## CONTENTS

**Preface**

**Chapter 1 Introduction**
- Mainframe Enablers and Consistency Groups .................................................. 18
- Licensing ........................................................................................................... 18
- Introduction to Consistency Groups ................................................................. 19
- Purpose ............................................................................................................. 19
- Features ............................................................................................................ 24
- APIs and utilities ............................................................................................. 25
- Security............................................................................................................. 26
- Note for TDMF/FDRPAS users ....................................................................... 26

**Chapter 2 Getting Started**
- Post-installation ............................................................................................... 28
- Configuring ConGroup ..................................................................................... 28
  - Creating configuration file ........................................................................... 28
  - Configuration parameter types .................................................................. 29
- Global configuration parameters ..................................................................... 30
  - Syntax conventions ....................................................................................... 30
  - ALLOW_SHARED_R1S.................................................................................. 30
  - AUTO_REFRESH .......................................................................................... 31
  - CAXOPTS .................................................................................................... 31
  - CLOCK ........................................................................................................ 32
  - CLOCKN ..................................................................................................... 32
  - COUPLEDS_ALLOWED ............................................................................... 33
  - DISABLE_AT_SHUTDOWN .......................................................................... 33
  - DISABLE_ON_VERIFY_ERROR ..................................................................... 34
  - DISPLAY_CONGROUP_LISTOPT ............................................................... 34
  - DISPLAY_SAFAUTH_SUCCESS .................................................................... 35
  - GLOBAL ...................................................................................................... 35
  - MODE ......................................................................................................... 36
  - MSGLEVEL ................................................................................................ 37
  - PAGEDEV_ALLOWED .................................................................................. 37
  - PAVO .......................................................................................................... 38
  - REMSPLIT_INTERVAL ................................................................................ 38
  - REMSPLIT_OPTION ................................................................................... 38
  - RESUME_INTERVAL .................................................................................... 39
  - RESUME_OPTION ....................................................................................... 39
  - SAF_CLASS .................................................................................................. 40
  - SAF_PROFILE ............................................................................................. 40
  - SEMISYNC_ALLOWED ................................................................................. 40
  - USE_NR_ON_TO ......................................................................................... 41
  - VERIFY_INTERVAL ...................................................................................... 41
- Consistency group-specific configuration parameters ................................... 42
  - Syntax conventions ....................................................................................... 42
  - ALLOW_SHARED_R1S.................................................................................. 42
  - ALLOWABLE_MSS ...................................................................................... 42
  - CAX .............................................................................................................. 43
Contents

Adding devices to consistency group .................................................... 82
Adding/removing devices dynamically .................................................. 86
Using multiple subchannel sets .......................................................... 88
Protecting specific R1-R2 pairs .......................................................... 90
ConGroup in Concurrent SRDF ............................................................ 90
R1 sharing by mirror ........................................................................... 91
Example............................................................................................. 93
Verifying consistency groups .............................................................. 94
Manual verification ............................................................................. 94
Automatic verification ......................................................................... 94
Enabling/disabling consistency groups ............................................... 95
Enabling consistency groups............................................................... 95
Disabling consistency groups............................................................. 95
Monitoring consistency groups ........................................................... 97
What is ECA? ..................................................................................... 98
Monitoring with CAX enabled ............................................................ 98
Tripping consistency groups ............................................................... 99
Suspending consistency group operations ........................................... 100
Splitting remote BCVs ..................................................................... 101
Trip API ............................................................................................ 102
Resuming consistency group operations ............................................. 105
Resume prerequisites ......................................................................... 105
Resuming after consistency group trip ................................................. 106
Resume process ................................................................................ 106
Resume interval ................................................................................ 107
Resume messages ............................................................................. 107
Canceling a resume process ............................................................... 107

Chapter 4 Swapping Consistency Groups with CAX

Overview............................................................................................ 110
Site prerequisites................................................................................ 110
Configuring CAX ............................................................................... 111
Example............................................................................................ 112
Summary of operations ....................................................................... 113
CAX process ..................................................................................... 114
Validation ......................................................................................... 114
Monitoring ....................................................................................... 114
Handling unexpected conditions ....................................................... 115
Handling devices with Concurrent Copy/SNAP sessions .................. 117
Handling unknown online FROM devices ....................................... 117
Handling LRO online TO devices ...................................................... 118
Checking LPAR/path group counts .................................................... 119
Handling lost communications with owner LPAR ......................... 119
Routing CAX messages ................................................................... 120
Resuming operations after swap ....................................................... 121
Swapping back immediately ............................................................. 121
Swapping back later .......................................................................... 125

Chapter 5 Managing ConGroup Environment

Summary of operations ....................................................................... 130
Coordinating ConGroup instances ..................................................... 130
Managing owner LPARs ................................................................. 130
Adding/removing VMAX systems ..................................................... 130
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing gatekeepers</td>
<td>130</td>
</tr>
<tr>
<td>Viewing/updating configuration</td>
<td>131</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>131</td>
</tr>
<tr>
<td>Viewing ConGroup status</td>
<td>132</td>
</tr>
<tr>
<td>Coordinating ConGroup instances</td>
<td>133</td>
</tr>
<tr>
<td>Single/multi-LPAR mode</td>
<td>133</td>
</tr>
<tr>
<td>Managing multiple ConGroup instances</td>
<td>135</td>
</tr>
<tr>
<td>Managing owner LPARs</td>
<td>138</td>
</tr>
<tr>
<td>Owner LPAR requirements</td>
<td>138</td>
</tr>
<tr>
<td>Changing owner LPAR</td>
<td>138</td>
</tr>
<tr>
<td>Adding/removing VMAX systems</td>
<td>140</td>
</tr>
<tr>
<td>Adding VMAX system manually</td>
<td>140</td>
</tr>
<tr>
<td>Removing VMAX system manually</td>
<td>140</td>
</tr>
<tr>
<td>Managing gatekeepers</td>
<td>141</td>
</tr>
<tr>
<td>Pinning/unpinning gatekeepers</td>
<td>141</td>
</tr>
<tr>
<td>Gatekeeper server</td>
<td>141</td>
</tr>
<tr>
<td>Managing the gatekeeper pool</td>
<td>142</td>
</tr>
<tr>
<td>Refreshing ConGroup configuration</td>
<td>144</td>
</tr>
<tr>
<td>Refreshing automatically</td>
<td>144</td>
</tr>
<tr>
<td>Refreshing manually</td>
<td>146</td>
</tr>
<tr>
<td>Updating consistency group state</td>
<td>147</td>
</tr>
<tr>
<td>Cleaning up environment with ECGUTIL</td>
<td>148</td>
</tr>
<tr>
<td>Overview</td>
<td>148</td>
</tr>
<tr>
<td>Running ECGUTIL utility</td>
<td>149</td>
</tr>
<tr>
<td>Setting/clearing device flags with ECGUTIL</td>
<td>152</td>
</tr>
<tr>
<td>Diagnostics and troubleshooting</td>
<td>155</td>
</tr>
</tbody>
</table>

**Chapter 6**  
**ConGroup Reports**

- ConGroup Environment report ................................................................. 158  
  Report fields ....................................................................................... 159  
- Consistency Group — Summary report .................................................. 160  
  Variable GLOBAL SETTINGS lines .......................................................... 161  
  Report fields ....................................................................................... 162  
- Consistency Group — Details report .................................................... 163  
  Details section fields ......................................................................... 164  
  Summary section fields ........................................................................ 164  
- SMFID report .......................................................................................... 165  
  Gatekeeper report .................................................................................. 165  

**Chapter 7**  
**Database Considerations**

- Introduction ............................................................................................ 168  
- ConGroup and database I/O ..................................................................... 168  
- Database consistency group suspensions .............................................. 168  
- Device definitions .................................................................................. 169  
- Expanding to new devices ...................................................................... 169  
- Other configuration parameters .......................................................... 170  
- Recovery site startup ............................................................................. 170  
- DB2 considerations ................................................................................ 171  
  Device definitions .................................................................................. 171  
  DB2 consistency group example ............................................................. 172  
  Recovery site startup ............................................................................. 172  
- IMS considerations ................................................................................ 173  
  Device definitions .................................................................................. 173  

*Consistency Groups for z/OS 8.1 Product Guide*
### IMS Consistency Group Example

Consistency group example .......................................................... 173
Recovery site startup ............................................................... 174
Mixed database considerations .................................................. 175
Device definitions ........................................................................ 175
Hybrid consistency group example .............................................. 175
Recovery site startup ............................................................... 176

### Chapter 8 Command Reference

Syntax conventions ....................................................................... 178
ConGroup commands .................................................................. 179
ADD ............................................................................................. 179
#ADD CONTROLLER ................................................................. 181
CANCEL ....................................................................................... 182
DAS ............................................................................................... 183
DELETE ........................................................................................ 184
#DELETE CONTROLLER ........................................................... 185
DISABLE ...................................................................................... 186
DISPLAY CONGROUP ................................................................. 187
DISPLAY ENVIRONMENT ............................................................ 188
#DISPLAY GATEKEEPERS .......................................................... 188
ENABLE ....................................................................................... 189
HELP ........................................................................................... 190
LA ............................................................................................... 190
MOVEOWNER ............................................................................. 191
#PIN ........................................................................................... 192
QUERY CON .............................................................................. 192
REFRESH ..................................................................................... 193
REMSPLIT .................................................................................. 193
RESET .......................................................................................... 194
RESUME ..................................................................................... 195
SET VERIFY_INTERVAL .............................................................. 196
STOP ........................................................................................... 196
TAKEOVER .................................................................................. 197
TRIP .............................................................................................. 197
#UNPIN ...................................................................................... 198
VERIFY ....................................................................................... 198
ECGUUTIL commands ............................................................... 199
ADD DEVICES ............................................................................. 200
AG ............................................................................................... 201
CG ............................................................................................... 201
DG ............................................................................................... 202
SET GROUP ............................................................................... 202
SET MSGLEVEL ....................................................................... 203
STOP ........................................................................................... 203
FIGURES

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependent write logic</td>
<td>19</td>
</tr>
<tr>
<td>Single VMAX configuration</td>
<td>20</td>
</tr>
<tr>
<td>Communication failure in multiple VMAX configuration</td>
<td>21</td>
</tr>
<tr>
<td>Basic consistency group</td>
<td>22</td>
</tr>
<tr>
<td>Using ConGroup in Concurrent SRDF configurations</td>
<td>23</td>
</tr>
<tr>
<td>Extended consistency group</td>
<td>68</td>
</tr>
<tr>
<td>Basic consistency group</td>
<td>22</td>
</tr>
<tr>
<td>Consistency group lifetime</td>
<td>69</td>
</tr>
<tr>
<td>Active state</td>
<td>71</td>
</tr>
<tr>
<td>Suspend pending state</td>
<td>71</td>
</tr>
<tr>
<td>Suspended state</td>
<td>72</td>
</tr>
<tr>
<td>Remote split in process state</td>
<td>72</td>
</tr>
<tr>
<td>Resume in process state</td>
<td>72</td>
</tr>
<tr>
<td>Using ConGroup in Concurrent SRDF configurations</td>
<td>90</td>
</tr>
<tr>
<td>Verify statement and output</td>
<td>94</td>
</tr>
<tr>
<td>Verbose verification information</td>
<td>94</td>
</tr>
<tr>
<td>SWAP command output</td>
<td>122</td>
</tr>
<tr>
<td>ConGroup Environment report</td>
<td>158</td>
</tr>
<tr>
<td>Consistency Group — Summary report</td>
<td>160</td>
</tr>
<tr>
<td>Consistency Group — Details report</td>
<td>163</td>
</tr>
<tr>
<td>Consistency Group — Details report, Summary section</td>
<td>164</td>
</tr>
<tr>
<td>SMFID report</td>
<td>165</td>
</tr>
<tr>
<td>Gatekeeper report</td>
<td>165</td>
</tr>
<tr>
<td>DB2 consistency group example</td>
<td>172</td>
</tr>
<tr>
<td>IMS consistency group example</td>
<td>173</td>
</tr>
<tr>
<td>Hybrid consistency group example</td>
<td>175</td>
</tr>
</tbody>
</table>
Figures
# TABLES

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Managing consistency groups</td>
<td>74</td>
</tr>
<tr>
<td>2 Managing consistency group members</td>
<td>75</td>
</tr>
<tr>
<td>3 Protecting specific R1-R2 pairs</td>
<td>75</td>
</tr>
<tr>
<td>4 CAX operations</td>
<td>113</td>
</tr>
<tr>
<td>5 Coordinating consistency groups</td>
<td>130</td>
</tr>
<tr>
<td>6 Coordinating consistency groups</td>
<td>130</td>
</tr>
<tr>
<td>7 Adding/removing VMAX systems</td>
<td>130</td>
</tr>
<tr>
<td>8 Managing gatekeepers</td>
<td>130</td>
</tr>
<tr>
<td>9 Viewing/updating configuration</td>
<td>131</td>
</tr>
<tr>
<td>10 Miscellaneous operations</td>
<td>131</td>
</tr>
<tr>
<td>11 ConGroup Environment report fields</td>
<td>159</td>
</tr>
<tr>
<td>12 Consistency Group — Summary report, swap option values</td>
<td>161</td>
</tr>
<tr>
<td>13 Consistency Group — Summary report fields</td>
<td>162</td>
</tr>
<tr>
<td>14 Consistency Group — Details report, details section fields</td>
<td>164</td>
</tr>
</tbody>
</table>
As part of an effort to improve its product lines, 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 on product features.

Contact your 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 EMC online support website. Check the EMC online support website to ensure that you are using the latest version of this document.

Purpose

The *EMC Consistency Groups for z/OS Product Guide* is for use with the following products:

- EMC Consistency Groups for z/OS
- EMC AutoSwap for z/OS

Coverage

This document describes Consistency Groups for z/OS when used in the following VMAX operating environments supported by Mainframe Enablers 8.1:

- HYPERMAX OS 5977
- Enginuity 5876
- Enginuity 5773

Note: Refer to prior versions of the *Consistency Groups for z/OS Product Guide* for information pertaining to other Enginuity levels.

Related documentation

The following documents provide additional information about Mainframe Enablers:

- *Mainframe Enablers Release Notes*
- *Mainframe Enablers Installation and Customization Guide*
- *Mainframe Enablers Message Guide*
- *Consistency Groups for z/OS Product Guide*
- *ResourcePak Base for z/OS Product Guide*
- *SRDF Host Component for z/OS Product Guide*
- *TimeFinder SnapVX and zDP Product Guide*
- *TimeFinder/Clone Mainframe Snap Facility Product Guide*

---

1. Enginuity 5773 is not supported in SRDF configurations that include a VMAX system running HYPERMAX OS 5977.
The following documents provide additional information:

- **EMC VMAX All Flash Product Guide** — Documents the features and functions of the VMAX All Flash arrays.
- **HYPERMAX OS for EMC VMAX All Flash and EMC VMAX3 Family Release Notes** — Describe new features and any known limitations.
- **EMC VMAX3 Family with HYPERMAX OS Product Guide** — Documents the features and functions of the VMAX3 100K, 200K, and 400K arrays.
- **EMC VMAX Family with Enginuity Product Guide** — Documents the features and functions of the VMAX 10K, 20K, and 40K arrays.
- **EMC VMAX Family Viewer for Desktop and iPad®** — Illustrates system hardware, incrementally scalable system configurations, and available host connectivity offered for VMAX arrays.
- **E-Lab™ Interoperability Navigator (ELN)** — Provides a web-based interoperability and solution search portal. You can find the ELN at [https://elabnavigator.EMC.com](https://elabnavigator.EMC.com).
- **SolVe Desktop** — Provides links to documentation, procedures for common tasks, and connectivity information for 2-site and 3-site SRDF configurations. To download the SolVe Desktop tool, go to EMC Online Support at [https://support.EMC.com](https://support.EMC.com) and search for SolVe Desktop. Download the SolVe Desktop and load the VMAX Family and DMX procedure generator.

You need to authenticate (authorize) your SolVe Desktop. Once it is installed, please familiarize yourself with the information under Help tab.

### Conventions used in this document

EMC uses the following conventions for special notices:

<table>
<thead>
<tr>
<th><strong>CAUTION</strong></th>
<th>CAUTION, used with the safety alert symbol, indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note</strong></td>
<td>A note presents information that is important, but not hazard-related.</td>
</tr>
<tr>
<td><strong>IMPORTANT</strong></td>
<td>An important notice contains information essential to software or hardware operation.</td>
</tr>
</tbody>
</table>
Typographical conventions

EMC uses the following type style conventions in this document:

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

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

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

**Italic**  
Used in all text (including procedures) for:
- Full titles of publications referenced in text
- Emphasis, for example, a new term
- Variables

**Courier**  
Used for:
- System output, such as an error message or script
- URLs, complete paths, filenames, prompts, and syntax when shown outside of running text

**Courier bold**  
Used for specific user input, such as commands

**Courier italic**  
Used in procedures for:
- Variables on the command line
- User input variables

< >  
Angle brackets enclose parameter or variable values supplied by the user

[ ]  
Square brackets enclose optional values

{}  
Braces enclose content that the user must specify, such as $x$ or $y$ or $z$

|  
Vertical bar indicates alternate selections — the bar means “or”

...  
Ellipses indicate nonessential information omitted from the example

Where to get help

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

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

Product information

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

https://support.EMC.com
Technical support

EMC offers a variety of support options.

Support by Product — EMC offers consolidated, product-specific information on the Web at:

https://support.EMC.com/products

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

EMC Live Chat. — Open a Chat or instant message session with an EMC Support Engineer.

eLicensing support

To activate your entitlements and obtain your VMAX license files, visit the Service Center on https://support.EMC.com, as directed on your License Authorization Code (LAC) letter emailed to you.

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

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

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

◆ North America, Latin America, APJK, Australia, New Zealand: SVC4EMC (800-782-4362) and follow the voice prompts.

◆ EMEA: +353 (0) 21 4879862 and follow the voice prompts.

Your comments

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

VMAXContentFeedback@emc.com
CHAPTER 1
Introduction

This chapter covers the following topics:

- Mainframe Enablers and Consistency Groups .......................................................... 18
- Introduction to Consistency Groups ........................................................................ 19
Mainframe Enablers and Consistency Groups

EMC® Consistency Groups is one of the EMC Mainframe Enablers. The EMC Mainframe Enablers include the following components that you can use to monitor and manage your storage:

- 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

Mainframe Enablers includes the software for all of these components. When you install the Mainframe Enablers, you install the software for all the components.

Licensing

Refer to the following documents for information about licensing:

- Mainframe Enablers Installation and Customization Guide
- VMAX All Flash Product Guide
- VMAX3 Family Product Guide
- VMAX Family Product Guide

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

EMC Consistency Groups (ConGroup) is designed to protect and maintain consistency of remotely mirrored data in an SRDF/S configuration. ConGroup supports all valid SRDF topologies\(^1\), including Concurrent and Switched SRDF, as well as STD devices with their established BCVs\(^2\).

Purpose

Dependent writes

Many applications (especially transaction-oriented systems or database management systems) use *dependent write logic* for data consistency. Dependent write logic means that an attempt by an application to issue a given I/O request depends on the prior successful completion of another I/O operation. Figure 1 shows this process:

![Dependent write logic diagram](image)

**Figure 1** Dependent write logic

As Figure 1 shows, the application takes the following steps:

1. Writes a record of what it is going to do to the transaction log.
2. Writes the data to the actual database.
3. Writes another record to the transaction log to indicate that the data was updated successfully.

These three writes (log, database, and log again) are related. Each write request is not issued until the previous related write has completed.

---

1. The *VMAX All Flash Product Guide* and *VMAX3 Family Product Guide* describe SRDF topologies.
2. STDs and BCVs are used for local replication. The *TimeFinder/Mirror for z/OS Product Guide* describes STDs and BCVs.
Device or link failures

When an application using dependent write logic is under SRDF, the application is protected by the *Transparent Link Outage* feature in normal circumstances. When an SRDF link not running with the *Domino*\(^1\) attribute fails or is intentionally disabled, the application writing to the R1 is not aware of the failure of writes to the R2. When the link is restored, the invalid R2 tracks are automatically filled in by the more up-to-date R1 tracks.

This solution works well if both Site A and Site B have only one VMAX system, as shown in Figure 2.

---

1. The *VMAX All Flash Product Guide* and *VMAX3 Family Product Guide* describe the Domino mode.
However, if the configuration includes multiple VMAX systems at Site A or Site B (and therefore, multiple links), something more is needed. Consider the situation shown in Figure 3.

**Figure 3** Communication failure in multiple VMAX configuration

In Figure 3, data at Site A is remotely mirrored across more than one link. If one or more of these links fail, and the remaining links continue to operate, the result is a mixture of “stale” and “fresh” data at Site B. Restarting operations using data at Site B could result in integrity problems, such as:

- Indexes pointing to the wrong data
- Records being linked incorrectly
- Corrupted database structures
ConGroup role

To solve possible data consistency problems, ConGroup takes the following actions:

- Prevents loss or corruption of data when there is a communications failure or device failure in configurations of multiple VMAX systems at the source or target site.
- Ensures logically consistent, restartable data copies at the remote side of the SRDF configuration.

Basic consistency group

ConGroup ensures data consistency by protecting devices associated with a *consistency group*.

A basic consistency group is illustrated in Figure 4.

*Figure 4* Basic consistency group

*Figure 6 on page 68* illustrates an extended consistency group.

“Managing Consistency Groups” on page 67 provides detailed information on how to create and manage consistency groups.
Concurrent SRDF

In a Concurrent SRDF configuration, a single primary device (R11) is mirrored to two R2s concurrently.

**Note:** The *VMAX All Flash Product Guide* and *VMAX3 Family Product Guide* describe Concurrent SRDF.

When using ConGroup with Concurrent SRDF, you can have one or both of the R2s protected by ConGroup.

*Figure 5* illustrates a Concurrent SRDF configuration for ConGroup.

![Figure 5 Using ConGroup in Concurrent SRDF configurations](image)

“Protecting specific R1-R2 pairs” on page 90 discusses how to set up protection with Concurrent SRDF.
Introduction

Features

The key ConGroup features include:

- Monitoring of consistency groups
- Tripping consistency groups
- Swapping consistency groups with CAX

Monitoring of consistency groups

ConGroup provides protection for user-defined groups of devices called consistency groups.

“Managing Consistency Groups” on page 67 describes consistency groups.

During daily operation, ConGroup monitors the consistency group for signs that a write is unable to propagate to R2. ConGroup uses the Enginuity Consistency Assist (ECA) technology to monitor consistency groups.

“Monitoring consistency groups” on page 97 describes ConGroup monitoring.

Tripping consistency groups

When an SRDF link failure or an R2 device failure occurs, ConGroup trips the consistency group. This includes stalling I/O, suspending SRDF links, and splitting remote STDs from their BCVs while maintaining data consistency in the group.

“Tripping consistency groups” on page 99 describes consistency group tripping.

“Resuming consistency group operations” on page 105 discusses methods to resume operations after a trip.

Swapping consistency groups with CAX

If you have enabled CAX (ConGroup AutoSwap Extension), ConGroup also monitors the consistency group for host I/O (channel) failures and R1 device failures. When such a failure occurs, it can automatically swap I/O from R1s to R2s of your consistency group.

“Swapping Consistency Groups with CAX” on page 109 describes CAX.

“Resuming operations after swap” on page 121 discusses methods to resume operations after a swap.
APIs and utilities

ConGroup includes the following utilities:

◆ Gatekeeper server
◆ Trip API
◆ ECGUTIL utility
◆ CGRPQDEV Callable Service

Gatekeeper server

The gatekeeper server uses the ResourcePak Base Gatekeeper API to acquire a list of SCF gatekeepers. It dynamically allocates them to internal requesters for all syscalls and API calls within ConGroup.

“Gatekeeper server” on page 141 describes using the gatekeeper server in ConGroup.

Trip API

The Trip API causes a trip of a specified consistency group. The Trip API is principally intended for use by 3rd party developers, but is available to all ConGroup users.

“Trip API” on page 102 describes the Trip API.

ECGUTIL utility

The ECGUTIL utility allows you to clean up the environment in situations when devices are left in an incorrect state.

“Cleaning up environment with ECGUTIL” on page 148 describes the ECGUTIL utility.

CGRPQDEV Callable Service

CGRPQDEV is one of the routines in the Symmetrix Application Programming Interface (SymmAPI™). SymmAPI is a set of routines that can interface with VMAX systems attached to hosts that are running in a z/OS environment.

CGRPQDEV is a callable service used to query the ConGroup address space. Queries can be made to determine the names of all currently defined consistency groups, to find out which devices are part of a named consistency group, and to check the state of a device in a consistency group.

CGRPQDEV is principally intended for use by EMC and 3rd party developers.
Introduction

Security

ConGroup provides for security by allowing you to validate authorization to access ConGroup resources through the mainframe EMCSAFI interface.

**Note:** The *Mainframe Enablers Installation and Customization Guide* describes how to use EMCSAFI.

As a result, any of the following SAF-compliant security products can be used to ensure proper user authorization:

- RACF
- CA-ACF2
- CA-Top Secret

The security product you select must be compatible with RACF release 1.9 or higher.

**Note for TDMF/FDRPAS users**

If you are using ConGroup with IBM (Softek) Transparent Data Migration Facility (TDMF), or with FDRPAS by Innovation Data Processing, keep the following points in mind:

- Ensure that TDMF or FDRPAS does not migrate/swap into or out of a consistency group.
- Define the TDMF or FDRPAS source and target devices so that they are in the same consistency group.
- Do not use volser masking with TDMF or FDRPAS source and target devices.
- Do not allow a TDMF or FDRPAS source or target device to be added to the CAX-enabled consistency group.
CHAPTER 2
Getting Started

This chapter covers the following topics:

- Post-installation ................................................................. 28
- Configuring ConGroup ......................................................... 28
- Global configuration parameters ........................................ 30
- Consistency group-specific configuration parameters .......... 42
- CAX options ........................................................................ 53
- Running ConGroup ............................................................ 63
Post-installation

To use ConGroup, you need to install Mainframe Enablers and enable ConGroup, as described in the *Mainframe Enablers Installation and Customization Guide*.

Configuring ConGroup

Before using ConGroup, you have to create a configuration file containing your customized parameter settings.

The configuration file is specified using the CONFIG DD statement of the ConGroup started task, as described in “Customizing ConGroup started task” on page 63.

You can find a sample configuration file in the CGRPCFG0 member of the Mainframe Enablers SAMPLIB (SMP/E DD name: CGPSAMP).

If you run multiple ConGroup instances, EMC recommends that all ConGroup instances use the same configuration file by having their CONFIG DD statements point to the same dataset.

Creating configuration file

Generally, ConGroup configuration parameters are specified as a series of statements in the following format:

```plaintext
<keyword>=<value>
<keyword>=<value>
...
<keyword>=<value>
```

*Note:* If a parameter has a different syntax, it is provided in the parameter description.

- For each parameter statement, the keyword begins in column 1.

  *Note:* An asterisk (*) in column 1 denotes a comment record.

- Each statement in the ConGroup configuration file begins on a separate line.

- Each statement can have a maximum length of 70 bytes and may not span lines. Columns 71 to 80 are ignored.

- You can include a comment on a parameter line as follows:

  ```plaintext
  keyword=value /* comment text */
  ```

In the configuration file, a consistency group name is defined first, immediately followed by the devices in the consistency group.
Configuration parameter types

There are two types of configuration parameters:

- Global configuration parameters that determine general ConGroup behavior. The global parameters are listed in “Global configuration parameters” on page 30.
- Consistency-group specific configuration parameters that define a particular consistency group. The group-specific parameters are listed in “Consistency group-specific configuration parameters” on page 42.

In addition, you can use CAX options described in “CAX options” on page 53. The CAX options are specified as subparameters after the CAXOPTS keyword, as described in “CAXOPTS” on page 31.
Global configuration parameters

Global configuration parameters affect all consistency groups used with the current ConGroup instance.

Place global configuration parameters first in the configuration file, followed by consistency group-specific configuration parameters that define one or more consistency groups.

Syntax conventions

The parameters follow these syntax conventions:

- Keywords appear in uppercase (for example, `ADD`). They must be spelled exactly as shown.
- For easy reference, command keywords are supplemented by lowercase letters to form a meaningful word (for example, `DEVi ces`). When typing a command, use only CAPITALIZED characters of any keyword.
- Variables appear in lowercase and italics (for example, `cngr p`). They represent user-supplied names or values in the syntax.
- Square brackets `[]` indicate an optional entry (for example, `cuu[-cuu]`).
- The vertical bar `|` indicates alternative argument values (for example, `SUSPEND|REMSPLIT`).
- Curly brackets `{}` are used to group a series of alternative values that can be used with a single keyword, for example: `{SUSPEND|REMSPLIT|RESUME}`.
- Aside from the square and curly brackets and the vertical bar characters, you must type all other characters that are shown in the syntax statements.
- Default values are indicated by an underline. For example, if the parameter has the following option, `(WAIT|NOWAIT)`, the underlined NOWAIT indicates the default value.

ALLOW_SHARED_R1S

Allows or disallows R1 sharing by mirror for all consistency groups.

Note: “R1 sharing by mirror” on page 91 describes R1 sharing by mirror.

Valid values

YES | NO

Default

NO
**AUTO_REFRESH**

Enables you to automatically refresh the ConGroup configuration by generating REFRESH command statements automatically.

**Note:** "Refreshing automatically" on page 144 describes automatic refreshes of ConGroup configuration.

AUTO_REFRESH is ignored if the consistency group is a CAX group.

**Note:** "Swapping Consistency Groups with CAX" on page 109 describes CAX.

**Valid values**

YES | NO

**Default**

NO

**CAXOPTS**

Defines a set of CAX options under an option set name.

Each option set is a collected series of parameter statements that specify how ConGroup handles devices for CAX processing.

**Note:** "Swapping Consistency Groups with CAX" on page 109 describes CAX.

After defining an option set, you can associate the option set with a consistency group using the CAX parameter, as described in “CAX” on page 43.

**Syntax**

```plaintext
CAXOPTS option_set=(cax_option[,cax_option[,...]])
```

**Parameters**

`cax_option`

The CAX options included in the option set.

“CAX options” on page 53 lists the CAX options that are supported by ConGroup and/or are appropriate for ConGroup usage.

If you use multiple `cax_option` parameters, enclose them in parentheses and separate them with commas or intervening spaces.

`option_set`

The name of the option set being defined.

The option set name can be from 1 to 8 characters in length.

**Example**

```plaintext
CAX=(CAXOPTS=CAXOPT1)
CAXOPTS CAXOPT1=(AUTOCOND=NOPATHS, CFW=OFFVAL, LOSTO ONSWAP=SYSRESET(OFF0), ASCM)
```
CLOCKE

Sets the multiplier of the clock tick interval after which unreleased coordination locks are expired.

*Note:* “Coordination timing” on page 137 describes coordination timing.

For example, if the clock tick is 5 seconds and you use the default value of 20 for CLOCKE, the unreleased coordination locks expire in $5 \times 20 = 100$ seconds.

**IMPORTANT**

For normal operation, use the default value of 20.

**Syntax**

```
CLOCKE(\textit{n})
```

*Where:*

\textit{n}

The multiplier value. Valid values are from 4 to 9999. The default value is 20.

**Example**

CLOCKE(25)

CLOCKN

Determines the time, in hundredths of a second, between coordination activities.

*Note:* “Coordination timing” on page 137 describes coordination timing.

**CAUTION**

For normal operation, do not change the default value of CLOCKN.

**Syntax**

```
CLOCKN(\textit{n})
```

*Where:*

\textit{n}

The interval (in hundredths of a second) between ConGroup coordination activities. Valid values are from 100 to 30000. The default value is 1000 (10 seconds).
**COUPLEDS_ALLOWED**

Enables or disables explicitly adding volumes with couple datasets to a consistency group.

*Note:* “Couple datasets” on page 80 discusses handling of couple datasets in consistency groups.

If you include DEVICE_LIST=ALL and also specify COUPLEDS_ALLOWED=YES, then volumes with couple datasets are *not* added to the consistency group.

**Valid values**

YES | NO

**Default**

NO

**DISABLE_AT_SHUTDOWN**

Disables all consistency groups when ConGroup is shutdown.

*Note:* “Disabling consistency groups” on page 95 describes disabling consistency groups.

**Valid values**

YES | NO

**Default**

NO
**DISABLE_ON_VERIFY_ERROR**

Disables an enabled consistency group if the automatic verification logic finds that the devices in the consistency group are not in the expected state.

*Note:* “Disabling consistency groups” on page 95 describes disabling consistency groups.

**CAUTION**

When set to YES, a successful consistency group trip cannot be guaranteed because sometimes AUTO_VERIFY may disable the group first.

If you do not specify the DISABLE_ON_VERIFY_ERROR parameter, a DISPLAY ENVIRONMENT command (described in “DISPLAY ENVIRONMENT” on page 188) shows that DISABLE_ON_VERIFY_ERROR is set to NO.

DISABLE_ON_VERIFY_ERROR is always NO for consistency groups that also have a swap group associated with them. If you make an R1 Not Ready in such cases, automatic verification still functions, but does not disable the consistency group. Disabling the consistency group would delete the swap group.

Valid values

YES | NO

Default

NO

**DISPLAY_CONGROUP_LISTOPT**

Sets the default behavior for the DISPLAY CONGROUP command described in “DISPLAY CONGROUP” on page 187:

- **(Default) LIST** — Lists all devices in the consistency group.
- **NOLIST** — Suppress listing of all devices in the consistency group.

Valid values

LIST | NOLIST

Default

LIST
DISPLAY_SAFAUTH_SUCCESS

Determines whether the CGRP097I message is issued for authorized SAF calls:

- If YES is specified, the command is executed and the CGRP097I message is issued.
- If NO is specified or defaulted, the command is executed without displaying the CGRP097I message.

**Note:** The *Mainframe Enablers Message Guide* describes message CGRP097I and the conditions when it is displayed.

Valid values

YES | NO

Default

NO

GLOBAL

Sets the owner LPAR to control the consistency group during CAX operations.

**Note:** “Managing owner LPARs” on page 138 describes owner LPARs. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

The GLOBAL parameter is required in the multi-LPAR mode.

**Note:** “Multi-LPAR mode” on page 134 describes the multi-LPAR mode.

The GLOBAL parameter is effectively ignored by REFRESH.

Syntax

GLOBAL=(OWNER=smfid)

Where:

*smfid*

The SMFID of the owner LPAR (up to 4 characters).

This value must be the same in ConGroup configuration files for all ConGroup instances that use the consistency group.
MODE

Sets the single-LPAR or multi-LPAR mode for ConGroup.

Note: “Single/multi-LPAR mode” on page 133 describes the single-LPAR and multi-LPAR modes.

Syntax

MODE={SINGLE|MULTI}[,CGSET\text{nn}]

Parameters

CGSET\text{nn}

Sets the global storage name token anchor to ensure ConGroup and CAX cross-address space and/or LPAR isolation.

Note: “CG sets” on page 135 describes CG sets.

Valid values for \text{nn} are from 00 to 06. The default value is CGSET00.

MULTI

Sets the multi-LPAR mode.

Note: The multiple ConGroup instances must be running at the same release level (for example, 8.1).

IMPORTANT

If you set MODE=MULTI, always specify the GLOBAL configuration parameter in the ConGroup configuration file, as described in “GLOBAL” on page 35.

SINGLE

(\text{Default}) Sets the single-LPAR mode.

Note: The single-LPAR mode is not supported with CAX.

Example

To enable multi-LPAR mode and define CGSET06 as the global storage name token anchor:

MODE=MULTI, CGSET06
**MSGLEVEL**

Sets the minimum verbosity level that a message must have to be sent to SYSLOG. When a message is issued, the verbosity level associated with it is compared with the value of the MSGLEVEL parameter. If the message verbosity level is equal or lower than the specified MSGLEVEL value, the message is queued for display.

For example, when MSGLEVEL is set to 7, any message with a verbosity level from 1 to 7 is sent to SYSLOG.

**Syntax**

```plaintext
MSGLEVEL(n)
```

Where:

- `n`
  
  The verbosity level for SYSLOG messages. Valid values are from 1 to 9, where 9 is the most verbose. The default value is 5, which should be sufficient for most purposes.

⚠️ **CAUTION**

Do not change the default setting unless directed by EMC representatives.

**PAGEDEV_ALLOWED**

Determines whether devices that contain page datasets can be defined as part of a consistency group.

**Note:** “Page datasets” on page 81 discusses page datasets in consistency groups.

⚠️ **CAUTION**

EMC does not recommend using page datasets in a non-CAX consistency group.

**Valid values**

- YES | NO

**Default**

- NO
PAVO

Enables (YES) or disables (NO) zBoost™ PAV Optimizer support for ConGroup. When set to YES, ConGroup notifies zBoost PAV Optimizer prior to a consistency group trip.

**Note:** The *ResourcePak Base for z/OS Product Guide* discusses zBoost PAV Optimizer.

**Valid values**

YES | NO

**Default**

YES

REMSPLIT_INTERVAL

Sets the time (in seconds) to wait before retrying a remote split.

**Note:** “Remote split retries” on page 101 describes remote split retries.

**Valid values**

Number of seconds

**Default**

10

REMSPLIT_OPTION

Instructs ConGroup to generate the CGRP314E message during remote split processing, thus notifying the operator about all R2s that do not have a BCV attached.

**Note:** “R2s without BCVs” on page 101 discusses processing of R2s without BCVs during a remote split.

When REMSPLIT_OPTION is specified without the NOESTERR keyword or is not included in the configuration file, R2s without a BCV are ignored.

**Valid values**

NOESTERR
**RESUME_INTERVAL**

Sets the time (in seconds) for the RESUME command (described in “RESUME” on page 195) to wait before checking whether all the devices in the consistency group are synchronized.

**Note:** “Resume interval” on page 107 discusses the resume waiting interval.

**Valid values**

Number of seconds

**Default**

10

**RESUME_OPTION**

Controls information that is displayed during RESUME and VERIFY processing.

**Note:** “Resuming consistency group operations” on page 105 discusses RESUME processing. “Verifying consistency groups” on page 94 discusses VERIFY processing.

The available options can be combined using the plus (+) sign.

**Syntax**

RESUME_OPTION={NOTRNMSG|RSMALLIT}

RESUME_OPTION={NOTRNMSG+RSMALLIT|RSMALLIT+NOTRNMSG}

**Parameters**

**NOTRNMSG**

Suppresses transient state messages that are generated during resume processing.

**RSMALLIT**

Displays invalid tracks for all devices in the consistency group during each resume interval. If not specified, only messages for the first device found with invalid tracks are issued on each interval.

**Note:** “Resume interval” on page 107 describes resume intervals.

**Example**

To specify both the NOTRNMSG and the RSMALLIT values:

RESUME_OPTION=NOTRNMSG+RSMALLIT

or

RESUME_OPTION=RSMALLIT+NOTRNMSG
SAF_CLASS

Defines the queue name (QNAME) that is used to perform security checks for the ConGroup operator commands.

Valid values

Any valid queue name that is defined to the security subsystem

Default

XFACILIT

SAF_PROFILE

Defines the base RNAME that is used to perform security access checks for the ConGroup operator commands.

Before making a security access check, ConGroup appends a command-specific suffix to the value of this parameter.

Valid values

Any valid resource name for the security subsystem. The resource name can be up to 35 characters long.

Note: The Mainframe Enablers Installation and Customization Guide provides a list of the resource names.

Default

EMC.ADMIN.CMD.CG

SEMISYNC_ALLOWED

Determines whether semi-synchronous devices can be used in a consistency group definition.

Note: “Semi-synchronous devices” on page 78 discusses use of semi-synchronous devices in consistency groups.

Valid values

YES | NO

Default

NO
USE_NR_ON_TO

Determines whether to make devices of the specified type Not Ready if the ECA Window times out on the devices.

**Note:** “What is ECA?” on page 98 describes ECA.

The timeout may occur if the host system loses power during a DR event. The VMAX system will then independently set the specified class of devices to Not Ready in order to preserve data integrity. This is useful in situations where you do not want a Open Systems application to continue to use R1s while an AutoSwap application on a mainframe system swaps to R2s.

Specifying ALL includes both FBA and CKD devices. Specifying NO includes neither of the device types.

Valid values

FBA | CKD | ALL | NO

VERIFY_INTERVAL

Sets the time interval (in seconds) before the automatic verification subtask attempts to verify the state of all enabled/active consistency groups.

**Note:** “Automatic verification” on page 94 describes automatic verification.

The verification interval is to ensure that all consistency group devices are still in the expected state.

If you do not specify a verification interval, ConGroup uses a default of 300 seconds. If you specify zero (0), the automatic verification logic is disabled.

Valid values

From 0 to 99999999

Default

300
Consistency group-specific configuration parameters

In the ConGroup configuration file, you place consistency-group specific configuration parameters after the global configuration statements. You use consistency group-specific configuration parameters to declare a consistency group and define its characteristics.

Syntax conventions

Refer to “Syntax conventions” on page 30.

ALLOW_SHARED_R1S

Allows or prohibits R1 sharing by mirror for the consistency group being defined.

Note: “R1 sharing by mirror” on page 91 describes R1 sharing by mirror.

Valid values

YES | NO

Default

NO

ALLOWABLE_MSS

Determines subchannel sets from which R1s are allowed.

Note: “Using multiple subchannel sets” on page 88 discusses using multiple subchannel sets in ConGroup.

The ALLOWABLE_MSS parameter (together with the PAIR subparameter of the CONGROUP statement) is required when using CAX and if any of the device pairs “straddle” two subchannel sets.

Note: “Swapping Consistency Groups with CAX” on page 109 describes CAX.

If ALLOWABLE_MSS is specified, it must at least always specify subchannel set 0.

Syntax

ALLOWABLE_MSS (n[,n])

Where:

(n[,n])

Defines the allowable subchannel sets.

Example

1. To allow R1s from subchannel set 0 or 1:

   ALLOWABLE_MSS (0,1)

2. To allow R1s from subchannel set 0 only:

   ALLOWABLE_MSS (0)
CAX

Activates CAX for a consistency group and determines the CAX option set you want to use for that group.

Note: “Swapping Consistency Groups with CAX” on page 109 describes CAX.

The CAX parameter requires that the CAX feature is available in ConGroup.

Syntax

CAX=(CAXOPTS=option_set)

Parameters

option_set

The name of the CAX option set created using the CAXOPTS global configuration parameter.

Note: Multiple CAX statements may refer to the same CAXOPTS statement.

Example

CAX=(CAXOPTS=CASOPT1)
CONGROUP

Defines the name of a consistency group.

Note: “Creating consistency groups” on page 76 describes consistency groups.

You can also use the SRDF_CONGROUP parameter, which has the same meaning. You can substitute one for the other.

Syntax

CONGROUP=cngrp[,PAIR(pairname)]

Parameters

cngrp
The name of the consistency group. It can be a string from 1 to 8 characters.

PAIR(pairname)
Assigns a name to an R1-R2 pair if any of the device pairs are located on different subchannel sets.

Note: “Using multiple subchannel sets” on page 88 discusses using multiple subchannel sets in ConGroup.

The PAIR parameter can be specified at any time, but is only required if the device pairs are on alternate subchannel sets. For example, one device, R1-side, is on subchannel 0 and the other, R2-side, is on subchannel 1.

Example

CONGROUP=GROUPA,PAIR(MYPAIR)
DEVICE_LIST

Defines the devices that belong to the consistency group most recently defined with the CONGROUP (or SRDF_CONGROUP) parameter. For a given consistency group, there can be one or more DEVICE_LIST statements.

**Note:** “Adding devices to consistency group” on page 82 describes adding devices to consistency groups.

The devices are identified with their CUUs or volsers.

**Syntax**

```plaintext
DEVICE_LIST={device_list|ALL}
```

**Parameters**

**ALL**

- Includes all devices.

**device_list**

- The list of devices identified with their CUUs or volsers.

You can specify individual CUUs or ranges, as well as volsers. Specify ranges with a hyphen, for example 280-28F. Separate individual CUUs, ranges of CUUs and volsers with commas. You can mix CUUs and volsers in the same statement.

**Note:** When using volsers, ensure that all included devices are online to all LPARs participating in the consistency group.

You can also use a volser mask. You can use the mask character (*) after any number of characters.

**Note:** The specified volsers cannot contain hyphens.

**Example**

```plaintext
DEVICE_LIST=200-27F,290,B000-B007
DEVICE_LIST=900,901,90A,DB2001,DB2002
DEVICE_LIST=IMS*
DEVICE_LIST=X*,QW*,EMC*,ABCO*,CDE01*
DEVICE_LIST=200,300-310,ABC*,SYM001,EMC*
```
DEVICE_LIST_STD

Defines STD devices that belong to the consistency group most recently defined with the CONGROUP (or SRDF_CONGROUP) parameter. For a given consistency group, there can be one or more DEVICE_LIST_STD statements.

**Note:** The *TimeFinder/Mirror for z/OS Product Guide* describes STD devices. “STD devices” on page 79 discusses STD devices in a consistency group.

The devices are identified with their CUUs or volser s.

**Syntax**

```
DEVICE_LIST_STD=device_list
```

**Parameters**

device_list

The list of devices identified with their CUUs or volser s.

You can specify individual CUUs or ranges, as well as volser s. Specify ranges with a hyphen, for example 280-28F. Separate individual CUUs, ranges of CUUs and volser s with commas. You can mix CUUs and volser s in the same statement.

You can also use a volser mask. You can use the mask character (*) after any number of characters.

**Note:** The specified volser s cannot contain hyphens.

**Example**

```
DEVICE_LIST_STD=200,300-303
DEVICE_LIST_STD=STD001,STD002
DEVICE_LIST_STD=EMC*
DEVICE_LIST_STD=X*,QW*,EMC*,ABCO*,CDE01*
DEVICE_LIST_STD=200,300-310,ABCO*,SYM001,EMC*
```
**EXCLUDE**

Excludes devices from a consistency group definition. For a given consistency group, there can be one or more EXCLUDE statements.

**Note:** “Adding devices to consistency group” on page 82 describes adding devices to consistency groups.

Excluded devices have precedence over devices included by the DEVICE_LIST, DEVICE_LIST_STD, and SMS_GROUP parameters. This means that an excluded device is always left out, whereas an included device can be preempted by an EXCLUDE statement.

The EXCLUDE parameter does not apply to devices defined with the SYMM_DEV# parameter.

**Syntax**

```
EXCLUDE=device_list
```

**Parameters**

`device_list`

The list of devices identified with their CUUs or volsers.

You can specify individual CUUs or ranges. Specify ranges with a hyphen, for example 280-28F. Separate individual CUUs, ranges of CUUs and volsers with commas. You can mix CUUs and volsers in the same statement.

**Note:** The specified volsers cannot contain hyphens.

**Example**

```
EXCLUDE=200-21F
```
SCFG

Adds all devices belonging to the specified GNS group to the consistency group.

**Note:** The *ResourcePak Base for z/OS Product Guide* describes GNS groups. “Adding devices by GNS group” on page 82 discusses adding a GNS group to a consistency group.

The GNS group must be of the Enterprise R1 type.

**Syntax**

```
SCFG(gnsgrp)
```

**Parameters**

* gnsgrp
  
  The name of the GNS group.
  
  Each GNS group name can be from 1 to 65 alphanumeric characters long. Each apostrophe, if used, reduces the maximum possible length of `gnsgrp` by 1.
  
  Enclose the name in apostrophes if it includes any character that is neither an uppercase letter nor a numeric digit. If a GNS group name you use in ConGroup includes blank spaces, enclose it in double quotes:

  ```
  SCFG("West Coast Stores")
  ```

  However, be aware that even if the GNS group name is enclosed in apostrophes, non-alphanumeric characters or lowercase alphabetic characters may be translated to blanks or converted to uppercase when presented as input to ConGroup, possibly causing the GNS group not to be found or possibly resulting in processing of an unintended GNS group.

SRDF_CONGROUP

Same as CONGROUP described in “CONGROUP” on page 44.

SMS_GROUP

Adds all devices belonging to the specified SMS group to the consistency group most recently defined with the CONGROUP (or SRDF_CONGROUP) parameter.

**Valid values**

Any valid SMS group name from 1 to 8 characters in length.
**SUSPEND_FAILURE**

Determines what ConGroup does when a failure occurs in the suspend process:

- **FAIL** — Immediately resumes I/O to the devices in the consistency group without retrying the failed suspend process. The remote devices are in an inconsistent state.
- **RETRY** — Continues retrying the failed suspend process until successful completion, or until the timeout value specified with the SUSPEND_TIMEOUT parameter is reached.
- **WTOR** — Enables you to retry the failed suspend process by entering R in response to the WTOR message displayed.

**Note:** WTOR is valid only if ConGroup is running with SUB=MSTR, as described in “Starting ConGroup with SUB=MSTR” on page 65.

SUSPEND_FAILURE is a required parameter. Include SUSPEND_FAILURE for each consistency group you define in the ConGroup configuration file. If you do not specify the SUSPEND_FAILURE parameter for a consistency group, you receive the CGRP336E and CGRP125E messages. ConGroup terminates without completing initialization or fails its refresh if it encounters CGRP336E during a refresh.

Valid values

FAIL | WTOR | RETRY

**SUSPEND_RETRY_TIMEOUT**

Sets the period of time (in seconds) the suspend process is allowed to retry.

**Note:** “Suspending consistency group operations” on page 100 describes suspending operations.

When the period expires, the suspend process is terminated.

If you do not specify a value, SUSPEND_TIMEOUT defaults to zero (0), meaning that the suspend operation is never automatically canceled.

Valid values

Number of seconds

Default value

0

**SUSPEND_TIMEOUT**

Same as SUSPEND_RETRY_TIMEOUT described in “SUSPEND_RETRY_TIMEOUT” on page 49.
SYMM_DEV#

Defines the devices that belong to the consistency group most recently defined with the CONGROUP (or SRDF_CONGROUP) parameter. For a given consistency group, there can be one or more SYMM_DEV# statements.

Note: “Adding devices to consistency group” on page 82 describes adding devices to consistency groups.

The devices are identified with their VMAX device numbers.

For FBA meta devices, you can specify both heads and members, but the members are ignored. ConGroup determines the members based on the heads that it finds in the specified list.

Specifying CKD devices results in message CGRP140E.

Syntax

SYMM_DEV#=device_list,
{CUU=cuu|NAME=Symmname|SER=SymmID}

Parameters

CUU=cuu

Specifies the z/OS device used to identify the correct VMAX system.

The device identified with the CUU does not need to be in a consistency group, nor is it implicitly included in a consistency group by this statement.

device_list

The list of devices identified with their VMAX device numbers. You can specify individual VMAX device numbers or ranges. Specify ranges with a hyphen.

Note: The specified devices must reside on the same physical VMAX system.

NAME=Symmname

The name of the VMAX system on which the devices reside.

If the name has embedded blanks, enclose it in quotes.

SER=SymmID

The 12-digit serial number of the VMAX system on which the devices reside.

Example

SYMM_DEV#=100-1FF,200,300,CUU=298
SYMM_DEV#=100-1FF,200,300,NAME='My VMAX'
SYMM_DEV#=100-1FF,200,300,SER=000123409754
SYMGROUP

Limits the devices added to the consistency group to those that also have mirrors on the specified SRDF groups.

**Note:** “Protecting specific R1-R2 pairs” on page 90 provides information on limiting protection to specific R1-R2 pairs.

The SYMGROUP parameter does not include devices in an SRDF group. It only specifies R1s (or R1 mirrors, if R1 sharing by mirror is allowed) to be monitored. Given a set of R1s, ConGroup uses the SRDF group number to find the connected R2s. If no connected R2s exists on the given SRDF group number(s), ConGroup initialization (or Refresh) fails.

If you specify any SYMGROUP statements, then you must define *all* consistent SRDF groups with the SYMGROUP statements. The SYMGROUP parameter applies only to devices defined with the DEVICE_LIST, SMS_GROUP, and SYMM_DEV# parameters.

The SYMGROUP parameter applies to the most recently defined consistency group. The SYMGROUP statement must come before the corresponding DEVICE_LIST statements.

**Note:** If the consistency group includes devices that are not specified by the SCFG parameter (GNS) and if any of those devices use SRDF groups that are not used by any of the GNS devices, then one or more SYMGROUP statements must be coded to support the non-GNS devices.

Syntax

```
SYMGROUP=({SymmID|Symmname},{srdfgrp_list|ALL})
```

Parameters

**ALL**

Selects all SRDF groups.

**SymmID**

The 12 character serial number of the VMAX system on which the R2s reside.

**Symmname**

The name of the VMAX system on which the R2s reside.

The name can be a string of up to 64 characters. If the name is made up of mixed-case characters or contains spaces, enclose it in quotation marks.

**srdfgrp_list**

The SRDF group or groups identified with their 1-2 hex character IDs.

You can specify comma-separated individual values or ranges.
Example

1. To specify SRDF groups 2, 4, and 6 on VMAX system 000184000345:
   SYMGROUP=(000184000345,2,4,6)

2. To specify SRDF groups 2, 3, and 4 on VMAX system named BOSTON:
   SYMGROUP=(BOSTON,02-04)

3. To specify all SRDF groups on VMAX system 000184000345:
   SYMGROUP=(000184000345,ALL)

4. To specify all SRDF groups on the VMAX system WAREHSE4:
   SYMGROUP=(WAREHSE4,ALL)
CAX options

CAX options are specified after the CAX and CAXOPTS keywords, as shown in the following example:

```
CAX=(CAXOPTS=CAXOPT1)
CAXOPTS CAXOPT1=(AUTOCOND=NOPATHS, CFW=OFFVAL, LOSTO ONSWAP=SYSRESET(OFF0), ASCM)
```

“CAX” on page 43 describes the CAX parameter. “CAXOPTS” on page 31 describes the CAXOPTS parameter.

Syntax conventions

Refer to “Syntax conventions” on page 30.

AllowConcurrentCopy

Allows or prohibits swaps of devices with Concurrent Copy sessions.

---

**Note:** “Handling devices with Concurrent Copy/SNAP sessions” on page 117 discusses processing of devices with Concurrent Copy sessions.

The AllowConcurrentCopy option is mutually exclusive with the NOCHECKConcurrentCopy option described in “NOCHECKConcurrentCopy” on page 59.

**Syntax**

```
{AllowConcurrentCopy | NOAllowConcurrentCopy}
```

**Where:**

- **(Default)** AllowConcurrentCopy allows swaps of devices with Concurrent Copy sessions.
- **NOAllowConcurrentCopy** prohibits swaps of devices with Concurrent Copy sessions.

---

**Note:** You can abbreviate AllowConcurrentCopy as ACC.
AllowOnlineToDevice

Allows or prohibits using HRO (Host Read Only) online target (TO) devices in a CAX group or swap group.

Nota:*"Handling HRO online TO devices” on page 118 discusses processing of HRO online TO devices. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

Syntax

{AllowOnlineToDevice|NOAllowOnlineToDevice}

Where:
- AllowOnlineToDevice allows using online target (TO) devices in CAX or swap groups.
- (Default) NOAllowOnlineToDevice prohibits using online target (TO) devices in CAX or swap groups.

Nota:* You can abbreviate AllowOnlineToDevice as AllOnTo.

AllowOnlineUndefinedDevice

Allows or prohibits use of source (FROM) devices, which are online but excluded or not discovered by ResourcePak Base, in a CAX or swap group.

This can occur if ResourcePak Base was unable to discover the device due to I/O timeouts or some other issues, or when an exclude list of devices is defined in ResourcePak Base using the SCF.DEV.EXCLUDE configuration parameter, as described in the ResourcePak Base for z/OS Product Guide.

⚠️ CAUTION

Use AllowOnlineUndefinedDevices with caution, as it could result in loss of access to a device after a swap.

Syntax

{AllowOnlineUndefinedDevice|NOAllowOnlineUndefinedDevice}

Where:
- AllowOnlineUndefinedDevice allows using undefined source (FROM) devices in a CAX or swap group.
- (Default) NOAllowOnlineUndefinedDevice prohibits using undefined source (FROM) devices in a CAX or swap group. When an online undefined device is present in the group, the group fails validation, and the CGRS587E message is displayed.

Nota:* You can abbreviate AllowOnlineUndefinedDevice as ALLONUNDEFDEV.
AllowSnapSession

Allows or prohibits swaps of devices with Snap sessions.

Note: “Handling devices with Concurrent Copy/SNAP sessions” on page 117 discusses processing of devices with Snap sessions.

The AllowSnapSession option is mutually exclusive with the NOCHECKSnapSessions option described in “NOCHECKSnapSessions” on page 59.

Syntax

{AllowSnapSession | NOAllowSnapSession}

◆ (Default) AllowSnapSession allows swaps of devices with Snap sessions.
◆ NOAllowSnapSession prohibits swaps of devices with Snap sessions.

Note: You can abbreviate AllowSnapSession as ALLSNAP.

AllowSystemsCountMismatch

 Allows or prohibits mismatches in the count of participating LPARs at swap time and the count of path groups for the device established at validation time.

Note: “Checking LPAR/path group counts” on page 119 describes LPAR/path group count mismatches.

Syntax

{AllowSystemsCountMismatch | NOAllowSystemsCountMismatch}

Where:

◆ (Default) AllowSystemsCountMismatch allows system count mismatch. In case of mismatch, the swap continues.
◆ NOAllowSystemsCountMismatch prohibits system count mismatch. In case of mismatch, the swap fails.

AUTOSWAPCONDITIONS

Same as UNPLANNEDCONDITIONS described in “UNPLANNEDCONDITIONs” on page 61.

The UNPLANNEDCONDITIONS option replaces and performs the same function as the AUTOSWAPCONDITIONS option used in previous releases of ConGroup. For backward compatibility, however, ConGroup still recognizes AUTOSWAPCONDITIONS.
BypassSystemsCount

Same as AllowSystemsCountMismatch described in “AllowSystemsCountMismatch” on page 55.

Syntax

```
{BypassSystemsCount|NOBypassSystemsCount}
```

Where:

- **(Default)** BypassSystemsCount allows system count mismatch.
- **NOBypassSystemsCount** prohibits system count mismatch.

CFW

Controls Cache Fast Write (CFW).

**CAUTION**

Do not use CFW on devices that may be swapped. When you use CFW, additional integrity checks may be involved. In some cases, the subsystem ID is checked. If the device has been swapped, the SSID is different and the job is terminated.

Syntax

```
CFW={OFF|VALIDation|NO|ALLOW}
```

Parameters

**OFF\_VALIDation**

*(Default)* CFW is disabled during ConGroup validation processing.

No CFW processing is attempted during a CAX swap. If you subsequently reactivate CFW, and a CAX swap occurs, jobs using CFW fail. You must rerun these jobs after the swap.

EMC recommends that you use CFW=OFFVAL, followed by the VALIDATE command. CFW=OFFVAL turns off CFW for the devices in the group. VALIDATE detects and warns you about any prior use of CFW.

**NO**

Prohibits swapping if CFW is active. If CFW is active on any devices in the consistency group, validation of the consistency group fails and CAX processing is disabled.

**ALLOW**

CFW is checked during ConGroup validation processing. However, no change in the device CFW state is made. If a CAX swap occurs, jobs using CFW fail. You must rerun these jobs after the swap.
CROSSSYSTEMTIMEOUT

Sets the cross-system timeout value.
This is the time CAX waits for acknowledgment from other CAX instances on other hosts.

Syntax

CROSSSYSTEMTIMEOUT=\textit{time\_interval}

Parameters

\textit{time\_interval}

Number of seconds. Valid values are from 60 to 9999. The default value is 300.

CSD

Determines the post-swap state of R1s that were swapped by setting or overriding the default of the RNR (SRDF Not Ready) state at swap completion.

Syntax

CSD={\textit{NRDY} | \textit{USRNRDY}}

Parameters

\textit{NRDY}

\textbf{(Default)} Sets R1s to the RNR state at swap completion.

\textit{USRNRDY}

Sets R1s to the User-Not-Ready state at swap completion.
LOSTOwnerpolicy ONSWAP

Specifies the action for CAX to take if all communication with the owner LPAR or controlling system is lost during the swap process.

Note: “Handling lost communications with owner LPAR” on page 119 discusses losing communication with the owner LPAR.

LOSTOwnerpolicy affects system behavior only during the swap process.

You can abbreviate LOSTOwnerpolicy as LOP.

Syntax

LOSTOwnerpolicy ONSWAP={OPERATOR|BACKOUT|SYSRESET[(wait_code)]}

Parameters

BACKOUT

Specifies that the swap operation is undone on the system that has lost connectivity with the owner LPAR.

The BACKOUT option is also made available in the ESWP485A WTOR message issued when LOSTOwnerpolicy ONSWAP=OPERATOR is specified.

OPERATOR

(Defualt) Lets operator decide on the action to be taken by issuing WTOR message ESWP485A. You can choose the appropriate policy by responding to the message. The Mainframe Enablers Message Guide describes the options available in the message.

CAUTION

Ensure that the owner is really lost and not just isolated before using this option.

SYSRESET[(wait_code)]

Specifies that a non-restartable wait state occurs on the system that has lost connectivity with the owner LPAR.

The SYSTESET option is also made available in the ESWP485A WTOR message issued when LOSTOwnerpolicy ONSWAP=OPERATOR is specified.

wait_code

The code of the wait state. Valid values are from X'FF0' to X'FFE'. The default value is 0FF0.
NOCHECKConcurrentCopy

Bypasses verification of devices with Concurrent Copy sessions.

**Note:** “Handling devices with Concurrent Copy/SNAP sessions” on page 117 discusses processing of devices with Concurrent Copy sessions.

The NOCHECKConcurrentCopy option is mutually exclusive with the AllowConcurrentCopy option described in “AllowConcurrentCopy” on page 53.

**Syntax**

NOCHECKConcurrentCopy

**Note:** You can abbreviate NOCHECKConcurrentCopy as NOCHKCC.

NOCHECKSnapSessions

Bypasses verification of devices with Snap sessions.

**Note:** “Handling devices with Concurrent Copy/SNAP sessions” on page 117 discusses processing of devices with Snap sessions.

The NOCHECKSnapSessions option is mutually exclusive with the AllowSnapSession option described in “AllowSnapSession” on page 55.

**Syntax**

NOCHECKSnapSessions

**Note:** You can abbreviate NOCHECKSnapSessions as NOCHKSNAP.

QUIESCETimeout

Sets the quiesce timeout.

During I/O halt processing, this is the time that CAX waits for any outstanding I/O to complete on a device. If I/O is still active on the device following this timeout, then a swap backout occurs.

**Syntax**

QUIESCETimeout={MIH\[time_interval\]}

**Parameters**

MIH

*(Default)* Makes CAX use the z/OS Missing Interrupt Handler (MIH) timeout interval.

time_interval

Number of seconds. Valid values are from 0 to 300. A value of zero (0) is equivalent to specifying MIH.

If the resolved timeout value is less than 15 seconds, then 15 seconds is used.
ROUTEMESSAGetoowner

Enables or disables routing of CAX messages to the SYSLOG of the owner system.

Note: “Routing CAX messages” on page 120 describes routing CAX messages.

If you use ROUTEMESSAGetoowner with no keywords, only error (E) messages are routed to the owner system SYSLOG.

Syntax

{ROUTEMESSAGetoowner|NOROUTEMESSAGetoowner} [ALL|Warn|Error]

Where:
- ROUTEMESSAGetoowner enables CAX message routing.
- NOROUTEMESSAGetoowner disables CAX message routing.

Parameters

ALL
All non-owner CAX messages are routed to the owner system SYSLOG.

Error
Only error (E) messages are routed to the owner system SYSLOG.

Warn
Only warning (W) and error (E) messages are routed to the owner system SYSLOG.
UNPLANNEDCONDiotions

Determines the action CAX takes when the specified condition occurs for a device in the consistency group.

*Note:* “Handling unexpected conditions” on page 115 discusses handling unexpected conditions in CAX.

This is a required parameter. There is no default value.

You can specify UNPLANNEDCONDiotions options in a series or all together:

```
UNPLANCOND=INTERVENTIONREQUIRED
UNPLANCOND=NOPATHS
UNPLANCOND=SYNCLINKFAILURE
UNPLANCOND=INTERVENTIONREQUIRED NOPATHS SYNCLINKFAILURE
```

You can also disable unplanned swaps by setting UNPLANCOND to null:

```
UNPLANCOND=
```

In this case, all possible CAX conditions are disabled. There are no unplanned swaps if a link failure occurs.

*Note:* Setting UNPLANCOND to null is for the life of the group; this means that an unplanned swap is not going to be performed for this group. Opposed to this, the SETSWAP DISABLE command is used to temporary stop unplanned (or planned) swaps during some window of time; for example, when you might be doing some configuration changes and do not want to accidentally have an unplanned swap.

**Syntax**

```
UNPLANNEDCONDiotions={INTERVENTIONREQUIRED|NOPATHS|SYNCLINKFAILURE ALLOWTRIPONVAULT|VAULTONLY}
```

**Parameters**

**ALLOWTRIPONVAULT**

Directs ConGroup to allow a trip event to proceed even if that trip event occurred during a VMAX vaulting process. This overrides the default behavior of converting a trip into a swap during vaulting. Other parameters are not affected.

**INTERVENTIONREQUIRED**

Performs a swap when any device in the consistency group “drops Ready.”

You can abbreviate INTERVENTIONREQUIRED as INTREQ.

**NOPATHS**

Performs a swap when access to a device is lost.

**SYNCLINKFAILURE**

Causes a swap by programmatically setting devices to *Not Ready* whenever an SRDF link failure occurs.

Use of SYNCLINKFAILURE requires Enginuity patch 36705.
Use SYNCLINKFAILURE with caution. SYNCLINKFAILURE is not sensitive to Concurrent SRDF or SRDF/Star configurations.

VAULTONLY

Overrides of INTERVENTIONREQUIRED or NOPATHS (if specified) and specifies that the only time an AutoSwap may occur is in the case of a detected trip during vaulting. In that case, the trip will be converted into a swap.

UNPLANNEDOPTIONS

Makes FBA R2 devices Not Ready on the channel.

This parameter is only intended for unplanned swaps including FBA devices.

Syntax

UNPLANNEDOPTIONS=FBAUSRNRDY
Running ConGroup

Start ConGroup as soon after IPL as possible and keep it running at all times.

EMC recommends that you include activation of ConGroup in the IPL procedures on all your systems.

Run the ConGroup started task at the same priority as VTAM or JES or at a higher priority.

Customizing ConGroup started task

ConGroup is intended to run as a started task, although you can also run it as a batch job.

The sample started task for ConGroup is available in the EMCCGRP member of the Mainframe Enablers SAMPLIB.

Customize EMCCGRP according to your site requirements. When finished, copy the customized EMCCGRP member to a system PROCLIB that is used for started task START commands.

Sample JCL

The JCL to run ConGroup is as follows:

```
//CONGROUP  EXEC  PGM=CGRPMAIN,REGION=0M
//STEPLIB   DD  DISP=SHR,DSN=ds_prefix.LINKLIB
//SCF$nnnn  DD  DUMMY
//CONFIG    DD  DISP=SHR,DSN=config_file
//CONFIGCA  DD  DISP=SHR,DSN=swap_config_file
```

Where:

- `ds_prefix` is the product dataset name prefix you specified during installation of Mainframe Enablers as described in the *Mainframe Enablers Installation and Customization Guide*.
- `nnnn` identifies the ResourcePak Base task that the ConGroup job runs against.
- `config_file` is the name of the configuration file, as described in “Configuring ConGroup” on page 28.
- `swap_config_file` is the name of the dataset that contains the definition statements for the swap (complement) group. “Defining swap (complement) groups” on page 123 describes swap (complement) groups.
Starting SCF

ConGroup requires the Symmetrix Control Facility (SCF) to be fully active before ConGroup initializes.

**Note:** SCF is the key component of ResourcePak Base. The *ResourcePak Base for z/OS Product Guide* describes SCF.

Ensure that SCF is running before you start ConGroup. To start SCF, if necessary, follow the instructions provided in the *ResourcePak Base for z/OS Product Guide*.

- If SCF is not running when you start ConGroup, you receive the CGRP513E message.
- If SCF has been started, but not fully initialized when you start ConGroup, then tests at one-minute intervals are performed for 30 minutes. If SCF becomes active during the 30 minutes, ConGroup continues normal initialization. If ConGroup has not detected an active SCF after 30 minutes have passed, ConGroup shuts down.

This behavior may provide the misleading impression that ConGroup protection is in force simply because ConGroup is running, when in fact ConGroup is waiting for SCF to become active in order to proceed with its own initialization. A fully initialized ConGroup is evidenced by CGRP149I messages for each specified group of devices being protected.

- If SCF is not running while ConGroup is running, any ConGroup commands issued receive the following message:

```
EMC SCF IS NOT AVAILABLE - reason
```

Where:

- `reason` is one of the following:
  - SERVICE EMCSAI FAILED
  - SERVICE SAICALL FAILED

Starting ConGroup

To start ConGroup, type the following command at a console:

```
S emccgrp[,REUSASID=YES][,SUB=MSTR]
```

Where:

- `emccgrp` is the name of the ConGroup started task, normally EMCCGRP.
- `REUSASID=YES` avoids permanent loss of ASIDs\(^1\) during the life of an IPL.
- `SUB=MSTR` enables you to swap page datasets, JES2 checkpoint datasets, and couple datasets. Refer to “Starting ConGroup with SUB=MSTR” on page 65 for additional information.

---

1. Address Space Identifier
Starting ConGroup with SUB=MSTR

EMC recommends running ConGroup with SUB=MSTR to avoid any potential deadlock conditions with JES during consistency group suspend operations.

**IMPORTANT**

Use SUB=MSTR if you are running CAX. Running SUB=MSTR is not required on GDDR C-systems.

If any of your consistency groups could ever include devices containing JES2 or JES3 datasets, you must run ConGroup with SUB=MSTR.

If you do not, you may obtain an undesirable result if you run ConGroup under JES, and ask ConGroup to control JES datasets. In such a case, ConGroup may request services from JES (such as WTO) at the same time as it suspends I/O to JES devices. Running with SUB=MSTR resolves this issue and also gives the ConGroup address space higher priority. This can be advantageous in heavily loaded systems.

*Note:* When ConGroup is installed and started with SUB=MSTR, then SCF should also be started with SUB=MSTR, as described in the *ResourcePak Base for z/OS Product Guide.*

Running ConGroup on multiple LPARs

ConGroup can run on one LPAR or on multiple LPARs. If you install ConGroup on multiple LPARs, you can have ConGroup instances on several LPARs protect the same consistency groups or have several ConGroup instances on different LPARs protecting different consistency groups.

To enable the multi-LPAR mode, use the MODE configuration parameter, as described in “MODE” on page 36.

In the multi-LPAR mode, a ConGroup address space (task) only needs to be on every LPAR if CAX is used.

*Note:* “Swapping Consistency Groups with CAX” on page 109 describes CAX.

Running multiple ConGroup instances on single LPAR

**CAUTION**

EMC does not recommend running multiple ConGroup instances on a single LPAR.

You can execute up to seven ConGroup started tasks (all being version 7.2 and higher), plus one additional ConGroup started task (version 7.0) – all in the same LPAR. Each ConGroup started task can manage multiple consistency groups within the LPAR.

When running multiple ConGroup instances on a single LPAR, it is imperative that their configuration files do not interfere with each other. They should protect different devices.
Stopping ConGroup

To stop ConGroup, issue either the z/OS STOP (P) command or the functionally equivalent MODIFY (F):

```
P emccgrp
```

or

```
F emccgrp, STOP
```

Where:

◆ `emccgrp` is the name of the ConGroup started task.

If the ALL-CONGROUPS lock is currently held, shutdown will continually check for the release of the lock and shutdown when released. If it has not been released within the five minute wait period, ConGroup terminates anyway at the end of five minutes.

You can also use the following syntax with the z/OS MODIFY command to force stop ConGroup:

```
F emccgrp,STOP,FORCE
```

Swap service messages

ConGroup can generate swap service messages. The AutoSwap for z/OS Product Guide describes the format of swap service messages. The Mainframe Enablers Message Guide message guide lists the messages.
CHAPTER 3
Managing Consistency Groups

This chapter covers the following topics:

- Overview................................................................................................................. 68
- Summary of operations........................................................................................... 74
- Creating consistency groups................................................................................... 76
- Managing consistency group members ................................................................. 77
- Protecting specific R1-R2 pairs............................................................................. 90
- Verifying consistency groups.................................................................................. 94
- Enabling/disabling consistency groups ................................................................. 95
- Monitoring consistency groups............................................................................... 97
- Tripping consistency groups.................................................................................. 99
- Resuming consistency group operations ................................................................ 105
Overview

What is consistency group?

A *consistency group* is a group of devices for which ConGroup ensures data consistency, allowing you to restart applications from the remote site in case of an outage.

From the configuration point of view, a basic consistency group includes R1s, as shown on Figure 4 on page 22.

However, from a functional point of view, ConGroup ensures consistency of data that reside on the associated R2s.

The R2s are not specified in the consistency group configuration parameters explicitly. Instead, they are implied by the SRDF group numbers of the R1 remote mirrors.

In addition to ensuring consistency on R2s, ConGroup protects the data residing on the BCVs attached to the R1s or R2s (if any). In case of a write failure, ConGroup can split the BCVs from their STDs to maintain a consistent copy of data on the BCVs.

*Note:* Local and remote STDs/BCVs are configured and processed differently. “STD devices” on page 79 describes handling of local STDs and their BCVs. “Splitting remote BCVs” on page 101 discusses protection of remote BCVs attached to R2s.

An extended consistency group is illustrated in Figure 6.

![Extended consistency group](image)

*Figure 6*  Extended consistency group

*Note:* When CAX is enabled for a consistency group, the consistency group is referred to as a *CAX-enabled consistency group* or simply a *CAX group.*
Consistency group lifecycle

Figure 7 provides an overview of basic steps in the consistency group lifecycle.

The consistency group lifecycle includes the following steps:

Create consistency group — You define your consistency group using ConGroup configuration parameters. This process is described in “Creating consistency groups” on page 76. After the group is defined, you fill it with devices as described in “Managing consistency group members” on page 77.

Verify consistency group — You can verify configuration of your consistency group either automatically or manually, as described in “Verifying consistency groups” on page 94. This step is optional.

Enable consistency group — To have ConGroup maintain data consistency in your consistency group, enable it as described in “Enabling/disabling consistency groups” on page 95. Note that you can deactivate ConGroup protection at any time by disabling the consistency group.

Monitor consistency group — ConGroup monitors devices in the consistency group for communication or device failures. The monitoring process is described in “Monitoring consistency groups” on page 97. For CAX-enabled groups, see “Monitoring” on page 114.
**Write failure** — A write failure occurs when data is unable to be propagated to the R2. This can be caused by communication issues or device failures.

- When ConGroup is used without CAX, it detects SRDF link failures and R2 device failures. When such failure is detected, ConGroup stops monitoring and trips the consistency group.

- When CAX is enabled, ConGroup additionally detects host I/O (channel) failures and R1 device failures. When such failure is detected, ConGroup stops monitoring and swaps I/O to the R2s.

**Trip consistency group** — Tripping includes halting I/O to R1s and suspending SRDF transfer while maintaining data consistency, as described in “Tripping consistency groups” on page 99.

**Swap I/O to R2s** — This step is for CAX-enabled groups only. When CAX is enabled and a host I/O or R1 device failure occurs, ConGroup automatically swaps I/O from the R1s in your consistency group to R2s, as described in “Swapping Consistency Groups with CAX” on page 109.

**Resume operations** — Specific procedures you perform to resume operations depend on whether CAX is enabled for the consistency group:

- If CAX is not used, refer to “Resuming consistency group operations” on page 105.

- For CAX-enabled consistency groups, you can either leave I/O on the R2s or swap back to R1s, as described in “Resuming operations after swap” on page 121.
Consistency group states

A consistency group can be either enabled or disabled. When the group is enabled, ConGroup protection is switched on for the group. Disabling the consistency group means there is no ConGroup protection active; the consistency group is essentially a defined group of devices with nothing to do.

Note: “Enabling/disabling consistency groups” on page 95 describes enabling and disabling a consistency group.

Substates of enabled consistency group

An enabled consistency group can be in one of the following substates:

- **Active**

  This is the normal state for a consistency group. Data is flowing from R1s to R2s; the consistency group is monitored and eligible to be suspended.

  Figure 8 illustrates the active state.

  ![Figure 8 - Active state](image)

  **Figure 8**  Active state

- **Suspend pending**

  An error has occurred, causing ConGroup to suspend the consistency group; that is, to stop the data flow to R2s. While in this state, I/O to R1s is held until the suspend process completes.

  Figure 9 illustrates the suspend pending state.

  ![Figure 9 - Suspend pending state](image)

  **Figure 9**  Suspend pending state

Note: “Suspending consistency group operations” on page 100 describes suspending operations.
Managing Consistency Groups

- **Suspended**

  The suspend process has completed. Data has stopped flowing to R2s, and I/O has resumed to R1s. The R2s are in a consistent state.

  Figure 10 illustrates the suspended state.

  ![Figure 10 Suspended state](image)

  **Note:** “Suspending consistency group operations” on page 100 describes suspending operations.

- **Remote split in process**

  ConGroup is in the process of splitting all R2s devices from their BCVs, if present. This state implies that the consistency group is suspended.

  Figure 11 illustrates the remote split in process state.

  ![Figure 11 Remote split in process state](image)

  **Note:** “Splitting remote BCVs” on page 101 describes remote splits.

- **Resume in process**

  ConGroup is in the process of resuming the data flow to R2s, or has completed the process and is now waiting for all R1s in the consistency group to synchronize with their R2s.

  Figure 12 illustrates the resume in process state.

  ![Figure 12 Resume in process state](image)

  **Note:** “Resuming consistency group operations” on page 105 describes resume operations.
Blocked consistency group

A consistency group can be blocked when one of the following occurs:

- An abend occurs within the address space, which does not cause the ConGroup started task to end.
- The consistency group fails CAX validation.

**Note:** "Swapping Consistency Groups with CAX" on page 109 describes CAX.

A periodic health checker eventually detects that an abend has occurred and then sets every consistency group to BLOCKED. The blocked state is also checked whenever an MSC\(^1\) registration request is attempted. The BLOCKED state is displayed when you issue the ConGroup DISPLAY command, for example:

```
CONGROUP= CGRPX1B  DISABLED  ACTIVE  BLOCKED
```

The BLOCKED state may be cleared with the ConGroup RESET command. This can be done if and when the user determines that the prior abend or CAX validation issue has appropriately recovered or does not affect this particular group.

---

1. MSC stands for Multi-Session Consistency. The *SRDF Host Component for z/OS Product Guide* describes MSC.
# Summary of operations

## Managing consistency groups

Table 1 lists operations for managing consistency groups.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create consistency group</td>
<td>CONGROUP configuration parameter</td>
</tr>
<tr>
<td>Enable consistency group</td>
<td>ENABLE command</td>
</tr>
<tr>
<td>Verify consistency group</td>
<td>• VERIFY command</td>
</tr>
<tr>
<td></td>
<td>• RESET command, VERIFY parameter</td>
</tr>
<tr>
<td>Set verification interval</td>
<td>• VERIFY_INTERVAL configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• SET VERIFY_INTERVAL command</td>
</tr>
<tr>
<td>Whether to make devices Not Ready when ECA Window times out</td>
<td>USE_NR_ON_TO configuration parameter</td>
</tr>
<tr>
<td>Trip consistency group</td>
<td>TRIP command</td>
</tr>
<tr>
<td>Set response to suspend failure</td>
<td>SUSPEND_FAILURE configuration parameter</td>
</tr>
<tr>
<td>Set suspend retry period</td>
<td>SUSPEND_RETRY_TIMEOUT configuration parameter</td>
</tr>
<tr>
<td>Split remote (R2) STDs from their BCVs</td>
<td>REMSPLIT command</td>
</tr>
<tr>
<td>Set remote split retry interval</td>
<td>REMSPLIT_INTERVAL configuration parameter</td>
</tr>
<tr>
<td>Whether to issue message CGRP314E during remote split</td>
<td>REMSPLIT_OPTION configuration parameter</td>
</tr>
<tr>
<td>Resume operations for consistency group</td>
<td>RESUME command</td>
</tr>
<tr>
<td>Set synchronization check interval for resume</td>
<td>RESUME_INTERVAL configuration parameter</td>
</tr>
<tr>
<td>Set up messages issued during resume</td>
<td>RESUME_OPTION configuration parameter</td>
</tr>
<tr>
<td>Disable consistency group</td>
<td>DISABLE command</td>
</tr>
<tr>
<td>Disable consistency group that fails verification</td>
<td>DISABLE_ON_VERIFY_ERROR configuration parameter</td>
</tr>
<tr>
<td>Disable all consistency groups at ConGroup shutdown</td>
<td>DISABLE_AT_SHUTDOWN configuration parameter</td>
</tr>
</tbody>
</table>
Managing Consistency Groups

Managing consistency group members

Table 2 lists operations for managing consistency group members.

Table 2  Managing consistency group members

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add devices to consistency group</td>
<td>• ADD command</td>
</tr>
<tr>
<td></td>
<td>• DEVICE_LIST configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• DEVICE_LIST_STD configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• SYMM_DEV# configuration parameter</td>
</tr>
<tr>
<td>Remove devices from consistency group</td>
<td>• DELETE command</td>
</tr>
<tr>
<td></td>
<td>• EXCLUDE configuration parameter</td>
</tr>
<tr>
<td>Add GNS group to consistency group</td>
<td>SCFG configuration parameter</td>
</tr>
<tr>
<td>Add SMS group to consistency group</td>
<td>SMS_GROUP configuration parameter</td>
</tr>
<tr>
<td>Allow/prohibit couple datasets in consistency groups</td>
<td>COUPLEDS_ALLOWED configuration parameter</td>
</tr>
<tr>
<td>Allow/prohibit page datasets in consistency groups</td>
<td>PAGEDEV_ALLOWED configuration parameter</td>
</tr>
<tr>
<td>Allow/prohibit semi-synchronous devices in consistency groups</td>
<td>SEMISYNC_ALLOWED configuration parameter</td>
</tr>
<tr>
<td>Set allowed subchannels for R1s</td>
<td>ALLOWABLE_MSS configuration parameter</td>
</tr>
</tbody>
</table>

Protecting specific R1-R2 pairs

Table 3 lists operations for protecting specific R1-R2 pairs.

Table 3  Protecting specific R1-R2 pairs

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit protection to specific R1-R2 pairs</td>
<td>SYMGROUP configuration parameter</td>
</tr>
<tr>
<td>Allow/prohibit R1 sharing by mirror</td>
<td>• ALLOW_SHARED_R1S configuration parameter (global)</td>
</tr>
<tr>
<td></td>
<td>• ALLOW_SHARED_R1S configuration parameter (group-specific)</td>
</tr>
</tbody>
</table>
Creating consistency groups

You create consistency groups using the CONGROUP (or SRDF_CONGROUP) configuration parameter described in “CONGROUP” on page 44. After defining a group, you specify the group devices, as described in “Adding devices to consistency group” on page 82.

In the ConGroup configuration file, consistency group definitions come after the global configuration parameters. First comes the statement that defines the con group name. Then you specify statements to include devices and set other group-specific parameters.

Examples

The following examples illustrate different methods of creating a consistency group and filling it with devices. For details, refer to the relevant method in “Adding devices to consistency group” on page 82 and descriptions of the configuration parameters in “Consistency group-specific configuration parameters” on page 42.

1. To create a consistency group with the specified devices:

   CONGROUP=SALES
   DEVICE_LIST=900,C00-C2F,A200-A203

2. To create two consistency groups and add devices using both CUUs and volsers:

   CONGROUP=DB2TEST
   DEVICE_LIST=100-10F,STA001,STA002
   DEVICE_LIST=EC0-EC4
   DEVICE_LIST=B90-B9F
   CONGROUP=DB2PROD
   DEVICE_LIST=FIN001,FIN002,FIN003

3. To create a consistency group that includes an SMS group:

   CONGROUP=ENNGNEER
   DEVICE_LIST=AC0-ACF
   SMS_GROUP=SMSENG

4. To create a consistency group limited to R1-R2 pairs in SRDF group 2:

   CONGROUP=MYGROUP
   SYMGROUP=(000184000345,2)
   DEVICE_LIST=AC0-ACF

5. To create a consistency group that includes both R1s and local STDs:

   CONGROUP=MIXED
   DEVICE_LIST=100-10F,STA001,STA002
   DEVICE_LIST_STD=STD000,STD001
Managing consistency group members

Device requirements

Devices in a consistency group meet the following requirements:

- A device is on a VMAX system that meets the requirements listed for VMAX systems in the *Mainframe Enablers Installation and Customization Guide*.
- A device can be an Open System or a mainframe device.
- A device is an R1 associated with its R2 or a local STD established with its BCV.

*Note:* The STD cannot be any type of SRDF device (R1, R2, and so on). Additionally, there must be at least one R1 in the group.

**IMPORTANT**

A consistency group cannot include R2s. At startup, ConGroup ignores consistency groups that have any R2s, but continues to initialize. ConGroup recognizes R21s and R22s, but treats them as R2s.

Location of devices

The devices in a consistency group can be on different VMAX systems, but all devices must be at the same physical site (see Figure 4 on page 22). You cannot include devices from different physical sites in the same consistency group.

*Note:* The exception is placing STDs in a consistency group with the R2s over a channel extender. In this case, the devices are physically connected locally, but logically they are co-resident with the R2s at the remote site.

Devices added to a consistency group by their VMAX device numbers must reside on the same physical VMAX system.
Device types

A consistency group can contain the following items:

- R1 devices
- Semi-synchronous devices
- STD devices
- FBA devices
- Couple datasets
- Page datasets
- JES checkpoint and spool volumes

R1 devices

A consistency group contains R1s that reside on the primary side of an SRDF configuration.

**Note:** The *SRDF Host Component for z/OS Product Guide* describes R1s.

To add R1s to the consistency group, use the following configuration parameters:

- DEVICE_LIST described in “DEVICE_LIST” on page 45
- SCFG described in “SCFG” on page 48
- SYMM_DEV# described in “SYMM_DEV#” on page 50

You can also exclude devices from processing using the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.

ConGroup supports R1 sharing by mirror. For more information, refer to “R1 sharing by mirror” on page 91.

Semi-synchronous devices

**Note:** Semi-synchronous devices are available under Enginuity 5773.

The SRDF semi-synchronous mode allows the R1s and R2s to be out of synchronization by one write I/O operation.

Use of semi-synchronous devices with consistency groups requires special consideration and knowledge of the nature of the applications concerned.

To determine whether semi-synchronous devices can be part of a consistency group, use the SEMISYNC_ALLOWED configuration parameter described in “SEMISYNC_ALLOWED” on page 40.

You cannot use semi-synchronous devices if you have any CAX groups defined in your configuration file. In this case, the SEMISYNC_ALLOWED parameter is ignored. If a semi-synchronous device appears in a CAX group, ConGroup issues the CGRP273E message.
Managing Consistency Groups

STD devices

A consistency group can include local STDs established with their BCVs.

**Note:** The *TimeFinder/Mirror for z/OS Product Guide* describes STDs and BCVs.

When contained in the consistency group, the STDs are protected by their BCVs rather than R2s. The STD cannot be any type of SRDF device (R1, R2, and so on). Otherwise, ConGroup issues the CGRP199E error message and does not start.

The consistency group is enabled only if the STDs are established with their BCVs.

To add local STDs, use the DEVICE_LIST_STD configuration parameter described in “DEVICE_LIST_STD” on page 46.

Note that local STDs are not monitored for the inability to write to their established BCVs. The devices cannot cause a ConGroup trip, but they are split when a ConGroup trip occurs.

**Note:** “Tripping consistency groups” on page 99 describes ConGroup trips.

FBA devices

You can add FBA (Fixed Block Architecture) devices to your consistency group as follows:

- **(Recommended)** Through GNS, as described in “Adding devices by GNS group” on page 82.

- Using the SYMM_DEV# configuration parameter described in “SYMM_DEV#” on page 50. For FBA meta devices, you can specify both heads and members, but the members are ignored. ConGroup determines the members based on the heads that it finds in the specified list.

- Using the ADD or DELETE command described in “ADD” on page 179 and “DELETE” on page 184. In this case, both meta heads and all members must be specified.
Couple datasets

You can allow or prohibit explicitly adding volumes that contain couple datasets to your consistency group. To do this, use the COUPLEDS_ALLOWED configuration parameter described in “COUPLEDS_ALLOWED” on page 33.

When a consistency group trips, ConGroup sets volumes with couple datasets to high IOS levels. This could cause problems if the wrong couple datasets are added to the consistency group.

**Note:** “Tripping consistency groups” on page 99 describes ConGroup trips.

There are seven types of couple datasets:

- SYSPLEX
- CFRM (Coupling Facility Resource Manager)
- WLM (Workload Manager)
- OMVS
- ARM (Automatic Restart Manager)
- SFM (Sysplex Failure Manager)
- LOGR (System Logger)

ConGroup does not support active XCF couple datasets, with the exception of LOGR couple datasets.

If you are using CICS journaling/logging through the system logger, consider including only the primary LOGR couple dataset in a consistency group. If a LOGR couple dataset is to be included in a consistency group, there must be no other couple datasets on the volume with the primary LOGR couple dataset.

The COUPLEDS_ALLOWED configuration parameter allows you to add the primary couple dataset used by the system logger to the consistency group. No determination as to which couple dataset is added is made, so you should isolate the logger datasets on volume(s) separate from the SYSPLEX couple datasets.

If the primary LOGR couple dataset is to be used as part of a consistency group, you should set up the LOGR structure in the Coupling Facility for duplexing to staging datasets. After the consistency group has tripped, the contents of the primary LOGR couple dataset, the LOGR staging datasets, and the LOGR log stream datasets contain the information necessary to restart CICS.
Page datasets

You can allow or prohibit explicitly adding volumes that contain page datasets to your consistency group. To do this, use the PAGEDEV_ALLOWED configuration parameter described in “PAGEDEV_ALLOWED” on page 37.

⚠️ **CAUTION**

**EMC does not recommend using page datasets in a non-CAX consistency group.**

---

**Note:** The *AutoSwap for z/OS Product Guide* provides more information about page datasets in a swap group.

I/O to devices with page datasets cannot be halted during the suspend processing. This means that during a ConGroup trip, I/O may complete to the remote mirror of a paging volume prior to the SRDF SUSPEND command completing against that volume.

---

**Note:** “Tripping consistency groups” on page 99 describes ConGroup trips. “Suspending consistency group operations” on page 100 describes suspending operations.

Consistency groups that contain devices with page datasets must also include at least one non-page dataset device.

Ensure that devices containing page datasets do not contain data which has dependencies with data on other devices in the consistency group. Failure to do so may result in inconsistent data on the remote side.

If you use the ALL option of the DEVICE_LIST configuration parameter, as described in “DEVICE_LIST” on page 45, then volumes with page datasets are not added to the consistency group unless you take the following steps:

1. Specify the page devices to be included to the consistency group by their CUUs using the DEVICE_LIST configuration parameter.

2. Specify PAGEDEV_ALLOWED=YES.

---

**Note:** CAX operates upon page datasets only when they are on SRDF R1 devices.

**JES checkpoint and spool volumes**

When configuring JES checkpoint and spool volumes in a consistency group, the ConGroup and SCF started tasks must be started with the SUB=MSTR parameter, as described in “Starting ConGroup with SUB=MSTR” on page 65 and the *ResourcePak Base for z/OS Product Guide*. 
Adding devices to consistency group

You can add devices to your consistency group using the following methods:

- Adding devices by GNS group
- Adding devices by SMS group
- Adding devices by CUUs
- Adding devices by VMAX device numbers
- Adding devices by volser
- Adding “ALL” devices

You can combine these methods for a single consistency group. For example, the definition of any consistency group can be composed of one or more device CUUs specified with the DEVICE_LIST configuration parameter described in “DEVICE_LIST” on page 45, one or more VMAX device numbers specified with the SYMM Dev# configuration parameter described in “SYMM_DEV#” on page 50, and one or more GNS groups.

Note: “Examples” on page 76 provides some examples of how to add devices to a consistency group.

Adding devices by GNS group

You can fill your consistency group with devices using the GNS (Group Name Services) feature of ResourcePak Base.

Note: The ResourcePak Base for z/OS Product Guide describes GNS.

The GNS group can be of any type, as long as it contains R1s to be included to the consistency group. The GNS group can include FBA devices.

Advantages

Using GNS to add devices to a consistency group provides the following advantages:

- You can define consistency groups that span multiple LPARs.
- You can add devices in other LPARs and devices currently offline to an LPAR running ConGroup.
- You can update consistency group definitions “on the fly” without having to update the ConGroup configuration file, as described in “Dynamically changing groups”.
- For meta devices, you can let ConGroup determine correct members based on the specified head.
Specifying GNS group

To add devices by GNS group:

1. Create a GNS group, as described in the ResourcePak Base for z/OS Product Guide.
2. Add the GNS group to your consistency group using the SCFG configuration parameter, as described in “SCFG” on page 48. Your consistency group configuration may be as follows:

```
CONGROUP=consistency_group_name
SCFG(gns_group_name)
```

If a GNS group name you use in ConGroup includes blanks, enclose the name in double quotes, for example:

```
SCFG(“West Coast Stores”)
```

Each group name can be from 1 to 65 alphanumeric characters long. Each apostrophe, if used, reduces the maximum possible length of the group name by 1.

You can also use the GNS EXCLUDE parameter (described in the ResourcePak Base for z/OS Product Guide) to exclude certain devices when defining a consistency group. For a given consistency group, there can be one or more EXCLUDE statements.

If ConGroup does not find a GNS group name when it attempts to include the devices of that group into a consistency group, ConGroup ignores the GNS group when starting. The rest of the devices in the consistency group are ConGroup-protected.

Dynamically changing groups

Using GNS groups simplifies the process of changing the contents of a consistency group. To change the contents of a consistency group:

1. Use GNS commands to change the definition of the GNS group, as described in the ResourcePak Base for z/OS Product Guide.
2. In ConGroup, issue the REFRESH command described in “REFRESH” on page 193 to update the consistency group configuration.

**Note:** As far as GNS group name is the same, no changes are required to the ConGroup configuration file.

Adding devices by SMS group

You can add an entire SMS group to your consistency group by using the SMS_GROUP configuration parameter described in “SMS_GROUP” on page 48.

Adding devices by CUUs

You can add individual devices or device ranges to your consistency group by device CUU.

To add devices by CUU, use one of the following configuration parameters:

- For R1s, use DEVICE_LIST described in “DEVICE_LIST” on page 45.
- For local STDs, use DEVICE_LIST_STD described in “DEVICE_LIST_STD” on page 46.

You can also exclude devices from ConGroup processing using the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.
Adding devices by VMAX device numbers

You can add individual devices or device ranges to your consistency group by VMAX device number.

ConGroup supports 4-byte VMAX device addresses.

VMAX device numbers and ranges of VMAX device numbers can include FBA devices. For FBA meta devices, only heads are required.

**IMPORTANT**

Devices added by their VMAX device numbers must reside on the same physical VMAX system.

To add devices by VMAX device number, use the SYMM_DEV# configuration parameter described in “SYMM_DEV#” on page 50.

Adding devices by volser

When using volsers, ensure that all the devices being added to a consistency group are online to all LPARs participating in the consistency group. This guarantees that a consistency group trip includes all of the devices on every LPAR.

**Note:** “Tripping consistency groups” on page 99 describes consistency group trips.

To add devices by volsers, use one of the following configuration parameters:

- For R1s, use DEVICE_LIST described in “DEVICE_LIST” on page 45.
- For local STDs, use DEVICE_LIST_STD described in “DEVICE_LIST_STD” on page 46.

You can also exclude devices from ConGroup processing using the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.

ConGroup also supports volser masks for specifying devices to be included in a consistency group. Volsers for devices to be included in a consistency group cannot contain hyphens.
Adding “ALL” devices

Instead of specifying particular devices, you can use the ALL option provided with the DEVICE_LIST configuration parameter, as described in “DEVICE_LIST” on page 45.

You can use ALL to add all devices and then exclude some of devices using the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.

Every device included with the ALL option will meet the following criteria:

- The device is defined on the mainframe host.
- A device is not on the exclude list.

**Note:** You can create an exclude list using the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.

- A device meets the requirements listed in “Device requirements” on page 77.
- A device is not a meta device.
- A device does not contain a couple dataset.
- A device does not contain a page dataset.
Adding/removing devices dynamically

Dynamic addition and removal of devices in ConGroup allow for modification of a running configuration without issuing the REFRESH command and without disruption to the affected group or to other groups. This helps achieve the capability of 24x7 hour operation of ConGroup.

- To add devices dynamically, use the ADD command described in “ADD” on page 179.

  **IMPORTANT**
  Devices are required to be in sync before executing the ADD command.

  Devices can only be added to existing consistency groups.

  If a range specified for the ADD command includes non-existent devices (path offline), those devices are skipped and ignored. The remaining devices are then tested for qualification. If, after the path-offline addresses are excluded, any range includes any non-qualifying devices, the entire request fails and no devices are added.

  The ADD command can be executed against any ConGroup address space participating in the same CG set.

  **Note:** “CG sets” on page 135 describes CG sets.

  For SRDF/Star configurations, execution of the ADD command must follow the resume processing of devices that were moved into the production Star SRDF/S group, but precede the movement of devices into the SRDF/A group.

  **Note:** The VMAX All Flash Product Guide and VMAX3 Family Product Guide describe SRDF/Star.

- To remove devices dynamically, use the DELETE command described in “DELETE” on page 184.

  The ADD and DELETE commands must pass a SAF test to be executed. ConGroup performs a RACF check against the XFACILIT class and a resource name EMC.ADMIN.CMD.CG.ADDDEL.

  Dynamic addition and removal modify the running ConGroup configuration. This is true for one or more connected ConGroup address spaces. Dynamically added or deleted devices are not reflected in the new configuration. If a restart of ConGroup (or a refresh) occurs, the dynamic changes will be lost. In other words, dynamic add/delete is not persistent.

  Because of this, once a dynamic ADD/DELETE is done, ConGroup no longer permits a new ConGroup address space to join the network of existing connected ConGroup address spaces. This is because a new ConGroup could possibly be using a configuration file that does not match the running configuration. If a new ConGroup address space is started, it will be forced (by the other ConGroups) to shut down immediately. Message CGRP653E will be issued by the interloping ConGroup before it comes down.
Limitations include:

- ADD or DELETE is not allowed if the ALLOW_SHARED_R1S configuration parameter is set to YES, as described in “ALLOW_SHARED_R1S” on page 30.
- If any devices fail validation, the entire request fails and no devices are added.
- Devices that have been added using the ADD command are not persistent. A subsequent REFRESH will remove dynamically added devices unless the configuration file is first updated to include the added devices.

**Note**: If you are using GNS groups defined by inclusion of RA groups, then any devices dynamically added to ConGroup with ConGroup’s dynamic ADD function (that are also new to the RA group) will automatically be picked up by GNS at the next REFRESH (by virtue of now being in the RA group). Additionally, if you add a device to a GNS group by device number, it will also be there after the next REFRESH.
Using multiple subchannel sets

Multiple Subchannel Set (MSS) is a z/Architecture and z/OS operating feature that allows applications to access devices in the traditional addressable 4-digit device number range from 0000 to FEFF and, at the same time, allows system addressable devices to be contained in additional subchannel sets. The range of the subchannel set numbers is 0-3, although presently only 0 to 2 are used.

Devices are addressed by z/OS in a particular subchannel set using a 5th digit on the device number. When 5-digit device numbers are displayed or required for command input, they are usually in the form sdddd, where s is the subchannel set number and dddd is the 4-digit z/OS device number. For example, subchannel set 0 devices are 00000-0FEFF, subchannel set 1 devices are 10000-1FEFF, and so on.

Note: To enable multiple subchannel addressing for your system, use the SCF.DEV.MULTSS initialization parameter of SCF described in the ResourcePak Base for z/OS Product Guide.

The base, or active, subchannel set are those devices that are accessible to applications. These are the devices that are online and available.

The alternate subchannel set devices are those not online and are only available for system or “special” use. Along with the current 3390A PAV alias devices, z/OS allows 3390S and 3390D devices to be defined in the alternate subchannel sets. 3390S and 3390D device types are known as SPECIAL devices.

Using MSS with CAX

When using CAX, CAX allows 3390D (secondary) devices to be defined as the target (TO) devices in a swap group. The source (FROM) devices are always in the base, or active, subchannel set.

The 4-digit device portion of the device number is identical for the same device pair. If the source (FROM) was device 01234, then the corresponding target (TO) device would be 11234. In a swap group, all of the 3390D devices must be contained in the same subchannel set. Some system volumes, like the IPL device, cannot be contained within a subchannel set other than 0.

Once the swap is complete, the target (TO) devices are now the base devices even though a non-0 subchannel set is still used. The source (FROM) devices are now the 3390D SPECIAL device type. They are inaccessible to any normal application usage.

When an IPL is required following a swap to devices in subchannel sets 1-3, the correct devices need to be selected by z/OS as the base devices during IPL. The IODF\(^1\) statement of the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB allows devices in a subchannel set other than 0 to be selected as the base, or active, devices.

---

1. Input/Output Definition File
Related parameters

The ALLOWABLE_MSS configuration parameter described in “ALLOWABLE_MSS” on page 42 (and the related PAIR subparameter of the CONGROUP statement described in “CONGROUP” on page 44) is required when using CAX and if any of the device pairs “straddle” two subchannel sets. The subparameters define what subchannel sets on the R1 devices are allowed. They may also be specified whether or not the device pairs straddle subchannel sets.

As always, R2s seen in a group always cause the group to be bypassed. But with R2s on an alternate subchannel, ConGroup does not see them as R2s because they are the same channel address as the R1s. The PAIR and ALLOWABLE_MSS parameters are necessary to force a bypass when an R1 subchannel set does not match the “allowable” sets. Also, if the first group of a pair is not bypassed, then the 2nd group in the pair is always bypassed.

Specifying the PAIR and ALLOWABLE_MSS parameters is the only way to always correctly bypass one of the two groups, regardless of statement ordering, when using an alternate subchannel set. The PAIR and ALLOWABLE_MSS parameters provide an “allowable” view of which alternate subchannels should be treated as part of this ConGroup pair.

Examples

1. To create a consistency group named GROUPA and add it to pair MYPAIR, allowing R1s from subchannel set 0 only:

   CONGROUP=GROUPA, PAIR(MYPAIR)
   ALLOWABLE_MSS(0)
   ...device definitions...

2. To create a consistency group named GROUPB and add it to pair MYPAIR, allowing R1s from subchannel set 0 or 1:

   CONGROUP=GROUPB, PAIR(MYPAIR)
   ALLOWABLE_MSS(0,1)
   ...device definitions...
Protecting specific R1-R2 pairs

ConGroup in Concurrent SRDF

When using ConGroup with Concurrent SRDF, you can have one or both of the R2s (paired to the R11 that is part of your consistency group) to be protected. To determine the R2 to be protected, specify the SRDF group that includes the R2 using the SYMGROUP configuration parameter described in “SYMGROUP” on page 51.

You are required to have at least one R1-R2 pair in the consistency group.

The protected leg is required to be in the SRDF synchronous or semi-synchronous mode, while the unprotected leg can be set to the Adaptive Copy mode.

**Note:** The VMAX All Flash Product Guide and VMAX3 Family Product Guide describe SRDF modes and topologies.

Example

*Figure 13* illustrates use of ConGroup in the Concurrent SRDF configuration.

As shown in *Figure 13*, ConGroup is protecting the R1-R2 device pairs connected by SRDF groups 2 and 4, while the R1-R2 pairs on SRDF group 7 are not protected.

To implement this configuration, specify SRDF groups 2 and 4 in the SYMGROUP configuration parameter described in “SYMGROUP” on page 51.
R1 sharing by mirror

What is R1 sharing by mirror?

R1s with two concurrent synchronous legs to be protected by two independent consistency groups.

You can assign specific mirrors of an R1 to different consistency groups using GNS statements or with the SYMGROUP configuration parameter described in “SYMGROUP” on page 51.

Eligibility for R1 sharing by mirror is determined by the order in which groups are specified in the ConGroup configuration file. After an R1 mirror is assigned to a consistency group, that mirror is no longer eligible for assignment to any subsequently defined consistency group.

Therefore, an R1 can only be shared if it is concurrent (that is, R11). One of the mirrors is assigned to one group and the other mirror is assigned to another group. This means that each of the sharing groups must restrict their assignment of R1 mirrors to those mirrors not assigned by another group.

Enabling R1 sharing by mirror

To enable R1 sharing by mirror, use the global ALLOW_SHARED_R1S configuration parameter described in “ALLOW_SHARED_R1S” on page 30 or set the group-specific ALLOW_SHARED_R1S configuration parameter described in “ALLOW_SHARED_R1S” on page 42 for a specific group. The value set for the consistency group-specific ALLOW_SHARED_R1S overrides the value of the global ALLOW_SHARED_R1S for the current consistency group.

**Note:** If you attempt to share an R1 with an Enginuity level that does not support R1 sharing by mirror, ConGroup returns an error.

If you enable R1 sharing by mirror, duplicate device checking is replaced by duplicate checking at the mirror level. Any explicit or implicit SYMGROUP statements defined for the consistency group or groups are used to assign mirrors to the group(s).

If R1 sharing by mirror is disabled (the default), ConGroup does not use duplicate checking at the mirror level.
Using SYMGROUP with R1 sharing by mirror

When R1 sharing by mirror is enabled and ConGroup protection is not limited to specific R1-R2 pairs using the SYMGROUP configuration parameter, as described in "SYMGROUP" on page 51, ConGroup protects both legs. This is equivalent to specifying SYMGROUP and listing both SRDF groups.

If you enable R1 sharing by mirror for a specific consistency group, you can use the SYMGROUP configuration parameter to set up protection for the legs as follows:

- If you apply SYMGROUP to both consistency groups, ConGroup protects both legs.
- If you omit SYMGROUP completely, ConGroup also protects both legs.
- If you apply SYMGROUP to only one of the groups, only that group is protected.

Example

Consider the following example using consistency groups 04 and 05 in VMAX system MEDREC:

CONGROUP=MEDREC
SUSPEND_FAILURE=FAIL
SUSPEND_RETRY_TIMEOUT=45
SYMGROUP=(DLR1,04)
DEVICE_LIST=8A00-8A0F

"SYMGROUP=(DLR1,04)" means that the remote mirrors of devices 8A00-8A0F that are associated with SRDF group 04 are protected. A write failure on leg 04 suspends leg 04, but not leg 05.

Protection is not enabled for leg 05 of the consistency group. For consistency purposes, write failures detected on leg 04 or leg 05 have no effect on leg 05.
Example

1. In the following example, two consistency groups, MSFTIGO1 and MSFTIGO2, are established for the same device list, 8702-870A. R1 sharing by mirror is enabled, and the SYMGROUP statement is used to protect remote mirrors of devices associated with the specified SRDF groups in both consistency groups:

   CONGROUP=MSFTIGO1
   ALLOW_SHARED_R1S=YES
   SUSPEND_FAILURE=FAIL
   SUSPEND_RETRY_TIMEOUT=50
   SYMGROUP=(000192600304,80)
   DEVICE_LIST=8702-870A
   CONGROUP=MSFTIGO2
   ALLOW_SHARED_R1S=YES
   SUSPEND_FAILURE=FAIL
   SUSPEND_RETRY_TIMEOUT=50
   SYMGROUP=(000192600304,C0)
   DEVICE_LIST=8702-870A

2. In the following example, the two consistency groups, MSFTIGO1 and MSFTIGO2, are established for the same device list, 8702-870A. R1 sharing by mirror is enabled, but no SYMGROUP statements are used. Therefore, remote mirrors of devices associated with the specified SRDF groups in both consistency groups are protected:

   CONGROUP=MSFTIGO1
   ALLOW_SHARED_R1S=YES
   SUSPEND_FAILURE=FAIL
   SUSPEND_RETRY_TIMEOUT=50
   DEVICE_LIST=8702-870A
   CONGROUP=MSFTIGO2
   ALLOW_SHARED_R1S=YES
   SUSPEND_FAILURE=FAIL
   SUSPEND_RETRY_TIMEOUT=50
   DEVICE_LIST=8702-870A
Verifying consistency groups

You can verify your consistency group either automatically or manually.

Manual verification

You can run a set of verification tests for your consistency group before enabling the group configuration. For each device experiencing an error, a message is displayed explaining the reason for the error.

To verify a consistency group, use the VERIFY command described in “VERIFY” on page 198. To display detailed verification information, specify VERBOSE.

Figure 14 shows a verify statement for consistency group TVCGCS and sample output:

```
VERIFY TVCGCS
CGRP282I VERIFY TVCGCS
CGRP170I ALL DEVICES FOR CONGROUP TVCGCS HAVE BEEN VERIFIED
```

Figure 14  Verify statement and output

Figure 15 shows verbose verification information for consistency group TVCGCS:

```
VERIFY TVCGCS VERBOSE
CGRP282I VERIFY TVCGCS VERBOSE
CGRP163W R2 DEVICE FOR DEV# 0000DD IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 0000DE IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 0000DF IS TARGET NOT READY
CGRP159W R2 DEVICE FOR DEV# 0000A7 HAS  344 INVALID TRACKS
CGRP163W R2 DEVICE FOR DEV# 000923 IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 000E50 IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 000E51 IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 000E52 IS TARGET NOT READY
CGRP163W R2 DEVICE FOR DEV# 000E53 IS TARGET NOT READY
CGRP170I ALL DEVICES FOR CONGROUP TVCGCS HAVE BEEN VERIFIED
```

Figure 15  Verbose verification information

Automatic verification

ConGroup provides an automatic verification subtask that periodically verifies the state of all enabled/active consistency groups to ensure that all consistency group devices are still in the expected state.

You can set the automatic verification interval using the VERIFY_INTERVAL configuration parameter described in “VERIFY_INTERVAL” on page 41 or the SET VERIFY_INTERVAL command described in “SET VERIFY_INTERVAL” on page 196.

Automatic verification is enabled by default. To disable automatic verification, set the verification interval to 0 (zero).

Note: The automatic verification logic is disabled for swapped or swapping consistency groups.
Enabling/disabling consistency groups

Enabling consistency groups

Enabling a consistency group activates ConGroup protection for the group, including dynamic creation of a related swap group if necessary.

By default, when ConGroup starts, all consistency groups are in the enabled state.

**Note:** Swap groups are used with CAX. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

If ConGroup encounters a problem and cannot enable the consistency group, it puts the consistency group in the disabled state. When the problem is resolved, you can use the ENABLE command described in “ENABLE” on page 189 to enable the consistency group.

In the multi-LPAR mode, ConGroup automatically coordinates ENABLE requests on all LPARs. In the single-LPAR mode, you have to manually issue ENABLE requests on all relevant LPARs.

**Note:** “Single/multi-LPAR mode” on page 133 describes single-LPAR and multi-LPAR modes.

Disabling consistency groups

Disabling manually

Disabling a consistency group means removing ConGroup protection from the group.

To disable a consistency group, use the DISABLE command described in “DISABLE” on page 186.

The DISABLE command changes the status of the devices in the consistency group. It removes both ConGroup and ECA protection and also deletes any related swap group.

Disabling a consistency group turns off ConGroup protection for all the devices in that consistency group, regardless of which hosts are connected to the devices.

If CAX is not used, a DISABLE command issued on one LPAR disables the consistency group on all LPARs. However, when using CAX, you should issue the DISABLE command on the owner LPAR to delete the swap group.

If you issue a DISABLE command on a non-owner LPAR when using CAX, the ConGroup bits would be turned off, the AutoSwap validation group would be marked invalid and a subsequent swap would fail.

**Note:** “Swapping Consistency Groups with CAX” on page 109 describes CAX.
Managing Consistency Groups

Disabling due to verification errors

You can set ConGroup to disable an enabled consistency group if the automatic verification logic finds that the devices in the consistency group are not in the expected state.

Note: “Automatic verification” on page 94 describes automatic verification of consistency groups.

To do this, use the DISABLE_ON_VERIFY_ERROR configuration parameter described in “DISABLE_ON_VERIFY_ERROR” on page 34.

DISABLE_ON_VERIFY_ERROR is always NO for consistency groups that also have a swap group associated with them. If you make an R1 Not Ready in such cases, automatic verification still functions, but does not disable the consistency group. Disabling the consistency group would delete the swap group.

Disabling at shutdown

You can also make ConGroup remove ConGroup protection from all consistency groups when ConGroup is shutdown.

To do this, use the DISABLE_AT_SHUTDOWN configuration parameter described in “DISABLE_AT_SHUTDOWN” on page 33.

When ConGroup is shut down, all consistency groups are disabled. If you use the multi-LPAR mode, the last ConGroup instance to be shut down disables the groups. You must shut down the owner system last.

If you need to shutdown a non-owner system last, first issue the TAKEOVER command to change the ownership and then proceed with the shutdown on the now-current owner LPAR.

Note: Disabling at shutdown is particularly important when ConGroup is active on multiple z/OS images.
Monitoring consistency groups

During normal operation, ConGroup monitors the consistency group for signs that a write is unable to propagate to the R2; that is, that the ending status for the write has not yet been presented to the application.

When the VMAX system finds such a problem, it defers completion of the current write operation and sets the ECA Window state to Open.

**Note:** “What is ECA?” on page 98 describes ECA.

If ConGroup finds a device in the ECA Window Open state, ConGroup opens the ECA Window on every other device in the consistency group.

All devices defined in this consistency group enter deferred write completion until the ECA Window is closed.

ConGroup then suspends SRDF/S for all members of the consistency group.

**Note:** “Suspending consistency group operations” on page 100 describes suspending operations.

After all R1-R2 pairs are *Target Not Ready*, ConGroup closes the ECA Windows on all the affected devices, thus allowing I/Os to resume (albeit only to the R1s).

**Note:** “Resuming consistency group operations” on page 105 describes resuming operations.

This ensures that all R2s in this consistency group are dependent write consistent.
What is ECA?

ConGroup uses the ECA (Enginuity Consistency Assist) technology to monitor consistency groups.

ECA is a HYPERMAX OS feature that signals to the host software when to open an ECA Window for the group. While host software makes sure that all members have the ECA Window open, HYPERMAX OS blocks host I/O when the ECA Window is open and, in case of SRDF/S, brings R1 SRDF mirrors to the Not Ready state.

The type of ECA used with ConGroup, \textit{SRDF ECA}\textsuperscript{1}, provides enterprise-level consistency protection for synchronous devices by suspending operations across all SRDF/S devices in a consistency group.

SRDF ECA for consistency groups operates within the VMAX system.

ConGroup dispatches batches of service tasks from an internal worker pool (on a schedule determined by ConGroup) to determine the ECA Window state in all the VMAX systems defined in a consistency group.

With SRDF ECA, the VMAX system, rather than the host processor, suspends writes during a ConGroup incident. When a VMAX system determines that a write is not able to propagate to the R2 side, it defers completion of the current write operation and enters the ECA Window Open state.

When a ConGroup service task finds a VMAX system in the ECA Window Open state, ConGroup alerts all other VMAX systems defined in the consistency group to enter the ECA Window Open state. All devices defined in this consistency group enter deferred write completion until the ECA Window is closed.

ConGroup then suspends SRDF/S protection for all members of the consistency group. When the SRDF/S suspends are complete, ConGroup closes the ECA Windows, which allows write I/O operations to resume. All devices on the target (R2) side of SRDF/S relationships in this consistency group definition are dependent write consistent.

SRDF ECA can detect R2 write failures for both CKD and FBA devices.

R1s protected with SRDF ECA can be in the synchronous or semi-synchronous mode. R1s that do not need ECA protection can be set to the Adaptive Copy mode.

\textbf{Note}: The \textit{VMAX All Flash Product Guide} and \textit{VMAX3 Family Product Guide} describe the SRDF modes.

Monitoring with CAX enabled

Refer to “Monitoring” on page 114.

\textsuperscript{1} This type of ECA can also be referred to as “RDF-ECA”.
Tripping consistency groups

ConGroup monitors the R1s in the consistency group for failures to complete a successful write to their R2s. An SRDF link failure or an R2 device failure results in the *consistency group trip*.

**Note:** Consistency group tripping and CAX swap processing are mutually exclusive. Once begun, either one precludes the other.

The consistency group trip includes the following:

1. I/O at the VMAX system is temporarily stalled.
2. Standard SRDF ECA processing is applied to maintain group consistency.
   
   **Note:** “What is ECA?” on page 98 describes ECA.

3. ConGroup suspends all R1 and R2 relationships in the consistency group, as described in “Suspending consistency group operations” on page 100.

4. ConGroup splits the remote BCVs from their STDs (if any), as described in “Splitting remote BCVs” on page 101.
   
   If the consistency group includes local STDs established with their BCVs, ConGroup splits them as well.
   
   **Note:** “STD devices” on page 79 describes local STDs and BCVs in a consistency group.

In Concurrent SRDF configurations, only the devices on the protected leg are suspended during a ConGroup trip.

If a ConGroup trip occurs, you can resume operations as described in “Resuming after consistency group trip” on page 106.

After a consistency group is tripped, that group must be resumed before it can be tripped again. If a trip is attempted against a tripped group, the CGRP640I message is issued.

You can also trip a consistency group manually with the TRIP command described in “TRIP” on page 197.

The TRIP command uses the Trip API described in “Trip API” on page 102. The TRIP command must pass the same RACF test as the Trip API. In both cases, ConGroup does a RACF check against facility EMC ADMIN FNC CG TRIP.

**Note:** The *Mainframe Enablers Installation and Customization Guide* provides more information about the security system.
Suspending consistency group operations

When a consistency group is in the process of being suspended, I/O to the devices in the consistency group is halted until the suspend process is complete. While in this state, the applications using these devices are hung, waiting for the I/O to complete. This is a critical time, and if the error is severe, the suspend process could take minutes to complete.

When the suspend has completed, data has stopped flowing to the R2s, and I/O has resumed to the R1s. The R2s are in a consistent state.

Note: “Consistency group states” on page 71 illustrates the suspend pending and suspended states.

If a system problem is delaying the suspend process, you may wish to forego having consistent data and allow the applications using the consistency group devices to continue. To do this, cancel the suspend process with the CANCEL command described in “CANCEL” on page 182.

You can determine whether to retry a failed suspend process using the SUSPEND_FAILURE configuration parameter described in “SUSPEND_FAILURE” on page 49.

You can also set the timeout for retrying with the SUSPEND_RETRY_TIMEOUT parameter described in “SUSPEND_RETRY_TIMEOUT” on page 49. After the timeout occurs, the suspend process is terminated, I/O to the devices is resumed, and the suspend operation is deemed unsuccessful. ConGroup issues the CGRP070E message followed by the CGRP073E message.
Splitting remote BCVs

Splitting remote BCVs from their STDs, which are the R2s in your consistency group, ensures that there is a consistent copy of data in the event of a failure when resuming operations. This is called remote split.

**Note:** “Consistency group states” on page 71 illustrates the remote split in process state.

To perform a remote split for the consistency group, use the REMSPLIT command described in “REMSPLIT” on page 193. Alternatively, you can use the SPLIT parameter of the RESUME command described in “RESUME” on page 195 to perform a remote split followed by a resume process.

REMSPLIT starts an asynchronous process to split the BCVs. For a remote split to be honored, the consistency group must be enabled and suspended. The consistency group cannot have a resume or remote split process already in progress.

Issue REMSPLIT just before you issue a RESUME command for the consistency group so that you can save a separate copy of the consistent data.

**Note:** Remember that during the resume process, data consistency on the remote side is delayed until the resume is complete. By creating copies of the data with REMSPLIT, you ensure that there is always a copy of consistent data on which you can rely.

Successful completion of the remote split is indicated with the CGRP203I message. Unsuccessful completion is indicated with either the CGRP204E or CGRP205E message.

R2s without BCVs

By default, ConGroup ignores R2s without attached BCVs during a remote split.

However, you can make ConGroup issue the CGRP314E error message to notify the operator about all R2s that do not have a BCV attached to them. To do so, set the REMSPLIT_OPTION configuration parameter to NOESTERR, as described in “REMSPLIT_OPTION” on page 38.

Remote split retries

Depending on the number and state of the devices, a remote split process can take a long time to complete. When the remote split process encounters temporary errors, it enters a retry loop. In each iteration of the loop, the remote split process takes the following steps:

1. Waits the number of seconds specified in the REMSPLIT_INTERVAL configuration parameter (described in “REMSPLIT_INTERVAL” on page 38).
2. Issues the CGRP206I message.
3. Retries the split.

You can set the retry interval using the REMSPLIT_INTERVAL configuration parameter described in “REMSPLIT_INTERVAL” on page 38.

Remote split is not retried in the following cases:

- ConGroup encounters an R2 without a BCV and the REMSPLIT_OPTION configuration parameter is set to NOESTERR, as described in “REMSPLIT_OPTION” on page 38.
- You cancel the remote split using the CANCEL command described in “CANCEL” on page 182.
Trip API

The Trip API consists of a macro (ECGAPI) and a program call (PC) routine and related code that runs in the ConGroup address space.

Consistency group trips are done using the ECGTRIP program. When invoked, the program ECGTRIP requests the trip of a named consistency group.

The Mainframe Enablers SAMPLIB includes a sample REXX exec named TRIP that can be used to call ECGTRIP. This REXX exec is provided in “Sample REXX exec” on page 104.

Installing ECGTRIP

To install ECGTRIP:
1. Copy the TRIP REXX exec to the appropriate CLIST library.
2. Modify the TRIP REXX exec to point to the load library of your choice.

Security

Before attempting to use ECGTRIP to trip a group, define the resource EMC.ADMIN.FNC.CG.TRIP (class XFACILIT) to RACF and authorize users to update that resource.

If you do not define the resource, access is allowed but ConGroup issues the CGRP642I, CGRP643I, and CGRP641I messages, indicating that access was allowed because the resource was not protected.

If the resource is protected and the user is authorized, the CGRP642I and CGRP643I messages are not displayed. Only CGRP641I is displayed.

Note: The Mainframe Enablers Installation and Customization Guide provides more information about the EMCSAFI Security Interface.

Invoking from TSO

You can invoke TRIP from a TSO session as follows:

TRIP cngrp,nnnn

Where:

- **cngrp** is the name of the consistency group.
- **nnnn** identifies the ResourcePak Base task that the job runs against (same as //SCF$nnnn DD DUMMY).

This command sends a request to ConGroup to trip the named group using the specified SCF instance.
Calling ECGTRIP

ECGTRIP can be called using one of the following methods:

- From a REXX exec, as shown in the sample provided in “Sample REXX exec” on page 104.
- Using JCL as follows:

  ```bash
  EXEC PGM=ECGTRIP, PARM=cngrp, nnnn
  ```

  Where:
  - `cngrp` is the name of the consistency group.
  - `nnnn` identifies the ResourcePak Base task that the job runs against (same as `/SCF$nnnn DD DUMMY`).
- Programmatically from an assembler program.

  Parameters on entry to ECGTRIP:

  - R1: Parm list - See PARMAREA below
  - R13: 72 byte save area provided by caller.
  - R14: Return address.
  - R15: Entry point address (address of ECGTRIP)

  Caller passed parameter area format pointed to by R1:

  ```assembly
  PARMAREA DC 0F parms
  DC A(GRPNAME) Groupname
  DC A(SCFNAME+X'80000000') SSID
  ...
  GRPNAME DC CL8'GRP1' 8 byte blank filled
  SCFNAME DC CL4'MY1' 4 byte blank filled
  ```

  Example via z/OS CALL macro:

  ```assembly
  CALL PGM=ECGTRIP, (GROUPNAME, SSID),VL
  ```

ECGTRIP return codes

ECGTRIP return codes are as folLows:

- R15 - 00  Trip successful (see notes)
- R15 - 08  See below RSN codes (7.2 and higher)
- R15 - 12  No parm specified
- R15 - 16  ConGroup not running

Reason codes associated with return code 8:

- RSN=8 — Group name not found
- RSN=9 — Failed security check
Sample REXX exec

The following is a sample REXX exec. You can find this sample in the Mainframe Enablers SAMPLIB.

**Note:** All members in the Mainframe Enablers SAMPLIB are subject to change and may differ slightly from published text.

```rexx
/* Rexx TRIP - Trip a consistency group.

Syntax:

>>--%TRIP--group[,ssid]<

where:

- **group** specifies the name of the consistency group to be tripped.
- **ssid** the 1-4 character subsystem id (as in //SCF$ssid DD DUMMY) of the target ConGroup/SCF set.

**Usages Notes:**

1. The DSN-prefix must be customized in the CALL statement below.
2. Requires ConGroup 6.3.
3. Supports both RDF-ECA and IOSlevel groups.
4. Scfname is required for ConGroup 7.2 and later. If supplied and running below release 7.2, the older interface will be attempted before failing.

* /

arg group .;
if group = '' then call exit 16, 'Syntax: %TRIP group[,ssid]';
signal on halt; signal on syntax;
"CALL 'DSN-prefix.LINKLIB(ECGTRIP) ' "group""";
select;
  when rc = 0 then
    call exit 0, 'Request for trip of group' group 'sent to ConGroup.';
  when rc = -3 then call exit 12, 'Command not found.';
  when rc < 0  then call exit 12, 'Abend S'd2x(-rc)'.';
  otherwise
    call exit 8, 'TRIP of' group 'failed with return code',
       rc // 65536', reason code' rc % 65536'.';
end;
ERROR: call exit 12, 'Error' rc 'in: "sourceline(sigl)"'.';
HALT: say time() 'Halted in: "sourceline(sigl)".';
  trace '?'R'; nop; call exit 12;
SYNTAX: say time() 'Syntax error' rc 'in: "sourceline(sigl)"'.';
  trace '?'R'; nop; call exit 12;
EXIT: parse arg $er, msg; if msg <> '' then say time() msg;
       exit $er;
```
Resuming consistency group operations

After you have resolved the issues that caused your consistency group to be suspended, you can resume operations for the consistency group.

To resume operations, use the RESUME command described in “RESUME” on page 195.

Resume prerequisites

The prerequisites to actual resumption of operations are as follows:

- When a consistency group is resumed, the R2s are in an inconsistent state until all invalid tracks have been transferred from the R1s to the R2s. The consistency group is not enabled until the R1s and R2s are synchronized.

  Note: The SRDF Host Component for z/OS Product Guide describes R1 and R2 synchronization.

- In Concurrent SRDF configurations, you have to consider which of the R2s you want to use for resumed operations.

  Note: “ConGroup in Concurrent SRDF” on page 90 discusses Concurrent SRDF.

- If STD devices are defined as part of the consistency group, the STDs must have BCVs established and synchronized to them prior to resuming operations. A consistency group cannot be resumed if there is an STD either without a BCV or with a BCV that is not synchronized.

  Note: The TimeFinder/Mirror for z/OS Product Guide instructs how to establish and synchronize STDs and BCVs.
Resuming after consistency group trip

To resume operations after a ConGroup trip, take the following steps:

Note: In the steps, *emccgrp* is the name of the ConGroup started task, and *cngrp* is the name of the consistency group.

1. Delete the swap group associated with your consistency group:

   ```f
   F emccgrp,DAS DELETE GROUP cngrp
   ```

   Note: In the multi-LPAR mode, issue this command from the owner LPAR.

2. Resume consistency group operations:

   ```f
   F emccgrp,RESUME cngrp
   ```

   Note: In the multi-LPAR mode, issue this command first from non-owner LPARs and then from the owner LPAR.

3. Verify that your consistency group is enabled/active:

   ```f
   F emccgrp,DAS DISPLAY CONGROUP cngrp
   ```

4. Verify that your consistency group has a swap group defined, and the devices can be swapped automatically:

   ```f
   F emccgrp,DAS DISPLAY GROUP cngrp DET
   ```

Resume process

After an internal RESUME request is issued to each VMAX system in the consistency group and the CGRP006I message is issued, the resume operation is not completed until the CGRP007I message is issued.

In the meantime, you can see the status of the resume process indicated with the CGRP022I message and a companion message that specifies why the resume process is not complete. The CGRP022I message is periodically issued until the resume process is complete.

If you use the multi-LPAR mode, ConGroup automatically implements the RESUME request on all LPARs using the same CSC listener port. In the single-LPAR mode, you have to manually issue the RESUME command on all relevant LPARs.

Note: “Single/multi-LPAR mode” on page 133 describes single-LPAR and multi-LPAR modes.
Resume interval

You can set the time a RESUME command waits before checking whether all the devices in the consistency group are synchronized.

To do so, use the RESUME_INTERVAL configuration parameter described in “RESUME_INTERVAL” on page 39.

The CGRP022I message is issued, followed by another message describing the state of the resume request. ConGroup continues to retry the request until either of the following occurs:

- The request succeeds.
- The request is canceled by the CANCEL command, as described in “CANCEL” on page 182.

Resume messages

You can control the amount of information displayed during the resume process with the RESUME_OPTION configuration parameter described in “RESUME_OPTION” on page 39.

Canceling a resume process

You can cancel a resume process using the CANCEL command described in “CANCEL” on page 182 with the RESUME parameter.

A CANCEL RESUME cancels the resume processing wherever it is detected, namely:

- Before any resume requests are issued.
- During the resume processing.
- When waiting for synchronization.

CANCEL RESUME does not stop the synchronization between R1s and R2s that have already had their data flow resumed. Any invalid tracks on the R1s continue to synchronize with their R2s.

CANCEL RESUME merely terminates the monitoring process that informs you when all the devices in a consistency group are synchronized and that the remote data is consistent. The CGRP007I message is not issued when the devices are synchronized because the resume process has not been terminated.
CHAPTER 4
Swapping Consistency Groups with CAX

This chapter covers the following topics:

- Overview ............................................................................................................... 110
- Configuring CAX .................................................................................................... 111
- Summary of operations ......................................................................................... 113
- CAX process .......................................................................................................... 114
- Resuming operations after swap ........................................................................... 121
Overview

ConGroup can optionally include the ConGroup AutoSwap Extension (CAX).

**Note:** CAX is one of the operation modes provided by the EMC AutoSwap product. For more information, refer to the *AutoSwap for z/OS Product Guide*.

CAX can move (swap) workloads from volumes in one set of VMAX systems to volumes in other VMAX systems without interruption of operations. Devices subject to a swap are called a *swap group*. CAX performs swaps while application workloads continue in conjunction with ConGroup, using the CSC component of ResourcePak Base for communication.

**Note:** CSC is a component of SCF in ResourcePak Base. The *ResourcePak Base for z/OS Product Guide* describes CSC.

Swaps can be initiated manually as planned events or automatically as unplanned events upon failure detection. The *AutoSwap for z/OS Product Guide* provides more information on planned and unplanned swaps.

Swapping back to the original configuration can be easily achieved using *complement groups*, a type of swap group, as described in “Defining swap (complement) groups” on page 123.

CAX is a licensed feature. To use CAX, you need an active license. The *Mainframe Enablers Installation and Customization Guide* provides information on licensing.

**Note:** CAX was originally called Symmetrix Dynamic Address Switching (S/DAS). Messages you receive from CAX may still refer to S/DAS.

Site prerequisites

To perform a non-disruptive swap from the primary (source) to secondary (target) devices, the following conditions must be met:

- Your site has SRDF/S (Synchronous) configured and operating.

  **Note:** The *SRDF Host Component for z/OS Product Guide* describes SRDF/S.

- Your site has independent channel paths to connected VMAX systems with the secondary (R2) volumes.

- Ensure that the ConGroup started task runs with the SUB=MSTR parameter, as described in “Starting ConGroup with SUB=MSTR” on page 65.
Configuring CAX

To configure CAX:

- Set the following ConGroup configuration parameters:
  
  - CAX described in “CAX” on page 43.
    
    The CAX parameter creates a named set of options that define CAX behavior, thus enabling you to use one definition for multiple consistency groups.

    **Note:** Each consistency group that needs CAX protection must include one CAX statement.

  - CAXOPTS described in “CAXOPTS” on page 31.
    
    The CAXOPTS parameter specifies CAX behavior using CAX options listed in “CAX options” on page 53. The settings determine, for example, which errors cause a swap, whether devices with Snap sessions are eligible for a swap, how to handle Cache Fast Write devices, or what is the lost owner policy.

- Ensure that each CAX-enabled consistency group has a single identified owner LPAR.

  **Note:** “Managing owner LPARs” on page 138 discusses owner LPARs.

- Check the current setting for the following ConGroup configuration parameters that are important for CAX:
  
  - GLOBAL described in “GLOBAL” on page 35
  - COUPLEDS_ALLOWED described in “COUPLEDS_ALLOWED” on page 33
  - PAGEDEV_ALLOWED described in “PAGEDEV_ALLOWED” on page 37

- When using a Concurrent SRDF configuration, narrow your protected mirror list to one mirror using the SYMGROUP configuration parameter, as described in “SYMGROUP” on page 51. CAX supports only one mirror in a Concurrent SRDF environment.
Example

Consider the following CAX settings in the consistency group:

```
CAX=(CAXOPTS=CAXOPT1)
CAXOPTS CAXOPT1=(AUTOCOND=NOPATHS,CFW=OFFVAL,LOSTOWNER
ONSwap=SYSRESET(Off0),ASCM)
```

The CAX statement declares that CAXOPT1 specifies the following CAX options for this consistency group:

- **AUTOCOND=NOPATHS** specifies that if an I/O failure with a “no paths available” condition for any of the devices in the consistency group occurs, that failure event actions begin.
- **CFW=OFFVAL** turns off the Cache Fast Write (CFW) at the time of group validation. No CFW processing is attempted during the swap.
- **LOSTOWNER policy ONSwap=SYSRESET(Off0)** specifies that if all communication (both channels and SRDF) is lost, processing continues, if possible, on the primary side and the secondary side enters a non-restartable wait state Off0.
- **ASCM** specifies that failure response actions do not occur even if the system count at the time of the event does not match the system count at validation time.

If the failure event is the inability to communicate from the R1 to the R2, a normal ConGroup trip occurs and CAX is disabled. This is because the R2 data is no longer up-to-date. A swap should never occur to “stale” data.
## Summary of operations

Table 4 lists CAX operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable and set up CAX</td>
<td>CAX configuration parameter</td>
</tr>
<tr>
<td>Name a set of CAX options for ConGroup</td>
<td>CAXOPTS configuration parameter</td>
</tr>
<tr>
<td>Enable/disable routing CAX messages to SYSLOG</td>
<td>ROUTEMeSsaGetoowner CAX option</td>
</tr>
<tr>
<td>Allow/prohibit swap of devices with Concurrent Copy sessions</td>
<td>AllowConcurrentCopy CAX option</td>
</tr>
<tr>
<td>Bypass verification of devices with Concurrent Copy sessions</td>
<td>NOCHECKConcurrentCopy CAX option</td>
</tr>
<tr>
<td>Allow/prohibit swap of devices with Snap sessions</td>
<td>AllowSnapSession CAX option</td>
</tr>
<tr>
<td>Bypass verification of devices with Snap sessions</td>
<td>NOCHECKSnapSessions CAX option</td>
</tr>
<tr>
<td>Allow/prohibit use of online target (TO) devices</td>
<td>AllowOnlineToDevice CAX option</td>
</tr>
<tr>
<td>Allow/prohibit use of undefined source (FROM) devices</td>
<td>AllowOnlineUndefinedDevice CAX option</td>
</tr>
<tr>
<td>Pass AutoSwap statements to CAX</td>
<td>DAS command</td>
</tr>
<tr>
<td>Allow/prohibit LPAR/group path count mismatch</td>
<td>AllowSystemsCountMismatch CAX option</td>
</tr>
<tr>
<td></td>
<td>BypassSystemsCount CAX option</td>
</tr>
<tr>
<td>Set cross-system timeout</td>
<td>CROSSSYSTEMTIMEOUT CAX option</td>
</tr>
<tr>
<td>Set response to device-specific conditions</td>
<td>AUTOSWAPCONDITIONS CAX option</td>
</tr>
<tr>
<td></td>
<td>UNPLANNEDCONDITIONs CAX option</td>
</tr>
<tr>
<td>Make FBA R2 devices <em>Not Ready</em> on the channel</td>
<td>UNPLANNEDOPTIONS CAX option</td>
</tr>
<tr>
<td>Control use of Cache Fast Write</td>
<td>CFW CAX option</td>
</tr>
<tr>
<td>Set the quiesce timeout</td>
<td>QUIESCETimeout CAX option</td>
</tr>
<tr>
<td>Set response to lost owner communication</td>
<td>LOSTOwnerpolicy ONSWAP CAX option</td>
</tr>
</tbody>
</table>
CAX process

Validation

At startup, ConGroup validates the group of devices subject to swapping (the swap group). Validation includes determining the number of participating LPARs and path groups. The result can be used to determine whether a swap can proceed.

Note: If LPARs and path groups for LPARs cannot be found at validation time, the CGRP195E message is issued.

Monitoring

During operation, ConGroup monitors the CAX-enabled consistency group. If a host I/O (channel) failure or an R1 device failure occurs, or you initiate a planned swap, ConGroup swaps operations from primary devices in the swap group to the secondary devices in the swap group on the secondary VMAX system.

Note: Swap processing and consistency group tripping are mutually exclusive. Once begun, either one precludes the other.

CAX coordinates swap processing with any other LPARs that have access to the ConGroup-protected devices.

After a swap occurs, you can resume operations as described in “Resuming operations after swap” on page 121.
Handling unexpected conditions

You can specify whether to perform a swap if one of the following unexpected conditions occurs:

- Any device in the consistency group “drops Ready.” The device signals this by the Unit Check status. This status indicates that the device is still addressable, but is unable to perform normal functions.

  To swap when a device “drops Ready,” set the UNPLANNEDCONDITION option to INTERVENTIONREQUIRED, as described in “UNPLANNEDCONDITION” on page 61. INTERVENTIONREQUIRED is recommended if you want the protection offered by unplanned swaps.

- Loss of access to a device. This includes loss of all channel paths to a device, a device being BOXed, or an intervention (INTREQ) condition generated by a paging device. INTREQ on a paging device is included in the swap trigger, as such a condition would normally result in a disabled WTOR and likely loss of an LPAR. If any of these conditions occurs to a device in the group, CAX swaps the group.

  This kind of swapping affects operations only when all paths are lost. If many paths fail, performance may be severely degraded, but an unplanned swap is not triggered. You can use the SWAP command of AutoSwap to transfer the workload to the alternate VMAX systems.

  **Note:** The *AutoSwap for z/OS Product Guide* describes the SWAP command of AutoSwap.

  To swap upon loss of access to a device, set the UNPLANNEDCONDITION option to NOPATHS, as described in “UNPLANNEDCONDITION” on page 61. NOPATHS is recommended if you want the protection offered by unplanned swaps.

- SRDF link failure. CAX causes a swap by programmatically setting devices to Not Ready whenever an SRDF link failure occurs. This is identical to the “device drops Ready” event and implies INTREQ internally. However, in this case, the Not Ready condition is set by ConGroup in response to the SRDF link failure.

  If an SRDF link failure occurs, a swap is attempted. If the swap is successful, host application I/O continues on the target volumes. If the swap is not successful, then the host applications halt because the source devices are Not Ready to the host.

  This allows work to continue if a successful swap occurs when configured with channel access to the target volumes, thus providing a solution for sites needing to ensure that no additional work is processed unless the I/O is represented on the target volume.

  ConGroup performs the following steps:

  1. An I/O triggers an ECA Window open condition.
  2. The I/O is stalled.
  3. ConGroup eventually detects the open window as a result of periodic polling.
  4. ConGroup opens the windows on all other devices in the consistency group.
5. When all the windows are open, ConGroup immediately closes all the windows with a special flag for HYPERMAX OS. This flag tells the VMAX system to set the devices *Not Ready* (NR) and to close the windows.

6. Upon seeing the NR condition, CAX initiates a swap.

7. CAX redirects the initial I/O to the channel-attached R2.

To swap upon SRDF link failures, set the UNPLANNEDCONDITIONs option of CAX to SYNCLINKFAILURE, as described in “UNPLANNEDCONDITIONs” on page 61.

**Note:** Use of SYNCLINKFAILURE requires Enginuity patch 36705.

⚠️ **CAUTION**

Use SYNCLINKFAILURE with caution. SYNCLINKFAILURE is not sensitive to Concurrent SRDF or SRDF/Star configurations.
Handling devices with Concurrent Copy/SNAP sessions

You can allow or prohibit swaps of devices that have an open Concurrent Copy or Snap session. To do this, use the AllowConcurrentCopy option described in “AllowConcurrentCopy” on page 53 or the AllowSnapSession option described in “AllowSnapSession” on page 55.

In addition, you can make CAX bypass verification of such devices by using the NOCHECKConcurrentCopy option described in “NOCHECKConcurrentCopy” on page 59 or the NOCHECKSnapSessions option described in “NOCHECKSnapSessions” on page 59.

Bypassing verification helps you reduce the overhead of verifying sessions. In this case, the additional per-device I/O required to determine that sessions are active is not performed. This means that no messages will be displayed where there are active sessions on the devices in the swap group. For this reason, bypass verification only if the outcome of the processing is not important.

Handling unknown online FROM devices

You can specify whether to allow or prohibit use of unknown online source (FROM) devices in a CAX or swap group.

Unknown devices are those excluded or not discovered by ResourcePak Base, for example, due to I/O timeouts or some other access issue.

CAX checks for unknown online FROM devices during swap group validation. If unknown online FROM devices are not allowed, the CGRS587E message is issued and group validation fails.

To allow or prohibit use of unknown online FROM devices, use the AllowOnlineUndefinedDevice option described in “AllowOnlineUndefinedDevice” on page 54.

⚠️ CAUTION

Use AllowOnlineUndefinedDevice with caution, as it could result in loss of access to a device after a swap.
Handling HRO online TO devices

You can specify whether to allow to prohibit use of HRO (Host Read Only) online target (TO) devices in a CAX or swap group.

- When online TO devices are not allowed, the CAX validation process fails the group activation if the group contains an online TO device.
- When online TO devices are allowed, an online TO device can be present in the group only if the device is in the list of HRO-managed devices specified in ResourcePak Base.

**Note:** The list of HRO-managed devices is specified using the SCF.DEV.ATTR.HRO.INCLUDE configuration parameter of ResourcePak Base, as described in the *ResourcePak Base for z/OS Product Guide.*

Where the R2 is online prior to a swap, VMAX ensures the Read-Only (R/O) state of the device. However, after a swap the R2 becomes Read-Write (R/W). Specifying the TO device in the list of HRO-managed devices ensures that the Read-Only state is preserved after the swap. HRO is local to the host where the list is specified and prevents any write processing occurring to the TO device.

- If the online TO device is in the list of HRO-managed devices, the CGRS617I message is issued to confirm that the device is acceptable.
- If the online TO device is not included in the list of HRO-managed devices, swap group validation fails with the CGRS274E message.

To allow or prohibit use of online TO devices, use the AllowOnlineToDevice option described in “AllowOnlineToDevice” on page 54.

Swapping back HRO-managed devices

There are special considerations on swapping back where HRO is specified.

When the target (TO) device is not swapped and remains online, the device is set to the Not Ready state due to the ChangeSourceDevice (CSD) specification and becomes unavailable. Additional processing is required following such a swapback to make the device available again for access.
Checking LPAR/path group counts

The system determines the number of participating LPARs and path groups first during validation and then at swap time.

Note: If LPARs and path groups for LPARs cannot be found at validation time, the CGRS195I message is displayed.

By default, the LPAR count at swap time does not have to match the count at validation time. This means that a swap proceeds regardless of the LPAR count.

However, you can prohibit swaps with an LPAR/group path count mismatch by setting the AllowSystemsCountMismatch option to NO, as described in “AllowSystemsCountMismatch” on page 55. In this case, the LPAR counts at swap time must be equal to or greater than the number of path groups for the device established at validation time. If the LPAR count is less than the number of established path groups for the device, the swap cannot proceed. The devices must be individually validated across all hosts to be certain that all hosts with paths to the devices correctly validate the device. A message is output for each device mismatch with an associated CGRS19I message.

EMC recommendations are as follows:

◆ Allow count mismatch when you are using CAX to protect against failure events or unplanned outages. This is because loss of communication with one or more LPARs may be part of the failure scenario. This should not disable CAX.

◆ Allow count mismatch with SWAPCONTROL=BYRANGE or BYGROUP, when you validate a swap group before a swap. This lets CAX optimize the validation processing.

◆ Prohibit count mismatch for planned swaps. If your swap is planned, and not the result of failure, the number of LPARs involved should not change. If it does, there has been either a failure or an operational error.

Handling lost communications with owner LPAR

Lost communication occurs when SRDF and channel paths to the consistency group devices become disconnected.

In such a situation, ensure that the owner host is really dead and not just isolated before you take any action, such as issuing the ConGroup TAKEOVER command. In some circumstances, the owner could back out of the swap without being able to contact the non-owners or the non-owners being able to contact the owner.

You can set the action for CAX to take if all communication with the owner LPAR or controlling system is lost during the swap process. To do this, use the LOSTOwnerpolicy ONSWAP option described in “LOSTOwnerpolicy ONSWAP” on page 58.

Possible actions include:

◆ Holding all I/O for devices in the swap group until IPL

◆ Undoing the swap operation on the lost LPAR

◆ Resetting the lost LPAR to a non-restartable wait state

◆ Taking over the ownership from the current LPAR
Routing CAX messages

You can have CAX messages consolidated in a single place, the SYSLOG of the owner LPAR. CAX enables you to specify whether you want a particular message type (warning or error) to be routed to the SYSLOG.

To set up message routing, use the ROUTEMESSAGEtoowner option, as described in “ROUTEMESSAGEtoowner” on page 60.

CAX messages have the LPAR name contained in them where the request sequence number would normally be displayed. For example, the following message is routed from the non-owner LPAR X06 to the owner LPAR. The source system is indicated by the >X06.

CGRS274E (>X06)(PID 00001) 'TO' device C4B9 has an invalid state, RS 00000003
Resuming operations after swap

Resuming operations after a swap requires careful planning.

At your site after the swap, the applications are still running (if the swap was caused by an unplanned outage) and the target (R2) devices have become the new primary devices.

After a swap has occurred, you can choose either of the following options:

- **Swapping back immediately**
- **Swapping back later**

Swapping back immediately

Suppose that a planned or unplanned swap occurred in your SRDF environment, resulting in I/O being redirected to R2s. Now you want to return I/O to R1s. To do this, you swap back to R1s immediately after the original swap.

**Note:** This scenario implies use of Dynamic SRDF capability, which is required for SRDF personality swaps performed for devices that contain page datasets. With page datasets, this scenario is not applicable for VMAX systems not supporting Dynamic SRDF.

To swap back to original R1s:

**Note:** In the steps, `emccgrp` is the name of the ConGroup started task, `cngrp` is the name of the consistency group, and `swapgrp` is the name of the swap (complement) group.

1. Define the swap (complement) group, as described in “Defining swap (complement) groups” on page 123.
2. Optional: If your consistency group includes page datasets, perform an SRDF personality swap on the page dataset devices.

   **Note:** The *SRDF Host Component for z/OS Product Guide* describes SRDF personality swaps.

3. Swap I/O in the swap group:

   ```
   F emccgrp,DAS SWAP GROUP swapgrp
   ```

   **Note:** After an unplanned swap that involved FBA devices, you have to set the CAX option UNPLANNEDOPTIONS to FBAUSRNRDY, as described in “UNPLANNEDOPTIONS” on page 62, to make the R2 devices *Not Ready* on the channel.

After the swapback, SRDF resynchronization is required so that I/O may begin on the restored primary R1 devices. When using the DAS SWAP GROUP command to perform the swapback, SRDF resynchronization is performed automatically.
4. Verify that all of the devices have swapped back. In the SWAP command output, the count of total devices must match the count of swapped devices, as shown in Figure 16:

```
CGRS292I (00009) Group NTC1NTH 646
Total Devices : 4 Highest PID : 4
   Valid : 0 Invalid : 0
Auto Swappable: 0 Auto Pending : 0
   Swapped : 4 Failed Swap : 0
   Offline : 0 Not Defined : 0
```

**Figure 16** SWAP command output

5. Optional: If using page datasets, perform another SRDF personality swap on the page dataset devices.

**Note:** The *SRDF Host Component for z/OS Product Guide* describes SRDF personality swaps.

6. Delete the swap group from the owner LPAR:

   ```
   f emccgrp,DAS DELETE GROUP cngrp
   ```

7. Enable the consistency group:

   ```
   f emccgrp,ENABLE cngrp FORCE
   ```

**Note:** In the multi-LPAR mode, issue the command first from non-owner LPARs and then from the owner LPAR.

Ensure that this step completes successfully. Otherwise, the consistency group may appear to be in an acceptable state but the swap group has not been defined.

8. Verify that the consistency group is enabled/active:

   ```
   f emccgrp,DISPLAY CONGROUP cngrp
   ```

9. Verify that your consistency group has a swap group defined, and the devices can be swapped automatically:

   ```
   f emccgrp,DAS DISPLAY GROUP cngrp DET
   ```
Defining swap (complement) groups

A swap group contains devices subject to a swap.

Note: The AutoSwap for z/OS Product Guide provides detailed information about swap groups.

When using the swap functionality through ConGroup, you do not define the swap group from stretch. Instead, you use device specifications of your consistency group. This type of group is called a complement group.

A complement group is a named set of devices to which normal operations for the devices in an associated consistency group are restored after a swap to the R2s.

You can define the complement group at any time after you have defined the original consistency group. When defined prior to a swap, the complement group is created, but is not populated with devices.

For a complement group, devices subject to swapback are determined automatically during the swap. This means that the complement group contains only the devices that were swapped automatically.

To create a complement group to the original CAX consistency group, choose one of the following methods:

- Using DAS command of ConGroup
- Using CONFIGCA DD statement of ConGroup
- Using DEFINE COMPLEMENT command of ResourcePak Base

Using DAS command of ConGroup

IMPORTANT

Use this method only if the ConGroup started task that executed the swap is still running.

This method allows you to create a complement group by passing the DEFINE GROUP command to AutoSwap.

Note: The AutoSwap for z/OS Product Guide provides detailed information about the DEFINE GROUP command of AutoSwap.

To create a complement group, issue the following command:

F emccgrp,DAS DEFINE GROUP cmpgrp COMPLEMENT cngrp [options]

Where:

- emccgrp is the name of the ConGroup started task.
- cmpgrp is the name of the complement group (1-8 characters).
- cngrp is the name for the original consistency group (1-8 characters).
- options are the optional parameters defined for the complement group, as described in the AutoSwap for z/OS Product Guide.
The options specified in this statement apply to the complement group, not the original consistency group.

The options for the original consistency group and its complement group do not have to be the same. For example, you can have by-group swap control in the complement group while having by-device in the consistency group.

**Note:** In the multi-LPAR mode, issue the command from the owner LPAR.

### Using CONFIGCA DD statement of ConGroup

**IMPORTANT**

EMC recommends using CONFIGGA DD statement to create a complement group as an alternative when the ConGroup task is not running.

To create a complement group, externalize the complement group definition statements in a dataset and then specify this dataset in the CONFIGCA DD statement of the ConGroup started task JCL, as described in “Customizing ConGroup started task” on page 63.

### Reprocessing CONFIGCA

To reprocess statements from a CONFIGCA DD file, issue the following command:

```
F emccgrp, DAS T PARMS
```

Where:

- `emccgrp` is the name of the ConGroup started task.

### Using DEFINE COMPLEMENT command of ResourcePak Base

You can also use DEFINE COMPLEMENT command of the EMCGROUP utility (GNS) in ResourcePak Base to create a complement group based on a GNS group. The ResourcePak Base for z/OS Product Guide describes the DEFINE COMPLEMENT command.
Swapping back later

Suppose that a planned or unplanned swap occurred in your SRDF environment, resulting in I/O being redirected to R2s. You want to keep the new configuration for a while with ConGroup protection re-enabled. To do this, you reconfigure SRDF, ConGroup, and CAX, as described in “Re-enabling ConGroup after swap” on page 125.

After a while, you want to return I/O to the R1s. To do this, you swap back to R1s, as described in “Swapping back to original R1s” on page 126.

Note: This scenario implies use of Dynamic SRDF capability, which is required for SRDF personality swaps. This scenario is not applicable for VMAX systems not supporting Dynamic SRDF.

Re-enabling ConGroup after swap

To re-enable ConGroup protection after a swap:

1. Delete the swap group from the owner LPAR:

   `F emccgrp,DAS DELETE GROUP cngrp`

   Note: “DAS” on page 183 describes the DAS command of ConGroup. The AutoSwap for z/OS Product Guide describes the DELETE GROUP command of AutoSwap.

2. Modify the ConGroup configuration file to replace original R1 devices with the devices to which I/O currently goes (former R2s).

   Note: “Creating configuration file” on page 28 describes the ConGroup configuration file.

3. Perform the SRDF personality swap:

   `#SC VOL,LCL(gatekeeper,srdfgrp),SWAP,sysdv#_list`

   Note: The SRDF Host Component for z/OS Product Guide describes SRDF personality swaps.

4. Resume SRDF:

   `#SC VOL,device_list,RDF-RSUM`

   Note: The SRDF Host Component for z/OS Product Guide describes the SC VOL RDF-RSUM command.

5. Wait for resynchronization. To verify that resynchronization is completed:

   `#SQ VOL,device_list,INV_TRKS`
6. Refresh ConGroup configuration information:

   `F emccgrp,REFRESH FORCE`

   **Note:** “REFRESH” on page 193 describes the REFRESH command of ConGroup.

   In the multi-LPAR mode, issue the command first on the non-owner LPARs and then on the owner LPAR.

7. Wait until ConGroup reports that refresh is complete and ConGroup is active. The CGRP149I, CGRP143I, CGRP200l messages confirm this.

8. Verify that the consistency group is enabled/active:

   `F emccgrp,DISPLAY CONGROUP cngrp`

   **Note:** “DISPLAY CONGROUP” on page 187 describes the DISPLAY CONGROUP command of ConGroup.

9. Verify that your consistency group has a swap group defined, and the devices can be swapped automatically:

   `F emccgrp,DAS DISPLAY GROUP cngrp DET`

   **Note:** “DAS” on page 183 describes the DAS command of ConGroup. The *AutoSwap for z/OS Product Guide* describes the DISPLAY GROUP command of AutoSwap.

---

**Swapping back to original R1s**

To swap back to original R1s:

1. Swap back:

   `F emccgrp DAS SWAP GROUP cngrp`

2. Delete the swap group from the owner LPAR:

   `F emccgrp,DAS DELETE GROUP cngrp`

   **Note:** “DAS” on page 183 describes the DAS command of ConGroup. The *AutoSwap for z/OS Product Guide* describes the DELETE GROUP command of AutoSwap.

3. Modify the ConGroup configuration file back to use original R1 devices (use same devices as prior to the automatic swap).

   **Note:** “Creating configuration file” on page 28 describes the ConGroup configuration file.
4. Perform the SRDF personality swap.

```bash
#SC VOL,LCL(gatekeeper,srdfgrp),SWAP,symdv#_list
```

**Note:** The *SRDF Host Component for z/OS Product Guide* describes SRDF personality swaps.

5. Resume SRDF:

```bash
#SC VOL,device_list,RDF-RSUM
```

**Note:** The *SRDF Host Component for z/OS Product Guide* describes the SC VOL RDF-RSUM command.

6. Wait for resynchronization. To verify that resynchronization is completed:

```bash
#SQ VOL,device_list,INV_TRKS
```

**Note:** The *SRDF Host Component for z/OS Product Guide* describes the SC VOL INV_TRKS command.

7. Refresh ConGroup configuration information:

```bash
F emccgrp,REFRESH FORCE
```

**Note:** “REFRESH” on page 193 describes the REFRESH command of ConGroup.

   In the multi-LPAR mode, issue the command first on the non-owner LPARs and then on the owner LPAR.

8. Wait until ConGroup reports that refresh is complete and ConGroup is active. The CGRP149I, CGRP143I, CGRP200I messages confirm this.

9. Verify that the consistency group is enabled/active:

```bash
F emccgrp,DISPLAY CONGROUP cngrp
```

**Note:** “DISPLAY CONGROUP” on page 187 describes the DISPLAY CONGROUP command of ConGroup.

10. Verify that your consistency group has a swap group defined, and the devices can be swapped automatically:

```bash
F emccgrp,DAS DISPLAY GROUP cngrp DET
```

**Note:** “DAS” on page 183 describes the DAS command of ConGroup. The *AutoSwap for z/OS Product Guide* describes the DISPLAY GROUP command of AutoSwap.
Swapping Consistency Groups with CAX
CHAPTER 5
Managing ConGroup Environment

This chapter covers the following topics:

◆ Summary of operations ................................................................. 130
◆ Viewing ConGroup status ............................................................. 132
◆ Coordinating ConGroup instances .............................................. 133
◆ Managing owner LPARs ............................................................... 138
◆ Adding/removing VMAX systems ............................................... 140
◆ Managing gatekeepers ................................................................. 141
◆ Refreshing ConGroup configuration .......................................... 144
◆ Updating consistency group state .............................................. 147
◆ Cleaning up environment with ECGUTIL ................................. 148
◆ Diagnostics and troubleshooting ................................................ 155
Summary of operations

Coordinating ConGroup instances

Table 5 lists operations for coordinating consistency groups.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set single-LPAR or multi-LPAR mode</td>
<td>MODE configuration parameter</td>
</tr>
<tr>
<td>View participating LPARs</td>
<td>LA command</td>
</tr>
<tr>
<td>Set interval between coordination activities</td>
<td>CLOCKN configuration parameter</td>
</tr>
<tr>
<td>Set clock tick interval multiplier</td>
<td>CLOCKE configuration parameter</td>
</tr>
</tbody>
</table>

Managing owner LPARs

Table 6 lists operations for coordinating consistency groups.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set owner LPAR</td>
<td>GLOBAL configuration parameter</td>
</tr>
<tr>
<td>Change owner LPAR</td>
<td>MOVEOWNER command</td>
</tr>
<tr>
<td></td>
<td>TAKEOVER command</td>
</tr>
</tbody>
</table>

Adding/removing VMAX systems

Table 7 lists operations for adding/removing VMAX systems.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add VMAX system to ConGroup</td>
<td>#ADD CONTROLLER command</td>
</tr>
<tr>
<td>Remove VMAX system from ConGroup</td>
<td>#DELETE CONTROLLER command</td>
</tr>
</tbody>
</table>

Managing gatekeepers

Table 8 lists operations for managing gatekeepers.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>View gatekeepers</td>
<td>#DISPLAY GATEKEEPERS command</td>
</tr>
<tr>
<td>Add gatekeepers</td>
<td>#PIN command</td>
</tr>
<tr>
<td>Remove gatekeepers</td>
<td>#UNPIN command</td>
</tr>
</tbody>
</table>
Viewing/updating configuration

Table 9 lists operations for viewing/updating configuration.

Table 9  Viewing/updating configuration

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>View consistency group/ConGroup environment</td>
<td>• DISPLAY CONGROUP command</td>
</tr>
<tr>
<td></td>
<td>• QUERY CON command</td>
</tr>
<tr>
<td>Update consistency group configuration</td>
<td>REFRESH command</td>
</tr>
<tr>
<td>Update consistency group configuration</td>
<td>AUTO_REFRESH configuration parameter</td>
</tr>
<tr>
<td>automatically</td>
<td></td>
</tr>
<tr>
<td>Update consistency group state</td>
<td>RESET command</td>
</tr>
</tbody>
</table>

Miscellaneous

Table 10 lists miscellaneous ConGroup operations.

Table 10  Miscellaneous operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>View list of ConGroup commands</td>
<td>HELP command</td>
</tr>
<tr>
<td>Set up SAF for ConGroup commands</td>
<td>• SAF_CLASS configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• SAF_PROFILE configuration parameter</td>
</tr>
<tr>
<td>Set up messaging</td>
<td>• MSGLEVEL configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• DISPLAY_SAFAUTH_SUCCESS configuration parameter</td>
</tr>
<tr>
<td>Enable/disable zBoost PAV Optimizer support</td>
<td>PAVO configuration parameter</td>
</tr>
<tr>
<td>Terminate ConGroup process</td>
<td>CANCEL command</td>
</tr>
</tbody>
</table>
Viewing ConGroup status

Typically, the following command series provides the most comprehensive information about ConGroup status:

- **LA**
  Shows which LPARs are also running ConGroup, and which LPAR is the current owner.
  
  **Note:** “LA” on page 190 instructs how to use the command; “SMFID report” on page 165 provides details about the command output.

- **DIS ENV**
  Displays key setup values.
  
  **Note:** “DISPLAY ENVIRONMENT” on page 188 instructs how to use the command; “ConGroup Environment report” on page 158 provides details about the command output.

- **DIS CON NOLIST**
  Displays key values for each consistency group.
  
  **Note:** “DISPLAY CONGROUP” on page 187 instructs how to use the command; “Consistency Group — Summary report” on page 160 provides details about the command output.

- **DIS CON LIST**
  Shows details about devices participating in the consistency group.
  
  **Note:** “DISPLAY CONGROUP” on page 187 instructs how to use the command; “Consistency Group — Details report” on page 163 provides details about the command output.

- **#DISPLAY GATEKEEPERS**
  Provides information which gatekeeper devices can be used by ConGroup, and which devices actually are in use.
  
  **Note:** “#DISPLAY GATEKEEPERS” on page 188 instructs how to use the command; “Gatekeeper report” on page 165 provides details about the command output.
Coordinating ConGroup instances

Single/multi-LPAR mode

ConGroup provides two operation modes: single-LPAR and multi-LPAR.

ConGroup modes enable ConGroup to run in one LPAR and be visible to all other LPARs. This means that there is only one ConGroup instance that consumes only one z/OS address space. This simplifies operation in multi-LPAR systems.

If CAX is enabled for any consistency group, ConGroup must run in all LPARs.

The single-LPAR mode is not supported with CAX. When you attempt to switch to the single-LPAR mode when using CAX, a warning is written to the log. ConGroup will operate in the single-LPAR mode, but multiple systems will be unable to correctly communicate. Therefore, EMC recommends that you always use the multi-LPAR mode.

Note: Whenever any R1-R2 pair defined in a configuration file is protected by more than one ConGroup instance, ensure that all such ConGroup instances use the same configuration file.

Single-LPAR mode

In the single-LPAR mode, there is no information sharing among ConGroup instances that are monitoring the same consistency groups. ConGroup instances are not communicating with one another. The ConGroup instances that come up do not attempt to talk to ConGroup instances already running.

The single-LPAR mode is enabled by default. To switch to the multi-LPAR mode, set the MODE configuration parameter to MULTI, as described in “MODE” on page 36.
Multi-LPAR mode

The multi-LPAR mode is an optional coordination mechanism based on distributed locking, message exchange, and the Cross System Communication (CSC) facility.

**Note:** CSC is a component of ResourcePak Base. The *ResourcePak Base for z/OS Product Guide* describes CSC.

The multi-LPAR mode provides shared information across all ConGroup instances monitoring the same consistency group(s) about the state of those consistency groups.

The multi-LPAR mode allows two or more ConGroup instances to communicate with each other. All ConGroup instances that come up can talk to ConGroup instances that are already running.

**Note:** To be able to communicate, the multiple ConGroup instances must be running at the same release level (for example, 8.1).

In the multi-LPAR mode, the commands you issue are globally communicated. This means that you do not need to issue a command on all LPARs. You can issue it on any involved LPAR instead. The exception to this is the DISABLE command, as described in “Disabling consistency groups” on page 95.

To use the multi-LPAR mode across systems and/or sysplexes, the system time must come from the same source. This could be the sysplex timer or a single CEC system clock.

ConGroup uses an internal lock, ALL-CONGROUPS, to serialize many global operations to ensure the integrity of ConGroup functionality.

To enable the multi-LPAR mode, set the MODE configuration parameter to MULTI, as described in “MODE” on page 36.

**CAUTION**

When in the multi-LPAR mode, do not start new ConGroup instances during the execution of the MOVEOWNER or TAKEOVER command statements. Starting a new ConGroup instance during MOVEOWNER or TAKEOVER is not currently supported and produces unpredictable results.
Managing multiple ConGroup instances

ConGroup uses the CSC facility of ResourcePak Base to identify other active instances of ConGroup.

Note: CSC is a component of ResourcePak Base. The ResourcePak Base for z/OS Product Guide describes CSC.

ConGroup instances using the same CSC port exchange information that is used to coordinate actions and sequence command execution.

As each ConGroup instance starts, it begins exchanging messages with existing members of the group. If CAX is enabled for a consistency group, the exchanged information includes the SMFID of the owner LPAR.

Note: “Swapping Consistency Groups with CAX” on page 109 describes CAX. “Managing owner LPARs” on page 130 provides information on owner LPARs.

CG sets

To ensure ConGroup and CAX cross-address space and/or LPAR isolation, you can create one or more CG sets.

Note: CG sets are for testing purposes only.

A CG set contains ConGroup instances that are assigned the same global storage name token anchor. In a multi-LPAR configuration, all ConGroup instances that wish to communicate must have the same storage name token anchor assigned to them. Adding of devices is coordinated among the participating ConGroup address spaces through CSC.

A given ConGroup instance can belong to one and only one CG set.

When running multiple ConGroup instances belonging to the same CG set on different LPARs, all the ConGroup instances must use the same configuration file by having their CONFIG DD statements point to the same dataset.

This prevents the intolerable situation of a new node joining the “set” with a different set of parameters. ConGroup checks for this and, if the configuration files are not identical in content, forces the new node to shutdown immediately.
Assigning storage name token anchor

To assign the global storage name token anchor, use the MODE configuration parameter with the CGSETnn option, as described in “MODE” on page 36.

Using CGSETnn allows ConGroup to modify a configuration without the disable/enable processing involved in the ConGroup REFRESH command (described in “REFRESH” on page 193), which has the negative effect of temporarily stopping protection of all consistency groups (star and non-star) in a ConGroup address space.

For multiple ConGroup instances running in the same LPAR, the storage name token anchor must be unique for each ConGroup instance running in the LPAR. For example, with the storage name token anchor defaulted to “00”, any additional ConGroup instance must have a value between “01” and 06”.

IMPORTANT

Even though you can run up to seven independent ConGroup instances in one LPAR, only one ConGroup instance can be associated with a given Symmetrix Control Facility (SCF).

Note: SCF is the key component of ResourcePak Base. The ResourcePak Base for z/OS Product Guide describes SCF.

For running ConGroup instances in different or separate LPARs, the storage name token anchor assigned in one LPAR has ConGroup “pair-up” with a ConGroup instance in a different LPAR that has the same storage name token anchor assigned to it. This provides redundancy support for a production ConGroup instance.

For example, if a ConGroup instance running in LPAR 1 uses storage name token anchor “01”, and another ConGroup instance running in LPAR 2 uses the same value of “01”, the two ConGroup instances in the two LPARs both monitor and protect their defined consistency groups. Of course, the two ConGroup instances must use the same configuration file. So, if LPAR 1 malfunctions, the ConGroup processing in LPAR 2 still provides protection for both defined ConGroups.

Possible cross-LPAR anchor conflicts are automatically resolved/prevented by independent CRC checks on the ConGroup configuration file at each node with subsequent comparison based on the CRC as embedded in inter-node message traffic.
Coordination timing

You can make the following ConGroup coordination timing settings:

- Set the multiple of the clock tick interval for heartbeat coordination using the CLOCKE configuration parameter described in “CLOCKE” on page 32.
- Set the time between coordination activities using the CLOCKN configuration parameter described in “CLOCKN” on page 32.

The number of seconds represented by CLOCKN must be equal to or greater than twice (after dividing by 100) the idle poll value of CSC.

Note: The CSC idle poll value is optionally specified in ResourcePak Base with the SCF.CSC.IDLEPOLL initialization parameter. The ResourcePak Base for z/OS Product Guide describes the SCF.CSC.IDLEPOLL parameter.

If the number of seconds represented by CLOCKN is less than double the CSC idle poll value, reliable coordination is not possible. In this case, an error message is issued, and initialization fails.

For normal operation, use the default value for both CLOCKN and the CSC idle poll value.

Because ConGroup functions often take multiple clock intervals, large values for CLOCKN can unnecessarily lengthen the amount of time ConGroup functions can take.
Managing ConGroup Environment

Managing owner LPARs

The owner LPAR is the LPAR that controls the consistency group during CAX operations. The owner LPAR is responsible for CAX activity coordination. If a failure occurs that results in all communication being lost between the primary and secondary sites, the owner LPAR proceeds.

You specify the owner LPAR using the GLOBAL configuration parameter described in “GLOBAL” on page 35.

If necessary, you can change the owner LPAR using the MOVEOWNER or TAKEOVER commands, as described in “Changing owner LPAR” on page 138.

Owner LPAR requirements

In the multi-LPAR mode, specification of the owner LPAR is obligatory.

**Note:** “Multi-LPAR mode” on page 134 describes the multi-LPAR mode.

This is because SRDF ECA management functions are only carried out on the designated owner LPAR. If the owner LPAR is not specified, ConGroup initialization is terminated and the CGRP085E message is issued. In this case, specify the owner LPAR and restart ConGroup.

All ConGroup instances that use the consistency group must have the same owner LPAR.

The group of consistency groups has only one owner LPAR and all ConGroup instances are aware of the current owner LPAR as a result of the message exchanges. The remainder of the ConGroup LPARs in the group are referred to as non-owners.

Changing owner LPAR

You may need to change ownership due to maintenance activities on the current owner LPAR or when the current owner LPAR goes down.

You can transfer ownership from one LPAR to another without the need to reinitialize all participating ConGroup instances manually.

To do this, use either the MOVEOWNER command described in “MOVEOWNER” on page 191 or the TAKEOVER command described in “TAKEOVER” on page 197.

The MOVEOWNER and TAKEOVER commands use synchronized message passing to coordinate the ConGroup activities on all LPARs in a manner that result in the passing of the ownership attribute from one LPAR to another.

The new ownership you assign lasts as long as any single participating ConGroup instance is active. If all ConGroup instances are shutdown and then restarted, the LPAR specified in the GLOBAL configuration parameter becomes the owner LPAR.

**Note:** When you recycle a new LPAR, it inherits the same status of the owner as at startup. You can either ensure that the current owner is in a completely enabled state (both consistency group and swap groups) or move ownership back and do a global ENABLE.

The MOVEOWNER and TAKEOVER commands are *not* available in the single-LPAR mode or for non-CAX consistency groups.
Using the MOVEOWNER command

The MOVEOWNER command is your first choice. The MOVEOWNER command transfers ownership from the current owner LPAR to another LPAR.

Note: Ownership is moved for every consistency group that ConGroup protects.

Use MOVEOWNER if both the current owner LPAR and the prospective new owner LPAR are running. MOVEOWNER lets you verify the identity of the existing owner before you use TAKEOVER.

MOVEOWNER is intended to be used during normal operations; for example, if you need to transfer ownership because the owner LPAR is subject to IPL.

You can issue MOVEOWNER from any LPAR running ConGroup. After MOVEOWNER is issued, but before ConGroup acts upon it, ConGroup validates the existence of the specified LPAR(s).

After successfully changing the owner LPAR with the MOVEOWNER command, the CGRP6501 message is issued.

Note: You cannot use the MOVEOVER command with shared R1 devices.

Using TAKEOVER command

Use the TAKEOVER command if the current owner LPAR is not running. The TAKEOVER command transfers ownership of a consistency group from the current owner LPAR to the LPAR from which the TAKEOVER command is issued.

You can issue the TAKEOVER command from any LPAR running ConGroup.

Note: If ConGroup is started with no owner LPAR, issue the TAKEOVER command to assign an owner LPAR and then issue the ENABLE command for the new owner to actually take ownership.
Managing ConGroup Environment

Adding/removing VMAX systems

When ConGroup starts, it queries its connected SCF instance to determine which VMAX systems are available.

Note: SCF is the key component of ResourcePak Base, described in the ResourcePak Base for z/OS Product Guide. To connect an SCF instance to ConGroup, you use the SCF$nnnn DD statement, as described in “Customizing ConGroup started task” on page 63.

The returned list of VMAX systems includes the identity of gatekeeper devices defined or defaulted in the ResourcePak Base configuration file. ConGroup creates a subtask for each VMAX system and a single subtask to manage the list of gatekeepers for all of the VMAX systems. Other processes in ConGroup then request gatekeepers from the gatekeeper subtask whenever they wish to access a particular VMAX system.

Note: “Pinning/unpinning gatekeepers” on page 141 describes pinning/unpinning gatekeepers.

When new VMAX systems are introduced to SCF (by modifying the ResourcePak Base configuration file or initiating an IODF1 configuration change), SCF notifies ConGroup. ConGroup then automatically incorporates any new VMAX systems by building new subtasks and by updating its gatekeeper list. Since new VMAX systems do not (initially) have any ConGroup protected devices, they will not have any pinned gatekeepers. However, as soon as any device on one of the new VMAX systems is added to a group, all the gatekeepers on that VMAX system are pinned. Deletions from the ResourcePak Base configuration file do not cause deletion of VMAX system subtasks and gatekeeper structures from ConGroup.

Adding VMAX system manually

Normally, VMAX systems are added to every ConGroup address space automatically, in response to ResourcePak Base configuration settings.

However, you can add a VMAX system to ConGroup manually, for example, if you had previously removed it manually from ConGroup on a given LPAR.

To add a VMAX system manually, use the #ADD CONTROLLER command described in “#ADD CONTROLLER” on page 181.

Before adding a VMAX system to ConGroup, ensure that the VMAX system is known to SCF, as described in the ResourcePak Base for z/OS Product Guide.

Removing VMAX system manually

To remove a Symmetrix system, use the #DELETE CONTROLLER command described in “#DELETE CONTROLLER” on page 185.

Typically, VMAX systems do not need to be removed manually.

---

1. Input/Output Definition File
Managing gatekeepers

Pinning/unpinning gatekeepers

ConGroup automatically pins and unpins gatekeepers based on the list of VMAX systems containing the R1s that are part of the consistency group or groups.

To protect actively managed groups from accidental disruption due to IODF ACTIVATEs of configurations to remove gatekeepers from z/OS, ConGroup pins its gatekeepers using the IBM UCBPIN service.

To view pinned gatekeepers, use the #DISPLAY GATEKEEPERS command described in “#DISPLAY GATEKEEPERS” on page 188.

All gatekeepers are kept pinned by ConGroup as long any devices on those VMAX systems are defined in any consistency group. If multiple gatekeepers are defined, ConGroup can use more than one concurrently.

To pin a gatekeeper manually, use the #PIN command described in “#PIN” on page 192.

To manually unpin a gatekeeper, use the #UNPIN command described in “#UNPIN” on page 198.

Unpinning can also occur when the consistency group (or groups) that uses the VMAX system is removed from the ConGroup configuration due to one of the following:

- All relevant devices are manually removed from the consistency group.
- The ConGroup configuration is refreshed with a non-existent consistency group.
- ConGroup shuts down.

**Note:** You cannot unpin all the gatekeepers to a VMAX system.

Gatekeeper server

As an internal performance aid, ConGroup uses the ResourcePak Base (SCF) Gatekeeper API to assist in managing the internal polling processes.

When started, ConGroup requests the list of available SCF gatekeepers. That list is then managed for other internal ConGroup processes by a dedicated gatekeeper server process.

After a worker process receives a poll request, it sends a request for a gatekeeper device.

ConGroup receives the request and finds an unused gatekeeper on the list and responds to the worker process with information about that gatekeeper.
Managing the gatekeeper pool

Normally, you can let the gatekeeper pool function automatically. However, there may be occasions when the gatekeeper pool is backed up with queuing and you have to change the gatekeeper pool.

Determining if there is gatekeeper queuing

To determine if there is gatekeeper queuing, run the ConGroup Environment report and check the HWM value. If the value is zero (0), no gatekeeper request ever has to wait for a free gatekeeper. However, the value is almost always non-zero.

Note: “ConGroup Environment report” on page 158 describes the ConGroup Environment report.

Changing the gatekeeper pool

You can change the gatekeeper pool while ConGroup is running. To do this, perform the following steps:

1. Change the ResourcePak Base configuration file to specify the desired gatekeepers.


2. Issue the SCF,INI,REFRESH command followed by the SCF,DEV,REFRESH GATEKEEPERS command.

   Result: The SCF0417I message is displayed.

   Note: The ResourcePak Base for z/S Product Guide describes the SCF,INI,REFRESH command.

3. Issue the following command to make ConGroup use the new gatekeepers:

   F emccgrp,CE GKREFR

   Where emccgrp is the name of the ConGroup started task.

   Result: ConGroup begins using the new gatekeepers.

   Note: CE is a special command that stands for Command Entry. CE routes the command tokens that follow it to a general-service command processor in the ConGroup address space. This command processor is distinct from the command processor that handles most traditional ConGroup commands (such as ENABLE or REFRESH). The use of CE is not tied to GKREF or any other specific command.
Gatekeeper pool recommendations

EMC recommends that you specify at least two gatekeepers in the ResourcePak Base configuration file for each VMAX system that ConGroup is protecting. The reason for this is simply to allow ConGroup to split the polling workload into parallel sub-workloads. The most parallelism ConGroup can achieve is 10 sub-workloads because ConGroup runs with 10 worker subtasks.

**Note:** Each additional gatekeeper provides progressively smaller marginal advantage. More than five achieves little advantage.

If you specify only one gatekeeper per VMAX system, potentially dozens of poll operations are single-threaded (with a corresponding high HWM value in the ConGroup Environment report described in “ConGroup Environment report” on page 158) even though there are 10 workers. Note that currently, each group definition generates independent polling, so four groups generate at least four times the amount of polling as one group. But the gatekeepers are shared (and are correspondingly busier) when you define multiple groups.
Refreshing ConGroup configuration

Refreshing automatically

ConGroup provides an automatic refresh mechanism that detects when a device is brought online. ConGroup checks to see if the device belongs to a consistency group, and automatically generates a REFRESH command statement if needed.

Note: “REFRESH” on page 193 describes the REFRESH command.

Automatic refresh is used when you specify volser masks for either the DEVICE_LIST parameter described in “DEVICE_LIST” on page 45 or the EXCLUDE parameter described in “EXCLUDE” on page 47, or when you specify SMS groups using the SMS_GROUP parameter described in “SMS_GROUP” on page 48. If you specify devices by their CUUs, the devices become part of the consistency group regardless of whether they are online or offline. A refresh is initiated when devices that you included by CUU are brought online or offline.

Offline devices excluded only by volser are not excluded as the volser does not exist during the time the device is offline. Clipping a device that you defined by CUU as part of the consistency group does not initiate a refresh as the device is already part of the group.

The REFRESH command statement is issued only if the state of the consistency group or groups allows it to be issued. If the REFRESH command cannot be issued immediately, ConGroup monitors the consistency groups and issues the REFRESH at an appropriate time.

To enable automatic refresh, use the AUTO_REFRESH configuration parameter described in “AUTO_REFRESH” on page 31.

Automatic refresh is not available for CAX groups. If you attempt to enable automatic refresh for a CAX group, the CGRP610W message is issued.

Note: “Swapping Consistency Groups with CAX” on page 109 describes CAX.
How automatic refresh works

When you define a consistency group, you can include devices in the consistency group by specifying an SMS group or a volser mask. If a device is brought online, and the device belongs to a specified SMS group or fits a specified volser mask, ConGroup generates a REFRESH command statement so that the new device is added to the consistency group.

For example, consider the following definition of consistency group devices:

DEVICE_LIST=EMC*

*Note: “DEVICE_LIST” on page 45 describes the DEVICE_LIST configuration parameter.*

When ConGroup starts, all online devices whose volsers begin with the characters “EMC” are added to the consistency group.

Suppose after startup, the DISPLAY CONGROUP command gave these results:

CONGROUP=DB2PROD ENABLED ACTIVE
USE_NR_ON_TO=FBA
SUSPEND_FAILURE=WTOR SUSPEND_RETRY_TIMEOUT=120
CTLR SER#=000190300647  uCode: 58760239 GroupId: 0003

*Note: “DISPLAY CONGROUP” on page 187 describes the DISPLAY CONGROUP command.*

If volume EMC005 is brought online, ConGroup recognizes this event, and determines that volume EMC005 fits the volser mask “EMC*.” ConGroup generates a REFRESH command statement so that volume EMC005 is added to consistency group DB2PROD.
Refreshing manually

You can refresh consistency group configuration manually, without terminating ConGroup, by issuing the RESRESH command described in “REFRESH” on page 193.

For example, you may need a manual refresh when you have defined a consistency group through GNS or updated the consistency group definition.

If any consistency group is suspended, disabled, or has a suspend, a resume, or a remote split process running, an error message is produced.

The state of the consistency group configuration after the REFRESH completes depends on the number and type of errors (if any) encountered during the refresh. The REFRESH is processed in two phases:

1. A syntax and verification phase
2. An enabling phase

Any errors during the syntax and verification phase terminate the refresh, with the old configuration still active. The CGRP201E message is issued signifying this type of error.

After the new configuration has passed verification, the REFRESH statement disables all the old consistency groups and attempts to enable the new consistency groups. The new consistency groups may not be enabled if one or more devices are in an improper state for enablement.

For example, if a device has invalid tracks, the consistency group to which it belongs is not enabled. Although the CGRP2001 message is issued indicating that the new configuration is active, you should take special care to verify that the new consistency groups have been enabled successfully.

To enable the new consistency configuration, even if all the consistency groups are not in an enable-ready state, use the FORCE parameter of the REFRESH command.

Note: If you are operating in multi-LPAR mode, ConGroup automatically issues the REFRESH statement on all LPARs using the same CSC listener port. In the single-LPAR mode, you have to manually issue the REFRESH command on all relevant LPARs.
Updating consistency group state

You can update ConGroup internal tables with the newest information on the consistency group state by issuing the RESET command described in “RESET” on page 194.

When you issue RESET, you allow ConGroup to recognize a change in the state of a consistency group that would ordinarily not be recognized until the next automatic verification cycle.

**Note:** “Automatic verification” on page 94 describes automatic verification.

ConGroup sets the state of the consistency group based on whether or not the consistency group was enabled and suspended/resumed by another LPAR.

Normally, RESET does not automatically verify all the attributes of each device (for example, synchronous versus Adoptive Copy, whether it contains a page dataset, and so on). If you want full verification, issue the RESET command with the VERIFY parameter.

RESET re-enables trip processing for a consistency group that has had a suspend timeout failure.

**Note:** “Tripping consistency groups” on page 99 describes consistency group trips.

After a suspend timeout failure, data at the remote site is not guaranteed to be consistent. Most likely, some devices are suspended, while other devices are not. You need to resolve the reason for the suspend timeout failure and then fully resume the consistency group before you issue the RESET command for the consistency group.

**Note:** “Resuming consistency group operations” on page 105 discusses how to resume a consistency group.
Cleaning up environment with ECGUTIL

Overview

Certain events may leave devices in your ConGroup environment in an incorrect state.

Suppose you have a channel failure that forces a swap. The swap leaves ConGroup-enabled R1 devices that ConGroup is now not able to disable.

In such situation, you can use the ECGUTIL utility to clean up the environment from a different host that can still reach the devices.

Security

Use of ECGUTIL requires proper RACF authorization. Define a RACF class/resource combination of XFACILIT/EMC.ADMIN.FNC.CG.ECGUTIL and give authorized users UPDATE access to it. If the resource is not defined, anyone may use the program. ECGUTIL displays the ECGC001I and ECGC002I messages if the resource is not protected.

Limitations

The ECGUTIL utility has the following limitations:

- Only locally attached VMAX systems are supported.
- Except for validating the VMAX system serial number or name, ECGUTIL does not perform any checks before issuing its syscalls. ECGUTIL simply issues the syscalls to the VMAX systems through gatekeepers it receives from ResourcePak Base (SCF). Because no locking or other cooperative protocols are followed, you could be adversely affecting other applications when running this utility.
- If you specify DEVICES=(000-FFF) (for example), ECGUTIL issues 4096 syscalls even if there is only one device of those named in the VMAX system. If you try to issue syscalls against non-existent devices, you receive a return code 8 and the ECGC006E message is written to the job log for each syscall in error.
- No post-syscall polling is done to confirm proper execution of the generated syscalls. Individual syscall results may be seen in HYPPRINT and in summary form with the final utility return code. Furthermore, because syscalls execute asynchronously in the VMAX system, the actual desired state changes can occur slightly after successful completion of the syscalls.
Running ECGUTIL utility

You can run ECGUTIL as a started task, as a batch job or as a called subroutine. In any case, all utility functions are driven concurrently by JCL DD statements and z/OS console input.

The following is an example of JCL for the ECGUTIL utility:

```bash
//STEP EXEC PGM=ECGUTIL[,PARM=parameters]
//STEPLIB DD DSN=ECGUTIL.LINKLIB,DISP=SHR
//HYPPRINT DD SYSOUT=A
//SCF$nnnn DD DUMMY
//COMMAND DD *     <-- ECGUTIL in-stream commands/PDS member
```

Where:

- **parameters** are described in “EXEC PARM parameters” on page 149.
- The HYPPRINT DD statement determines where trace messages are written to. When not specified, trace data is written to ResourcePak Base (SCF).
- **nnnn** identifies the ResourcePak Base task that the utility job runs against.
- **ECGUTIL in-stream commands** are described in “ECGUTIL commands” on page 199. You can specify the PDS member that contains the commands instead.

EXEC PARM parameters

**MSGLEVEL(n)**

Overrides the default message level of 5 for ECGUTIL HYPPRINT output with a value of 1 through 9.

Specifying MSGLEVEL(7) displays all syscalls (including successful ones) so hoplists can be seen.

**LOCAL|REMOTE|EITHER**

Determines location of the devices accessed by the ECGUTIL utility:

- **(Default) LOCAL** — Only access devices directly attached to a channel.
- **REMOTE** — Only access devices indirectly via a secondary VMAX system.
- **EITHER** — Attempt LOCAL mode, but drop back to REMOTE if no local paths are available.
Gatekeepers

ECGUTIL uses an internal gatekeeper server (which is also used by ConGroup, as described in “Gatekeeper server” on page 141) to determine which gatekeeper devices to use. The gatekeeper server obtains a list of gatekeepers from ResourcePak Base (SCF).

Use the SCF.GATEKEEPER.LIST initialization parameter of ResourcePak Base to specify one or more gatekeepers. If you specify multiple gatekeepers, ECGUTIL uses them in parallel (if you define multiple groups). All syscalls for a group are driven serially, but groups are handled in parallel. Each generated syscall object requests a free gatekeeper from the internal gatekeeper server “on the fly” before issuing the actual syscall. After syscall completion, the object returns the gatekeeper to the pool. This technique allows for high levels of parallelism whenever sufficient gatekeepers are available.

Automatic access using hop lists to target devices via secondary VMAX systems is available. It is sometimes useful to force ECGUTIL to access a VMAX system from another VMAX system (rather than directly across the channel).

This capability is dependent on the connected SCF configuration and the gatekeeper list provided to ECGUTIL at run time.

Note: The ResourcePak Base for z/OS Product Guide describes SCF and its initialization parameters.

Terminating ECGUTIL

You can terminate the ECGUTIL utility using the STOP command described in “STOP” on page 203.

If you issue an in-stream STOP command, ECGUTIL executes prior in-stream commands and then stops processing. If you do not issue STOP, ECGUTIL waits for further input from the console after executing any other in-stream commands.

Avoiding an in-stream STOP command allows you to specify groups and then operate upon the devices in the groups with console commands. After you finish issuing other console commands to the utility, issue STOP to direct the program to end.

Messages

ECGUTIL can generate messages with the prefix ECGC. You can find these messages described in the Mainframe Enablers Message Guide.

You can set the verbosity level for ECGUTIL messages using the SET MSGLEVEL command described in “SET MSGLEVEL” on page 203.
Return codes

The following are the return codes you can receive:

0  All generated syscalls succeeded.

**Note:** All syscalls might succeed and the final return code might not be zero (0). For example, return code 8 is returned if the ECGC004E message is issued (for an extraneous ADD statement for a non-existent group). But other statements might very well succeed.

4  At least one generated syscall received an application layer error. Search HYPPRINT DD for the string “RC=4” and then examine the data fields displayed after the preceding strings “PAYLOAD” and “SYSCALL CALLBACK” for details of the problem.

8, 12, 16  A variety of possible errors. Look for RC=8, 12, or 16 in HYPPRINT or error messages in the SYSLOG for details.
Setting/clearing device flags with ECGUTIL

To set or clear device flags, take the following steps:

1. Define a device group with the DG command described in “DG” on page 202.
   
   Note: “Defining a device group” on page 152 describes defining a device group.

2. Fill the group with devices using the ADD DEVICES command described in “ADD DEVICES” on page 200.
   
   Note: “Adding devices to a group” on page 153 describes adding devices to a device group.

3. Set or clear device flags with the SET GROUP command described in “SET GROUP” on page 202.
   
   Note: “Setting device flags” on page 153 describes setting devices flags.

4. Stop ECGUTIL with the STOP command described in “STOP” on page 203.
   
   Note: “Terminating ECGUTIL” on page 150 describes stopping UCGUTIL.

Defining a device group

You assign an arbitrary group name using the DG command described in “DG” on page 202.

The group allows you to set or clear flags for the devices in the group with one statement.

The group name does not have to have any relationship to any existing consistency group.

The group name used here and throughout the ECGUTIL utility has nothing to do with ConGroup or any other EMC product. This group exists only for the duration of ECGUTIL utility execution.
Adding devices to a group

After defining a group, you fill it with devices using the ADD DEVICES command described in “ADD DEVICES” on page 200.

The devices can be either CKD or FBA.

Each device group may consist of devices from different VMAX systems.

Any duplicate device specified within a single group is automatically ignored.

You can add the same devices to more than one group. The ability to add devices to more than one group is useful if you want to operate on subsets of a single group. Consider the following example:

DG ALL.
DG LEFT.
DG RIGHT.
DG MIDDLE.
DG 107ONLY.
ADD DEVICES={107} TO GROUP 107ONLY CNTRL 6185.
ADD DEVICES={107} TO GROUP ALL CNTRL 6185.
ADD DEVICES={100-10F} TO GROUP ALL CNTRL 6185.
ADD DEVICES={100-102} TO GROUP LEFT CNTRL 6185.
ADD DEVICES={103-10A} TO GROUP MIDDLE CNTRL 6185.
ADD DEVICES={10B-10F} TO GROUP RIGHT CNTRL 6185.

In this example, you can operate on subsets of group ALL by operating on groups LEFT, MIDDLE and RIGHT or 107ONLY.

Group 107ONLY consists of just device 107. It is also part of group ALL. Because it is singularly part of group 107ONLY, you can operate on just device 107 with commands specifying group 107ONLY.

You can restrict action to a certain leg of the device(s) added to the group. In the following example, the RA parameter restricts action to the 08 leg:

ADD DEVICES={158-15A} TO GROUP MYGRP CNTRL 000195700079 RA 08.

Setting device flags

You can set or clear device flags using the SET GROUP command described in “SET GROUP” on page 202.

Flags include the CG ENA bit, the SRDF ECA state, and the USR Ready/Not Ready state.

The ConGroup CG ENA state and the SRDF ECA state consist of fixed default values for group IDs, various masks, and other values because the ECGUTIL does not provide a way to specify these values. The “enabling” keywords exist in ECGUTIL only for testing purposes to allow conformation that the cleaning keywords function correctly.
Example

The following example illustrates how to use ECGUTIL:

DG TEST1.
ADD DEVICES=(20,220-22F)
   TO GROUP TEST1 CNTRL 000190100991.
SET GROUP TEST1 USRNRDY.
SET GROUP TEST1 USRRDY.
STOP.

This example defines the group TEST1, adds devices to it, and defines the serial number of the VMAX system. It then uses the two SET GROUP commands to clear NR. Finally, it issues a STOP command to end ECGUTIL.
Diagnostics and troubleshooting

ConGroup uses the following diagnostic facilities available through ConGroup and the operating system:

- Diagnostic messages
- z/OS dump services
- z/OS GTF tracing service
- z/OS LOGREC (EREP) data

Where a condition occurs that prevents continued operation, ConGroup takes the following steps:

- Suspends processing.
- Generates a symptom record in the LOGREC dataset.

If an ABEND is associated with the error, ConGroup formats VRA data and creates a LOGREC record.

To facilitate problem diagnosis, retain the LOGREC software EREP records for EMC Customer Support. You can format the software records using the following JCL. The example assumes that SYS1.LOGREC is being used.

```plaintext
//STEP1   EXEC PGM=IFCEREP1,PARM='CARD'
//EREPPT  DD  SYSOUT=*  
//TOURIST DD  SYSOUT=*  
//SERLOG  DD DSN=SYS1.LOGREC,DISP=SHR  
//ACCDEV  DD DUMMY  
//SYSIN   DD *  
  PRINT=PS  
  HIST=N  
  TYPE=S  
  ENDPARM
```

Note: The IBM publication, *EREP Reference*, provides more information about EREP records.
CHAPTER 6
ConGroup Reports

This chapter covers the following topics:

◆ ConGroup Environment report ................................................................. 158
◆ Consistency Group — Summary report ...................................................... 160
◆ Consistency Group — Details report .......................................................... 163
◆ SMFID report ............................................................................................ 165
◆ Gatekeeper report ...................................................................................... 165
ConGroup Environment report

**CAUTION**

This report is primarily designed for diagnostic purposes. It is intended for experienced users and EMC Support.

The ConGroup Environment report displays global configuration parameters currently set for your ConGroup environment, followed by information on the worker processes, the ConGroup mode, gatekeepers, and CAX. For a description of each field, refer to “Global configuration parameters” on page 30.

To produce the ConGroup Environment report, use the DISPLAY ENVIRONMENT command described in “DISPLAY ENVIRONMENT” on page 188.

A sample ConGroup Environment report is show in Figure 17.

TIME/NETID/GHA: 7150339/33168091/33168091
AUTO_REFRESH=ON
DISABLE_ON_VERIFY_ERROR=YES
DISPLAY_CONGROUP_LISTOPT=NOLIST
RESUME_OPTION=NOTNRMSG
SAF_CLASS=XFACILIT
SAF_PROFILE=EMC.ADMIN.CMD.CG
VERIFY_INTERVAL=0
Work Pool Size: 10 Free: 10 Busy: 0
5 Second Request History: 0 4 0 0 0
Gatekeeper Queue HWM: 0
DSK_RECVPORT=118
ConGroup is executing in MULTI mode
CONNECTED HOSTS: X004 X006 *X00B
AutoSwap Ownerid=X00B, LOCAL SYSTEM IS X00B
This is the owning system
AutoSwap Listencode=128

**Figure 17** ConGroup Environment report

**Note:** As global ConGroup settings may differ from site to site, the ConGroup Environment report at your site may show slightly different fields.
Report fields

In addition to current global settings, the ConGroup Environment report shows the fields listed in Table 11.

Table 11 ConGroup Environment report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME/NETID/GHA</td>
<td>An internal debugging value used by EMC.</td>
</tr>
<tr>
<td>Work Pool Size</td>
<td>The current work pool size shown as the number of active internal worker tasks. This field is for diagnostic use only.</td>
</tr>
<tr>
<td></td>
<td>• FREE indicates the number of worker tasks not currently working.</td>
</tr>
<tr>
<td></td>
<td>• BUSY indicates the number of worker tasks currently working. Because worker processes are overwhelmingly involved in issuing poll requests which require</td>
</tr>
<tr>
<td></td>
<td>gatekeepers, a sustained non-zero value for “busy” indicates sustained congestion.</td>
</tr>
<tr>
<td>5 Second Request</td>
<td>The number of work pool requests (typically polls) in each of the last 5 seconds is displayed as series of five numerals. The most recent second is at the left of the list.</td>
</tr>
<tr>
<td>History</td>
<td>Because most requests are very quick, even high values in this rolling history do not generally translate into a sustained non-zero busy value. “Catching” a non-zero busy value should be fairly rare in a normally functioning system. Of course, the busier the system, the more likely you are to find a non-zero busy value.</td>
</tr>
<tr>
<td>Gatekeeper Queue HWM</td>
<td>The maximum number of requests for a gatekeeper that could not be immediately satisfied. This value indicates the number of gatekeeper requests have had to wait for a free gatekeeper since startup or since the last issuance of the CE GKREFR command. (HWM stands for “high water mark.”)</td>
</tr>
<tr>
<td>CONNECTED HOSTS</td>
<td>A list of the SMFIDs of all connected hosts, separated by a space, at the time the command is issued. The Owner SMFID is prefixed by an asterisk (*). Multiple lines are displayed as necessary based upon the number of hosts.</td>
</tr>
</tbody>
</table>
ConGroup Reports

Consistency Group — Summary report

The Consistency Group — Summary report shows the following information:

- Global ConGroup parameters currently set for your environment
- Consistency group name and status
- Consistency group summary

To produce the Consistency Group — Summary report, use the DISPLAY CONGROUP command with the NOLIST parameter, as described in “DISPLAY CONGROUP” on page 187.

A sample Consistency Group — Summary report is shown in Figure 18:

GLOBAL SETTINGS:
COUPLEDS_ALLOWED=NO
DISABLE_AT_SHUTDOWN=OFF
PAGEDEV_ALLOWED=NO
REMSPLIT_INTERVAL=10
RESUME_INTERVAL=10
SEMISYNC_ALLOWED=YES
CONNECTED HOSTS: X004 X006 *X00B
CONGROUP= CGRPLR  ENABLED  ACTIVE
SUSPEND_FAILURE=FAIL  SUSPEND_RETRY_TIMEOUT=0
This is an AutoSwap group
SWAP CONDITIONS: INTREQ VAULT
Group Summary:
+-----------------------------------------------------+
|   CG   |  CAX   |  RECA  |  TNR   |  Dev#  |  Invt  |
+-----------------------------------------------------+
| ENABLED     NO    Y/ARMED    NONE         11        0 |

Figure 18  Consistency Group — Summary report
Variable GLOBAL SETTINGS lines

The following 2 lines are only displayed if an AutoSwap group is defined:

This is an AutoSwap group
SWAP CONDITIONS: [option values]

Table 13 lists possible swap option values.

Table 12 Consistency Group — Summary report, swap option values

<table>
<thead>
<tr>
<th>Option value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNCLINK FAILURE</td>
<td>Swap on any trip condition.</td>
</tr>
</tbody>
</table>
| NOPATHS | Normal unplanned swap condition. This value is the default if UNPLANNEDCONDITIONS is not specified.  

Note: “UNPLANNEDCONDITIONS” on page 61 describes the UNPLANNEDCONDITIONS CAX option. |
| INTREQ | Normal unplanned swap condition. |
| VAULT | Indicates that ConGroup will attempt to detect VMAX system shutdown in order to trigger a swap. This will always show up if either NOPATHS or INTREQ are specified.  

Note: Enginuity patch 80268 is still required on a system by system basis for this feature to work. |
| NONE | “UNPLANNEDCONDITIONS=” specified with no conditions following the “=” sign.  

Note: “UNPLANNEDCONDITIONS” on page 61 describes the UNPLANNEDCONDITIONS CAX option. |
ConGroup Reports

Report fields

Table 13 lists the fields of the Consistency Group — Summary report.

**Table 13  Consistency Group — Summary report fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| CG    | Consistency group status, indicated by any of the following values:  
  • ENABLED — All of the devices are ConGroup-enabled.  
  • DISABLED — All of the devices are ConGroup-disabled.  
  • MIXED — Some devices are enabled and some are disabled. |
| CAX   | Whether the consistency group is defined as a CAX group. The field can have any of the following values:  
  • YES — The consistency group is defined as a CAX group.  
  • NO — The consistency group is not defined as a CAX group.  
  • /DEFINED — The underlying swap group exists.  
  • /UNDEFINED — The underlying swap group does not exist. |
| RECA  | The SRDF ECA status for devices that are protected with SRDF ECA mode:  
  • Y/CLEAR — No SRDF ECA is defined.  
  • Y/ARMED — SRDF ECA is armed. This is the normal SRDF ECA status.  
  • Y/OPEN — The ECA Window is open as the result of a ConGroup trip.  
  • Y/CLOSED — The trip processing is complete. (Shows when tripped). The ECA Window is closed via SYSCALL or by timeout.  
  • Y/MIXED — The devices are not all in the same status. |
| TNR   | Whether the devices have the Target Not Ready (TRN) status. The field can have any of the following values:  
  • ALL — All protected mirrors are TRN.  
  • NONE — None of the protected mirrors are TRN.  
  • MIXED — Some mirrors are TRN and others are not. |
| Dev#  | The total R1 device count for the consistency group. |
| Invt  | The total number of invalid tracks in the consistency group. |
Consistency Group — Details report

The Consistency Group — Details report shows the following information:

◆ Global ConGroup parameters currently set for your environment
◆ Consistency group name, ID, and status

**Note:** All statuses in this display are polled results. If ConGroup cannot determine the status of a consistency group, ConGroup displays the last known status followed by “????”.

◆ Configuration parameters set for this consistency group
◆ VMAX system serial number and OS level
◆ Listing of devices in the consistency group

The Consistency Group — Details report consists of a details section and group summary. If there is more than one VMAX system in the consistency group, the report contains a series of detail sections followed by a single group summary section.

To produce the Consistency Group — Details report, use the DISPLAY CONGROUP command with the LIST parameter, as described in “DISPLAY CONGROUP” on page 187.

A sample Consistency Group — Details report is shown in Figure 19:

```
CGRP2821 D C LIST 204
*** Begin Display from system X04 ***
GLOBAL SETTINGS: COUPLEDS_ALLOWED=NO
DISABLE AT SHUTDOWN=OFF
PAGEDEV_ALLOWED=NO
REMSPLIT_INTERVAL=10
RESUME INTERVAL=10
SEMISYNC ALLOWED=YES
CONNECTED HOSTS: X004 X006 *X00B
CONGROUP= CGRP2821 ENABLED ACTIVE ...
SYNCLINKFAILURE=NO
SUSPEND.FAILURE=FAIL SUSPEND_RETRY_TIMEOUT=0
CTRL SER#=000190300344 uCode: 57720042 GroupId: 0001
USC1 = 2105 5772
| R1 | R1 | Volser|Rdf-| CG |Sync|InvT|RA GRP/Mirr| NR |
| Cue | Dev# z|or oth|1234|1234|1234|1234| 1 2 3 4|1234|
| 4D3A| 00007A|......|A...|..E.|..S.|....|..|..|23|..|....|
| 4D3B| 00007B|......|A...|..E.|..S.|....|..|..|23|..|....|
| 4D3C| 00007C|......|A...|..E.|..S.|....|..|..|23|..|....|
| 4D3D| 00007D|......|A...|..E.|..S.|....|..|..|23|..|....|
```

Figure 19  Consistency Group — Details report

If the consistency group you are displaying is defined as capable of being swapped, the display includes the line “This is an AutoSwap group.” For example:

```
CGRP2821 D C LIST 204
*** Begin Display from system X04 ***
GLOBAL SETTINGS: COUPLEDS_ALLOWED=NO
DISABLE_AT_SHUTDOWN=ON
PAGEDEV_ALLOWED=YES
REMSPLIT_INTERVAL=5
RESUME INTERVAL=30
SEMISYNC_ALLOWED=NO
CONNECTED HOSTS: X004 X006 *X00B
CONGROUP= CGR PX1B ENABLED ACTIVE ...
SUSPEND.FAILURE=RETRY SUSPEND_RETRY_TIMEOUT=0
This is an AutoSwap group
CTRL SER#=000190300344 uCode: 58760239 GroupId: 0001
```
Details section fields

Table 14 lists the fields in the details section of the Consistency Group — Details report.

**Table 14 Consistency Group — Details report, details section fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 Cuu</td>
<td>The z/OS device number.</td>
</tr>
<tr>
<td>R1 Dev#</td>
<td>The VMAX device number.</td>
</tr>
<tr>
<td>Volser or oth</td>
<td>The volser if the device is online, or “..” if the device is offline.</td>
</tr>
<tr>
<td>Rdf-</td>
<td>The device status:</td>
</tr>
<tr>
<td></td>
<td>• A — Armed, the normal status.</td>
</tr>
<tr>
<td></td>
<td>• T — Timeout, the ECA Window is closed by timeout.</td>
</tr>
<tr>
<td></td>
<td>• C — Closed, the trip processing is complete. (That is, shows when complete.)</td>
</tr>
<tr>
<td></td>
<td>• O — Open, the ECA Window that is open as the result of a ConGroup trip.</td>
</tr>
<tr>
<td></td>
<td>Otherwise “....”.</td>
</tr>
<tr>
<td>CG</td>
<td>The consistency group status:</td>
</tr>
<tr>
<td></td>
<td>• E — Enabled.</td>
</tr>
<tr>
<td></td>
<td>• D — Disabled. (This status is based on R1 polling only.)</td>
</tr>
<tr>
<td>Sync</td>
<td>The SRDF status for each mirror:</td>
</tr>
<tr>
<td></td>
<td>• S — Synchronized <em>(Ready).</em></td>
</tr>
<tr>
<td></td>
<td>• U — Unsynchronized <em>(Not Ready).</em></td>
</tr>
<tr>
<td></td>
<td>This is independent of whether ConGroup can protect the mirror.</td>
</tr>
<tr>
<td>InvT</td>
<td>The invalid track status for each mirror:</td>
</tr>
<tr>
<td></td>
<td>• I — For mirrors that have invalid tracks.</td>
</tr>
<tr>
<td></td>
<td>This status applies to any configured mirror.</td>
</tr>
<tr>
<td>RA GRP/Mirr</td>
<td>SRDF group and mirror mask numbers; otherwise, “.” if the corresponding mirror is not SRDF-protected.</td>
</tr>
<tr>
<td>NR</td>
<td>The <em>Not Ready</em> status for each mirror.</td>
</tr>
<tr>
<td></td>
<td>• N — For mirrors that are <em>Not Ready</em>. Only mirrors shown in the Prot column are tested. N in 1 is <em>User Not Ready</em>.</td>
</tr>
</tbody>
</table>

Summary section fields

The summary section looks as shown in Figure 20:

```
+-----------------------------------------------+--------+--------+--------+--------+--------+--------+--------+
|     CG      |    CAX |    RECA |    TNR |    Dev# |    InvT |     No |
+-----------------------------------------------+--------+--------+--------+--------+--------+--------+
|  ENABLED   |    NO  |   Y/ARMED |  NONE  |     11  |     0   |
+-----------------------------------------------+--------+--------+--------+--------+--------+--------+
```

Figure 20 Consistency Group — Details report, Summary section

For a description of fields, refer to “Report fields” on page 162.
SMFID report

The SMFID report lists SMFIDs of the participating ConGroup address spaces. It shows which LPARs are also running ConGroup, and which LPAR is the current owner.

Note: “Managing owner LPARs” on page 138 describes running ConGroup on different LPARs.

To produce the SMFID report, use the LA command described in “LA” on page 190.

A sample SMFID report is shown in Figure 21:

```
F CG61A,LA
EMCP001I LA
Hosts running ConGroup
Z06    <-- OWNER
Z05
Z04
```

Figure 21  SMFID report

Gatekeeper report

The Gatekeeper report shows which gatekeepers can be used by ConGroup, and which gatekeepers are actually in use. It displays the current gatekeeper status for each VMAX system that is known by the ConGroup gatekeeper server. The report provides the following information: the VMAX system serial number, OS level, channel address, location, type, pin status.

Note: “Managing gatekeepers” on page 141 discusses ConGroup gatekeepers.

To produce the Gatekeeper report, use the #DISPLAY GATEKEEPERS command described in “#DISPLAY GATEKEEPERS” on page 188.

A sample Gatekeeper report is shown in Figure 22:

```
000195600140 5076  8003 LOCAL  USR
000195700079 5076  3801 LOCAL  USR PINNED Active
000195700079 5076  3802 LOCAL  USR PINNED Active
000195700080 5076  3C00 LOCAL  SCF
```

Figure 22  Gatekeeper report

The “PINNED Active” status means that ConGroup is using this device as a gatekeeper, and I/O can happen any time.
ConGroup Reports
CHAPTER 7
Database Considerations

This chapter covers the following topics:

- Introduction .................................................................................................................. 168
- DB2 considerations ...................................................................................................... 171
- IMS considerations ...................................................................................................... 173
- Mixed database considerations ..................................................................................... 175
Introduction

When in SRDF remote mirroring in campus mode across two or more VMAX systems, a database management system (DMBS) can have its logs or data on different VMAX systems. When failures occur, they do not occur consistently. On one secondary (remote) device, a write failure or an SRDF link failure can occur while on another secondary (remote) device, the data flow continues from the primary (local) devices.

Then, the various elements in a dependent data stream are out of synchronization.

ConGroup and database I/O

ConGroup keeps dependent I/O synchronized when its data is spread across multiple VMAX systems. Typical database management, system-generated, dependent I/O consists of the following:

- Logs
- Application data
- Metadata
- System data

The dependent I/O occurs only after its predecessor I/O successfully completes. When such I/O spans multiple VMAX systems, ConGroup is to freeze dependent I/O when predecessor I/O is unsuccessful because of a device or link failure.

Note: In all cases, the devices potentially related by I/O dependency must be specified in the consistency group device list, as described in “Device definitions” on page 169.

Database consistency group suspensions

The effect of a consistency group suspension on a DBMS is similar to that of a z/OS power loss outage. The DBMS does not have an opportunity to properly flush buffers or close logs and database files. Under such circumstances, a warm or emergency startup of the DMBS resolves any data inconsistencies caused by the outage, as long as the DBMS has the necessary log records to perform backout to a prior point of consistency.
Device definitions

All devices used for related dependent I/O must be defined in one consistency group. If you are a database administrator, make sure that you include the following types of volumes in the consistency group device list:

- Volumes where log and system data resides.
- Volumes where application data resides.

**Note:** Group application data with the logs because application data I/O is dependent on the logs. Failure to do so negates the purpose of ConGroup protection.

- Volumes that contain the system data of any transaction processor, such as CICS, that you use to coordinate synchpoints, commits, or checkpoints (between either IMS, DB2, or both).

If your site does not use SMS, you, as the database administrator, must constantly monitor the migration of database data to ensure that the secondary devices remain within an active consistency group device list.

Expanding to new devices

When you expand I/O to devices outside an existing active consistency group, take the following steps:

- Update the device list with new devices.

  **Note:** “Adding devices to consistency group” on page 82 describes adding devices to a consistency group.

- Refresh consistency group configuration with the ConGroup REFRESH command.

  **Note:** “Refreshing ConGroup configuration” on page 144 describes refreshing the consistency group configuration.

Specifying an SMS group in the group device list greatly simplifies this task by keeping the contents of the consistency group device list static.

**Note:** “Adding devices by SMS group” on page 83 describes using SMS groups to add devices to the consistency group.
Other configuration parameters

EMC recommends that you use SUSPEND_FAILURE=RETRY or WTOR if remote data integrity must take precedence over local database availability.

**Note:** “SUSPEND_FAILURE” on page 49 describes the SUSPEND_FAILURE configuration parameter.

If a SUSPEND I/O fails because of an unavailable path after you use SUSPEND_FAILURE=RETRY, ConGroup suspends local I/O and retries the SUSPEND until the SUSPEND_TIMEOUT value is exceeded.

If you want local database availability to take precedence over remote data integrity, choose SUSPEND_FAILURE=FAIL. This parameter allows local I/O to resume immediately upon experiencing a SUSPEND I/O failure because of an unavailable VMAX link.

Recovery site startup

The device list configuration ensures that I/O to devices containing log data is never superseded by I/O to devices containing application or system data. Conversely, if a recovery site log has signaled that a unit-of-work (UOW) has completed, while I/O had not reached its remote target database, the incomplete UOW on the database would not be detected because the recovery would be driven by end-UOW records present on the log.

ConGroup ensures that both data and logs remain synchronized so that database management system startup can resolve such transient data inconsistencies. Database management system startup rolls back database updates to a prior checkpoint or commit point. Avoid any type of restart that suppresses this process, such as a cold start, if you want consistent data.
DB2 considerations

Device definitions

Volumes in the ConGroup device list for a DB2 group must include all volumes where DB2 system data and all other application tablespaces reside. DB2 system data whose devices must be included are:

- ICF CATALOGS
- ARCHIVE LOGS
- BSDS01
- BSDS02
- LOGCOPY1.DS01 - LOGCOPY1.DSnn
- LOGCOPY2.DS01 - LOGCOPY2.DSnn
- DSNDB01
- DSNDB04
- DSNDB06
- DSNRLST
- DSNDF (V4 or lower)
- DSNRGFDB (if used)

You should include devices containing linear datasets comprising all DB2 application data, vendor applications, and the Query Management Facility (QMF). To make this device list comprehensive, include all potentially usable devices found in SYSIBM.SYSVOLUMES that are pointed to by DB2 storgroups during the device list definition.

Storage groups can span to such volumes when currently used volumes no longer contain room for table spaces defined after the device list has been defined. Such table spaces would be unintentionally excluded from the consistency group if the currently unused devices in SYSIBM.SYSVOLUMES were not included. Of course, an ALTER STOGROUP sgname ADD VOLUMES (volser list) should also be reflected in an updated device list followed by a REFRESH command.

You may exclude DSNDB07 because it is not required for DB2 recovery site restart. You can exclude DSNDB07 with the EXCLUDE configuration parameter described in “EXCLUDE” on page 47.

If you exclude DSNDB07, no other DB2 data can share its devices. Sharing those devices may cause this data to be unintentionally unprotected. For the same reason, you must also remove this database’s devices from any SMS storage groups to prevent SMS from moving needed files to devices shared with this database. You can allocate DSNDB07 at the recovery site just before DB2 restart.
Database Considerations

DB2 consistency group example

Figure 23 shows an example of a configuration file entry for a DB2 consistency group:

```plaintext
RESUME_INTERVAL=60
CONGROUP=DB2PROD
* The following device list ranges represent devices where
* all DB2 system and application data resides.
DEVICE_LIST=2F0-2FB,DB2001,DB201F
SMS_GROUP=DB2GRP
* The following excludes DB2005, DB2007
* for DSNDDB07. No other critical data may share these devices.
EXCLUDE=DB2005,DB2007
SUSPEND_FAILURE=RETRY
SUSPEND_TIMEOUT=30
```

Figure 23  DB2 consistency group example

Recovery site startup

When a consistency group is tripped and DB2 startup commences at the recovery site, you need to permit a warm start to resolve in-progress, in-commit, in-abort or in-doubt units-of-recovery (URs) so that they are either committed or rolled back. This is the only way that ConGroup can guarantee data consistency. The DB2 log contains the data it needs to perform DB2 roll-forward and backward recovery of incomplete unit-of-recovery-IDs (URIDs) during the warm start process.

EMC does not recommend that you use a cold start, or conditional restart, which suppresses the UR recovery. Use cold starts with extreme caution, and only after you examine the DB2 active logs to determine if incomplete URIDs do not exist.

Whether or not you use ConGroup protection, avoid long-running update tasks with low commit-to-update ratios. Such tasks cause extended restart times.
IMS considerations

Device definitions

Depending on the IMS configuration, include devices containing only IMS system datasets in the consistency group definition, in addition to those containing application databases, as follows:

- DFSOLP00 - DFSOLPnn
- DFSOLS00 - DFSOLSnn
- DFSWADS0 - DFSWADSn
- RECON1
- RECON2
- RECON3
- MODSTAT
- MODBLKS
- MATRIX
- ACBLIB
- FORMAT
- RDS
- QBLKS
- SHMSG
- LGMSG

IMS consistency group example

Figure 24 shows an example of a configuration file entry for an IMS consistency group:

RESUME_INTERVAL=60
CONGROUP=IMSPROD
* The following device list ranges represent devices where
* all IMS system and application data resides.
DEVICE_LIST=2E0-2EB,IMS001,IMS01F
SMS_GROUP=IMSAPPS
SMS_GROUP=IMSWADS
SMS_GROUP=IMSOOLDS
* The following excludes IMS005, IMS007.
* No other critical data may share these devices.
EXCLUDE=IMS005,IMS007
SUSPEND_FAILURE=FAIL

Figure 24  IMS consistency group example
Recovery site startup

When a consistency group is suspended and IMS startup begins at the recovery site, you need to allow IMS to undergo an emergency restart. The emergency restart permits IMS to close the previously active log and backout databases to a prior checkpoint encountered on the WADS and OLDS.

To permit an emergency restart, issue the command:

/ERE OVERRIDE

Then, wait for the DFS3257I message, followed by the DFS994I message.

EMC does not recommend an IMS cold start. A cold start does not perform all log closes and database backouts. Use a cold start with extreme caution only after the following:

- Examining the OLDS by running DFSULTR0 in PSB mode
- Obtaining a list of active PSBs
- Performing batch backout against WADs and OLDS
Mixed database considerations

In the case of hybrid applications which use multiple DBMS (for example, IMS and DB2) within the same unit of work, use one common consistency group definition.

Device definitions

If there is no logical relationship between data in multiple DBMS, you can use separate consistency group definitions. However, if these separate consistency group definitions contain common devices, they do, in effect, behave as a single consistency group.

**Note:** Different consistency groups containing common devices are called intersecting consistency groups.

This is because the SRDF suspension of the intersecting devices caused by one group triggering also causes the other group to trigger.

Hybrid consistency group example

*Figure 25* shows an example of a configuration file entry for a hybrid consistency group:

```
RESUME_INTERVAL=60
CONGROUP=COMMON
  * The following device list ranges represent devices where
  * all IMS system and application data resides.
DEVICE_LIST=2E0-2EB,IMS001-IMS01F
SMS_GROUP=IMSAPPS
SMS_GROUP=IMSWADS
SMS_GROUP=IMSOLDS
  * The following excludes IMS005, IMS007.
  * No other critical data may share these devices.
EXCLUDE=IMS005,IMS007
  * The following device list ranges represent devices where
  * all CICS system data and journals resides.
DEVICE_LIST=2D0-2DB,CIC001-CIC01F
SMS_GROUP=CICJRL
  * The following excludes a subset of range DB2005,DB2007
  * for DSNDB07. No other critical data may share these devices.
EXCLUDE=DB2005,DB2007
SUSPEND_FAILURE=RETRY
SUSPEND_TIMEOUT=45
```

*Figure 25*  Hybrid consistency group example
Recovery site startup

In addition to DBMS-specific restart issues, you should consider the following recovery site operational issues.

- DB2 restart may produce *indoubt threads* that must be resolved with the DB2 command:
  
  ```
  RECOVER INDOUBT connection name ACTION (ABORT)
  ```

  You can identify indoubt threads by issuing the command:

  ```
  DISPLAY THREAD (*) TYPE (INDOUBT)
  ```

- IMS restart may also produce *indoubt threads* that you can identify by issuing either of the following commands:

  ```
  /DIS CCTL ALL INDOUBT
  ```

  or

  ```
  /DIS OASN SUBSYS ALL
  ```

Following recovery site restart, such scenarios result when both the transaction manager and the DBMS logs have not recorded completion of a two-phased commit on the target (R2) devices at the recovery site. Use of separate DBMS-specific consistency groups are more likely to cause this problem.
CHAPTER 8
Command Reference

This chapter covers the following topics:

◆ Syntax conventions ................................................................. 178
◆ ConGroup commands ............................................................... 179
◆ ECGUTIL commands .............................................................. 199
Syntax conventions

The commands follow these syntax conventions:

- Keywords appear in uppercase (for example, **ADD**). They must be spelled exactly as shown.
- For easy reference, command keywords are supplemented by lowercase letters to form a meaningful word (for example, **DEVices**). When typing a command, use only CAPITALIZED characters of any keyword.
- Variables appear in lowercase and italics (for example, **cngrp**). They represent user-supplied names or values in the syntax.
- Square brackets [ ] indicate an optional entry (for example, **cuu [cuu]**).
- The vertical bar | indicates alternative argument values (for example, **SUSPEND | REMSPLIT**).
- Curly brackets { } are used to group a series of alternative values that can be used with a single keyword, for example: **{SUSPEND | REMSPLIT | RESUME}**.
- Aside from the square and curly brackets and the vertical bar characters, you must type all other characters that are shown in the syntax statements.
- Default values are indicated by an underline. For example, if the parameter has the following option, (**WAIT | NOWAIT**), the underlined **NOWAIT** indicates the default value.
ConGroup commands

ConGroup operator commands are issued in the following format:

`F emccgrp,command[parameter[parameter[...]]]`

Where:

◆ `emccgrp` is the name of the ConGroup started task.
◆ `command` specifies a ConGroup command.
◆ `parameter` specifies a ConGroup command parameter (if any).

**ADD**

Adds devices to an existing consistency group.

*Note:* “Adding devices to consistency group” on page 82 describes adding devices to the consistency group.

For an FBA device, both meta heads and all members must be specified.

**Syntax**

```
ADD
{CUUs(cuu[-cuu],[,cuu[-cuu]]...[,cuu[-cuu]])|
DEVices(symdv#[-symdv#],[,symdv#[-symdv#]]...[,symdv#[-symdv#]])
CNTRL(SymmID) }
Group(cngrp)
RAgroup(srdfgrp[,srdfgrp])
[TIMEout(seconds)]
```

**Parameters**

**CNTRL(SymmID)**

The serial number of the VMAX system where the devices exist.

*Note:* The CNTRL parameter is only used with DEVICES, and is not compatible with the CUU parameter.

**CUUs(cuu[-cuu],[,cuu[-cuu]]...[,cuu[-cuu]])**

A device or range of devices identified with their CUUs.

You can specify up to 10 devices or device ranges separated by commas.

All devices are inserted into the running configuration using the SRDF groups specified with the RAGROUPS parameter.
DEVices(symdv#[-symdv#][,symdv#[-symdv#]]...[,symdv#[-symdv#]])
A device or range of devices identified with their VMAX device numbers.
You can specify up to 10 devices or device ranges separated by commas.

**Note:** If one or more devices are added using the DEVICES parameter of the ADD command that has a CUU assigned to it, the CUU is not displayed. It would be as if it were defined using the SYMM_DEV# configuration parameter described in “SYMM_DEV#” on page 50. This should be taken into consideration when updating your parameters after the ADD command is issued.

GROUP(cngrp)
The name of the consistency group to which the devices are added.

RAgroup(srdfgrp[, srdfgrp])
The SRDF group identified with its 2-character group ID.
You can specify up to 2 SRDF groups.
All devices to be added must have operational mirrors on the specified SRDF group or groups. Remote mirrors are assigned the same SRDF ECA group ID as the rest of the group on that VMAX system.
If CAX is used, two SRDF groups are not allowed. With CAX, only the first SRDF group is used for all devices.
The purpose of the RAgroup parameter is to validate the existence of SRDF group(s) on the VMAX system for use by a device. If the device has one leg, only one SRDF group is needed. If the device is concurrent, then two SRDF groups are needed. Since concurrent make no sense in the context of CAX, only the first SRDF group is used. There is no capability to match separate device ranges to different SRDF groups.

TIMEout(seconds)
Sets the maximum time (in seconds) to wait before giving up on the ADD.
Valid values are from 1 to 3600. The default value is 120 seconds.

**Note:** Change the default value only for especially busy systems to give the dynamic add time to complete.

Example
ADD DEVICES(F9-FF) CNTRL(000192600399) RAGROUP(90) GROUP(MSFCASST)
#ADD CONTROLLER

Adds a VMAX system to the ConGroup address space.

Note: “Adding/removing VMAX systems” on page 140 describes adding VMAX systems to ConGroup.

Syntax

```
#ADD [CONTROLLER] SER(SymmID)
```

Note: The CONTROLLER keyword is optional and assumed if not specified.

Parameters

```
SER(SymmID)
```

The VMAX system identified with its serial number.

Example

```
#ADD CONTROLLER SER(000812004096)
#ADD SER(000812004096)
```
CANCEL

Terminates an ongoing ConGroup process that may have been started by a ConGroup command or by external factors.

**IMPORTANT**
After you issue CANCEL, you must issue DISABLE to disable ConGroup. Only after ConGroup is disabled can you issue SRDF Host Component commands to suspend SRDF or change the mode.

**Syntax**

```
CANCEL {SUSPEND|REMSPLIT|RESUME} [cngrp]
```

**Parameters**

- `cngrp`
  - The name of the consistency group.

  **Note:** If you do not specify a consistency group name, all consistency groups are terminated.

- **REMSPLIT**
  - Terminates a REMSPLIT process or processes.

- **RESUME**
  - Terminates a RESUME process or processes.

- **SUSPEND**
  - Terminates a SUSPEND process or processes.

**Example**

1. To terminate all SUSPEND processes:
   ```
   CANCEL SUSPEND
   ```

2. To terminate the REMSPLIT process for consistency group DB2PROD:
   ```
   CANCEL REMSPLIT DB2PROD
   ```
DAS

Passes AutoSwap statements to CAX.

**Note:** The *AutoSwap for z/OS Product Guide* describes AutoSwap and its statements. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

This command can be used for consistency groups with CAX enabled.

**Syntax**

```
DAS statement
```

**Parameter**

```
statement

The AutoSwap statement (command and parameters) to be executed.

When specifying AutoSwap statements, you can use a continuation character to begin a new line. Continuations are indicated with a `-` after the last value on a line, for example:

```
DEF GRP PROD INC CUU=312C-312F -
  CFW=RES PREVAL PROCCNT=3 RETAIN REPLACE
```
```
Example

1. To view current AutoSwap settings:
   
   DAS DISPLAY OPT

2. To swap devices in the group named PATREC:
   
   DAS SWAP GROUP PATREC
DELETE

Deletes devices from an existing consistency group.

**Note:** “Managing consistency group members” on page 77 describes addition and deletion of devices.

For an FBA device, both meta heads and all members must be specified.

If any specified ranges include unprotected devices, the entire request fails.

**Syntax**

```
DELETE
{CUUs(cuu[-cuu] [, cuu[-cuu]] ...) [, cuu[-cuu]]}
DEVices(symdv#[-symdv#] [, symdv#[-symdv#]] ...) [, symdv#[-symdv#]]
CNTRL(SymmID)
Group(cngrp)
[TIMEout(seconds)]
```

**Parameters**

**CNTRL(SymmID)**

The serial number of the VMAX system where the devices exist.

**Note:** The CNTRL parameter is only used with DEVICES, and is not compatible with the CUU parameter.

**CUUs(cuu[-cuu] [, cuu[-cuu]] ...) [, cuu[-cuu]]**

A device or range of devices identified with their CUUs.

You can specify up to 10 devices or device ranges separated by commas.

**DEVices(symdv#[-symdv#] [, symdv#[-symdv#]] ...) [, symdv#[-symdv#]]**

A device or range of devices identified with their VMAX device numbers.

You can specify up to 10 devices or device ranges separated by commas.

**GROUP(cngrp)**

The name of the consistency group from which the devices are deleted.

**TIMEout(seconds)**

Sets the maximum time (in seconds) to wait before giving up on the ADD.

Valid values are from 1 to 3600. The default value is 120 seconds.

**Note:** Change the default value only for especially busy systems to give the dynamic delete time to complete.

**Example**

```
DELETE DEVICES(2B2-2B4) CNTRL(000192600399) GROUP(EMCGRP)
```
#DELETE CONTROLLER

Removes a VMAX system from a particular ConGroup address space.

**Note:** “Removing VMAX system manually” on page 140 describes removing VMAX systems from ConGroup manually.

⚠️ **CAUTION**

Use this command only when directed by EMC Support.

**Syntax**

```
#DELETE [CONTROLLER] SER(SymmID)
```

**Note:** The CONTROLLER keyword is optional and assumed if not specified.

**Parameters**

```
SER(SymmID)
```

The VMAX system identified with its serial number.

**Example**

```
#DELETE CONTROLLER SER(000812004096)
#DELETE SER(000812004096)
```
DISABLE

Disables a consistency group.

**Note:** "Disabling consistency groups" on page 95 describes disabling consistency groups.

**Syntax**

DISABLE cngrp [{NOWAIT|WAIT}]

**Parameters**

*cngrp*

The name of the consistency group.

*NOWAIT*

*(Default)* Executes the command immediately.

If the ALL-CONGROUPS lock is already held when the DISABLE command is issued, the command fails and the CGRP387W message is issued.

*WAIT*

Waits until the ALL-CONGROUPS lock is available. Meanwhile, the issued commands are "stacked".

**Example**

1. To disable consistency group SALES:

   ```
   DISABLE SALES
   ```

2. To disable consistency group PROFIT and wait for the ALL-CONGROUPS lock:

   ```
   DISABLE PROFIT WAIT
   ```
DISPLAY CONGROUP

Produces one of the consistency group reports:

- Consistency Group — Summary report described in “Consistency Group — Summary report” on page 160
- Consistency Group — Details report described in “Consistency Group — Details report” on page 163

You can set the default behavior for the DISPLAY CONGROUP command using the DISPLAY_CONGROUP_LISTOPT configuration parameter described in “DISPLAY_CONGROUP_LISTOPT” on page 34. The LIST|NOLIST options allow you to override the default for a specific DISPLAY command.

**Note:** You can abbreviate DISPLAY CONGROUP as D C.

**Syntax**

```
DISPLAY CONgroup[cngrp]{{LIST|NOLIST}}
```

**Parameters**

- **cngrp**
  The name of the consistency group.
  
  If no consistency group is specified, information for all consistency groups is displayed.

- **LIST**
  Produces the Consistency Group — Details report.

- **NOLIST**
  Produces the Consistency Group — Summary report.
DISPLAY ENVIRONMENT

Produces a ConGroup Environment report.

*Note:* “ConGroup Environment report” on page 158 describes the ConGroup Environment report.

Syntax

```
Display Environment
```

#DISPLAY GATEKEEPERS

Produces a Gatekeeper report.

*Note:* “Gatekeeper report” on page 165 describes the Gatekeeper report.

Syntax

```
#DISPLAY GATEKEEPERS
```
ENABLE

Enables a consistency group.

**Note:** “Enabling consistency groups” on page 95 describes enabling consistency groups.

**Syntax**

```plaintext
ENABLE cngrp [FORCE] [{NOWAIT|WAIT}]
```

**Parameters**

- `cngrp`  
  The name of the consistency group.

- `FORCE`
  Reenables the consistency group after a manual swap back of the devices.

  **Note:** “Resuming operations after swap” on page 121 describes swapping back devices.

  The FORCE parameter is intended to be used after a manual swap back so that operators are made aware that they must perform a swap back before enabling.

  An ENABLE `cngrp` FORCE command fails if all of the devices in the consistency group have not been swapped back.

- `NOWAIT`
  *(Default)* Executes the command immediately.

  If the ALL-CONGROUPS lock is already held when the DISABLE command is issued, the command fails and message CGRP387W is issued.

- `WAIT`
  Waits until the ALL-CONGROUPS lock is available. Meanwhile, the issued commands are “stacked”.

**Examples**

1. To enable consistency group PROD:
   ```plaintext
   ENABLE PROD
   ```

2. To enable consistency group TEST and wait for the ALL_CONGROUPS lock:
   ```plaintext
   ENABLE TEST WAIT
   ```
HELP

Displays available ConGroup commands with a short explanation. You can get help information on various general topics as well as on specific commands and parameters.

Issuing the HELP command with no keywords provides a brief overview of the HELP command parameters.

Issuing the HELP command with any of the following keywords gives you more information. Further “drilling down” is possible for some commands by using the DETAIL keyword with specific commands.

Syntax

HELP [{COMMANDS|CONFIG|MISC}|{command [DETAIL]}]

Parameters

command
A valid ConGroup command as listed in the HELP COMMANDS output.

COMMANDS
Lists ConGroup commands.

CONFIG
Displays configuration file statements.

DETAIL
Provides more details on the specified command if available.

MISC
Displays other help topics.

Example

1. To display a brief overview of HELP parameters:
   HELP

2. To view a brief overview of the ADD command:
   HELP ADD

3. To display a detailed description of the DELETE command:
   HELP DELETE DETAIL

LA

Produces the SMFID report.

Note: “SMFID report” on page 165 describes the SMFID report.

Syntax

LA
MOVEOWNER

Changes the current owner LPAR to another LPAR for CAX operations.

**Note:** “Managing owner LPARs” on page 138 describes owner LPARs. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

You can use the MOVEOWNER command only if you are using the multi-LPAR mode.

**Note:** “Multi-LPAR mode” on page 134 describes the multi-LPAR mode.

You can issue MOVEOWNER from any LPAR running ConGroup.

**Syntax**

MOVEOWNER[,]{LOCAL|target_smfid}

**Parameters**

LOCAL

Assigns ownership to the LPAR from which you issued the MOVEOWNER command.

target_smfid

The SMFID of the new owner LPAR (up to four characters).

**Example**

1. To move ownership from LPAR SYSA to SYSB:

   MOVEOWNER,SYSA,SYSB

2. To move ownership from the current owner LPAR to the LPAR the command is issued:

   MOVEOWNER,LOCAL
#PIN

Pins a ConGroup gatekeeper server-managed CUU.

Note: “Pinning/unpinning gatekeepers” on page 141 describes pinning gatekeepers.

Syntax

#PIN CUU(cuu)

Parameters

CUU(cuu)

The CUU of the gatekeeper device.

Example

#PIN CUU(03C8)

QUERY CON

The QUERY CON command is an alias for the DISPLAY command. You can use QUERY CON anywhere you would use the DISPLAY command.

Syntax

QUERY CONgroup [cngrp]

Parameters

cngrp

The name of the consistency group.

If no consistency group is specified, information for all consistency groups is displayed.

Example

1. To view all consistency groups:

   QUERY CON

2. To view all devices in consistency group DB2PROD:

   QUERY CON DB2PROD
REFRESH

Dynamically updates a consistency group configuration without terminating ConGroup.

**Note:** “Refreshing ConGroup configuration” on page 144 describes refreshing ConGroup configuration.

Syntax

```
REFresh [FORCE]
```

Parameters

FORCE

   Enables the new consistency configuration even if all consistency groups are not in a proper state for enablement.

REMSPLIT

Remotely splits all target (R2) STDs from their BCVs.

**Note:** The TimeFinder/Mirror for z/OS Product Guide describes STDs and BCVs. “Splitting remote BCVs” on page 101 describes performing a remote split in the ConGroup environment.

Syntax

```
REMSPLIT cngrp
```

Parameters

```
cngrp
```

   The name of the consistency group.

Example

To split target (R2) devices from their BCVs for consistency group PROD:

```
REMSPLIT PROD
```
RESET

Updates the state of a consistency group in ConGroup internal tables.

Note: “Updating consistency group state” on page 147 describes updating the consistency group state.

Syntax

RESET cngrp [VERIFY]

Parameters

cngrp
    The name of the consistency group.

VERIFY
    Verifies the consistency group whose state is being updated.
    Specifying VERIFY is equivalent to issuing the RESET command followed by the VERIFY command.

    Note: “Verifying consistency groups” on page 94 describes verification of consistency groups.

Example

To reset consistency group SALES:

RESET SALES
RESUME

Resumes operations for a consistency group.

**Note:** "Resuming consistency group operations" on page 105 describes resuming operations.

**Syntax**

```
RESume cngrp [SPLIT][{NOWAIT|WAIT}]
```

**Parameters**

- **cngrp**
  - The name of the consistency group.

- **NOWAIT**
  - *(Default)* Executes the command immediately.

  If the ALL-CONGROUPS lock is already held when the RESUME command is issued, the command fails and the CGRP387W message is issued.

- **SPLIT**
  - Specifies that all remote devices for the consistency group are to be split from their BCVs before the resume process starts.

  **Note:** The *TimeFinder/Mirror for z/OS Product Guide* describes STDs and BCVs. "Splitting remote BCVs" on page 101 describes performing a remote split in the ConGroup environment.

  Specifying SPLIT is equivalent to issuing the REMSPLIT command followed by the RESUME command.

  The SPLIT parameter does not split BCVs established to local STDs. The SPLIT parameter applies only to BCVs attached to the R2s.

  If you issued a RESUME SPLIT command, you have to issue the CANCEL REMSPLIT and the CANCEL RESUME command before resuming operations. If you just issue a CANCEL RESUME, ConGroup does not process the CANCEL RESUME until split processing is complete.

- **WAIT**
  - Waits until the ALL-CONGROUPS lock is available. Meanwhile, the issued commands are "stacked".

**Example**

To resume operations for consistency group SALES:

```
RESUME SALES
```
**SET VERIFY_INTERVAL**

Sets the time interval (in seconds) before the automatic verification subtask attempts to verify the state of all enabled/active consistency groups. This is to ensure that all consistency group devices are still in the expected state.

**Note:** “Verifying consistency groups” on page 94 describes consistency group verification.

**Syntax**

\[
\text{SET VERIFY_INTERVAL=}interval
\]

**Parameters**

(interval) The time interval (in seconds). Valid values are from 0 to 99999999. If you specify zero (0), you disable automatic verification.

**Example**

To set the verification interval to 600 seconds:

\[
\text{SET VERIFY_INTERVAL=}600
\]

**STOP**

Stops ConGroup.

**Note:** “Stopping ConGroup” on page 66 provides details about stopping ConGroup.

**Syntax**

\[
\text{STOP \ [FORCE]}
\]

**Parameters**

(FORCE) Forces ConGroup termination.
TAKEOVER

Changes the current owner LPAR to the LPAR from which the TAKEOVER command is issued for CAX operations.

Note: “Managing owner LPARs” on page 138 describes owner LPARs. “Swapping Consistency Groups with CAX” on page 109 describes CAX.

You can issue TAKEOVER from any LPAR running ConGroup.

Note: If ConGroup is started with no owner LPAR, issue the TAKEOVER command to assign an owner LPAR and then issue the ENABLE command for the new owner to actually take ownership.

Syntax

TAKEOVER

TRIP

Trips a consistency group.

Note: “Tripping consistency groups” on page 99 describes tripping consistency groups.

You can issue the TRIP command from any LPAR. You do not have to trip a consistency group from the owner LPAR.

Syntax

TRIP cngrp

Parameters

cngrp

The name of the consistency group.

Example

To trip consistency group DB2PROD:

TRIP DB2PROD
#UNPIN

Unpins a ConGroup gatekeeper server-managed CUU.

**Note:** “Pinning/unpinning gatekeepers” on page 141 describes unpinning gatekeepers.

The #UNPIN command only works against pinned gatekeepers if at least one gatekeeper to the VMAX system remains after the unpin. In other words, you cannot unpin all the gatekeepers to a VMAX system.

**Syntax**

```
#UNPIN CUU(cuu)
```

**Parameters**

- `CUU(cuu)`: The CUU of the gatekeeper device.

**Example**

```
#UNPIN CUU(03C8)
```

VERIFY

Verifies the consistency group configuration.

**Note:** “Verifying consistency groups” on page 94 describes consistency group verification.

**Syntax**

```
VERIFY cngrp [VERBOSE]
```

**Parameters**

- `cngrp`: The name of the consistency group.
- `VERBOSE`: This optional parameter overrides the RESUME_OPTION=NOTRNMSG parameter (if specified) to produce the normal CGRP163W messages issued during VERIFY operation.

**Example**

```
To verify consistency group DB2PROD:

VERIFY DB2PROD
```
ECGUTIL commands

The ECGUTIL utility accepts in-stream or console commands.

In-stream commands can be any length and may span lines. No continuation characters are needed. Columns 72-80 are ignored.

You can also enter all commands through console MODIFY commands. Console commands are identical to in-stream commands, but no delimiting period is necessary when entered through the console.

Note: ECGUTIL is case insensitive. Upper and lower case work identically because all lower case text is converted to upper case.
ADD DEVICES

Adds devices to the device group.

**Note:** “Adding devices to a group” on page 153 describes adding devices to device groups.

All the devices in a single ADD DEVICES statement must reside on a single VMAX system. If necessary, you can specify multiple ADD DEVICES statements.

**Syntax**

```
ADD DEVICES=(device_list) TO GROUP grp
CNTRL {SymmID|Symmname}
[RA srdfgrp]
```

**Parameters**

*device_list*

The list of devices identified with their VMAX device numbers.

You can specify comma-separated individual devices or ranges, using either upper or lowercase characters.

*grp*

The name of the device group you previously defined with the DG command. The name can be from 1 to 8 characters. The name can start with any alphanumeric character.

*srdfgrp*

The SRDF group identified with its 2-character hexadecimal group ID.

*SymmID*

The serial number of the VMAX system on which the devices reside.

Normally, you use 12-digit serial numbers. If you specify a value that is less than 12 digits, the value is prepended with zeros to make a 12-digit number.

*Symmname*

The name of the VMAX system on which the devices reside, up to 64 characters.

If the name has mixed-case characters or embedded blanks, it must be enclosed in quotes.

**Example**

```
ADD DEVICES=(4-FF,5,11e,C0-11B) TO GROUP MYGRP CNTRL 000195700079 RA 08.
```
AG

**CAUTION**
The AG command is for testing purposes only. Improper use may cause operational disruption.

Executes the following two commands for a device group:
- SET GROUP grp CGENA
- SET GROUP grp USRNRY

Before using the AG command, define a device group using the DG command and fill it with devices using the ADD DEVICES command.

**Note:** AG stands for ARM GROUP.

**Syntax**
AG grp

**Parameters**

`grp`

The name of the device group you previously defined with the DG command. The name can be from 1 to 8 characters. The name can start with any alphanumeric character.

CG

Executes the following three commands for a device group:
- SET GROUP grp CGDIS
- SET GROUP grp RECACLR
- SET GROUP grp USRRDY

**Note:** CG stands for CLEAN GROUP.

**Syntax**
CG grp

**Parameters**

`grp`

The name of the device group you previously defined with the DG command. The name can be from 1 to 8 characters. The name can start with any alphanumeric character.
DG

Creates a group of devices.

**Note:** “Defining a device group” on page 152 describes defining a device group.

You can specify multiple DG statements.

**Note:** DG stands for DEFINE GROUP.

**Syntax**

DG *grp*

**Parameters**

*grp*

The name of the device group. The name can be from 1 to 8 characters. The name can start with any alphanumeric character.

SET GROUP

Sets or clears characteristic flags for all devices in the specified device group.

**Note:** The “setting” keywords exist in ECGUTIL only for testing purposes to allow conformation that the clearing keywords function correctly.

**Syntax**

SET GROUP *grp* [CGDIS|CGENA|RECACLR|USRNRDY|USRRDY]

**Parameters**

CGDIS

Clears the CG ENA bit on the devices.

CGENA

Sets the CG ENA bit on the devices.

*grp*

The name of the device group you previously defined with the DG command. The name can be from 1 to 8 characters. The name can start with any alphanumeric character.

RECACLR

Clears the SRDF ECA status on the devices.

USRNRDY

Sets the USR Not Ready flag for the devices.

USRRDY

Sets the USR Ready flag for the devices.
SET MSGLEVEL

Sets the verbosity level of the messages returned from ECGUTIL.

Syntax

SET MSGLEVEL number

Parameters

number

A number from 1 to 9. The higher number you use, the more verbose messages you receive. If you do not specify SET MSGLEVEL, ECGUTIL uses a default value of five (5).

STOP

Terminates the ECGUTIL utility.

Syntax

STOP