRQ# EQU 2  .  2 = SEGMENT ID FOR REQUEST
DS XL1
*
SRQTYPE DS XL2  REQUEST TYPE
SRQTYPE#DSN EQU 1  .  COPY DATASET
SRQTYPE#VOL EQU 2  .  COPY VOLUME
SRQTYPE#RST EQU 3  .  RESET EXTENT TRACK
SRQTYPE#CLN EQU 4  .  CLEANUP EXTENT TRACK
SRQTYPE#DBG EQU 5  .  DEBUG DATASET
SRQTYPE#STD EQU 6  .  STOP SNAP TO DATASET
SRQTYPE#STV EQU 7  .  STOP SNAP TO VOLUME
SRQTYPE#RVD EQU 8  .  RESTORE VIRTUAL DEVICE
SRQTYPE#ACT EQU 9  .  ACTIVATE
SRQTYPE#QDS EQU 10  .  QUERY DATASET
SRQTYPE#QSD EQU 11  .  QUERY SAVEDEV
SRQTYPE#QVD EQU 12  .  QUERY VDEV
SRQTYPE#CFG EQU 13  .  CONFIG
SRQTYPE#QVL EQU 14  .  QUERY VOLUME
SRQTYPE#SER EQU 15  .  SERIAL
SRQTYPE#PAR EQU 16  .  PARALLEL
SRQTYPE#CPL EQU 17  .  CONFIGPOOL (SEE SRQSACTN)¹
SRQTYPE#QGR EQU 18  .  QUERY GROUP
SRQTYPE#FGR EQU 19  .  DEFINE GROUP
SRQTYPE#EGR EQU 20  .  END GROUP
SRQTYPE#DGR EQU 21  .  DELETE GROUP
SRQTYPE#GRP EQU 22  .  EXTERNAL GRP RQST (SEE SRQSACTN)
SRQTYPE#QGL EQU 23  .  QUERY GLOBAL
*
*
*  IF SRQTYPE = SRQTYPE#CPL, THEN USE THE FOLLOWING VALUES
**  IF SRQTYPE = SRQTYPE#GRP, THEN USE SRQTYPE IN THIS FIELD
SRQSACTN DS XL2  SUB-ACTION
SRQSACTN_ADD EQU 01  .  ADD
SRQSACTN_CREATE EQU 02  .  CREATE
SRQSACTN_DELETE EQU 03  .  DELETE
SRQSACTN_DISABLE EQU 04  .  DISABLE
SRQSACTN_DISPLAY EQU 05  .  DISPLAY
SRQSACTN_ENABLE EQU 06  .  ENABLE
SRQSACTN_REMOVE EQU 07  .  REMOVE
SRQSACTN_DRAIN EQU 08  .  DRAIN
SRQSACTN_UNDRAIN EQU 09  .  UNDRAIN
*
*
SRQSTMT# DS A  STATEMENT NUMBER
SRQRC DS H  HIGHEST RETURN CODE
SRQRE DS H  ASSOCIATED REASON CODE
SRQSTIME DS D  COPY START TIME (TIME BIN)
SRQETIME DS D  COPY END TIME (TIME BIN)
SRQPSTIM DS D  PARSE START TIME (TIME STCK)
SRQPETIM DS D  PARSE END TIME (TIME STCK)
SRQESTIM DS D  EXECUTION START TIME (TIME STCK)
SRQSETIM DS D  EXECUTION SUSPEND TIME (TIME STCK)
SRQRESTIM DS D  EXECUTION RESUME TIME (TIME STCK)
SRQSETIM DS D  EXECUTION END TIME (TIME STCK)
SNSMFREQ_LEN EQU *-SNSMFREQ LENGTH OF SNSMFREQ
*
*  OPTIONS INFORMATION
*
SNSMFOPT DSECT ,
SOPSGLEN DS H  SEGMENT LENGTH
SOPSGID DS XL1  SEGMENT ID

¹ The GPM command CONFIGPOOL is no longer supported.
SMF Record Layout (Mainframe Enablers 8.0)

SNSMFOP# EQU 3  . 3 = SEGMENT ID FOR OPTIONS
DS XL1
*
SOPFLG01 DS XL1  FLAG BYTE 1
SOP_SRC_WILD EQU B'10000000'  -  SOURCE IS WILD
SOP_TGT_WILD EQU B'01000000'  -  TARGET IS WILD
SOP_EXCLUDE_HERE EQU B'00100000'  -  EXCLUDE IS PRESENT
SOP_RELATE_HERE EQU B'00010000'  -  RELATE IS PRESENT
SOP_RELATE_MADE EQU B'00001000'  -  RELATE WAS DERIVED FROM SOURCE
SOP_SPHERE EQU B'00000100'  -  SPHERE (YES)
SOP_EXECUTED EQU B'00000010'  -  REQUEST WAS EXECUTED
SOP_ERRDISP_KEEP EQU B'00000001'  -  ERRDISP(KEEP)
SOPFLG02 DS XL1  FLAG BYTE 2
SOP_PERMINDIRECT EQU B'10000000'  -  PERMANENTINDIRECT(Y) REQUESTED
SOP_NO_BACKGRND EQU B'01000000'  -  BACKGROUNDCOPY(N) REQUESTED
SOP_WAIT4COMPLET EQU B'00100000'  -  WAITFORCOMPLETION(MSG) REQUESTED
SOP_UCODE_FULL EQU B'00010000'  -  SNAP "FULL" VOLUME REQUESTED
SOP_SYMM_CYL EQU B'00001000'  -  SYMMETRIX_CYLINDER(Y) REQUESTED
SOP_VIRTUAL EQU B'00000100'  -  VIRTUAL_DEVICE(Y) REQUESTED
SOP_MIG_IGNORE EQU B'00000010'  -  MIGRATE(RECALL-IGNORE) SPECIFIED
SOP_R1R2SYNC EQU B'00000001'  -  WAITFORCOMPLETION(R1R2SYNC) SPEC
SOPFLG03 DS XL1  FLAG BYTE 3
SOP_SRC_DSNAME EQU B'10000000'  -  SOURCE DSNAME PRESENT
SOP_SRC_DDNAME EQU B'01000000'  -  SOURCE DDNAME PRESENT
SOP_TGT_DSNAME EQU B'00100000'  -  TARGET DSNAME PRESENT
SOP_TGT_DDNAME EQU B'00010000'  -  TARGET DDNAME PRESENT
SOP_SRC_DD_JCL EQU B'00001000'  -  SOURCE DDNAME ORIGINALLY PRESENT
SOP_TGT_DD_JCL EQU B'00000100'  -  TARGET DDNAME ORIGINALLY PRESENT
SOPFLG04 DS XL1  FLAG BYTE 4
SOP_REPLACE EQU B'10000000'  -  REPLACE(YES)
SOP_REUSE EQU B'01000000'  -  REUSE(YES)
SOP_COPYVOLID EQU B'00100000'  -  COPYVOLID(YES)
SOP_FORCE EQU B'00010000'  -  FORCE(YES)
SOP_ENQFAIL EQU B'00001000'  -  TOLERATE ENQ FAILURE (YES)
SOP_ALLOFAIL EQU B'00000100'  -  TOLERATE ALLOCATION FAILURE(YES)
SOP_COPYFAIL EQU B'00000010'  -  TOLERATE COPY FAILURE (YES)
SOP_TRUNC EQU B'00000001'  -  TOLERATE TRUNCATION (YES)
SOPFLG05 DS XL1  FLAG BYTE 5 (DDNAME SETTINGS)
SOP_SRC_DSNNAME EQU B'10000000'  -  SOURCE DSNNAME PRESENT
SOP_SRC_DDNNAME EQU B'01000000'  -  SOURCE DDNNAME PRESENT
SOP_TGT_DSNNAME EQU B'00100000'  -  TARGET DSNNAME PRESENT
SOP_TGT_DDNNAME EQU B'00010000'  -  TARGET DDNNAME PRESENT
SOP_SRC_DD_JCL EQU B'00001000'  -  SOURCE DSNNAME ORIGINALLY PRESENT
SOP_TGT_DD_JCL EQU B'00000100'  -  TARGET DSNNAME ORIGINALLY PRESENT
SOPFLG06 DS XL1  FLAG BYTE 6
SOP_CONSIST EQU B'10000000'  -  CONSISTENT(YES)
SOP_VIBBLD EQU B'01000000'  -  BUILD_VTOCIX(YES)
SOP_VALIDATE EQU B'00100000'  -  VALIDATE(YES)
SOP_VSAMFAIL EQU B'00010000'  -  TOLERATE VSAMENQ FAILURE (YES)
SOP_CHK_BCVHOLD EQU B'00001000'  -  CHECKBCVHOLDSTATUS(YES)
SOP_SCRATCH EQU B'00000100'  -  SCRATCH(YES)
SOP_VCLOSE EQU B'00000010'  -  VCLOSE(YES)
SOP_DPDS_ADMIN EQU B'00000001'  -  DFDSS_ADMIN(YES)
SOPFLG07 DS XL1  FLAG BYTE 7
SOP_CATALOG EQU B'10000000'  -  CATALOG(YES)
SOP_ENQWAIT EQU B'01000000'  -  ENQWAIT(YES)
SOP_REFTOC EQU B'00100000'  -  REFTOC(YES)
SOP_CONDITIONVOLUME_ALL EQU B'00010000'  -  CONDITIONVOLUME(ALL) OFF=LABEL
SOP_BCVONLY EQU B'00001000'  -  BCVONLY(YES)
SOP_CHKONLINEPATH EQU B'00000100'  -  CHECKONLINEPATHSTATUS(YES)
SOP_DPDS_CC EQU B'00000010'  -  DFDSS_CC(YES)
SOP_RECALC_FREE EQU B'00000001'  -  RECALCULATE_FREESPACE(YES)
SOPFLG08 DS XL1  FLAG BYTE 8
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOP_SRCDSORG_VS</td>
<td>EQU B'10000000' - INDSORG (VS) PRESENT</td>
</tr>
<tr>
<td>SOP_TOTDSORG_PS</td>
<td>EQU B'01000000' - OUTDSORG (PS) PRESENT</td>
</tr>
<tr>
<td>SOP_SPACECYL</td>
<td>EQU B'00100000' - NONVSAMSPACE (CYL)</td>
</tr>
<tr>
<td>SOP_SPACETRK</td>
<td>EQU B'00010000' - NONVSAMSPACE (TRK)</td>
</tr>
<tr>
<td>SOP_DATACYL</td>
<td>EQU B'00001000' - DATASPACE (CYL)</td>
</tr>
<tr>
<td>SOP_DATATRK</td>
<td>EQU B'00000100' - DATASPACE (TRK)</td>
</tr>
<tr>
<td>SOP_INDEXCYL</td>
<td>EQU B'00000010' - INDEXSPACE (CYL)</td>
</tr>
<tr>
<td>SOP_INDEXTRK</td>
<td>EQU B'00000001' - INDEXSPACE (TRK)</td>
</tr>
<tr>
<td>SOP_NOTIFY_DATASET</td>
<td>EQU B'10000000' - NOTIFY(DATASET)</td>
</tr>
<tr>
<td>SOP_NOTIFY_JOB</td>
<td>EQU B'01000000' - NOTIFY(JOB)</td>
</tr>
<tr>
<td>SOP_NOTIFY_SNAP</td>
<td>EQU B'00100000' - NOTIFY(SNAP)</td>
</tr>
<tr>
<td>SOP_NOTIFY_STEP</td>
<td>EQU B'00010000' - NOTIFY(STEP)</td>
</tr>
<tr>
<td>SOP_VARYOFF</td>
<td>EQU B'00001000' - VARY_OFFLINE(NEVER)</td>
</tr>
<tr>
<td>SOP_FREESPC</td>
<td>EQU B'00000100' - FREESPACE(YES)</td>
</tr>
<tr>
<td>SOP_ORDER_NAME</td>
<td>EQU B'00000010' - ORDER(NAME)</td>
</tr>
<tr>
<td>SOP_ORDER_SIZE</td>
<td>EQU B'00000001' - ORDER(SIZE)</td>
</tr>
<tr>
<td>SOPFLG09 DS</td>
<td>FLAG BYTE 9</td>
</tr>
<tr>
<td>SOP_COPY_ERROR</td>
<td>EQU B'10000000' - COPYTRACKS HAD AN ERROR</td>
</tr>
<tr>
<td>SOP_REUSFAIL</td>
<td>EQU B'01000000' - TOLERATE REUSE FAILURE (YES)</td>
</tr>
<tr>
<td>SOP_CHKONL_NEVER</td>
<td>EQU B'00100000' - CHECKONLINEPATHSTATUS(NEVER)</td>
</tr>
<tr>
<td>SOP_MSG_DISPLAY</td>
<td>EQU B'00010000' - MESSAGE(DISPLAY)</td>
</tr>
<tr>
<td>SOP_MSG_PROMPT</td>
<td>EQU B'00001000' - MESSAGE(PROMPT)</td>
</tr>
<tr>
<td>SOP_QRY_CNTLr</td>
<td>EQU B'00000100' - CONTROLLER(#) SUPPLIED FOR QUERY</td>
</tr>
<tr>
<td>SOP_COND_VOL_DUMP</td>
<td>EQU B'00000010' - CONDITIONVOLUME(DUMP)</td>
</tr>
<tr>
<td>SOP_AUTO_EXPAND</td>
<td>EQU B'00000001' - REUSE_AUTO_EXPAND(YES)</td>
</tr>
<tr>
<td>SOPFLG11 DS</td>
<td>FLAG BYTE 11</td>
</tr>
<tr>
<td>SOP_DONE_EMCSNAP</td>
<td>EQU B'10000000' - EMCSNAP USED FOR COPY</td>
</tr>
<tr>
<td>SOP_DONE_EMCCOPY</td>
<td>EQU B'01000000' - EMCCOPY USED FOR COPY</td>
</tr>
<tr>
<td>SOP_DONE_IBMSNAP</td>
<td>EQU B'00100000' - IBMSNAP USED FOR COPY</td>
</tr>
<tr>
<td>SOP_DONE_IBMFLSH</td>
<td>EQU B'00010000' - IBM FLASH COPY USED FOR COPY</td>
</tr>
<tr>
<td>SOP_DONEUTILITY</td>
<td>EQU B'00001000' - UTILITY PROGRAM USED FOR COPY</td>
</tr>
<tr>
<td>SOP_DONE_EMCFLSH</td>
<td>EQU B'00000100' - EMCFLASH USED FOR COPY</td>
</tr>
<tr>
<td>SOPFLG12 DS</td>
<td>FLAG BYTE 12</td>
</tr>
<tr>
<td>SOP_VARYON</td>
<td>EQU B'10000000' - VARY_ONLINE(YES)</td>
</tr>
<tr>
<td>SOP_VARYON_AUTO</td>
<td>EQU B'01000000' - VARY_ONLINE(AUTO)</td>
</tr>
<tr>
<td>SOP_ADMIN</td>
<td>EQU B'00010000' - ADMINISTRATOR(YES)</td>
</tr>
<tr>
<td>SOP_ALLO_UNUSED</td>
<td>EQU B'00001000' - ALLOCATE_UNUSED_SPACE(YES)</td>
</tr>
<tr>
<td>SOP_SNAP_UNUSED</td>
<td>EQU B'00000100' - SNAP_UNUSED_SPACE(YES)</td>
</tr>
<tr>
<td>SOP_RELEASE_YES</td>
<td>EQU B'00000010' - RELEASE(YES)</td>
</tr>
<tr>
<td>SOP_RELEASE_NO</td>
<td>EQU B'00000001' - RELEASE(NO)</td>
</tr>
<tr>
<td>SOPFLG13 DS</td>
<td>FLAG BYTE 13</td>
</tr>
<tr>
<td>SOP_PERSIST</td>
<td>EQU B'10000000' - RESTORE PERSISTENT(YES)</td>
</tr>
<tr>
<td>SOP_AUTORLSE</td>
<td>EQU B'01000000' - AUTOMATIC_RELEASE(YES)</td>
</tr>
<tr>
<td>SOP_AUTODEAL</td>
<td>EQU B'00100000' - AUTOMATIC_DEALLOC(YES)</td>
</tr>
<tr>
<td>SOP_R1FULLCOPY</td>
<td>EQU B'00010000' - R1FULLCOPYONLY(YES)</td>
</tr>
<tr>
<td>SOP_AUTOCLEAN</td>
<td>EQU B'00001000' - AUTOMATIC_CLEANUP(YES)</td>
</tr>
<tr>
<td>SOP_DIFF_DSN</td>
<td>EQU B'00000100' - DIFFERENTIAL_DATASET(YES)</td>
</tr>
</tbody>
</table>

**SMF Record Layout (Mainframe Enablers 8.0)**

FLAG BYTE 9:
- SOP_NOTIFY_DATASET: EQU B'10000000' - NOTIFY(DATASET)
- SOP_NOTIFY_JOB: EQU B'01000000' - NOTIFY(JOB)
- SOP_NOTIFY_SNAP: EQU B'00100000' - NOTIFY(SNAP)
- SOP_NOTIFY_STEP: EQU B'00010000' - NOTIFY(STEP)
- SOP_VARYOFF: EQU B'00001000' - VARY_OFFLINE(NEVER)
- SOP_FREESPC: EQU B'00000100' - FREESPACE(YES)
- SOP_ORDER_NAME: EQU B'00000010' - ORDER(NAME)
- SOP_ORDER_SIZE: EQU B'00000001' - ORDER(SIZE)

FLAG BYTE 10:
- SOP_EXTALLOC_YES: EQU B'10000000' - EXTENT_ALLOCATION(YES)
- SOP_EXTALLOC_NO: EQU B'01000000' - EXTENT_ALLOCATION(NO)
- SOP_EXTXPAND: EQU B'00100000' - EXTENT_EXPAND(YES)
- SOP_ACTIVATE: EQU B'00010000' - WAIT FOR ACTIVATE
- SOP_ACTIVATE1: EQU B'00001000' - ACTIVATE PHASE 1 DONE
- SOP_ACTIVATE2: EQU B'00000100' - ACTIVATE PHASE 2 DONE
- SOP_DIFF_VOL: EQU B'00000010' - DIFFERENTIAL(YES)
- SOP_DISBAND: EQU B'00000001' - DISBAND(YES)

FLAG BYTE 11:
- SOP_COPY_ERROR: EQU B'10000000' - COPYTRACKS HAD AN ERROR
- SOP_REUSFAIL: EQU B'01000000' - TOLERATE REUSE FAILURE (YES)
- SOP_CHKONL_NEVER: EQU B'00100000' - CHECKONLINEPATHSTATUS(NEVER)
- SOP_MSG_DISPLAY: EQU B'00010000' - MESSAGE(DISPLAY)
- SOP_MSG_PROMPT: EQU B'00001000' - MESSAGE(PROMPT)
- SOP_QRY_CNTLr: EQU B'00000100' - CONTROLLER(#) SUPPLIED FOR QUERY
- SOP_COND_VOL_DUMP: EQU B'00000010' - CONDITIONVOLUME(DUMP)
- SOP_AUTO_EXPAND: EQU B'00000001' - REUSE_AUTO_EXPAND(YES)

FLAG BYTE 12:
- SOP_VARYON: EQU B'10000000' - VARY_ONLINE(YES)
- SOP_VARYON_AUTO: EQU B'01000000' - VARY_ONLINE(AUTO)
- SOP_ADMIN: EQU B'00010000' - ADMINISTRATOR(YES)
- SOP_ALLO_UNUSED: EQU B'00001000' - ALLOCATE_UNUSED_SPACE(YES)
- SOP_SNAP_UNUSED: EQU B'00000100' - SNAP_UNUSED_SPACE(YES)
- SOP_RELEASE_YES: EQU B'00000010' - RELEASE(YES)
- SOP_RELEASE_NO: EQU B'00000001' - RELEASE(NO)

FLAG BYTE 13:
- SOP_PERSIST: EQU B'10000000' - RESTORE PERSISTENT(YES)
- SOP_AUTORLSE: EQU B'01000000' - AUTOMATIC_RELEASE(YES)
- SOP_AUTODEAL: EQU B'00100000' - AUTOMATIC_DEALLOC(YES)
- SOP_R1FULLCOPY: EQU B'00010000' - R1FULLCOPYONLY(YES)
- SOP_AUTOCLEAN: EQU B'00001000' - AUTOMATIC_CLEANUP(YES)
- SOP_DIFF_DSN: EQU B'00000100' - DIFFERENTIAL_DATASET(YES)

FLAG BYTE 14:
- SOP.ready_YES: EQU B'10000000' - READY(YES)
- SOP.ready_NO: EQU B'01000000' - READY(NO)
- SOP.SAVEDEV_NR: EQU B'00100000' - SAVEDEV_FULL(NO)
- SOP.MOVRE_ACTIVE: EQU B'00010000' - DATAMOVER ACTIVE FOR REQUEST
- SOP.MOVN_SRCSR: EQU B'00001000' - DATAMOVER - NO SOURCE RESERVE
- SOP.MOVN_NTG: EQU B'00001000' - DATAMOVER - NO TARGET RESERVE
- SOP.BIET_NODT: EQU B'00000100' - BUILD EXTENTS WITH NO TARGET
- SOP.PARALLEL_REQ: EQU B'00000010' - PARALLEL_REQUESTED

FLAG BYTE 15:
- SOP.PERSIST: EQU B'10000000' - RESTORE PERSISTENT(YES)
- SOP.AUTO_RELEASE: EQU B'01000000' - AUTOMATIC_RELEASE(YES)
- SOP.AUTO_RELEASE: EQU B'00100000' - AUTOMATIC_DEALLOC(YES)
- SOP.R1FULLCOPY: EQU B'00010000' - R1FULLCOPYONLY(YES)
- SOP.AUTO_CLEANUP: EQU B'00001000' - AUTOMATIC_CLEANUP(YES)
- SOP.DIFF_DSN: EQU B'00000100' - DIFFERENTIAL_DATASET(YES)
SMF Record Layout (Mainframe Enablers 8.0)

**SOP_MODE_COPY** EQU B'00000010' - MODE(COPY) REQUESTED

**SOP_MODE_NOCOPY** EQU B'00000001' - MODE(NOCOPY) REQUESTED

**SOPFLG16** DS XL1 FLAG BYTE 16

**SOP_VOL_LEVEL** EQU B'10000000' - VOLUME LEVEL REQUEST

**SOP_DSN_LEVEL** EQU B'01000000' - DATASET LEVEL REQUEST

**SOP_CTRL_LEVEL** EQU B'00100000' - CONTROLLER LEVEL REQUEST

**SOP_COLLAPSE_NV** EQU B'00010000' - COLLAPSE DATASET EXTENTS (NV)

**SOP_COLLAPSE_VS** EQU B'00001000' - COLLAPSE DATASET EXTENTS (VS)

**SOP_PRECOPY** EQU B'00000100' - PRECOPY

**SOP_DUMPTRK** EQU B'00000010' - DUMP_TRACK_CONTENTS

**SOP_MBRENABLE** EQU B'00000001' - MEMBERSTATE(ENABLE)

**SOPFLG17** DS XL1 FLAG BYTE 17

**SOP_SRC_SYMDV** EQU B'10000000' - SOURCE SYMDV# PRESENT

**SOP_TGT_SYMDV** EQU B'01000000' - TARGET SYMDV# PRESENT

**SOP_TERMSESS** EQU B'00100000' - TERMINATE_SESSION

**SOP_CTRL_LEVEL** EQU B'00100000' - CONTROLLER LEVEL REQUEST

**SOP_COLLAPSE_NV** EQU B'00010000' - COLLAPSE_DATASET_EXTENTS (NV)

**SOP_COLLAPSE_VS** EQU B'00001000' - COLLAPSE_DATASET_EXTENTS (VS)

**SOP_PRECOPY** EQU B'00000100' - PRECOPY

**SOP_DUMPTRK** EQU B'00000010' - DUMP_TRACK_CONTENTS

**SOP_MBRENABLE** EQU B'00000001' - MEMBERSTATE(ENABLE)

**SOPFLG17** DS XL1 FLAG BYTE 17

**SOP_SRC_SYMDV** EQU B'10000000' - SOURCE SYMDV# PRESENT

**SOP_TGT_SYMDV** EQU B'01000000' - TARGET SYMDV# PRESENT

**SOP_TERMSESS** EQU B'00100000' - TERMINATE_SESSION

**SOP_CTRL_LEVEL** EQU B'00100000' - CONTROLLER LEVEL REQUEST

**SOP_COLLAPSE_NV** EQU B'00010000' - COLLAPSE_DATASET_EXTENTS (NV)

**SOP_COLLAPSE_VS** EQU B'00001000' - COLLAPSE_DATASET_EXTENTS (VS)

**SOP_PRECOPY** EQU B'00000100' - PRECOPY

**SOP_DUMPTRK** EQU B'00000010' - DUMP_TRACK_CONTENTS

**SOP_MBRENABLE** EQU B'00000001' - MEMBERSTATE(ENABLE)

**SOPFLG17** DS XL1 FLAG BYTE 17

**SOP_SRC_SYMDV** EQU B'10000000' - SOURCE SYMDV# PRESENT

**SOP_TGT_SYMDV** EQU B'01000000' - TARGET SYMDV# PRESENT

**SOP_TERMSESS** EQU B'00100000' - TERMINATE_SESSION

**SOP_CTRL_LEVEL** EQU B'00100000' - CONTROLLER LEVEL REQUEST

**SOP_COLLAPSE_NV** EQU B'00010000' - COLLAPSE_DATASET_EXTENTS (NV)

**SOP_COLLAPSE_VS** EQU B'00001000' - COLLAPSE_DATASET_EXTENTS (VS)

**SOP_PRECOPY** EQU B'00000100' - PRECOPY

**SOP_DUMPTRK** EQU B'00000010' - DUMP_TRACK_CONTENTS

**SOP_MBRENABLE** EQU B'00000001' - MEMBERSTATE(ENABLE)

**SOPFLG17** DS XL1 FLAG BYTE 17

**SOP_SRC_SYMDV** EQU B'10000000' - SOURCE SYMDV# PRESENT

**SOP_TGT_SYMDV** EQU B'01000000' - TARGET SYMDV# PRESENT

**SOP_TERMSESS** EQU B'00100000' - TERMINATE_SESSION

**SOP_CTRL_LEVEL** EQU B'00100000' - CONTROLLER LEVEL REQUEST

**SOP_COLLAPSE_NV** EQU B'00010000' - COLLAPSE_DATASET_EXTENTS (NV)

**SOP_COLLAPSE_VS** EQU B'00001000' - COLLAPSE_DATASET_EXTENTS (VS)

**SOP_PRECOPY** EQU B'00000100' - PRECOPY

**SOP_DUMPTRK** EQU B'00000010' - DUMP_TRACK_CONTENTS

**SOP_MBRENABLE** EQU B'00000001' - MEMBERSTATE(ENABLE)
SMF Record Layout (Mainframe Enablers 8.0)

SOP_SEL_Multiple EQU B'01000000' - SELECTMULTI(ANY)
SOP_SEL_All EQU B'00100000' - SELECTMULTI(ALL)
SOP_LISTStmt EQU B'00010000' - LIST(STATEMENTS)
SOP_LISTHIST EQU B'00001000' - LIST(HISTORY)
SOP_GROUP EQU B'00000100' - PART OF GROUP INVOCATION
SOP_POSTSNAP EQU B'00000010' - POSTSNAP(YES)
SOP_PRESNAP EQU B'00000001' - PRESNAP(YES)

SOPFLG24 DS XL1 FLAG BYTE 24
SOP_EXTXPVOL_ADDNEW EQU B'10000000' - EXTXPAND(ADDNEW)
SOP_IGNORE_RDF EQU B'01000000' - IGNORE_RDF(YES)
SOP_EMONLY EQU B'00100000' - EMC_ONLY(YES)
SOP_LDMF_DIVERT EQU B'00010000' - LDMF_DIVERT(YES)
SOP_SRDFAR1_INFO EQU B'00001000' - SRDFA-R1 SEVERITY=I (RC=0)
SOP_SRDFSR1_INFO EQU B'00000100' - SRDFA-R1 SEVERITY=I (RC=0)
SOP_IGNORE_SYMDV# EQU B'00000010' - IGNORE SYMDV#
SOP_CCUU_NM EQU B'00000001' - CCUU(N:M) SPECIFIED

SOPFLG25 DS XL1 FLAG BYTE 25
SOP_USESVL4LG EQU B'10000000' - USE_SVL_FOR_LOGINDYNAM
SOP_USESVL4SV EQU B'01000000' - USE_SVL_FOR_LOGINDYNAM
SOP_EXTALLOC_CONSALL EQU B'00100000' - EXTALLOC(CONSALL)
SOP_EXTALLOC_CONSVOL EQU B'00010000' - EXTALLOC(CONSVOL)
SOP_EXTALLOC_TRKALIGN EQU B'00001000' - ALIGN_TRACKS
SOP_VDEVWAIT EQU B'00000100' - VDEVWAIT(YES)
SOP_SESSDIFF EQU B'00000010' - SESSION_LIST W/DIFFERENTIAL
SOP_PROCgate EQU B'00000001' - PROCESSING GATEKEEPER

SOPFLG26 DS XL1 FLAG BYTE 26
SOP_GRPAPIVER EQU B'10000000' - GROUP_EMCQCAPI_VERIFY
SOP_GRPDEVRDY EQU B'01000000' - GROUP_DEVICE_READY_STATE
SOP_UNIT_RANGE EQU B'00100000' - 1ST OF UNIT RANGE
SOP_UNIT_NRange EQU B'00010000' - NOT 1ST OF UNIT RANGE
SOP_SYMDV_RANGE EQU B'00001000' - 1ST OF SYMDV RANGE
SOP_SYMDV_NR ange EQU B'00000100' - NOT 1ST OF SYMDV RANGE
SOP_TVOL_PRESENT EQU B'00000010' - SOURCE VOLUME SPECIFIED
SOP_TVOL_RECEIVE EQU B'00000001' - TARGET VOLUME SPECIFIED

SOPFLG27 DS XL1 FLAG BYTE 27
SOP_MODE_NOCOPYRD EQU B'00100000' - MODE(NOCOPYRD) REQUESTED
SOP_NO_BACKGRNDRD EQU B'00010000' - BACKGROUND(NOCOPYRD)
SOP_VDEV_DEL_LOCK EQU B'00001000' - VDEV DEL LOCK ACQUIRED
SOP)prepare EQU B'00000100' - PREPARE FOR SNAP
SOP_SKIP EQU B'00000010' - SKIP EXECUTION OF THIS REQ
SOP_ERRCHK_REduced EQU B'00000010' - REDUCED ERROR CHECKING
SOP_ERRREC_ENHANCED EQU B'00000001' - ENHANCED ERROR RECOVERY
SOP_CTRLNAME EQU B'00000001' - CONTROLLER NAME SUPPLIED

SOPFLG28 DS XL1 FLAG BYTE 28
SOP_CG_IGNORE EQU B'10000000' - NO CONGROUP CHECKING
SOP_CG_SAME EQU B'01000000' - CONGROUP SRC/TGT SAME
SOP_CG_ANY EQU B'00100000' - CONGROUP SRC/TGT ANY
SOP_CG_TARGET EQU B'00010000' - CONGROUP TGT REQUIRED
SOP_CG_WARNING EQU B'00001000' - CONGROUP SRC/TGT WARNING
SOP_CG_NONE EQU B'00000100' - CONGROUP NONE REQUIRED
SOP_PRECOPY_WAIT EQU B'00000010' - WAITFORPRECOPYPASS1
SOP_Forcecmp EQU B'00000001' - FORCE_COMPLETION

SOPFLG29 DS XL1 FLAG BYTE 29
SOP_LOCAL EQU B'10000000' - LOCAL KEYWORD
SOP_REMOTE EQU B'01000000' - REMOTE KEYWORD
SOP_LDMF_CLUSTER EQU B'00100000' - LDMF_COMPONENT(CLUSTER)
SOP_LDMF_DATA EQU B'00010000' - LDMF_COMPONENT(DATA)
SOP_LDMF_INDEX EQU B'00001000' - LDMF_COMPONENT(INDEX)
SOP_EXTALLOC_MRGEXIST EQU B'00000100' - EXTALLOC(MRGEXIST)
SOP_EXTALLOC_XNTBNDRY EQU B'00000010' - EXTALLOC(XNTBNDRY)
SOP_VERIFY_NEVER EQU B'00000001' - VERIFY(NEVER)

SOPFLG30 DS XL1 FLAG BYTE 30
SOP_SRC_SYMDV#_OK EQU B'10000000' - SOURCE SYMDV# SECURITY OK
SOP_TGT_SYMDV#_OK EQU B'01000000' - TARGET SYMDV# SECURITY OK
SOP_INVALIDATE_PDSE EQU B'00100000' - INVALIDATE PDSE BUFFER
SOP_REMOVE_REMOTE EQU B'00010000' - REMOVE_REMOTE_EXTENT_SESSIONS
SOP_OPT_TDEV   EQU B'00001000'  - QUERY VOLUME TDEV OPTION
SOP_EXAMINE_INDEX EQU B'00000100'  - EXAMINE(INDEXTEST)
SOP_EXAMINE_DATA EQU B'00000010'  - EXAMINE(DATATEST)
SOP_EXAMINE_NEVER EQU B'00000001'  - EXAMINE(NEVER)
SOPFLG31 DS XL1  FLAG BYTE 31
SOP_MULTIVDEV EQU B'10000000'  - MULTI_VIRTUAL(YES)
SOP_EATTR_NO EQU B'01000000'  - EATTR(NO)
SOP_EATTR_OPT EQU B'00100000'  - EATTR(OPT)
SOPFLG32 DS XL1  FLAG BYTE 32
SOP_POOLUSE EQU B'01000000'  - CHECK_POOL_USAGE(YES)
SOP_OPT_THINPOOL EQU B'00100000'  - QUERY VOLUME THINPOOL
SOP_CMRPSV EQU B'00001000'  - RESERVE(YES)
SOP_SRCVDEV EQU B'00000100'  - VDEV(UNIT()/VOL()) REQUESTED
SOP_ALLOWFC_YES EQU B'00000010'  - ALLOW_FLASHCOPY(YES)
SOP_ALLOWFC_NO EQU B'00000001'  - ALLOW_FLASHCOPY(NO)
SOPFLG33 DS XL1  FLAG BYTE 33
SOP_PCLONE EQU B'10000000'  - PARALLEL_CLONE(YES)
SOP_NOUCODE EQU B'01000000'  - DATAMOVER(NOUCODE)
SOP_PCLONE_REQ EQU B'00100000'  - PARALLEL_CLONE(REQUIRED)
SOP_PCLONE_PREF EQU B'00010000'  - PARALLEL_CLONE(PREFERRED)
SOP_INLINE_PRINT EQU B'00001000'  - AVOID PRINT BUFFERING
SOP.Activate_OFF EQU B'00000100'  - ACTIVATE?
SOP_MLQ EQU B'00000010'  - MULTI_LINE_QUERY(YES)
SOP.RPTDIFF EQU B'00000001'  - REPORT(DIFFERENTIAL(YES))
SOPFLG34 DS XL1  FLAG BYTE 34
SOP_TOL_COMPACT EQU B'10000000'  - TOLERATEDATACLASSCOMPACTATION...
SOP_MODELS_WARN EQU B'01000000'  - ESNP220(WARNING)
SOP_TOL_EXTENDED EQU B'00100000'  - TOLERATEDATACLASSEXTENDED...
SOP_EXPLAIN_VOL EQU B'00010000'  - EXPLAIN(VOLUMESALECTION)
SOP_CLEAN_R2 EQU B'00001000'  - AUTOCLEANR2
SOP_CRC_COMPARE EQU B'00000100'  - CRC_COMPARE(ALWAYS)
SOP_CRC_COMPARE EQU B'00000010'  - CRC_COMPARE(NEVER)
SOPFLG35 DS XL1  FLAG BYTE 35
SOPFLG36 DS XL1  FLAG BYTE 36
SOPFLG37 DS XL1  FLAG BYTE 37
SOPFLG38 DS XL1  FLAG BYTE 38
SOPFLG39 DS XL1  FLAG BYTE 39
SOPFLG40 DS XL1  FLAG BYTE 40
SOPFLG41 DS XL1  FLAG BYTE 41
SOPFLG42 DS XL1  FLAG BYTE 42
SOPFLG43 DS XL1  FLAG BYTE 43
SOPFLG44 DS XL1  FLAG BYTE 44
SOPFLG45 DS XL1  FLAG BYTE 45
SOPFLG46 DS XL1  FLAG BYTE 46
SOPFLG47 DS XL1  FLAG BYTE 47
SOPFLG48 DS XL1  FLAG BYTE 48
*  
SOP_DEBUG EQU B'10000000'  - DEBUG(ON)
SOP_TRACE EQU B'01000000'  - TRACE(ON)
SOP_ERROR EQU B'00100000'  - DEBUG(ERROR)
SOP_EXTRA EQU B'00010000'  - DEBUG(EXTRA)
SOP_QCAPI EQU B'00001000'  - DEBUG(EMCQCAPI)
SOPALLOC EQU B'00000100'  - DEBUG(EMCALLOC)
SOP_DBGSKIP EQU B'00000010'  - DEBUG(SKIP)
SOPMOVER DS CL8 DATA MOVER NAME
SOPMOVY DS YL2 DATA MOVER KEYWORD
SOPMOVY_NONE EQU 1  - NO DATA MOVER SPECIFIED
SOPMOVY_EMCCOPY EQU 2  - EMCCOPY SPECIFIED
SOPMOVY_COPYTRK EQU 3  - COPYTRK SPECIFIED
SOPMOVY_COPYCYL EQU 4  - COPYCYL SPECIFIED
SOPMOVY_DPDSSEQ EQU 5  - DFDSSEQ SPECIFIED
SOPMOVY_FDSSEQ EQU 6  - FDSSEQ SPECIFIED
SOPMOVY_IDCAMZ EQU 7  - IDCAMS SPECIFIED
SOPMOVYUTILITY EQU 99  - GENERIC UTILITY SPECIFIED
SMF Record Layout (Mainframe Enablers 8.0)

SOPWAIT DS FL4 WAIT TIME FOR API
SOPCMPLT DS FL4 WAIT TIME FOR SNAP COMPLETION
SOPVOLCT DS YL2 VOLCOUNT FOR MULTI-VOLUME ALLOC
SOPRELAT DS CL44 RELATE DATA SET NAME FOR AIX
SOPBVCVP DS CL8 TARGET BCVGROUP
SOPUNITN DS CL8 TARGET UNITNAME
SOPDATCL DS CL8 DATA CLASS
SOPSTGCL DS CL8 STORAGE CLASS
SOPMGTCI DS CL8 MANAGEMENT CLASS
SOPSTGRP DS CL8 STORAGE GROUP
SOPENQ DS CL1 ENQ = EXC/SHR/NONE
SOPDFDSS_OPT DS AL1 DFDSS_OPTIMIZE
SOPVENQ DS CL1 VSAMENQMODE = EXC/SHR/NONE
SOPNTGRP DS CL44 NOTIFY(GROUP(XXX))
SOPNEWVL DS CL6 NEW VOLID
SOPSCFGP DS CL66 TARGET SCFGROUP
SOPGRPNM DS CL8 GROUP NAME
SOPMGTCL DS CL8 MANAGEMENT CLASS
SOPSTGCL DS CL8 STORAGE CLASS
SOPUNITN DS CL8 UNITNAME
SOP_DATCL DS CL8 DATA CLASS
SOPSTGCL DS CL8 STORAGE CLASS
SOPUNTN DS CL8 STORAGE GROUP
SOPENQ DS CL1 ENQ = EXC/SHR/NONE
SOPACT_SUBTASK# DS XL1 CONSISTENT SUBTASK COUNT
SOPTIMEOUT DS FL4 CONSISTENT TIMEOUT VALUE
SOPPOOL DS CL12 SAVEDEV POOL NAME
SOPCPPLTY DS Y CONFIGPOOL TYPE
SOPCPPLTY_SAVEDDEV EQU X'0101'
SOPSRVL DS CL6 SOURCE_VOLUME_LIST NAME
SOPDESC DS CL64 GROUP DESCRIPTION
SOPCNTLR12 DS CL12 QUERY - BY CONTROLLER - 12 DIGIT
SNSMF.opts EQU *.SNSMFOPT LENGTH OF SNSMFOPT
*S SOURCE DATASET INFORMATION *
SNSMFSRC DSECT ,
SSDSDELN DS H SEGMENT LENGTH
SSDSID DS XL1 SEGMENT ID
SNSMFSD# EQU 4 4 = SEGMENT ID FOR SOURCE DATASET
DS XL1 *
SSDSQNAM DS CL44 SOURCE DATASET NAME
SSDDNAM DS CL8 SOURCE DNAME
SSDCATNM DS CL44 SOURCE CATALOG NAME
SSDDATCL DS CL8 SOURCE - DATA CLASS
SSDGMTCL DS CL8 SOURCE - MANAGEMENT CLASS
SSDSTGCL DS CL8 SOURCE - STORAGE CLASS
SSDTRK# DS F NUMBER OF TRACKS IN SOURCE FILE
SSDUSED# DS F NUMBER OF USED TRACKS IN SOURCE
SSDHARBA DS F VSAM HIGH ALLOCATED RBA
SSDHURBA DS F VSAM HIGH USED RBA
SSDHKURA DS F VSAM HIGH KEY RBA
SSDXTN# DS Y SOURCE NUMBER OF EXTENTS
SSDXTN# DS Y SOURCE NUMBER OF EXTENTS
SSDTYPE DS CL1 DATA SET TYPE (SEE EXWAATYP)
* EXWAATYP_NVSNM EQU C'A' NONVSAM DATASET
* EXWAATYP_GDG EQU C'B' GDG BASE
* EXWAATYP_CLUSTER EQU C'C' CLUSTER
* EXWAATYP_DATA EQU C'D' DATA COMPONENT
* EXWAATYP_ALIAI EQU C'E' ALTERNATE INDEX
* EXWAATYP_INDEX EQU C'I' INDEX COMPONENT
* EXWAATYP_PATH EQU C'R' PATH
* EXWAATYP_ALIAS EQU C'X' ALIAS NAME
SSDSTRP DS XL1 SOURCE STRIPE COUNT
SSDSSG DS XL2 SOURCE VOLUNIT
SSDLRCU DS XL2 SOURCE VOLUNIT
SSDBLZ DS XL2 SOURCE VOLUNIT
SSDIFM DS XL1 SOURCE VOLUNIT
SSD_CG DS CL8 SOURCE CONGROUP NAMES

1. The GPM command CONFIGPOOL is no longer supported.
SMF Record Layout (Mainframe Enablers 8.0)

SSD_CG2 DS CL8 . . .
SNSMFSRC_LEN EQU *-SNSMFSRC LENGTH OF SNSMFSRC
*
* TARGET DATASET INFORMATION *
*
SNSMFDTG DSECT ,
STDGLEN DS H SEGMENT LENGTH
STDGID DS XL1 SEGMENT ID
SNSMFDT# EQU 5 . 5 = SEGMENT ID FOR TARGET DATASET
DS XL1
*
STDDSNAM DS CL44 TARGET DATASET NAME
STDDNAM DS CL8 TARGET DDNAME
STDCATNM DS CL44 TARGET CATALOG NAME
STDDATCL DS CL8 NEW DATA CLASS (FROM ACS ROUTINES)
STDDMCTCL DS CL8 NEW MGMT CLASS (FROM ACS ROUTINES)
STDDSTGCL DS CL8 NEW STG CLASS (FROM ACS ROUTINES)
STDTRK# DS F NUMBER OF TRACKS IN TARGET FILE
STDXTN# DS Y TARGET NUMBER OF EXTENTS
STDSTG# DS XL1 TARGET STRIPE COUNT
STDDSG DS XL2 DSORG
STDLRCL DS XL2 LRECL
STDBLKZ DS XL2 BLOCKSIZE
STDRCM DS XL1 RECFM
STD_CG DS CL8 TARGET CONGROUP NAMES
STD_CG2 DS CL8 . . .
SNSMFDTG_LEN EQU *-SNSMFDTG LENGTH OF SNSMFDTG
*
* GATEKEEPER INFORMATION *
*
SNSMGTK DSECT ,
SGSKLEN DS H SEGMENT LENGTH
SGSKGID DS XL1 SEGMENT ID
SNSMFGK# EQU 6 . 6 = SEGMENT ID FOR GATEKEEPER
DS XL1
SGKVOL DS CL6 SPECIFIED GATEKEEPER VOLUME
SGKUNIT DS XL2 SPECIFIED GATEKEEPER UNIT ADDRESS
SGKUNIT DS CL8 SPECIFIED GATEKEEPER DNAME
SGKMHOP DS XL8 SPECIFIED GATEKEEPER MULTI-HOP LIST X
. (ALL X'FF' IF LOCAL GATEKEEPER)
SGKAVOL DS CL8 ACTUAL GATEKEEPER VOLUME
SGKUNIT DS XL2 ACTUAL GATEKEEPER UNIT ADDRESS
SNSMGTK_LEN EQU *-SNSMGTK LENGTH OF SNSMGTK
*
* SOURCE EXTENT LIST *
*
SNSMFSXL DSECT ,
SSXSOKEN LEN SEGMENT LENGTH
SSXSOLID DS XL1 SEGMENT ID
SNSMFSX# EQU 7 . 7 = SEGMENT ID FOR SOURCE EXTENTS
DS XL1
SSXTFRK# DS FL4 NUMBER OF TRACKS IN EXTENT
SSXUCB@ DS AL4 UCB ADDRESS
SSXVOL DS CL6 MVS VOLUME FOR THIS EXTENT
SSXCCUU DS XL2 CCUU
SSXBCCHH DS XL4 BEGINNING CCHH OF EXTENT
SSXECCHH DS XL4 ENDING CCHH OF EXTENT
SSXXINT# DS XL2 ENTENT NUMBER (RELATIVE TO 1)
SSXXTRK# DS FL4 STARTING TRACK NUMBER (REL. TO 0)
SSXVOL# DS XL1 VOLUME NUMBER (RELATIVE TO 1) X
. OR STRIPE NUMBER
SSXXER# DS CL12 SYMMETRIX SERIAL NUMBER (ALL 12)
SSXDEV# DS XL4 PHYSICAL DEVICE NUMBER
SSXCODE# DS XL2 MICROCODE LEVEL - 5065, 5066...
SNSMFSXL_LEN EQU *-SNSMFSXL LENGTH OF SNSMFSXL

364 TimeFinder/Clone Mainframe Snap Facility 8.0 and Higher Product Guide
* * TARGET EXTENT LIST *

SNSMFTXL DSECT ,
STXSGLEN DS H SEGMENT LENGTH
STXSGID DS XL1 SEGMENT ID
SNSMFTX# EQU 8 . 8 = SEGMENT ID FOR TARGET EXTENTS
  DS XL1
STXTRK# DS FL4 NUMBER OF TRACKS IN EXTENT
STXUCB@ DS AL4 UCB ADDRESS
STXVOL DS CL6 VOLSER FOR THIS EXTENT
STXCCUU DS XL2 MVS CCUU
STXECCHH DS XL4 BEGINNING CCHH OF EXTENT
STXECCHH DS XL4 ENDING CCHH OF EXTENT
STXTXTN# DS XL2 EXTENT NUMBER (RELATIVE TO 1)
STXSTRK# DS FL4 STARTING TRACK NUMBER (REL. TO 0)
STXTXTN# DS XL1 VOLUME NUMBER (RELATIVE TO 1) X
  OR STRIPE NUMBER
STXSER# DS CL12 SYMMETRIX SERIAL NUMBER (ALL 12)
STXDEV# DS XL4 PHYSICAL DEVICE NUMBER
STXMODE DS XL2 MICROCODE LEVEL - 5065, 5066...
SNSMFTXL_LEN EQU *-SNSMFTXL LENGTH OF SNSMFTXL

* * COPY EXTENT LIST *

SNSFCXCL DSECT ,
SCXSIGLEN DS H SEGMENT LENGTH
SCXSIGID DS XL1 SEGMENT ID
SNSFCX# EQU 9 . 9 = SEGMENT ID FOR COPY EXTENTS
  DS XL1
SCXSUCB@ DS AL4 SOURCE UCB ADDRESS
SCXCCUUU DS XL2 SOURCE UCB CCUU
SCXSYMDS DS XL4 SOURCE INTERNAL DEVICE NUMBER
SCXTUCB@ DS AL4 TARGET UCB ADDRESS
SCXTCCUU DS XL2 SOURCE UCB CCUU
SCXTSYMDS DS XL4 TARGET INTERNAL DEVICE NUMBER
SCXMHOP DS XL8 REMOTE MULTI-HOP LIST
SCXASSTR DS XL4 SOURCE CCCCCCCH (ASCENDING)
SCXATSTR DS XL4 TARGET CCCCCCCH (ASCENDING)
SCXSTR DS XL4 TARGET CCHH (IBM FORMAT)
SCXTRK DS XL4 SOURCE EXTENT NUMBER
SCXXTXTN# DS XL2 SOURCE EXTENT NUMBER
SCXPTTDS DS XL1 COPY TYPE NECESSARY
SCXCPYTY_SKIP EQU 0 . SKIP COPY
SCXCPYTY_MCODE EQU 10 . USE MICROCODE TO COPY
SCXCPYTY_EMFLASH EQU 20 . USE FLASH MICROCODE TO COPY
SCXCPYTY_EMCCOPY EQU 30 . USE EMCCOPY TO COPY
SCXCPYTY_EMMSNAP EQU 40 . USE ANTRQST-SNAPSHOT TO COPY
SCXCPYTY_IBMFLASH2 EQU 50 . USE ANTRQST-FLASHCOPY V2
SCXCPYTY_IBMFLASH EQU 60 . USE ANTRQST-FLASHCOPY TO COPY
SCXCPYTY_COPYCYL EQU 70 . USE COPYCYL TO COPY
SCXCPYTY_UTILITY EQU 80 . USE UTILITY TO COPY
SCXSTXTN# DS XL2 SOURCE EXTENT NUMBER
SCXXTXTN# DS XL2 TARGET EXTENT NUMBER
SNSFCXCL_LEN EQU *-SNSFCXCL LENGTH OF SNSFCXCL

MEND ,
SMF record sub-sections and TimeFinder actions

The following tables identify which sub-sections of the SMF record are present for each TimeFinder action. In determining segment length values, check for the values embedded in the record content.

Table 31 SMF Record sub-sections and TimeFinder actions

<table>
<thead>
<tr>
<th>CLEANUP EXTENT TRACK</th>
<th>ACTIVATE</th>
<th>QUERY VDEV</th>
<th>RESTORE VIRTUAL DEVICE</th>
<th>DESTROY EXTENT TRACK</th>
<th>QUERY DATASET</th>
<th>SNAP DATASET</th>
<th>STOP SNAP TO DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
<td>SCMSGLEN=42</td>
</tr>
<tr>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGIDN=1</td>
</tr>
<tr>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
<td>SCMSGLEN=80</td>
</tr>
<tr>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
<td>SCMSGIDN=2</td>
</tr>
<tr>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
<td>SCMSGLEN=428</td>
</tr>
<tr>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGIDN=3</td>
</tr>
<tr>
<td>SCMSGLEN=60</td>
<td>SCMSGLEN=60</td>
<td>SCMSGLEN=60</td>
<td>SCMSGLen=60</td>
<td>SCMSGLEN=60</td>
<td>SCMSGLEN=60</td>
<td>SCMSGLEN=60</td>
<td>SCMSGLEN=60</td>
</tr>
<tr>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
</tr>
<tr>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
<td>SCMSGLEN=154</td>
</tr>
<tr>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGIDN=5</td>
</tr>
<tr>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
<td>SCMSGLEN=40</td>
</tr>
<tr>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGIDN=6</td>
</tr>
<tr>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
</tr>
<tr>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
</tr>
<tr>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
<td>SCMSGLEN=53</td>
</tr>
<tr>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGIDN=7</td>
</tr>
<tr>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNAP VOLUME</th>
<th>STOP SNAP TO VOLUME</th>
<th>CONFIG</th>
<th>QUERY VOLUME</th>
<th>CLEANUP EXTENT TRACK</th>
<th>ACTIVATE</th>
<th>QUERY VDEV</th>
<th>RESTORE VIRTUAL DEVICE</th>
<th>RESET EXTENT TRACK</th>
<th>QUERY DATASET</th>
<th>SNAP DATASET</th>
<th>STOP SNAP TO DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMSGLEN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGIDN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=1</td>
</tr>
<tr>
<td>SCMSGIDN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=1</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
<td>SCMSGLEN=2</td>
</tr>
<tr>
<td>SCMSGLEN=3</td>
<td>SCMSGIDN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
<td>SCMSGLEN=3</td>
</tr>
<tr>
<td>SCMSGLEN=4</td>
<td>SCMSGIDN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
<td>SCMSGLEN=4</td>
</tr>
<tr>
<td>SCMSGLEN=5</td>
<td>SCMSGIDN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
<td>SCMSGLEN=5</td>
</tr>
<tr>
<td>SCMSGLEN=6</td>
<td>SCMSGIDN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
<td>SCMSGLEN=6</td>
</tr>
<tr>
<td>SCMSGLEN=7</td>
<td>SCMSGIDN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
<td>SCMSGLEN=7</td>
</tr>
<tr>
<td>SCMSGLEN=8</td>
<td>SCMSGIDN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
<td>SCMSGLEN=8</td>
</tr>
<tr>
<td>SCMSGLEN=9</td>
<td>SCMSGIDN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
<td>SCMSGLEN=9</td>
</tr>
</tbody>
</table>

Note: There may be multiple type 7, 8, or 9 segments.
APPENDIX D
TimeFinder REXX EXITS

This appendix contains the following topics:

- TimeFinder REXX Interface EXITS ................................................................. 368
- REXX keywords and parameters ................................................................. 369
- Using REXX EXECs ..................................................................................... 371
- REXX Examples ............................................................................................ 377
- Examples of REXX statement present in EMCSNAPO ................................. 379
**TimeFinder REXX Interface EXITS**

EMCSNAP allows users to either influence standard Dell EMC processing or to perform related external user-defined processing, using Dell EMC’s REXX Interface. This avoids the common problem in the customer environment where no one may understand how to take advantage of EXITS using the Assembler language.

The implementation points in the code will call a common REXX interface module, which will pass control to the customer REXX script, and then resume when the REXX script completes.

The parameters passed to the customer REXX script will be appropriate for the TimeFinder action being performed. The responses allowed from the REXX script are limited to the EXIT purpose.

The REXX scripts may be applied in any of the following ways:

1. They may be assembled into the site options table (EMCSNAPO) and will be executed in memory.
2. They may be present in a dataset (PDS or PDSE) identified in the site options table (EMCSNAPO) and will be executed from that dataset.
3. They may be present in the //SYSEXEC statement allocated to each job.

Some potential benefits of using REXX scripts include:

- Providing default GLOBAL statements for various users or job streams.
- Verifying that an entered pool name is appropriate for the user, or automatically providing the pool name for specific users or jobs.
- Verifying that the user-entered SMS class names are appropriate for the user, or automatically providing the SMS class names for specific users or jobs.
- Pass control to a user exit before and after the activate process, which allows them to potentially auto-quietce databases or systems for only the short period (seconds only) necessary to issue the activate request.
- Verifying that the target dataset name (from SNAP DATASET) is appropriate for the users, or automatically providing a dataset name.
- Verifying that a user is allowed to scratch dataset.
- View the SMF records as they are being written.
- Receive control when a device VARY command is being issued.
Macro EMCSNAPO is changed to support the specification of the REXX exits to be invoked. There is also a new MACRO associated with EMCSNAPO that is used to specify the “in-memory” REXX statements (REXXSTMT).

Table 32  REXX TF command parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Allowed Values</th>
<th>REXX EXEC Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;REXX_EXEC_DSNAME</td>
<td>44-char dataset name</td>
<td>N/A</td>
<td>Single dataset name that is dynamically allocated to SYSEXEC during EMCSNAP initialization. This dataset contains all of the exits to be invoked that are specified as &quot;YES&quot;.</td>
</tr>
<tr>
<td>&amp;REXX_ACTIVATE</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_DATASET_NEWNAME</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_GLOBAL</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_SCRATCH</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_SMS_DATACLAS</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
</tbody>
</table>
### Table 32  REXX TF command parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Allowed Values</th>
<th>REXX EXEC Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;REXX_SMS_MGMTCLAS</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_SMS_STORCLAS</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_TDEV_POOL</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_VARY_DEVICE</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_VDEV_POOL</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
<tr>
<td>&amp;REXX_WRITE_SMF</td>
<td>NO</td>
<td>YES</td>
<td>REXXSTMT label</td>
</tr>
</tbody>
</table>
Using REXX EXECs

REXX EXEC Name convention

The actual REXX code may be located in one of two places. The most efficient is to place it in the EMCSNAP Site Options Table. Then it is in resident in memory and no external I/O is required in order to execute the REXX.

As an alternative, the REXX code may be placed in a user-supplied library dataset (PDS or PDSE). Each exit has a pre-assigned name that will be executed from the dataset.

Activate (before and after) - ESNAACTV

There are several calls made to the activate exit. Each call is passed a unique ACTION/SERIAL# combination based on the call.

Single PRE-ACTIVATE: ACTION=ACT0 SERIAL#=blanks
This call is the first of the pre-activate calls. It is made just before the activate is going to occur all validation and setup is complete.

Controller PRE-ACTIVATE: ACTION=ACT2 SERIAL#=Symm Serial#
This call is made after the ACT0 call, and before the ACT4 call. The exit is called one time for each storage system that is involved in the activate.

Single PRE-ACTIVATE: ACTION=ACT4 SERIAL#=blanks
This call completes the pre-activate calls.

Single POST-ACTIVATE: ACTION=ACT5 SERIAL#=blanks
This call is the first of the post-activate calls. It is made immediately after the activate has occurred.

Controller POST-ACTIVATE: ACTION=ACT7 SERIAL#=Symm Serial#
This call is made after the ACT5 call, and before the ACT9 call. The exit is called one time for each storage system that is involved in the activate.

SINGLE POST-ACTIVATE: ACTION=ACT9 SERIAL#=blanks
This call completes the post-activate calls.

ESNAACTV arguments

Table 33 ESNAACTV arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>ACTION</td>
<td></td>
</tr>
<tr>
<td>ARG(2)</td>
<td>SERIAL#</td>
<td>Storage system serial number or blanks</td>
</tr>
</tbody>
</table>

No return values.
Global Statement - ESNAGBL

No arguments are supplied. Return values are allowed.

A valid GLOBAL statement that will be included in the user input stream must start with the word "GLOBAL" and follow the rules for syntax for the GLOBAL statement.

Dataset Newname Validation - ESNANEWD

Table 34  ESNANEWD arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>REQUEST</td>
<td>'SNAP DATASET' 'SNAP VOLUME' 'DESTROY' 'CLEANUP' 'DEBUG' 'STOP DATASET' 'STOP VOLUME' 'RESTORE' 'ACTIVATE' 'QUERY DATASET' 'QUERY SAVEDEV' 'QUERY VDEV' 'CONFIG' 'QUERY VOLUME' 'SERIAL' 'PARALLEL' 'CONFIGPOOL'a 'QUERY GROUP' 'DEFINE GROUP' 'END GROUP' 'DELETE GROUP' 'GROUP' 'QUERY GLOBAL'</td>
</tr>
<tr>
<td>ARG(2)</td>
<td>SDSNAME</td>
<td>Source dataset name</td>
</tr>
<tr>
<td>ARG(3)</td>
<td>TDSNAME</td>
<td>Target dataset name</td>
</tr>
<tr>
<td>ARG(4)</td>
<td>DSNTYPE</td>
<td>'NONVSAM' 'GDG' 'CLUSTER' 'DATA' 'INDEX' 'AIX' 'MSTRCAT' 'PATH' 'USERCAT' 'ALIAS'</td>
</tr>
</tbody>
</table>

a. The GPM command CONFIGPOOL is no longer supported.

Return Values are allowed. The return value may be the replacement Target Dataset Name, or blanks to leave it unchanged.
## Scratch Dataset - ESNASCRA

### Table 35  ESNASCRA arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ARG(1)   | REQUEST | 'SNAP DATASET'  
|          |        | 'SNAP VOLUME'  
|          |        | 'DESTROY'  
|          |        | 'CLEANUP'  
|          |        | 'DEBUG'  
|          |        | 'STOP DATASET'  
|          |        | 'STOP VOLUME'  
|          |        | 'RESTORE'  
|          |        | 'ACTIVATE'  
|          |        | 'QUERY DATASET'  
|          |        | 'QUERY SAVEDEV'  
|          |        | 'QUERY VDEV'  
|          |        | 'CONFIG'  
|          |        | 'QUERY VOLUME'  
|          |        | 'SERIAL'  
|          |        | 'PARALLEL'  
|          |        | 'CONFIGPOOL'  
|          |        | 'QUERY GROUP'  
|          |        | 'DEFINE GROUP'  
|          |        | 'END GROUP'  
|          |        | 'DELETE GROUP'  
|          |        | 'GROUP'  
|          |        | 'QUERY GLOBAL'  |

| ARG(2)   | DSNAME | Name of dataset that is being scratched. |
| ARG(3)   | VOLSER#1 | The first volser of the dataset being scratched. |
| ARG(4)   | DATADSN | For clusters, the name of the data component. |
| ARG(5)   | DATAVOL#1 | The first volser of the data component. |
| ARG(6)   | INDXDSN | For clusters, the name of the index component. |
| ARG(7)   | INDXVOL#1 | The first volser of the index component. |

- The GPM command CONFIGPOOL is no longer supported.

### Return Values are allowed:

- 0 = No action, continue.
- 4 = Do not scratch the dataset. Issue message ESNP071W. Sets the job high return code = 4.
- 8 = Do not scratch the dataset. Issue message ESNP071E. Sets the job high return code = 8.
**Table 36** contains arguments for the three SMS exits: ESNASDAT (SMS Dataclas), ESNASMGTS (SMS Mgmtclas), ESNASTG (SMS Storclas).

### Table 36 SMS Class Name Validation arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>REQUEST</td>
<td>'SNAP DATASET', 'SNAP VOLUME', 'DESTROY', 'CLEANUP', 'DEBUG', 'STOP DATASET', 'STOP VOLUME', 'RESTORE', 'ACTIVATE', 'QUERY DATASET', 'QUERY SAVEDEV', 'QUERY VDEV', 'CONFIG', 'QUERY VOLUME', 'SERIAL', 'PARALLEL', 'CONFIGPOOL', 'QUERY GROUP', 'DEFINE GROUP', 'END GROUP', 'DELETE GROUP', 'GROUP', 'QUERY GLOBAL'</td>
</tr>
<tr>
<td>ARG(2)</td>
<td>CLASSNAME</td>
<td>SMS class name as inputted by the user. May be blank if not specified. May be 'COPYSMS' if COPYSMS was requested by the user.</td>
</tr>
<tr>
<td>ARG(3)</td>
<td>SDSNAME</td>
<td>Source dataset name as inputted by the user. (Before expanding wildcards and resolving names). May be blank if DDNAME was specified.</td>
</tr>
<tr>
<td>ARG(4)</td>
<td>SDDNAME</td>
<td>Source DDNAME as inputted by the user. May be blank if DSNAME was specified.</td>
</tr>
<tr>
<td>ARG(5)</td>
<td>TDSNAME</td>
<td>Target dataset name as inputted by the user. (Before expanding wildcards and resolving names). May be blank if DDNAME was specified.</td>
</tr>
<tr>
<td>ARG(6)</td>
<td>TDDNAME</td>
<td>Target DDNAME as inputted by the user. May be blank if DSNAME was specified.</td>
</tr>
</tbody>
</table>

a. The GPM command CONFIGPOOL is no longer supported.

Return Values are allowed:

- Blank = make no changes
'COPYSMS' = specify COPYSMS for the Classname

String = use this as the classname for this request. Must be a valid SMS classname.

TDEV Pool Name Validation

Table 37 TDEV Pool Name Validation arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>POOLNAME</td>
<td>User specified poolname</td>
</tr>
<tr>
<td>ARG(2)</td>
<td>SRCVOL</td>
<td>The source volser (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
<tr>
<td>ARG(3)</td>
<td>SRCUNIT</td>
<td>The source CCUU (if specified) or zeros ('0000') if not specified.</td>
</tr>
<tr>
<td>ARG(4)</td>
<td>TGTVOl</td>
<td>The target volser (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
<tr>
<td>ARG(5)</td>
<td>TGTUNIT</td>
<td>The target CCUU (if specified) or zeros ('0000') if not specified.</td>
</tr>
<tr>
<td>ARG(6)</td>
<td>NEWVOL</td>
<td>The NEWVOLSER (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
</tbody>
</table>

Return Values are allowed.

The result returned may be blank, or a valid poolname to be used. If blank, the user-supplied poolname is not changed.

VDEV Pool Name Validation

Table 38 VDEV Pool Name Validation arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>POOLNAME</td>
<td>User specified poolname</td>
</tr>
<tr>
<td>ARG(2)</td>
<td>SRCVOL</td>
<td>The source volser (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
<tr>
<td>ARG(3)</td>
<td>SRCUNIT</td>
<td>The source CCUU (if specified) or zeros ('0000') if not specified.</td>
</tr>
<tr>
<td>ARG(4)</td>
<td>TGTVOl</td>
<td>The target volser (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
<tr>
<td>ARG(5)</td>
<td>TGTUNIT</td>
<td>The target CCUU (if specified) or zeros ('0000') if not specified.</td>
</tr>
<tr>
<td>ARG(6)</td>
<td>NEWVOL</td>
<td>The NEWVOLSER (if specified) or '<em>OMIT</em>' if not specified.</td>
</tr>
</tbody>
</table>

Return values are allowed.

The result returned may be blank, or a valid poolname to be used. If blank, the user-supplied poolname is not changed.
VARY Device Online/Offline

Table 39  VARY Device Online/Offline arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>REQUEST</td>
<td>'SNAP DATASET'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'SNAP VOLUME'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'DESTROY'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'CLEANUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'DEBUG'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'STOP DATASET'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'STOP VOLUME'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'RESTORE'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'ACTIVATE'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY DATASET'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY SAVEDEV'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY VDEV'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'CONFIG'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY VOLUME'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'SERIAL'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'PARALLEL'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'CONFIGPOOL'(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY GROUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'DEFINE GROUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'END GROUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'DELETE GROUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'GROUP'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'QUERY GLOBAL'</td>
</tr>
<tr>
<td>ARG(2)</td>
<td>OPERATION</td>
<td>&quot;ONLINE&quot; or &quot;OFFLINE&quot;</td>
</tr>
<tr>
<td>ARG(3)</td>
<td>COMMAND</td>
<td>The actual VARY command that will be issued.</td>
</tr>
<tr>
<td>ARG(4)</td>
<td>VOLSER</td>
<td>The device volser.</td>
</tr>
<tr>
<td>ARG(5)</td>
<td>CCUU</td>
<td>The device CCUU.</td>
</tr>
<tr>
<td>ARG(6)</td>
<td>UCB@</td>
<td>The device UCB@ (in display hex).</td>
</tr>
</tbody>
</table>

a. The GPM command CONFIGPOOL is no longer supported.

Return values are allowed.

The VARY command to be used MUST be returned. If no command is returned, no vary will be issued. The returned command must be less than 32 characters.
Write SMF

Table 40 Write SMF arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG(1)</td>
<td>@SMF_RECORD</td>
<td>The address of the SMF record that is to be written. The address is in display hex format (that is, the address x'12345678' will be presented as a character string c'12345678').</td>
</tr>
</tbody>
</table>

Return values are allowed:

- 0 - Continue
- 4 - Do not write the SMF record.

REXX Examples

Sample ESNAGLBL

/* REXX */
SAY ARG()
DO N=1 TO ARG()
    SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
GLOBAL = ' GLOBAL DEBUG ( EXTRA ) '
RETURN GLOBAL

Sample ESNASCRA

/* REXX */
SAY ARG()
DO N=1 TO ARG()
    SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7
SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6 ARG7
REQUEST = ARG(1)
DSNAME = ARG(2)
VOLSER = ARG(3)
DATADSN = ARG(4)
DATAVOL = ARG(5)
INDXDSN = ARG(6)
INDXVOL = ARG(7)
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
IF DSNAME="HLQ.TESTING.PDS2"
    THEN RETURN 4
RETURN 0
Sample ESNATPOL

/* REXX */
SAY ARG()
DO N=1 TO ARG()
   SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
POOLNAME = ''
IF SUBSTR(JOBNAME,1,3) = 'HLQ' THEN
   POOLNAME = 'ABCDE'
RETURN POOLNAME

Sample ESNAVARY

/* REXX */
SAY ARG()
DO N=1 TO ARG()
   SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
REQUEST=ARG(1)
OPERATION=ARG(2)
COMMAND=ARG(3)
VOLSER=ARG(4)
CCUU=ARG(5)
UCB@=ARG(6)
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
RETURN COMMAND' TEST '

Sample ESNAVPOL

/* REXX */
SAY ARG()
DO N=1 TO ARG()
   SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
POOLNAME = ''
IF SUBSTR(JOBNAME,1,3) = 'HLQ' THEN
   POOLNAME = 'FGHIJ'
RETURN POOLNAME

Sample ESNAWSMF

/* REXX */
SAY ARG()
DO N=1 TO ARG()
   SAY 'PARM 'N' IS ' ARG(N)
END
PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6
JOBNAME = MVSVAR('SYMDEF','JOBNAME')
RETURN 0
Examples of REXX statement present in EMCSNAPO

```
*---------------------------------------------------------------------
* *
* PROGRAM DEFAULT OPTION SETTINGS - CHANGABLE BY CUSTOMER *
* *
*---------------------------------------------------------------------
*
EMCSNAPO DSECT=NO, X
  REXX_EXEC_NAME=HLQ.CLIST, X
  REXX_DATASET_NAME=NEWNAME, X
  REXX_GLOBAL=YES, X
  REXX_WRITE_SMF=SMF
** REXX_GLOBAL=GLOBAL,
** REXX_SCRATCH=SCRATCH,
** REXX_VARY_DEVICE=VARY_DEVICE
** REXX_TDEV_POOL=TDEV_POOL,
** REXX_VDEV_POOL=VDEV_POOL
NEWNAME REXXSTMT 'ESNANEWD: PROCEDURE'
  REXXSTMT 'SAY ARG()' 'DO N=1 TO ARG()' 'END'
  REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6'
  REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6'
  REXXSTMT 'REQUEST=ARG(1)'
  REXXSTMT 'DSNAME=ARG(2)'
  REXXSTMT 'TDSNAME=ARG(3)'
  REXXSTMT 'DSNTYPE=ARG(4)'
  REXXSTMT 'NEWNAME = '''
  REXXSTMT 'JOBNAME = MVSVAR(''SYMDEF'',''JOBNAME'')'
  REXXSTMT 'IF SUBSTR(SDSNAME,1,9) = ''HLQ.KSDS4'' THEN'
  REXXSTMT ' NEWNAME = ''HLQ.KSDS5''SUBSTR(SDSNAME,10,44)'
  REXXSTMT 'IF SUBSTR(TDSNAME,1,9) = ''HLQ.KSDS5'' THEN'
  REXXSTMT ' NEWNAME = ''HLQ.KSDS6''SUBSTR(TDSNAME,10,44)'
  REXXSTMT 'NEWNAME = STRIP(NEWNAME)'
  REXXSTMT 'RETURN NEWNAME '
TDEV_POOL REXXSTMT 'ESNATPOL: PROCEDURE'
  REXXSTMT 'SAY ARG()' 'DO N=1 TO ARG()' 'END'
  REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6'
  REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6'
  REXXSTMT 'REQUEST=ARG(1)'
  REXXSTMT 'DSNAME=ARG(2)'
  REXXSTMT 'TDSNAME=ARG(3)'
  REXXSTMT 'DSNTYPE=ARG(4)'
  REXXSTMT 'POOLNAME = '''
  REXXSTMT 'JOBNAME = MVSVAR(''SYMDEF'',''JOBNAME'')'
  REXXSTMT 'IF SUBSTR(JOBNAME,1,3) = ''HLQ'' THEN'
  REXXSTMT ' POOLNAME = ''ABCDE'''
  REXXSTMT 'RETURN POOLNAME '
VDEV_POOL REXXSTMT 'ESNAVPOL: PROCEDURE'
  REXXSTMT 'SAY ARG()' 'DO N=1 TO ARG()' 'END'
  REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6'
  REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6'
  REXXSTMT 'REQUEST=ARG(1)'
  REXXSTMT 'DSNAME=ARG(2)'
  REXXSTMT 'TDSNAME=ARG(3)'
  REXXSTMT 'DSNTYPE=ARG(4)'
  REXXSTMT 'POOLNAME = '''
  REXXSTMT 'JOBNAME = MVSVAR(''SYMDEF'',''JOBNAME'')'
  REXXSTMT 'IF SUBSTR(JOBNAME,1,3) = ''HLQ'' THEN'
  REXXSTMT ' POOLNAME = ''FGHIJ'''
  REXXSTMT 'RETURN POOLNAME '
VARY_DEVICE REXXSTMT 'ESNAVARY: PROCEDURE'
  REXXSTMT 'SAY ARG()' 'DO N=1 TO ARG()'
```

Examples of REXX statement present in EMCSNAPO 379
REXXSTMT ' SAY 'PARM'N' IS ' ARG(N)' 
REXXSTMT ' END ' 
REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6' 
REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6' 
REXXSTMT 'REQUEST=ARG(1)' 
REXXSTMT 'OPERATION=ARG(2)' 
REXXSTMT 'COMMAND=ARG(3)' 
REXXSTMT 'VOLSER=ARG(4)' 
REXXSTMT 'CCUU=ARG(5)' 
REXXSTMT 'UCB@=ARG(6)' 
REXXSTMT 'JOBNAME = MVSVAR(‘’SYMDEF’’,’’JOBNAME’’)' 
REXXSTMT 'RETURN COMMAND’ TEST ’ ' 

GLOBAL 
REXXSTMT 'ESNAGLBL: PROCEDURE' 
REXXSTMT 'SAY ARG()' 
REXXSTMT 'DO N=1 TO ARG()' 
REXXSTMT ' SAY 'PARM'N' IS ' ARG(N)' 
REXXSTMT ' END ' 
REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6' 
REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6' 
REXXSTMT 'REQUEST = ARG(1)' 
REXXSTMT 'DSNAME = ARG(2)' 
REXXSTMT 'VOLSER = ARG(3)' 
REXXSTMT 'DATADSN = ARG(4)' 
REXXSTMT 'INDEXDSN = ARG(5)' 
REXXSTMT 'INDEXVOL = ARG(6)' 
REXXSTMT 'INDEXXDSN = ARG(7)' 
REXXSTMT 'JOBNAME = MVSVAR(‘’SYMDEF’’,’’JOBNAME’’)' 
REXXSTMT 'GLOBAL =’ GLOBAL DEBUG ( EXTRA ) ’ ' 
REXXSTMT 'RETURN GLOBAL' 

SCRATCH 
REXXSTMT 'ESNASCRA: PROCEDURE' 
REXXSTMT 'SAY ARG()' 
REXXSTMT 'DO N=1 TO ARG()' 
REXXSTMT ' SAY 'PARM'N' IS ' ARG(N)' 
REXXSTMT ' END ' 
REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7' 
REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6 ARG7' 
REXXSTMT 'REQUEST = ARG(1)' 
REXXSTMT 'DSNAME = ARG(2)' 
REXXSTMT 'VOLSER = ARG(3)' 
REXXSTMT 'DATADSN = ARG(4)' 
REXXSTMT 'DATAVOL = ARG(5)' 
REXXSTMT 'INDEXDSN = ARG(6)' 
REXXSTMT 'INDEXVOL = ARG(7)' 
REXXSTMT 'INDEXXDSN = ARG(8)' 
REXXSTMT 'JOBNAME = MVSVAR(‘’SYMDEF’’,’’JOBNAME’’)' 
REXXSTMT 'IF DSNAME=“HLQ.TESTING.PDS2”' 
REXXSTMT ' THEN RETURN 4' 
REXXSTMT 'RETURN 0' 

WRITE_SMF 
REXXSTMT 'ESNAWSMF: PROCEDURE' 
REXXSTMT 'SAY ARG()' 
REXXSTMT 'DO N=1 TO ARG()' 
REXXSTMT ' SAY 'PARM'N' IS ' ARG(N)' 
REXXSTMT ' END ' 
REXXSTMT 'PARSE ARG ARG1,ARG2,ARG3,ARG4,ARG5,ARG6' 
REXXSTMT 'SAY ARG1 ARG2 ARG3 ARG4 ARG5 ARG6' 
REXXSTMT 'JOBNAME = MVSVAR(‘’SYMDEF’’,’’JOBNAME’’)' 
REXXSTMT 'RETURN 4 ’ - DON’T WRITE SMF RECORD !!
END